Robert Krimmer, Melanie Volkamer,
Bernhard Beckert, Ardita Driza Maurer, David Duenas-Cid, Stéphane Glondu,
Iuliia Krivonosova, Oksana Kulyk, Ralf Küsters, Beata Martin-Rozumilowicz,
Peter Rønne, Mihkel Solvak, Oliver Spycher (Eds.)

Fifth International Joint Conference on Electronic Voting

E-Vote-ID 2020

6-9 October 2020

Co-organized by:

Tallinn University of Technology
Ragnar Nurkse Department of Innovation and Governance
University of Tartu
Johan Skytte Institute of Political Studies
Karlsruhe Institute of Technology
Institute of Applied Informatics and Formal Description Methods
E-Voting.CC GmbH
Competence Center for Electronic Voting and Participation
Gesellschaft für Informatik
German Informatics Society, SIG SEC/ECOM
Kastel
Competence Center for Applied Security Technology

PROCEEDINGS

Robert Krimmer, Melanie Volkamer,
Bernhard Beckert, Ardita Driza Maurer, David Duenas-Cid, Stéphane Glondu, Iuliia Krivonosova, Oksana Kulyk, Ralf Küsters, Beata Martin-Rozumilowicz, Peter Rønne, Mihkel Solvak, Oliver Spycher (Eds.)

5th **Joint International Conference on Electronic Voting**

E-Vote-ID 2020

6-9 October 2020

Co-organized by the Tallinn University of Technology, University of Tartu, Karlsruhe Institute of Technology, E-Voting.CC, Gesellschaft für Informatik and Kastel

TAL TECH PRESS

Proceedings E-Vote-ID 2020
TALTECH Press

ISBN 978-9949-83-601-7 (pdf)

Volume Editors

Prof. Dr. Robert Krimmer
Tallinn University of Technology
Ragnar Nurkse Department of Innovation and Governance
Akadeemia tee 3
12618 Tallinn, Estonia
University of Tartu
Johan Skytte Institute of Political Studies
Lossi 36
51003 Tartu, Estonia
robert.krimmer@taltech.ee

Prof. Dr. Melanie Volkamer
Karlsruhe Institute of Technology
Institute of Applied Informatics and Formal Description Methods
Kaiserstr. 89
76131 Karlsruhe, Germany
melanie.volkamer@secuso.org

Bernhard Beckert
Karlsruhe Institute of Technology
E-mail: beckert@kit.edu

Ardita Driza-Maurer
Zentrum für Demokratie Aarau/Zurich University
E-mail: ardita.driza@sefanet.ch

David Duenas-Cid
Tallinn University of Technology / Kozminski University
E-mail: david.duenas@taltech.ee

Stéphane Glondu
Institut National de Recherche en Sciences et Technologies du Numérique
E-mail: stephane.glondu@inria.fr

Iuliia Krivonosova
Tallinn University of Technology
E-mail: Iuliia.krivonosova@taltech.ee

Oksana Kulyk
IT University of Copenhagen
E-mail: okku@itu.dk

Ralf Küsters
University of Stuttgart
E-mail: ralf.kuesters@sec.uni-stuttgart.de

Beata Martin-Rozumlowicz
International Foundation for Electoral Systems
E-mail: bmartinrozumilowicz@ifes.org

Peter Rønne
University of Luxembourg
E-mail: peter.roenne@gmail.com

Mihkel Solvak
University of Tartu
E-mail: mihkel.solvak@ut.ee

Oliver Spycher
Swiss Federal Chancellery
E-mail: spycher.oliver@gmail.com

printed by TALTECH Press, Tallinn (Estonia)

This conference is co-organized by:

Tallinn University of Technology - Ragnar Nurkse Department of Innovation and Governance

University of Tartu - Johan Skytte Institute of Political Studies

Karlsruhe Institute of Technology - Institute of Applied Informatics and Formal Description Methods

E-Voting.CC GmbH - Competence Center for Electronic Voting and Participation

Gesellschaft für Informatik, German Informatics Society, SIG SEC/ECOM

Kastel, Competence Center for Applied Security Technology

Supported by:

Regional Government of Vorarlberg

Swiss Federal Chancellery

General Chairs

Krimmer, Robert (Tallinn University of Technology - Ragnar Nurkse Department of Innovation and Governance and University of Tartu - Johan Skytte Institute of Political Studies, Estonia)
Volkamer, Melanie (Karlsruhe Institute of Technology - Institute of Applied Informatics and Formal Description Methods, Germany)

Track on Security, Usability and Technical Issues

Bernhard Beckert
Karlsruhe Institute of Technology, Germany
Ralf Küsters
University of Stuttgart, Germany
Oksana Kulyk
IT University of Copenhagen, Denmark

Track on Administrative, Legal, Political and Social Issues

David Duenas-Cid
Tallinn University of Technology, Estonia / Kozminski University, Poland
Mihkel Solvak
University of Tartu, Estonia

Track on Election and Practical Experiences

Spycher, Oliver
Swiss Federal Chancellery, Switzerland
Beata Martin-Rozumilowicz
International Foundation for Electoral Systems, USA

PhD Colloquium

Ardita Driza-Maurer
Zentrum für Demokratie Aarau/Zurich University, Switzerland
Iuliia Krivonosova
Tallinn University of Technology, Estonia

Poster and Demo Session

Rønne, Peter
University of Luxembourg, Luxembourg
Glondu, Stéphane
Institut National de Recherche en Sciences et Technologies du Numérique, France

Organizational Committee

Licht, Nathan
Castle, Salina
E-Voting.CC, Austria

Outreach Chairs

Rønne, Peter
University of Luxembourg, Luxembourg
Krivonosova, Iuliia
Tallinn University of Technology, Estonia

Preface

This volume contains papers presented at the 5th InternationalJoint Conference on Electronic Voting (E-Vote-ID 2020), held during October 6-9, 2020. Due to the extraordinary situation provoked by the Covid-19 pandemic, the conference was held online during this edition, instead of at the traditional venue in Bregenz, Austria. The E-Vote-ID conference resulted from the merging of EVOTE and Vote-ID, and now totaling 16 years since the 1st E-Vote conference in Austria. Since that conference in 2004, over 1,000 experts have attended the venue, including scholars, practitioners, authorities, electoral managers, vendors, and PhD students. The conference collected the most relevant debates on the development of electronic voting, from aspects relating to security and usability through to practical experiences and applications of voting systems, also including legal, social or political aspects, amongst others; turning out to be an important global referent in relation to this issue.

This year, the conference also consisted of the following tracks:

- Security, Usability and Technical Issues Track
- Administrative, Legal, Political and Social Issues Track
- Election and Practical Experiences Track
- PhD Colloquium, Poster and Demo Session (held on the day before the conference).

E-Vote-ID 2020 received 55 submissions, each of them being reviewed by three to five Program Committee members, using a double-blind review process. The selected papers cover a wide range of topics connected with electronic voting, including experiences and revisions of the real uses of e-voting systems and corresponding processes in elections. We would like to thank the German Informatics Society (Gesellschaft für Informatik) with its ECOM working group, and KASTEL for their partnership over many years. Further we would like to thank the Swiss Federal Chancellery for their kind support. Special thanks go to the members of the International Program Committee for their hard work in reviewing, discussing, and shepherding papers. They ensured the high quality of these proceedings with their knowledge and experience. We are also thankful for the financial support received through the European Union (H2020 Research and Innovation Programme, Grant Agreement No 857622).

October 2020

Robert Krimmer
Melanie Volkamer

Table of Contents

Confidence and Costs

Audits and Verification

PhD Colloquium

Posters & Demo Session

Program Committee

Marta Aranyossy	Corvinus University of Budapest
Jordi Barrat i Esteve	eVoting Legal Lab
Bernhard Beckert	Karlsruhe Institute of Technology
Josh Benaloh	Microsoft
Matthew Bernhard	University of Michigan
David Bismark	Votato
Nadja Braun Binder	University of Zurich
Christian Bull	The Norwegian Ministry of Local Government and Regional Development
Susanne Caarls	Election Consultant
Gianpiero Catozzi	UNDP
Thomas Chanussot	IFES
Riccardo Chelleri	European Union
Veronique Cortier	CNRS, Loria
Staffan Darnolf	IFES
Ardita Driza-Maurer	Zentrum für Demokratie Aarau/Zurich University
David Duenas-Cid	Tallinn University of Technology and Kozmisnki University
Helen Eenmaa-Dimitrieva	University of Tartu
Aleksander Essex	University of Western Ontario
Joshua Franklin	National Institute of Standards and Technology
Micha Germann	University of Bath
J Paul Gibson	Mines Télécom
Rosario Giustolisi	IT University of Copenhagen
Kristian Gjøsteen	Norwegian University of Science and Technology
Nicole Goodman	University of Toronto
Rajeev Gore	The Australian National University
Ruediger Grimm	University of Koblenz
Rolf Haenni	Bern University of Applied Sciences
Thomas Haines	Queensland University of Technology
Toby James	University of East Anglia
Robert Krimmer	Tallinn University of Technology and University of Tartu
Iuliia Krivonosova	Tallinn University of Technology
Ralf Kuesters	University of Stuttgart
Oksana Kulyk	IT University of Copenhagen
Leontine Loeber	University of East Anglia
Ryan Macias	RSM Election Solutions LLC
Beata Martin-Rozumilowicz	IFES
Ronan McDermott	Mcdis
Vladimir Misev	OSCE/ODIHR
Johannes Mueller	University of Luxembourg
Magdalena Musial-Karg	Adam Mickiewicz University

Andras Nemeslaki	Budapest University of Technology and Economics
Stephan Neumann	Landesbank Saar
Hannu Nurmi	University of Turku
Jon Pammett	Carleton University
Liisa Past	Riigikantselei
Olivier Pereira	UCLouvain
Goran Petrov	OSCE
Egger Philipp	Staatskanzlei Kanton St.Gallen
Stéphanie Plante	University of Ottawa
Josep MÂª Reniu	University of Barcelona
Peter Roenne	SnT, University of Luxembourg
Mark Ryan	University of Birmingham
P. Y. A. Ryan	University of Luxembourg
Peter Sasvari	National University of Public Service
Steve Schneider	University of Surrey
Berry Schoenmakers	Eindhoven University of Technology
Carsten Schuermann	IT University of Copenhagen
Uwe Serdült	Centre for Research on Direct Democracy
Delfina Soares	University of Minho
Mihkel Solvak	University of Tartu
Oliver Spycher	Swiss Federal Chancellery
Philip Stark	University of California, Berkeley
Vanessa Teague	Thinking Cybersecurity
Tomasz Truderung	University of Trier
Priit Vinkel	State Electoral Office of Estonia
Melanie Volkamer	Karlsruhe Institute of Technology
Kåre Vollan	Quality AS
Roland Wen	The University of New South Wales
Gregor Wenda	BMI
Jan Willemson	Cybernetica
Peter Wolf	International IDEA
Michael Yard	IFES

Author Index

Manipulation, Audits and Veto

Shifting the Balance-of-Power in STV Elections

Michelle Blom[1], Andrew Conway[2], Peter J. Stuckey[3], and Vanessa J. Teague[4]

[1] School of Computing and Information Systems, University of Melbourne
michelle.blom@unimelb.edu.au
[2] Silicon Econometrics Pty. Ltd
andrewelections@greatcactus.org
[3] Department of Data Science & AI, Monash University
Peter.Stuckey@monash.edu
[4] Thinking Cybersecurity Pty. Ltd.
vanessa@thinkingcybersecurity.com

Abstract. In the context of increasing automation of Australian electoral processes, and accusations of deliberate interference in elections in Europe and the USA, it is worthwhile understanding how little a change in the recorded ballots could change an election result. In this paper we construct manipulations of the ballots in order to change the overall balance of power in an Australian Federal Senate election – the upper house of Parliament. This gives, hopefully tight, overestimations of the Margin of Victory (MOV) for the party or coalition winning the Senate. This is critical information for determining how well we can trust the reported results, and how much auditing should be applied to the election process to be certain that it reports the true result. The challenge arising in Australian Federal Senate elections is that they use a complicated Single Transferable Vote (STV) method for which it is intractable to compute the true MOV, hence we must rely on greedy methods to find small manipulations.

Keywords: Single Transferable Vote · Balance of Power · Margin of Victory

1 Introduction

In a climate of increasing public mistrust in all governmental activities, assurances that the results of elections are correct are critical for democracies to function well. One critical statistic that helps to define how trustworthy an election result is, is the so called Margin of Victory (MOV), which indicates the minimal number of ballots that need to be modified to change the election result. If the MOV is small, then we should invest considerable effort in auditing the election processes, since the true result may differ if inevitable errors lead to changes greater then that MOV. If the MOV is large we require less auditing to be assured that the election outcome is likely to be correct.

The Australian Federal Parliament consists of two houses: the House of Representatives, which defines the executive part of government responsible for making new laws; and the Senate, a house of review. For laws to be enacted they must pass both houses, so the controller of the Senate has significant influence on what legislation can be enacted. Australian politics is dominated by two "parties": the Labor Party (progressive); and the Liberal/National Party Coalition (conservative), an enduring coalition of two

parties. Historically, one or other party has formed government. The Senate is more complicated as there is a greater number of smaller parties and independents. In some cases no party has held the balance of power in the Senate, though usually one or other party, with perhaps some agreements with minor parties, does.

Existing work [2] has examined how to compute the MOV for Australian House of Representatives elections, which makes use of Instant Runoff Voting (IRV). In this paper, we examine the much more challenging problem of *estimating* the MOV for Australian Federal Senate elections. The difficulty arises because the election uses Single Transferable Vote (STV) which is a complicated election methodology. Determining the MOV of an STV election is NP-hard [8]. While we can determine MOVs for small individual STV elections [3], these methods do not scale to the size of the elections that actually occur for the Australian Federal Senate.

An Australian Federal Senate election consists of a separate STV election in each of the six Australian states, and the two Australian territories. There are 76 seats in the Senate, with 12 seats awarded to each of the six states, and 2 to each of the two territories. In a regular election, 6 of the available 12 seats for each state, and both of the 2 seats for each territory, are up for re-election. In a double-dissolution election, all 76 seats are vacated. The party, or coalition of parties, that occupies the majority of seats in the Senate chamber (39 or greater) significantly influences the legislation that the government is able to pass. Legislation has to pass through both houses of Parliament (the lower and upper house) before it can become law. In the 2016 and 2019 Australian Federal elections, conservative politicians have formed the majority in both houses. This has limited the power of more progressive parties to shape legislation.

In this paper, we consider estimating the number of ballot changes required to change the outcome of such an election to give a particular coalition of parties the majority in the Senate. In 2016, we would have had to shift four Senate seats away from conservatives, to progressive candidates, to change the nature of the majority. In 2019, only two seats were required to change hands to achieve a progressive majority. We present a heuristic, combined with an integer program (IP), to compute an upper bound on the number of ballot changes required in order to award an additional n seats to a coalition of parties, $\mathbb{C}$. In other words, if a coalition $d \in \mathbb{C}$ was originally awarded N_d seats, we are interested in manipulations that would result in d being awarded $n + N_d$ seats. This implies that n seats are taken away from candidates outside of d.

An Australian Federal Senate election consists of a number of separate STV elections. Our approach is based on finding small manipulations of each individual s-seat STV election, that awards an additional $j = 1, \ldots, k$ seats to our desired coalition, where $k = \min(s, n)$. A knapsack problem is then solved to determine the combination of these manipulations that results in a combined n-seat shift to our coalition with the least number of required ballot changes. Existing work [1] considers the use of local search for finding small manipulations of a STV election that elects a specific, favoured candidate c to a seat. This paper moves beyond this to find an upper bound on the manipulation required to elect n additional candidates from a coalition of parties, across a set of individual STV elections responsible for allocating seats in a Senate.

We apply our method to data from the 2016 and 2019 Australian Federal Senate elections. In both elections, candidates from conservative parties form the majority in

the elected Senate. We consider a coalition of more centrist or left-leaning parties, forming our desired coalition d. We then use our approach to find an upper bound on the number of ballots we would have to change, across each of the state and territory STV elections that form part of the Senate election as a whole, to shift enough seats to candidates in d to change the nature of the majority. In the 2016 and 2019 elections, we want to shift $n = 4$, and $n = 2$, seats respectively. Our local search algorithm is used to compute candidate manipulations that shift $j = 1, \ldots, \min(s, n)$ seats in each individual s-seat STV election. A simple integer program (IP) is then applied to the results to select a least cost combination of manipulations to apply in each state and territory. We have found that we can give a progressive coalition d a majority by changing 40,008 ballots in the 2016 election, and 27,635 ballots in the 2019 election.

2 Preliminaries

STV is a preferential system of voting in which voters rank candidates or parties in order of preference, and candidates compete for s seats. In Australian Senate Elections voters may cast their ballot in two ways. First, they may vote *above the line*. At the top of each ballot is a sequence of boxes, one for each party and group of independent candidates. To vote above the line, a voter ranks at least 6 of these parties and grouped independents.Alternatively, a voter may vote *below the line*. Under each of the above the line party boxes is a list of candidates belonging to that party or group. To vote below the line, a voter ranks at least 14 individual candidates in order of preference. STV elections for the Australian Senate can involve over 100 candidates.

Definition 1 (STV Election) *An STV election $\mathcal{E}$ is a tuple $\mathcal{E} = (\mathcal{C}, \mathcal{P}, \mathcal{B}, Q, s)$ where $\mathcal{C}$ is a set of candidates, $\mathcal{P}$ is the set of parties or groups to which candidates belong, $\mathcal{B}$ the multiset of ballots cast, Q the election quota (the number of votes a candidate must attain to win a seat – the Droop quota – Eqn 1), and s the number of seats to be filled.*

$$Q = \left\lfloor \frac{|\mathcal{B}|}{s+1} \right\rfloor + 1 \tag{1}$$

We will use a small running example to describe the concepts in this section, and our n-seat shifting approach. In this example, two 3-seat STV elections, $\mathcal{E}_1$ and $\mathcal{E}_2$, are held to elect a total of 6 candidates to a small Senate. In each election, four parties (A, B, C, and D) field 2 candidates, resulting in a total of 8 candidates. The candidates of $\mathcal{E}_1$ are denoted a_{11}, a_{12}, …, d_{11} and d_{12}, and those of $\mathcal{E}_2$, a_{21}, a_{22}, …, d_{21} and d_{22}. Tables 1a and 2a define the ballot profiles of $\mathcal{E}_1$ and $\mathcal{E}_2$, listing the number of ballots cast with a range of different *above the line* party rankings. For each ranking, we state the equivalent *below the line* ranking, indicating how the ballot would pass from candidate to candidate if they were eliminated in that sequence. During the election counting process, valid above the line votes are treated exactly as their below the line equivalent.

The counting of ballots in an STV election starts by distributing each ballot to the tally pile of its first ranked candidate. An above the line vote with a first preference for party p is given to the first candidate of that party listed on the ballot. Candidates are awarded a seat if the number of votes in their tally reaches or exceeds a threshold,

Ranking ATL	Ranking BTL	Count
[A, B]	$[a_{11}, a_{12}, b_{11}, b_{12}]$	270
[B, A, D, C]	$[b_{11}, b_{12}, a_{11}, a_{12}, d_{11}, d_{12}, c_{11}, c_{12}]$	250
[C, D, A, B]	$[c_{11}, c_{12}, d_{11}, d_{12}, a_{11}, a_{12}, b_{11}, b_{12}]$	20
[D, C, A]	$[d_{11}, d_{12}, c_{11}, c_{12}, a_{11}, a_{12}]$	5

(a)

Seats: 3, Quota: 137

Candidate	Round 1	Rounds 2-3	Rounds 4-6	Rounds 7-8
		a_{11}, b_{11} elected 133 votes to a_{12} 113 votes to b_{12}	c_{12}, d_{12} eliminated d_{11} eliminated 50 votes to c_{11}	c_{11} eliminated 250 votes to a_{12} a_{12} elected
a_{11}	270	–	–	–
a_{12}	0	0+133 = 133	133	133 +25 = 158
b_{11}	250	–	–	–
b_{12}	0	0+113 = 113	113	113
c_{11}	20	20	20 +5 = 25	–
c_{12}	0	0	–	–
d_{11}	5	5	–	–
d_{12}	0	0	–	–

(b)

Table 1: STV election, $\mathcal{E}_1$, stating (a) the number of ballots cast with each listed above-the-line ranking over parties A to D, their equivalent below-the-line ranking over candidates a_{11} to d_{12}, and (b) the tallies after each round of election, and elimination.

called a *quota*. The value of the quota is based on the total number of ballots cast in the election, and the number of seats available (Eqn 1). The quotas of elections $\mathcal{E}_1$ and $\mathcal{E}_2$ are 137 votes ($1 + \lfloor 545 / (3 + 1) \rfloor = 137$) and 188 votes, respectively.

Counting proceeds by electing candidates whose *tallies* (Definition 2) reach or exceed the quota, and distributing their *surplus* to the candidates that remain standing. A candidate's surplus is equal to the difference between their tally value and the quota. The non-exhausted ballots in an elected candidates tally pile are distributed to eligible candidates at a *reduced value*. Each ballot, starting with a value of 1, is reduced in value so that the sum of the value of the transferred ballots is equal to the surplus.

Definition 2 (Tally $t_i(c)$) *The tally of a candidate $c \in \mathcal{C}$ in round i is the sum of the values of the ballots in c's tally pile. These are the ballots for which c is ranked first among the set of candidates still standing, S_i. Let $\mathcal{B}_{i,c}$ denote the subset of ballots sitting in c's tally pile, and $v_i(b)$ the value of ballot b, at the start of round i.*

$$t_i(c) = \sum_{b \in \mathcal{B}_{i,c}} v_i(b) \tag{2}$$

Ranking ATL	Ranking BTL	Count
[B, C, D, A]	$[b_{21}, b_{22}, c_{21}, c_{22}, d_{21}, d_{22}, a_{21}, a_{22}]$	2,000
[A, D, C, D]	$[a_{21}, a_{22}, d_{21}, d_{22}, c_{21}, c_{22}, d_{21}, d_{22}]$	2,100
[D, A, B, C]	$[d_{21}, d_{22}, a_{21}, a_{22}, b_{21}, b_{22}, c_{21}, c_{22}]$	1,700
[C, D, A, B]	$[c_{21}, c_{22}, d_{21}, d_{22}, a_{21}, a_{22}, b_{21}, b_{22}]$	1,700

(a)

Seats: 3, Quota: 188

Candidate	Round 1	Rounds 2-3	Rounds 4-6	Rounds 7-8
		a_{21}, b_{21} elected 22 votes to a_{22} 12 votes to b_{22}	d_{22}, c_{22} eliminated b_{22} eliminated 12 votes to c_{21}	a_{22} eliminated 22 votes to d_{21} d_{21} elected
a_{21}	200	–	–	–
a_{22}	0	0+22 = 22	22	–
b_{21}	210	–	–	–
b_{22}	0	0+12 = 12	–	–
c_{21}	170	170	170 +12 = 182	182
c_{22}	0	0	–	–
d_{21}	170	170	170	170 +22 = 192
d_{22}	0	0	–	–

(b)

Table 2: STV election, $\mathcal{E}_2$, stating (a) the number of ballots cast with each listed above-the-line ranking over parties A to D, their equivalent below-the-line ranking over candidates a_{21} to d_{22}, and (b) the tallies after each round of election, and elimination.

Table 1b shows that for $\mathcal{E}_1$, only the first listed candidate of each party have votes in their tallies after Round 1 of counting. This is because all voters have cast an above the line vote. The ballots sitting in a_{11}'s tally will pass to a_{12} when a_{11} is either elected or eliminated. Candidates a_{11} and b_{11} have a quota with tallies of 270 and 250 votes, and surpluses of 133 and 113 votes. They are elected to the first two available seats in Rounds 2-3 of counting. All 270 ballots in a_{11}'s tally pile are given to a_{12}, but they now have a combined value of 133 (each ballot now has a reduced value of 0.4926).

If no candidate has a quota, the candidate with the smallest tally is eliminated. In $\mathcal{E}_1$ (Table 1b), no candidate has a quota after the election of a_{11} and b_{11}. The candidates with the smallest tally, c_{12} and d_{12}, both with 0 votes, are eliminated in Rounds 4-5. In Round 6, d_{11}, with 5 votes, is eliminated. These 5 votes are transferred, at their current value, to c_{11}, as d_{12} is no longer standing. Candidate c_{11} now has 25 votes.

The STV counting process continues in rounds of electing candidates whose tallies have reached a quota, and elimination of candidates with the smallest tally. In $\mathcal{E}_1$, candidate c_{11} is eliminated in Round 7, with their votes distributed to a_{12}. In Round 8, a_{12} is elected to the final seat, with their tally having exceeded a quota's worth of votes.

Several STV variants exist, differing in the way that surpluses are distributed [7]. The method of reducing the value of transferred ballots described above is the Inclusive Gregory Method [5]. The precise rules used by the Australian Federal Senate for adjusting the values of transferred ballots are more complex, and outlined in legislation.

The approach we present in this paper searches for manipulations of the STV elections that form part of an Australian Federal Senate election that achieve a desired outcome. Given a favoured coalition of parties d, whose candidates have been awarded N_d seats in the un-manipulated election, we are interested in manipulations that award $N_d + n$ candidates in d a seat. We define a manipulation of an STV election as follows.

Definition 3 (Manipulation $\mathcal{M}$) *A manipulation for an election $\mathcal{E} = (\mathcal{C}, \mathcal{P}, \mathcal{B}, Q, s)$ is a tuple $\mathcal{M} = (\mathcal{B}^+, \mathcal{B}^-)$, where: $\mathcal{B}^+$ denotes a multiset of ballots to add to $\mathcal{B}$; $\mathcal{B}^-$ a multiset of ballots to remove from $\mathcal{B}$; and $|\mathcal{B}^+| \equiv |\mathcal{B}^-|$. The result of applying $\mathcal{M}$ to an election $\mathcal{E}$ is a modified election profile $\mathcal{E}' = (\mathcal{C}, \mathcal{P}, \hat{\mathcal{B}}, Q, s)$, where $\hat{\mathcal{B}}$ is the result of removing each ballot in $\mathcal{B}^-$ from $\mathcal{B}$, and then adding each ballot in $\mathcal{B}^+$ to $\mathcal{B}$.*

To assess whether a given manipulation $\mathcal{M}$ awards n additional seats to our favoured coalition d, we simulate the STV counting process on the manipulated election profile $\mathcal{E}'$, and count the number of seats awarded to d in the outcome. We use a simulator, denoted SIM-STV, that captures the intricate rules specific to the Australian Federal Senate election, as defined in legislation. All the manipulations we generate in our experiments are validated on the full federal rules [4] using SIM-STV, and ballot data published by the AEC, standardized at `https://vote.andrewconway.org/`. An example manipulation for the election of Table 1 is shown in Example 1.

Example 1. Consider election $\mathcal{E}_1$ of Table 1. If we replace 111 ballots with the above the line ranking [A, B] with the above the line ranking [D], we no longer elect candidate a_{12} but elect d_{11} in their place. Candidates a_{11} and b_{11} are elected in the first two rounds, as before. Candidates d_{12}, c_{12}, and c_{11}, are then eliminated. The reduced flow of votes from a_{11} to a_{12} leaves a_{12} on only 22 votes, compared to d_{11}'s 136 and b_{12}'s 135 votes.

The rules used for Australian Federal Senate elections [6] are close to that described above, with some idiosyncrasies. For example, when ballots are distributed from one candidate's tally pile to another, the total value of those ballots is rounded down to the nearest integer. This practice causes a number of votes to be lost over the course of the tallying process. For further details of SIM-STV, we refer the reader to [1].

3 Finding n-seat Senate Manipulations

We present an approach for computing a manipulation of one or more of the individual STV elections that form an Australian Federal Senate election, to shift the majority of seats away from an unfavoured coalition of parties to a favoured coalition. In the 2016 and 2019 Australian Federal Senate elections, a conservative coalition of parties had a 4 and 2 seat majority. Our approach looks for the best combination of manipulations to apply to the state and territory STV elections to realise a combined $n = 4$, and $n = 2$,

seat shift to a progressive coalition. A shift of 2 seats could be realised by shifting 1 seat to our favoured coalition in Victoria, for example, and 1 seat in New South Wales.

Our approach consists of two stages. The first stage looks at each constituent s-seat STV election individually. We use a local search heuristic, described in Section 3.1, to find small manipulations that shift varying numbers of seats ($k = 1, \ldots, \min(s, n)$) away from the undesired coalition $u \in \mathbb{C}$ to candidates that belong to a desired coalition of parties $d \in \mathbb{C}$. The local search method is not optimal – it may not find the smallest possible manipulation that shifts k seats to our favoured candidates.

As a result of this first stage, we have a series of manipulations, for each state and territory election, that shift varying numbers of seats to our desired coalition. The second stage of our approach solves a simple integer program (IP) to select the combination of manipulations that realises our n-seat shift with the smallest number of required ballot changes. To achieve a shift of 4 seats, for example, we may shift 1 seat in each of four state elections, or 2 seats in one state and 2 seats in another.

Example 2. In Example 1, we manipulated $\mathcal{E}_1$ by 111 ballots (shifted from party A to D), giving 1 seat to D at the expense of A. In order to give 2 seats to a coalition of parties C and D, we need to shift ballots away from A and B to C and D. Consider a manipulation that removes 116 ballots with ranking [A, B], replacing them with the ranking [C], and 131 ballots with ranking [B,A,D,C], replacing them with [D]. This manipulation results in both c_{11} and d_{11} being elected at the expense of a_{12} and b_{11}.

3.1 Finding Manipulations with Local Search

Given an s-seat STV election $\mathcal{E}$, we present a local search heuristic for finding manipulations that award an additional k seats to a candidates in a desired coalition d. In the original outcome of $\mathcal{E}$, N_d seats have been awarded to candidates from d. We seek a manipulation of $\mathcal{E}$ in which $N_d + k$ seats are awarded to candidates from d. The heuristic is provided as input k candidate pairs of unfavoured original winner w, and favoured original loser l. We then search for smaller and smaller manipulations that aim to rob each w of a seat, and elect l. We repeatedly apply this heuristic to different sets of k candidate pairs, returning the smallest found successful manipulation as a result.

To identify sets of k winner-loser pairs to consider, we start with: a set of unfavoured original winners $\mathcal{W}$; and a set of favoured original losers $\mathcal{L}_d$.

1. Let $\overline{\mathcal{W}}$ denote the set of all k-candidate subsets of $\mathcal{W}$, and $\overline{\mathcal{L}_d}$ the set of all k-candidate subsets of $\mathcal{L}_d$.
2. The k-candidate subsets in $\overline{\mathcal{W}}$ are sorted in order of the total tally of candidates upon their election, from smallest to largest. This is the sum of each candidates tally in the round in which they were elected to a seat. The first subset in the sorted $\overline{\mathcal{W}}$ consists of candidates who were elected to a seat with the smallest tallies.
3. The k-candidate subsets in $\overline{\mathcal{L}_d}$ are sorted in order of the total tally of candidates upon their elimination, from largest to smallest. This is the sum of each candidates tally in the round in which they were eliminated. The first subset in the sorted $\overline{\mathcal{L}_d}$ consists of candidates who were eliminated with the largest tallies.

4. To limit the complexity of our approach, we restrict our attention to the first M subsets in $\overline{\mathcal{W}}$ and $\overline{\mathcal{L}_d}$. For each $W \in \overline{\mathcal{W}}$, we consider each $L \in \overline{\mathcal{L}_d}$. We apply our local search heuristic with k winner-loser pairs formed by pairing the first winner in W with the first loser in L, the second winner in W with the second loser in L, and so on. We return the best (smallest) successful manipulation found by applying our local search heuristic to each set of the $M \times M$ generated k winner-loser pairs.

Given a list of k unfavoured winner, favoured loser, pairs, our method aims to replace each unfavoured winner with its paired loser. However, any manipulation that elects $N_d + k$ candidates from d is considered to be successful. Each application of the local search heuristic involves two phases.

Phase 1 The first stage finds an initial, but potentially quite large, successful manipulation that, upon simulation of the manipulated election profile, elects at least $k + N_d$ candidates from our desired coalition $d \in \mathbb{C}$. This manipulation is denoted M_0.

Phase 2 We then repeatedly search for a good 'size reducing' move to apply to M_0. These moves reduce the number of ballots shifted between candidates, while still ensuring that the manipulation successfully elects $k + N_d$ candidates from d. In each iteration, we examine the set of possible changes (moves) we could make to the current 'best found' manipulation, selecting the move that results in the largest reduction in the number of ballot changes. When no 'size reducing' move can be found, search terminates and returns the best (smallest) manipulation it has found.

A manipulation defines k sets of ballot shifts between pairs of candidates from W and L. For each such (w, l) pair, our goal is to find a manipulation that replaces a certain number of ballots that favour w – ballots that form part of w's tally at the point of their election – with ballots that favour l. We consider three different approaches for specifying the ranking of these l-favouring ballots, denoted BTL, ATL, and IW. The latter, IW, uses the set of original winners from our desired coalition d, denoted $\mathcal{W}_d$.

BTL A below the line vote that preferences l first, and each other loser in L subsequently, in the order they appear in L.

ATL An above the line vote that preferences l's party first, and the parties of all other candidates in L subsequently, in the order they appear in L.

IW A below the line vote that preferences l first, and each of the original winners from our desired coalition d, $\mathcal{W}_d$, subsequently.

Example 3. When seeking to elect $k = 2$ candidates from the coalition $d = \{C, D\}$ in $\mathcal{E}_1$, our list of original winners is $\mathcal{W} = \{a_{11}, a_{12}, b_{11}\}$ and favoured losers $\mathcal{L}_d = \{c_{11}, c_{12}, d_{11}, d_{12}\}$. Our winner subsets $\overline{\mathcal{W}}$, sorted in order of the total tally of the candidates, upon their election (smallest to largest), are $\overline{\mathcal{W}} = \{\{a_{12},b_{11}\}, \{a_{12},a_{11}\}, \{b_{11},a_{11}\}\}$. Our subsets of favoured losers, sorted in order of the total tally of candidates, upon their elimination (largest to smallest), are $\overline{\mathcal{L}_d} = \{\{c_{11},d_{11}\}, \{c_{11},d_{12}\}, \{c_{11},c_{12}\},\{d_{11},d_{12}\}, \{d_{11},c_{12}\}, \{d_{12},c_{12}\}\}$. Our approach will try to elect each pair of losers in $\overline{\mathcal{L}_d}$, at the expense of each pair of winners in $\overline{\mathcal{W}}$, starting with $\{c_{11},d_{11}\}$ and $\{a_{12},b_{11}\}$. For these winner-loser pairs, a_{12}-c_{11} and b_{11}-d_{11}, we find a M_0 that: replaces 226 ballots that sit

```
FINDINITIALMANIPULATION(k, N_d, W, L, t)
1   M_0 ← ∅
2   for i in 1..k do
3       w ← W[i]
4       l ← L[i]
5       Δ_{i,0} ← ⌈t[w] − t[l]⌉
6   verified ← Verify M_0 with SIM-STV.
7   if verified then
8       return M_0 as our initial manipulation
9   else
10      M'_0 ← M_0
11      while not verified do
12          M'_0 ← Increase each Δ_{i,0} by a factor of 2, capping each Δ_{i,0} by t[i].
13          verified ← Verify M'_0 with SIM-STV
14          if verified then
15              return M'_0 as our initial manipulation
16  return failure
```

Fig. 1: Phase 1: Find initial manipulation to achieve the election of $k + N_d$ candidates from a desired coalition, where: W denotes original winners that are not in our desired coalition; and L are original losers who are in our desired coalition. Note that $t[c]$ denotes the tally of candidate c upon their election (if $c \in W$) or elimination (if $c \in L$).

in a_{12}'s tally upon their election, with a ranking that favours c_{11}; and 249 ballots that sit in b_{11}'s tally upon their election, with a ranking that favours d_{11}. The total size of this manipulation is 475. The IW method would replace 226 of a_{12}'s ballots with the ranking $[c_{11}]$. The ATL method would replace these ballots with the ranking [C], where C is the party to which c_{11} belongs. The BTL method would form 226 ballots with ranking $[c_{11}, d_{11}, c_{12}, d_{12}]$, adding the remaining favoured losers after c_{11}.

Additional Notation We use notation $\Delta_{i,j}$ to denote the number of ballots shifted between the i^{th} of our k winner-loser pairs, (w, l), in a manipulation M_j. A 'shift' of $\Delta_{i,j}$ ballots between w and l replaces $\Delta_{i,j}$ ballots that sit in w's tally at the time they are elected with $\Delta_{i,j}$ ballots whose ranking has been specified according to one of the above methods (BTL, ATL, or IW). The notation $t[c]$ denotes the tally of candidate c upon their election (if they are an original winner) or elimination (if they lost).

Phase 1: Finding an Initial Manipulation We define an initial manipulation M_0 by assigning a suitably high value to $\Delta_{i,0}$ for each winner-loser pair $i = 1, \ldots, k$ (see Fig 1). We verify M_0 by simulating it with SIM-STV, and verifying that $k + N_d$ candidates from our coalition are elected in the manipulated election.

Phase 2: Reduce size of Manipulation In the case where $k = 1$, we have one winner w that we want to replace with a loser l. Our initial manipulation M_0 is iteratively reduced by only one type of move (as shown in Fig 2). A 'step size', δ, controls how we reduce the size of our manipulation. We first reduce the number of ballots shifted between

MINIMISEMANIPULATION$_{k=1}(M_0, k, N_d, \alpha, \gamma)$
1 $M_{best} \leftarrow M_0$
▷ Initialise step size δ based on size of initial manipulation
2 $\delta \leftarrow \lceil \frac{\Delta_{1,0}}{\gamma} \rceil$
3 **while** true **do**
4 $M_1 \leftarrow M_{best}$
5 $\Delta_{1,1} \leftarrow \Delta_{1,best} - \delta$
6 *verified* $\leftarrow$ Verify M_1 with SIM-STV
7 **if** *verified* **then**
8 $M_{best} \leftarrow M_1$
9 **else**
10 **if** $\delta \equiv 1$ **then return** M_{best}
11 $\delta \leftarrow \lceil \frac{\delta}{\alpha} \rceil$

Fig. 2: Phase 2 ($k = 1$): Reduce size of an initial manipulation M_0 by reducing the number of shifted votes Δ by a step size δ. The step size δ is reduced each time the manipulation becomes too small (i.e., fails to realise the election of $k + N_d$ candidates from our desired coalition). In the above, $\alpha \geq 2$ and $\gamma \geq 2$ are predefined constants.

our winner and loser by the step size δ. If that reduction does not lead to a successful manipulation, we reduce δ, and keep trying (until we fail to find a better manipulation by shifting 1 less ballot). If a successful, smaller manipulation is found, we increase δ.

In the case where $k > 1$, our heuristic applies one of three types of moves in each iteration: reduce the shift of votes between one pair unfavoured winner and favoured loser (MOVE$_1$); reduce the shift of votes between each unfavoured winner and favoured loser pair (MOVE$_2$); and reduce the number of ballots shifted between one winner-loser pair while increasing the shift of votes between each other winner-loser pair (MOVE$_3$).

We maintain a step size δ_i^m for each move type m and winner-loser pair i. When we first use a particular move type $m \in \{1, 2, 3\}$ to reduce the size of a shift of ballots between the i^{th} winner-loser pair, we reduce the number of ballots shifted by the step size δ_i^m. As in the $k = 1$ setting, if that reduction does not lead to a successful manipulation, we reduce δ. If a successful, smaller manipulation is found, we increase the step size for the next time this kind of move is applied. An interpretation of the steps sizes is that they are an estimate of how much we think we can reduce the size of a shift between two candidates, via each different move type, and achieve a successful manipulation.

We apply these moves iteratively, as follows. Pseudocode for each type of move is provided in Fig 3. The predefined constants $\gamma \geq 2$ and $\alpha \geq 2$ are used when initialising, and updating, step sizes. The constant γ is used to initialise our step size δ – the amount by which we reduce the size of a manipulation as we look for smaller and smaller successful manipulations. Given an initial, quite large, manipulation that shifts $\Delta_{i,0}$ ballots between winner-loser pair i, our step size δ_i is initialised to $\lceil \frac{\Delta_{i,0}}{\gamma} \rceil$. The constant α is used to reduce our step size as the algorithm progresses (Step 11 in Fig 2), allowing us to make more fine grained changes in the search for a minimal manipulation.

1. We maintain a running record of the best (smallest) manipulation found thus far, M_{best}, initialised to M_0.

2. Step sizes, δ_i^m, are first initialised to $\lceil \frac{\Delta_{i,0}}{\gamma} \rceil$ for $i = 1, \ldots, k$.
3. As per Fig. 3, we apply move type 1 (MOVE$_1$) to find a smaller manipulation than M_{best}, using the current set of step sizes δ_i^1. The result is a new manipulation M_1.
4. If $M_1 \neq \emptyset$, we have been able to reduce the vote shift between one winner-loser pair. We then apply move type 2 (MOVE$_2$ in Fig. 3) to M_{best} to find a smaller manipulation than M_1, denoted M_2, using the step sizes δ_i^2.
5. If either move type 1 or 2 were successful, we update M_{best} to the smallest of the two manipulations, M_1 or M_2, and return to Step 3.
6. If neither moves 1 and 2 were successful, we apply MOVE$_3$ to find a smaller manipulation than M_{best}, denoted M_3, using the current set of step sizes δ_i^3.
7. If $M_3 \equiv \emptyset$, we have failed to improve upon M_{best}, and return M_{best} as our best found manipulation. If $M_3 \neq \emptyset$, we replace M_{best} with M_3, reset our step sizes for move types 1 and 2 to their initial values, and return to Step 3.

Example 4. After finding an initial manipulation of 475 ballots to award candidates c_{11} and d_{11} a seat at the expense of a_{12} and b_{11}, we move to Phase 2 and try to find a smaller manipulation. In our initial manipulation M_0, we have $\Delta_{1,0} = 226$ and $\Delta_{2,0} = 249$, where winner-loser pair 1 is a_{12}-c_{11} and winner-loser pair 2 is b_{11}-d_{11}.

Using the parameters $\alpha = 5$ and $\gamma = 2$, we initialise our step sizes for each move and winner-loser pair combination as follows:

$$\delta_1^1 = \delta_1^2 = \delta_1^3 = \lceil 226/\gamma \rceil = 113 \qquad \delta_2^1 = \delta_2^2 = \delta_2^3 = \lceil 249/\gamma \rceil = 125$$

As per Fig. 3, we first apply MOVE$_1$ to reduce one of the shifts $\Delta_{1,0}$ and $\Delta_{2,0}$. We consider each $\Delta_{i,0}$ in turn. For pair 1, we can reduce $\Delta_{1,0}$ by the step size, from 226 to 113, and maintain a successful manipulation. Similarly, we can reduce $\Delta_{2,0}$ by its step size, from 249 to 124, leaving $\Delta_{1,0} = 226$, and successfully manipulate $\mathcal{E}_1$ to elect 2 candidates from C and D. We choose the downward shift that results in the largest reduction in ballot changes, and reduce $\Delta_{2,0}$ to 124. Our best found manipulation now shifts 350 ballots. We next consider MOVE$_2$ on our initial manipulation M_0. Here, we see if we can reduce both $\Delta_{i,0}$ by their step sizes δ_i^2, and still maintain a successful manipulation. We find we cannot, the resulting manipulation of 237 ballots is too small. After the first iteration of local search, we accept the best manipulation found across the three move types, MOVE$_1$ (of 350 ballots) in this case, and increase δ_2^1 by a factor of γ.[5] Note that we only consider MOVE$_3$ when neither MOVE$_1$ and MOVE$_2$ is successful.

In the next two iterations, MOVE$_2$ yields the largest reduction in manipulation size, resulting in a manipulation of 282 ballots ($\Delta_{1,best} = 193$ and $\Delta_{2,best} = 89$). In the fourth iteration, MOVE$_1$ and MOVE$_2$ are not successful, and we consider MOVE$_3$. We start by reducing $\Delta_{1,best}$ by δ_1^3, which is still 113 ballots, and increasing $\Delta_{2,best}$ by 112 ballots. The manipulation with $\Delta_{1,best} = 80$ and $\Delta_{2,best} = 201$ is successful, resulting in a new best manipulation size of 281. Reducing $\Delta_{2,best}$ and increasing $\Delta_{1,best}$ does not lead to a smaller manipulation, and we accept the shift of 80 and 201 ballots as our

[5] Where $\delta_i^m > \Delta_{i,j}$, we reset δ_i^m to $\lceil \Delta_{i,j}/\gamma \rceil$.

MOVE$_1$(S, $M_{current}$, k, N_d, α, γ)
1 $M_{best} \leftarrow \emptyset$, $S_{best} \leftarrow S$
2 **for** i in $1..k$ **do**
3 $M_1 \leftarrow M_{current}$
4 **while** true **do**
5 $\Delta_{i,1} \leftarrow \Delta_{i,1} - \delta_i^1$
▷ Consider M_1 only if it is smaller than the size of the current best, S_{best}
6 **if** $|M_1| \geq S_{best}$ **then break**
7 **if** manipulation M_1 is verified by SIM-STV **then**
8 $M_{best} \leftarrow M_1$, $S_{best} \leftarrow |M_1|$
9 $\delta_i^1 \leftarrow \gamma\, \delta_i^1$
10 **break**
11 **else if** $\delta_i^1 \equiv 1$ **then break else** $\delta_i^1 \leftarrow \lceil \frac{\delta_i^1}{\alpha} \rceil$
12 **return** M_{best}

MOVE$_2$(S, $M_{current}$, k, N_d, α, γ)
1 $M_{best} \leftarrow \emptyset$, $S_{best} \leftarrow S$, $M_2 \leftarrow M_{current}$
2 **while** true **do**
3 **if** $\sum_{i=1}^{k} \delta_i^2 \equiv 0$ **then break**
4 $\Delta_{i,2} \leftarrow \Delta_{i,2} - \delta_i^2$ for all $i \in \{1..k\}$
5 **if** $|M_2| \geq S_{best}$ **then break**
6 **if** manipulation M_2 is verified by SIM-STV **then**
7 $M_{best} \leftarrow M_2$, $S_{best} \leftarrow |M_2|$
8 $\delta_i^2 \leftarrow \gamma\, \delta_i^2$ for all $i \in \{1..k\}$
9 **break**
10 **else** Set all δ_i^2 that are smaller than γ to 0, and all remaining to $\lceil \frac{\delta_i^2}{\alpha} \rceil$
11 **return** M_{best}

MOVE$_3$(S, $M_{current}$, k, N_d, α, γ)
1 $M_{best} \leftarrow \emptyset$, $S_{best} \leftarrow S$
2 **for** i in $1..k$ **do**
3 $M_3 \leftarrow M_{current}$
4 **while** true **do**
5 $\Delta_{i,3} \leftarrow \Delta_{i,3} - \delta_i^3$
▷ Distribute decrease of δ_i^3 across other pairwise shifts
6 $\Delta_{j,3} \leftarrow \Delta_{j,3} + \max\left(0, \lceil \frac{\delta_i^3}{k-1} \rceil - 1\right)$ for $j \in \{1..k\} \setminus \{i\}$
7 **if** manipulation M_3 is verified by SIM-STV **then**
8 $M_{best} \leftarrow M_3$, $S_{best} \leftarrow |M_3|$
9 **break**
10 **else**
11 **if** $\delta_i^3 \equiv 1$ **then break**
12 $\delta_i^3 \leftarrow \lceil \frac{\delta_i^3}{\alpha} \rceil$
13 **return** M_{best}

Fig. 3: Algorithms for move types one to three, where: S denotes the size of the best found manipulation in the current iteration of local search; $M_{current}$ is the best found manipulation at the start of the current iteration; k is the number of additional candidates we wish to elect from our desired coalition; N_d is the number of candidates from our desired coalition originally elected; and $\alpha \geq 2$, $\gamma \geq 2$ are predefined constants.

new 'best found manipulation'. After applying MOVE_3, the step sizes associated with move types 1 and 2 are reset to their initial values.

After 9 iterations, we have reduced our overall manipulation size to 247 ballots with a MOVE_3. In the next iteration, we cannot reduce the size of this manipulation further and local search terminates. This process is repeated for different combinations of subsets in $\overline{\mathcal{W}}$ and $\overline{\mathcal{L}_d}$, returning the smallest found manipulation as our result. In this example, the smallest successful manipulation we can discover is 247 ballots.

3.2 Choosing a best combination of manipulations

Let $x_{i,k}$ denote a binary variable that takes on a value of 1 if we choose to apply a manipulation to election i that elects k additional candidates from our desired coalition d, and 0 otherwise. Let $|M_{i,k}|$ denote the size of the manipulation required to elect k additional candidates from d in election i. We formulate an integer program (IP), modelled as a knapsack problem, to select the best combination of manipulations that, when applied to their respective elections, realise a combined n-seat shift toward our coalition. Our objective is to minimise the total number of ballot changes required across all selected manipulations. We use s to denote the number of seats available in election i.

$$minimise \sum_{i} \sum_{k=1}^{min(s,n)} |M_{i,k}|\, x_{i,k} \tag{3}$$

subject to:

$$\sum_{i} \sum_{k=1}^{min(s,n)} k\, x_{i,k} = n \tag{4}$$

The constraint in Eqn 4 restricts the total number of seats shifted to our coalition, across the set of individual STV elections i, to n. Where our local search method was unable to find a manipulation that elects k additional favoured candidates to an election i, we fix $x_{i,k} = 0$. As we shall see in Section 4, there are a number of situations in which a k-seat shifting manipulation is not possible in a given election.

Example 5. For $\mathcal{E}_1$, the best found manipulation to award $k = 1$ extra seats to our coalition $d = \{\text{C, D}\}$ is 111 ballots in size. Awarding $k = 2$ extra seats to d requires 247 ballot changes, across all ballot replacement methods. For $\mathcal{E}_2$, 15 ballot changes are required to elect $k = 1$ more members from d (for IW, BTL, and ATL), and 121 ballots for $k = 2$ (using BTL). In the latter case, using IW and ATL result in a manipulation of 122 ballots. For our small 6-seat Senate, the best manipulation we can find to shift $n = 2$ seats to our coalition is to shift 2-seats in $\mathcal{E}_2$, with a cost of 121 ballots.

4 Case Studies

We use the 2016 and 2019 Australian Federal Senate elections as case studies. We have partitioned the set of parties taking part in these elections into two groups: conservative; and progressive. The conservative group includes parties such as the Liberal Party,

Table 3: For each of the individual STV elections forming part of the 2016 and 2019 Australian Federal Senate elections, we report the number of: seats available; candidates standing; and formal votes cast. We additionally state the quota, and number of candidates elected from our desired 'progressive' coalition d, for each election.

	2016 Senate Election					2019 Senate Election				
Region	**Seats**	$\|\mathcal{C}\|$	**Formal Votes Cast**	**Quota**	**Elected from d**	**Seats**	$\|\mathcal{C}\|$	**Formal Votes Cast**	**Quota**	**Elected from d**
ACT	2	22	254,767	84,923	1	2	17	270,231	90,078	1
NT	2	19	102,027	34,010	1	2	18	105,027	35,010	1
SA	12	64	1,061,165	81,629	7	6	42	1,094,823	156,404	3
VIC	12	116	3,500,237	269,250	6	6	82	3,739,443	534,207	3
QLD	12	122	2,723,166	209,475	5	6	83	2,901,464	414,495	2
NSW	12	151	4,492,197	345,554	5	6	105	4,695,326	670,761	3
WA	12	79	1,366,182	105,091	4	6	67	1,446,623	206,661	3
TAS	12	58	339,159	26,090	7	6	44	351,988	50,285	3

the Nationals, and One Nation. The progressive group contains parties such as the Australian Labor Party and the Greens. The conservative coalition attained a 4-seat majority in 2016, and a 2-seat majority in 2019. Consequently, we use the progressive group as our desired coalition d in our experiments, and seek to find as small as possible a manipulation to award 4, and 2 respectively, additional seats to candidates in d in the 2016 and 2019 elections. All experiments have been run with parameters $\gamma = 2$ and $\alpha = 4$.

Table 3 reports the number of candidates standing, seats available, and formal (valid) votes cast in each of the individual STV elections forming part of these two Senate elections. In addition, we report the quota and number of candidates elected from d.

We report in Table 4 the sizes of the smallest manipulations our local search approach was able to find to shift $k = 1, 2$ seats toward our favoured candidates in coalition d, in each state and territory STV election in 2019. A '–' indicates that no manipulation was found to achieve a given shift of seats. In the ACT and NT, for example, only 2 seats are available for election. In each case, 1 candidate from d has been elected to a seat in the original outcome. We can only award 1 additional seat to candidates in d. We report the number of ballot shifts required to shift 1, and 2, seats toward our favoured candidates when using the BTL, ATL, and IW ballot replacement methods. Overall, the IW method leads to smaller manipulations. Recall that the IW approach replaces ballots that favour an undesired winner with a below the line vote that preferences a favoured loser first, and each of the original winners from our desired coalition subsequently.

Table 5 states the sizes of the smallest manipulations our local search approach could find to shift $k = 1..4$ seats toward our favoured coalition d, in each state and territory STV election in 2016. We use the IW method of replacing ballots for each election. As in 2019, we can only award 1 additional seat to candidates in d in the ACT and NT. In SA and TAS, we were unable to find a manipulation that awarded 4 additional seats to candidates in d. Both Tables 4 and 5 show that the degree of manipulation required to shift k seats to desired candidates increases significantly as k increases.

We apply the IP of Section 3.2 to the available manipulations for 2019, listed in Table 4. The coefficients of our objective are obtained from reported manipula-

Table 4: Smallest manipulations found to elect 1 to 2 additional members of a centre-left leaning coalition of parties, in each state/territory for the 2019 Australian Federal Sentate election. For each region, the election quota, and number of ballot changes required to realise the desired change, are stated for each method of forming new ballots. We additionally state the number of ballot changes as a percentage of formal votes cast.

1 additional seat to desired coalition

Region	**Quota**	**Ballot Shifts Required**								
ACT	90,078	BTL	**12,938**	(4.8%)	ATL	**12,938**	(4.8%)	IW	**12,938**	(4.8%)
NT	35,010	BTL	**14,697**	(14%)	ATL	14,922	(14.2%)	IW	**14,697**	(14%)
SA	156,404	BTL	**50,535**	(4.6%)	ATL	50,695	(4.6%)	IW	**50,535**	(4.6%)
VIC	534,207	BTL	**126,906**	(3.4%)	ATL	127,068	(3.4%)	IW	**126,906**	(3.4%)
QLD	414,495	BTL	**56,913**	(2%)	ATL	**56,913**	(2%)	IW	58,605	(2%)
NSW	670,761	BTL	**296,472**	(6.3%)	ATL	297,389	(6.3%)	IW	**296,472**	(6.3%)
WA	206,661	BTL	**108,915**	(7.5%)	ATL	**108,915**	(7.5%)	IW	**108,915**	(7.5%)
TAS	50,285	BTL	**19,824**	(5.6%)	ATL	20,399	(5.8%)	IW	**19,824**	(5.6%)

2 additional seats to desired coalition

Region	**Quota**	**Ballot Shifts Required**								
ACT	90,078	BTL	–		ATL	–		IW	–	
NT	35,010	BTL	–		ATL	–		IW	–	
SA	156,404	BTL	177,554	(16.2%)	ATL	177,730	(16.2%)	IW	**177,504**	(16.2%)
VIC	534,207	BTL	559,035	(14.9%)	ATL	558,734	(14.9%)	IW	**558,521**	(14.9%)
QLD	414,495	BTL	370,091	(12.8%)	ATL	370,046	(12.8%)	IW	**353,692**	(12.8%)
NSW	670,761	BTL	835,217	(17.8%)	ATL	835,180	(17.8%)	IW	**832,314**	(17.8%)
WA	206,661	BTL	**294,005**	(20.3%)	ATL	**294,005**	(20.3%)	IW	**294,005**	(20.3%)
TAS	50,285	BTL	**53,617**	(15.2%)	ATL	55,275	(15.7%)	IW	54,295	(15.4%)

tion sizes. We use the smallest manipulation discovered for each state and territory, across the different ballot replacement methods. For example, $|M_{ACT,1}| = 12,938$ and $|M_{SA,2}| = 177,504$. The least cost way to shift 2 seats to our desired coalition is to shift 1 seat in ACT, with 12,938 ballot manipulations (4.8% of cast formal votes), and 1 seat in the NT, with 14,697 ballot changes (14% of the cast formal votes). The nature of the elected Senate in 2019 could have significantly changed with a change in 27,635 votes. If we chose to minimise the percentage of formal ballots cast in any manipulated election, in place of the total number of ballots changed, we would instead shift 1 seat in QLD (56,913 manipulations, 2% of formal votes) and 1 seat in VIC (126,906, 3.4% of formal votes). The total manipulation size is significantly larger, at 183,819 ballots, yet it involves a smaller percentage of changes (a maximum of 3.4%).

In 2016, the least cost combination of manipulations to shift 4 seats to our coalition d are: a 1 seat shift in SA, with 1,772 manipulations (0.17%); a 1 seat shift in the NT, with 11,245 manipulations (11%); a 1 seat shift in NSW, with 12,313 manipulations (0.27%); and a 1 seat shift in WA, with 14,678 manipulations (1.1%). The nature of the elected Senate in 2016 could have significantly changed with a change in 40,008 votes.

Table 5: Smallest manipulations found to elect 1 to 4 additional members of a centre-left leaning coalition of parties, in each state/territory for the 2016 Australian Federal Senate election. For each region, the election quota, and number of ballot changes required to realise the desired change (using the IW method of forming new ballots) are stated. We additionally state the number of ballot changes as a percentage of formal votes cast.

Region	**Quota**	**Ballot Shifts Required**			
		1 seat	2 seats	3 seats	4 seats
ACT	84,923	18,836 (7.4%)	–	–	–
NT	34,010	11,245 (11%)	–	–	–
SA	81,629	1,772 (0.17%)	57,607 (5.4%)	132,576 (12.5%)	–
VIC	269,250	45,046 (1.3%)	181,770 (5.2%)	420,880 (12%)	682,348 (19.5%)
QLD	209,475	49,829 (1.8%)	139,196 (5.1%)	354,475 (13%)	573,357 (21.1%)
NSW	345,554	12,313 (0.27%)	149,046 (3.3%)	386336 (8.6%)	731,280 (16.3%)
WA	105,091	14,678 (1.1%)	79,308 (5.8%)	161,963 (11.9%)	280,426 (20.5%)
TAS	26,090	21,692 (6.4%)	43,383 (12.8%)	65,698 (19.4%)	–

5 Conclusion

We have presented a local search heuristic that, in combination with an integer program, finds an upper bound on the number of ballot changes required to change the nature of the majority in an elected Senate. We have found that in two case study elections, a relatively small, but not insignificant, number of cast ballots need to be changed to shift the majority from a conservative coalition of parties to one that is more progressive. This number is a lot larger, however, than the number of ballot changes required to realise any change in outcome. For example, the 2016 results in Tasmania were very close, requiring only 71 ballot changes to change the result [1].

References

1. Blom, M., Conway, A., Stuckey, P.J., Teague, V.J.: Did that lost ballot box cost me a seat? computing manipulations of stv elections. In: IAAI (2020)
2. Blom, M., Stuckey, P.J., Teague, V.: Computing the margin of victory in preferential parliamentary elections. In: EVote-ID. LNCS, vol. 11143, pp. 1–16 (2018)
3. Blom, M., Stuckey, P.J., Teague, V.: Towards computing the margin of victory in STV elections. INFORMS Journal of Computing **31**(4), 636–653 (2019)
4. Conway, A.: Australian federal senate simulator. `https://github.com/SiliconEconometrics/PublicService` (2019), accessed: August 2019
5. Miragliotta, N.L.: Little differences, big effects: An example of the importance of choice of method for transferring surplus votes in PR-STV voting systems. Representation **41**, 15–24 (2004)
6. Australian federal senate election rules. `www.austlii.edu.au/au/legis/cth/consol_act/cea1918233/s273.html` (2019), accessed: Aug 2019
7. Weeks, L.: Tolerable Chance or Undesirable Arbitrariness? Distributing Surplus Votes Under PR-STV. Parliamentary Affairs **64**, 530–551 (2011)
8. Xia, L.: Computing the margin of victory for various voting rules. In: Proceedings of the 13th ACM Conference on Electronic Commerce. pp. 982–999. EC '12, ACM, New York, NY, USA (2012)

Random errors are not necessarily politically neutral

Michelle Blom[1 [0000−0002−0459−9917]], Andrew Conway[[0000−0001−6277−2442]], Peter J. Stuckey[2 [0000−0003−2186−0459]], Vanessa Teague[3,4 [0000−0003−2648−2565]], and Damjan Vukcevic[5,6 [0000−0001−7780−9586]]

[1] School of Computing and Information Systems, University of Melbourne, Parkville, Australia
[2] Faculty of Information Technology, Monash University, Clayton, Australia
[3] Thinking Cybersecurity Pty. Ltd.
[4] College of Engineering and Computer Science, Australian National University
[5] School of Mathematics and Statistics, University of Melbourne, Parkville, Australia
[6] Melbourne Integrative Genomics, University of Melbourne, Parkville, Australia

Abstract. Errors are inevitable in the implementation of any complex process. Here we examine the effect of random errors on Single Transferable Vote (STV) elections, a common approach to deciding multi-seat elections. It is usually expected that random errors should have nearly equal effects on all candidates, and thus be fair. We find to the contrary that random errors can introduce systematic bias into election results. This is because, even if the errors are random, votes for different candidates occur in different patterns that are affected differently by random errors. In the STV context, the most important effect of random errors is to invalidate the ballot. This removes far more votes for those candidates whose supporters tend to list a lot of preferences, because their ballots are much more likely to be invalidated by random error. Different validity rules for different voting styles mean that errors are much more likely to penalise some types of votes than others. For close elections this systematic bias can change the result of the election.

1 Introduction

We investigate the effects of random errors on election outcomes, in the context of preferential elections counted using the Single Transferable Vote (STV). It is often assumed that random errors (whether from human or manual counting) are unimportant because they are likely to have nearly equal effects on all candidates. In this paper we show that this is not the case, using simulated random errors introduced into real STV voting data. In some cases, this introduces a systematic bias against some candidates.

Random errors have a non-random effect because real votes are not random. Voters not only express different preferences, but express them in a different way, according to whom they choose to support.

In STV, some candidates are elected mainly on the strength of their party listing; others rely on gathering preference flows from other parties, or on their individual popularity relative to their party's other candidates. So when we look at the votes that contributed to the election of different candidates, we find that the types of votes chosen by their supporters may be very different. Hence a random error that affects different types of votes differently introduces a systemic change in the election result.

One obvious kind of error is to misrecord a number. Usually, this either invalidates the ballot completely, or invalidates preferences below the error. The more preferences there are on a ballot, the more likely that at least one of them is misrecorded. So as a general rule, candidates that are more dependent on later preferences or long preference lists are more severely disadvantaged by random errors.

Although these results are significant, and need to be taken into account for close contests, we find that reasonable error rates produce changes in only very few elections, which (so far) correspond only to those that are obviously very close. It is possible for STV elections to have hidden small margins, but this seems to be uncommon—in almost all the elections we simulated, no plausible error rate produced a change in outcome. Typical random error rates will affect election results when the election is close, but are not expected to do so when the election is not close.

We do not consider the errors necessary to alter the election result in a targeted way by altering specific carefully chosen votes—they would obviously be much smaller. Hence the results of this paper apply to random errors, but not deliberate electoral fraud.

The remainder of the paper is organized as follows. In the next section we explain STV elections, in particular in the case of Australian Senate elections, and discuss how the votes are digitised and counted. In Section 3 we describe our experiment design and introduce the three error models we explore. In Section 4 we provide a number of different approaches to estimate the likely error rate that occurs for Australian Senate elections. In Section 5 we examine the result of applying simulated errors to Australian Senate elections and discuss how these errors can change the result of the election. Finally in Section 6 we conclude.

2 Background on STV counting

2.1 The Single Transferable Vote (STV) counting algorithm

STV is a multi-winner preferential voting system. Candidates compete for s available seats. A candidate is awarded a seat when their tally reaches or exceeds the quota, Q, defined as a function of the number of ballots cast in the election, $|\mathcal{B}|$, and the number of seats, s. One popular definition is the Droop quota,

$$Q = \left\lfloor \frac{|\mathcal{B}|}{s+1} \right\rfloor + 1.$$

When a voter casts a ballot in one of these STV elections, they have the option of voting ‘above the line’ or ‘below the line’. Figure 1 shows an example of a ballot for a simple STV election in which candidates from three parties are competing for s seats. Each party or group of independents fielding candidates in the election have a box sitting ‘above the line’ (ATL). A voter may rank these parties and groups by placing a number in their corresponding box (Figure 1a). Alternatively, a voter may rank individual candidates by placing a number in their box, below the line (BTL) (Figure 1b).

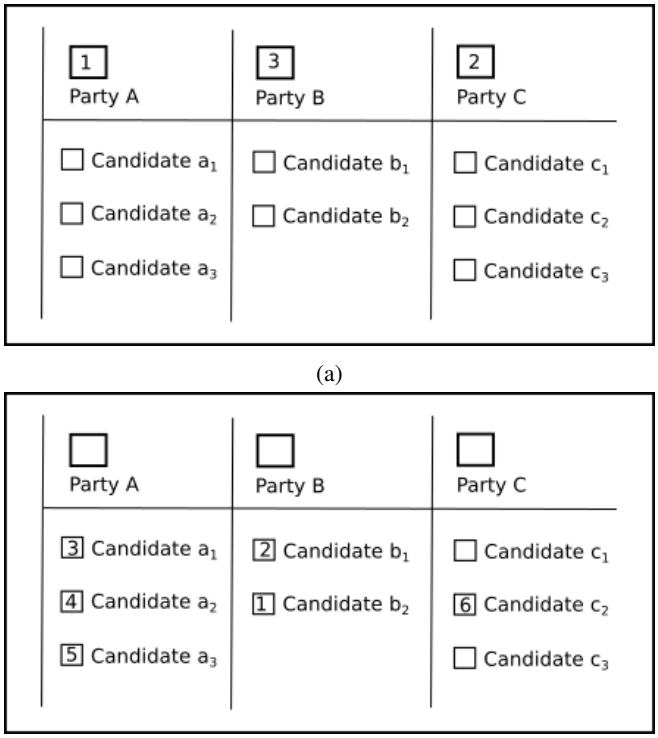


Fig. 1: **An example of two simple ballots for a 3-party STV election.** In (a), the voter has chosen to vote above the line, and in (b) they have voted below the line.

Tabulation starts by giving each candidate all the below-the-line ballots in which they have been ranked first. ATL ballots are awarded to the first candidate listed under the party that has been ranked first. For example, a ballot in which Party A has been ranked first sits in the first preference pile of candidate a_1. A BTL ballot in which candidate b_2 is ranked first sits in that candidate's first preference pile. Each ballot is assigned a weight, starting at 1, that changes as counting proceeds. The tally of a candidate is the sum of the weights of ballots sitting in their tally pile, possibly with some rounding.

Counting proceeds by awarding a seat to all candidates whose tallies have reached or exceeded Q. Their *surplus*—their tally after subtracting Q—is distributed to remaining eligible candidates. A candidate is eligible if they have not been eliminated, and their tally has not reached a quota's worth of votes. The ballots sitting in an elected candidate's tally pile are re-weighted so that their combined weight is equal to the candidate's surplus. These ballots are then given to the next most-preferred eligible candidate on their ranking. The ATL ballot in Figure 1a is given to candidate a_2 if a_1 is elected to a seat. If neither a_2 or a_3 are eligible, the ballot then moves to candidate c_1. The BTL ballot in Figure 1b is given to candidate b_1 if b_2 is elected or eliminated.

If no candidate has reached a quota, the candidate with the smallest tally is eliminated. The ballots in their tally pile are distributed to the next most-preferred eligible candidate in their rankings at their current weight.

Counting proceeds in rounds of election and elimination until all s seats are filled, or until the number of candidates that remain eligible equals the number of remaining seats. In this setting, each of the remaining candidates is awarded a seat.

2.2 Australian vote digitisation in practice

Australians cast their votes on paper ballots. The Australian Electoral Commission (AEC) digitises the preferences in a hybrid manual and automated process. Precise details about this process are unavailable, but most ballots seem to receive both automated digitisation and secondary human data entry. (Ballots that are judged blank are not re-examined.) It is possible that manual data entry is performed on ballot papers.[7] Other pamphlets suggest that only the images, not the paper ballots, are used.[8]

An automated system then checks, for each ballot, whether the automated digitisation matches the human interpretation. Obviously this does not defend against software errors or deliberate manipulation, particularly downstream of the process, but it probably does produce reasonably low random error rates, assuming that the human errors are not highly correlated with the errors of the automated system.

Ballots are required to have a minimum number of preferences before they are considered valid; such ballots are referred to as *formal* ballots. In the 2016 and 2019 elections, a BTL formal vote must have every preference from 1 to 6 inclusive present exactly once; an ATL formal vote requires the preference 1 to be present exactly once and a formal BTL vote not to be present. According to the information about the digitisation processes mentioned above, non-blank informal ballots seem to get a second human inspection automatically.

The AEC publishes on their website the complete digitised preferences for all Senate votes, excluding blanks and votes judged to be informal.

In summary, the published data could differ from the actual ballots for many reasons:

- random errors that match in both the automated and human digitisation process,
- random errors that occur in either the automated or human digitisation process, and are endorsed rather than corrected by the reconciliation process,
- erroneous exclusion of ballot papers judged to be informal,
- accidental alterations, duplicates or omissions caused by software bugs,
- deliberate manipulation by malicious actors, either of the images (before digitisation) or of the preference data (from digitisation to publication).

[7] https://www.aec.gov.au/Voting/counting/files/css-integrity.pdf
[8] https://www.aec.gov.au/Voting/counting/files/senate-count.pdf

Our investigation does not apply to the last two kinds of errors, which could be introduced in a non-random way that worked for or against a particular candidate. It does apply to the errors that are random. In particular, we show that digitisation errors that randomly cause some ballots to be judged informal can impact candidates differently.

3 Experimental design

Our analysis is performed on the AEC's published data for the 2016 and 2019 Australian federal elections for the Senate, i.e. the output of the process described in Section 2.2. Ideally our analysis would be based upon the actual marks that voters made on their ballots, or even what they intended to make, and the comparison with the AEC's output. However, these data are not available. Instead, we use the AEC's output as the 'actual' ballot data, and add simulated errors.

3.1 Analysis code

For logistical reasons, and to make it easy for anyone to replicate this experiment, we extract those preferences that are actually considered valid in the election. If a number is absent or repeated in the preference marks, then it and all subsequent preferences are disregarded. We have made available a standardised ".stv" file format based on the data published by the AEC[9]. This common format does unfortunately mean that we lose some (invalid) marks that could conceivably have become valid when we added new random errors, or which could, through errors, invalidate earlier preferences.

We used the Java pseudo-random number generator `java.util.Random` to generate random numbers, and ensured that different executions used different seeds. Our code is available for download[10].

3.2 Error models

We simulate the effect of errors by making random changes to the votes. We are not certain exactly what "random" failures in the scanning process would be, so we have devised three different models for simulated errors, in increasing order of complexity and plausibility. The first models an error where, somewhere in the list, something goes wrong that invalidates the rest of the preference list. The second models an error in which a digit is randomly misread as another digit, chosen uniformly. The final model recognises that some misreadings are much more likely than others—for example, a 3 is more likely to be confused with an 8 than a 1—so we use a model that includes a specific error probability for each digit and each potential misreading.

Each model applies to a valid list of preferences and treats either each number or each digit separately with random errors chosen independently.

1. For each preference, with probability ϵ, truncate the list at that preference.

[9] See the downloads section for each election at: https://vote.andrewconway.org
[10] https://github.com/SiliconEconometrics/PublicService

2. For each digit, with probability ϵ, replace that digit with a digit uniformly chosen from $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, which may be the original digit.
3. Start with a table of pairwise error ratios for digits such as Table 1 (that is, the probability that a certain digit is mistranscribed into a certain other digit). For each digit, change it into a different digit with the probability given in the table.

Note that in all three models, the probability of at least one error on the ballot increases with the number of preferences listed on the ballot. We are primarily motivated by machine errors, so per-digit or per-number random errors seem plausible, but it is worth noting that other errors might be important too, such as models that considered that some voters (those with bad handwriting) were much more likely to have their vote misinterpreted than others.

After applying errors, formality rules are checked again, reducing the number of ballots considered for the election.

4 What is a realistic error rate?

As far as we know, there are no publicly available results from any rigorous estimate of Senate scanning random errors in Australia. However, there are several independent estimates, which give us a per-digit error rate ranging from 0.01% to 0.38%. We define an error to be a discrepancy between the paper ballot and the electronic preference list output at the end of the process.

4.1 Using data from the Australian Electoral Commission

As far as we know, the AEC does not conduct, nor allow anyone else to conduct, a large random sample of Senate ballots for comparison between electronic and paper records. However, an Australian National Audit Office report[11] describes a process for gaining an estimate from a small sample. This process was conducted by AEC officials.

- A batch of 50 ballot papers was randomly selected and then six ballot papers from that batch were reviewed;
- Compliance inspectors recorded the first six preferences from the physical ballot paper on a checklist;
- Verification officers compared the preferences recorded on the checklist against those on the scanned image of the ballot paper and those in the related XML file;
- The IT security team compiled, investigated and reported on the findings.

The compliance inspection report outlined that a total of 1,510 ballot papers were inspected and 4 processing errors were identified. This seems to indicate an error rate of less than 0.3% per ballot. Although it wasn't recorded how many preferences were

[11] https://www.anao.gov.au/work/performance-audit/aec-procurement-services-conduct-2016-federal-election

on each ballot, it seems to indicate a very small per-digit error rate. However, a careful reading of that experimental description shows that the officials verified only the numbers from 1 to 6. Errors in later preferences were ignored. So this estimate may substantially underestimate the overall rate of error.

To estimate the per-digit error rate implied by these data, we assumed that all of the 1,510 ballot papers that were inspected had six preferences marked on them, giving a total of 9,060 digits. We also assumed that the 4 'processing errors' were each a single-digit error. This gave a per-digit error rate of 0.04%, with a 95% confidence interval of (0.01%, 0.11%).

In reality, some proportion of these ballot papers were likely to be informal and have fewer than six preferences marked. Adjusting the above assumptions based on reported rates of informality by the AEC[12] had negligible impact on these estimates.

4.2 Informal experiment

For the 2019 federal election, we conducted an informal experiment amongst 15 of our colleagues to get a rough estimate of the 'end to end' accuracy of the Senate vote digitisation process. Each of our colleagues decided on their Senate vote ahead of the election and made a private record of it for later comparison. On polling day, they each carefully completed their Senate ballot paper in accordance with their planned vote. After the election, it was possible to compare these against the electronic file of ballots published by the AEC. Each of our colleagues searched for a vote that matched their own vote either exactly or very closely.

All of our colleagues voted below the line in Victoria. Due to the very large number of possible ways to vote below the line, each of their votes was extremely likely to be unique. In addition, the electronic file from the AEC also recorded the polling booth for each ballot. These two facts together allowed each of our colleagues to easily identify their own ballot paper in the file and be confident that it was indeed their own. This was true even if the match were not exact, since the next 'closest' matching ballot would typically vary substantially from each person's private record.

Of our 15 colleagues, 12 found ballots in the file that exactly matched their own records. This indicates perfectly accurate digitisation. The remaining 3 found a mismatch: each of them had a single one of their preferences recorded differently in the file than in their private record. These mismatches could be due to an error in the AEC digitisation process or to a transcription error on the part of our colleagues. However, they do give us at least a rough estimate of accuracy.

What per-digit error rate does this imply? We use the following assumptions: a) Each ballot had votes below the line; b) All boxes below the line were numbered; c) All of the reported errors were for a single digit. These assumptions maximise the number of possible digits and minimise the number of errors, and thus will give the lowest possible error rate estimate. There were 82 candidates for Victoria. This gives $9 + 73 \times 2 = 155$

[12] For example, https://www.aec.gov.au/Voting/Informal_Voting/senate/

Table 1: **Pairwise error digit rates.** The entry for row x and column y gives the percentage chance of (mis)recognizing a digit y as a digit x. A dash '$-$' indicates less than 0.01% chance of misrecognition.

		Actual									
	Digit	0	1	2	3	4	5	6	7	8	9
Predicted	0	99.22	–	0.08	0.02	0.10	0.04	0.14	–	0.06	0.20
	1	–	98.75	0.14	–	–	–	–	0.40	0.04	0.08
	2	0.12	0.28	99.56	0.24	–	–	–	0.18	0.02	0.10
	3	–	–	0.22	99.50	–	–	–	0.24	0.14	0.22
	4	0.16	0.16	–	–	98.65	0.08	0.10	–	0.12	0.30
	5	–	0.02	–	–	–	99.52	0.22	0.10	0.18	0.12
	6	0.10	0.12	–	–	0.06	0.08	99.48	–	0.14	–
	7	0.08	0.42	–	0.16	–	0.02	–	98.90	–	0.38
	8	0.10	0.06	–	–	0.48	–	–	–	99.16	0.26
	9	0.22	0.20	–	0.08	0.72	0.26	0.06	0.18	0.14	98.34

digits per ballot, which is $155 \times 15 = 2,325$ digits in total. Out of these, we have 3 single-digit errors. These give a per-digit error rate of 0.13%, and a 95% confidence interval of (0.03%, 0.38%). The error rate here captures any errors either by a voter or by the digitisation process, so it provides a rough upper bound on the latter's error rate.

4.3 What is the state of the art in digit recognition error rate?

Accurately recognizing handwritten digits by computer is an important consideration for many applications where data crosses from the physical world into the digital. The MNIST (Modified National Institute of Standards and Technology) database is a large database of handwritten digits that is commonly used for training image processing systems. The database consists of digits written by high school students and American Census Bureau employees, and normalised to be represented as grayscale images of size 28×28 pixels. The current state of the art approach [1] to this dataset has an error rate of 0.18%.[13] Care must be taken with this result, which is on a well studied and well curated data set. While Australian ballot papers have boxes marked where each number should be filled in, not all digits written in practice fall completely within the box. Nevertheless, this gives an accurate lower bound on pure computer-based digit recognition accuracy. The AEC process involves human inspection which means that it may be able to achieve better overall digit recognition accuracy.

The errors in digit recognition are not uniform: some digits are easier to confuse, for example 1 and 7. Most work on digit recognition does not publish the cross-digit confusion rates. Table 1 gives a confusion table showing the percentage of each actual digit versus its predicted value from experiments reported by Toghi & Grover [3]. The overall digit recognition error in this work is 0.89%, which is substantially greater than the best results reported above.

[13] There is unpublished work claiming 0.17%.

Table 2: **Counts of ballot papers with repeated and missed preferences.** Tasmanian ballots with BTL marks, 2016.

Preference	1	2	3	4	5	6	7	8	9	10	11	12	13
Ballots with preference repeated	573	385	303	231	212	211	492	494	542	372	256	250	122
Ballots with preference skipped	240	43	54	49	45	37	130	133	134	193	203	45	44

4.4 Analysing the election data (NOT simulations) to infer the error rate

We only have the reported ballots, not the ones that were ruled informal. (Except of course we cannot distinguish human mistakes from scanning errors.) Errors that make the vote informal are hidden.

Recall that the formality rules require at least 6 unambiguous preferences below the line, and that informal votes are not reported. We can estimate the number of hidden informal votes by observing the erroneous but formal ones. We use the number of repeated or missing numbers greater than 6 to approximate the number of repeated or missing numbers less than or equal to 6.

Table 2 shows the data, for BTL votes cast in Tasmania for the 2016 Senate election. The first column is the preference p on the ballot. The second column is the number of ballot papers that contain p more than once. The final column shows the number of ballots missing that preference, showing preference $p-1$ and $p+1$ but not p. A 0 is not required for $p = 1$. Note that there is a sudden drop at 12 because voters were instructed to list at least 12 preferences, so many people listed exactly 12. If the 12th preference was miswritten or misrecorded, then it did not count in our table (there being no 13).

There would be no informal BTL ballots at all, and perfect zeros in the first 6 rows of Table 2, except for one special formality rule: if there is *also* a valid ATL vote present on the same ballot paper, then it is counted instead, and both the valid ATL vote and the invalid BTL markings are reported in the final database. Hence we expect that the numbers in the first 6 rows are only a small fraction of the ballots rendered informal by either human or scanning errors. There is a sudden increase at the 7th preference, because BTL votes with a repeated or omitted 7th preference are still included in the tally, as long as their first 6 preferences are unambiguous.

There are 97,685 published votes with BTL markings. Most of these were valid BTL votes but some were only published because they had valid ATL votes as well. The most representative preferences are probably 7 to 9, being single digits whose count is not artificially suppressed due to repetitions in them causing the BTL vote to be informal and thus usually not published. For these preference numbers, the observed repetitions are on the order of 0.5%. This doesn't prove that the scanning process introduces errors at a rate of 0.5% per digit, because they could be caused by voter error. It could also underestimate the scanner error rate because it includes only those not rendered informal. Nevertheless this provides an estimate of voter plus process error.

5 Results

5.1 Results from truncation and digit error models

We simulated counts with errors using the ballot data for all 8 states and territories from both the 2016 and 2019 Senate elections. We used both the truncation and digit error models, across a wide range of error probabilities. For any given choice of model and error probability, we simulated 1,000 elections (each with their own random errors under that model).

For error rates between 0% and 1%, the only election for which we observed any change in the elected candidates was for Tasmania in 2016. This election was somewhat unusual in three ways. First, it was a very close election, with the difference in tallies between the final two remaining candidates, Nick McKim and Kate McCulloch, being only 141 votes. For comparison, 285 votes were lost due to rounding. Second, there was a popular labor candidate, Lisa Singh, who won a seat despite being placed fourth on the party ticket, and the candidate above her not winning a seat. This means she received many BTL votes specifically for her, rather than relying on ATL votes for the party. Finally, the 2016 election was a double dissolution, which means that twelve candidates were elected rather than the usual six.

In the real election, the 12th (final) candidate that was elected was Nick McKim. In our simulations, once we introduced a small amount of error we saw that a different candidate, Kate McCulloch, was sometimes elected instead. As we increased the per-digit error rate from 0% to 1%, we saw a complete shift from one candidate to the other, see Figure 2. The truncation error model led to the same outcome (data not shown).

5.2 Pairwise digit error model

We ran 1,000 simulations for Tasmania 2016 using the pairwise digit error model. Unlike the other models, we did not have a parameter to set but simply used the pairwise error rate matrix shown in Table 1. This model has an average per-digit error rate of 0.89%. Across the 1,000 simulations, we observed Kate McCulloch being elected 99.5% of the time, and Nick McKim for the remaining 0.5%. This is consistent with the simple per-digit error model, which also resulted in Nick McKim occasionally being elected when the per-digit error was comparable.

5.3 Sharp transitions

The fact that such a sharp transition happens from electing one candidate to another was initially surprising to us. Rather than simply ‘adding noise’ and leading to randomness in which candidates got elected, the noise seems to be leading to a systematic bias in favour of or against specific candidates. This behaviour can be seen more clearly as the error rate is increased to larger values (beyond values that would be plausible in practice), see Figure 2, where sharp transitions are visible also at 28%, 36%, 62%, 68%, 82%, 86% and 97%.

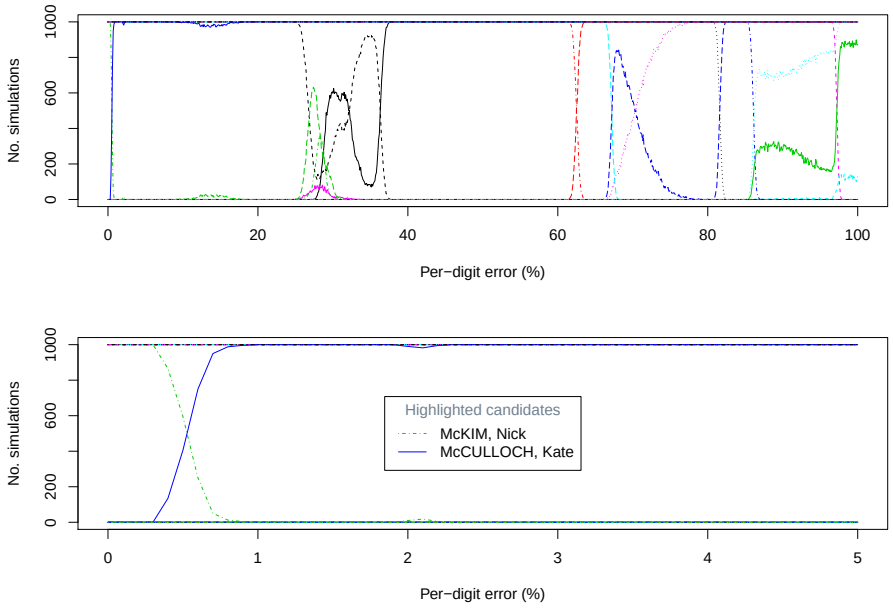


Fig. 2: **Changing election outcomes as a function of error rate, Tasmanian Senate election 2016.** The lower graph shows a complete reversal for a small error rate (about 0.5%), between the state in which McKim wins consistently (no error) and that in which McCulloch wins consistently (1% or greater error). The upper graph shows similar behaviour for larger error rates—with error rates of more than 20% there are sharp transitions between different election outcomes.

To investigate possible reasons for this, we looked at how individual ballots were affected by the simulated errors. Compared to the no-error scenario, two broad types of outcome are possible:

- The ballot becomes informal and is not counted. This will happen when it does not meet the formality requirements, e.g., does not have at least a single first preference above the line or consecutive preferences numbered 1 to 6 below the line.

- The ballot ends up exhausting before reaching a candidate. This will happen if the preference order becomes disrupted due to an error, which has the effect of truncating the preferences and not enabling the ballot to be counted in favour of any candidates further down the preference list.

We investigated these effects in the context of the Tasmanian 2016 election; we report on this in the next few sections. We found that the first type of effect was the dominant factor in determining the election outcome.

Table 3: **Partition of the Tasmanian 2016 ballots.** The number of ballots split by whether it is an above-the-line (ATL) or below-the-line (BTL) vote, and which candidate (if any) out of Kate McCulloch or Nick McKim is preferred over the other.

	McCulloch	McKim	Neither
ATL	73,975	97,331	72,468
BTL	17,066	42,170	36,149

5.4 Why random errors affect different candidates differently (Tasmania 2016)

We saw earlier that for small error rates, we have either Nick McKim (from the Australian Greens party) or Kate McCulloch (from the One Nation Party) elected as the final candidate. There were 339,159 formal ballots for this election. For each one, we looked at the preferences to see:

- whether it was an ATL or a BTL vote,
- which of the above two candidates (or their respective parties, if it was an ATL vote) was more highly preferred, or neither one.

Table 3 shows how the ballots split into these categories. The most important fact to note is the relative number of ATL and BTL votes in favour of each candidate: more than 80% of the ballots in favour of McCulloch were ATL votes, while for McKim it was less than 70%.

When errors are introduced, ballots that were BTL votes were much more likely to become informal. Figure 3 illustrates this: the larger the error rate, the greater the disparity in how many of the ATL or BTL ballots became informal. This on its own is enough to explain the systematic shift from McKim to McCulloch as error rates increase.

For more insight, we took a closer look at the simulations that used a per-digit error rate of 1%. For each ballot, we define the *formality rate* to be the proportion of simulations for which it remained formal. Figure 4 shows the distribution of the formality rate across different types of ballots. The left panel shows the clear disparity between ATL and BTL votes. This reiterates the difference we saw on average from Figure 3, but in addition we see that this disparity is very consistent across individual ballots (from the very little overlap for the ATL and BTL ballots).

When we further divided the ballots based on where in the preference list the voters placed their preferred candidate out of McKim or McCulloch, the distribution of formality rates was relatively consistent (right panel of Figure 4). This indicates that the major factor leading to McCulloch replacing McKim is simply the lower formality rate for BTL votes, after random errors were added, coupled with the fact that a larger proportion of ballots in favour of McKim were BTL votes.

For the less plausible larger errors, the sharp transitions came from new effects causing biases against major parties, who lost out as randomisation of preferences reduced their typical large first preference collection. This also caused major parties to not get

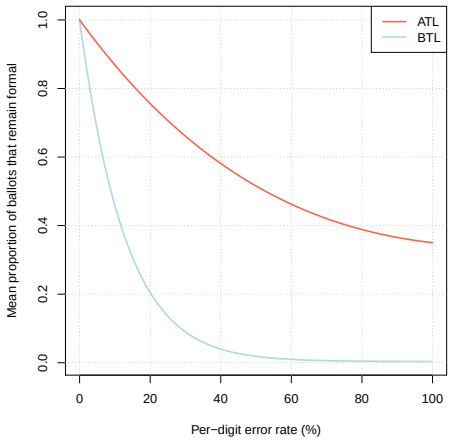


Fig. 3: **Effect of the per-digit error rate on the formality of votes.** The impact on above-the-line (ATL) and below-the-line (BTL) votes are shown separately.

multiple candidates elected in the first counting round, which meant that major party candidates low down on the party ticket tended to get eliminated before they could get preferences passed on to them, as they were reliant on BTL votes to avoid being eliminated before the first candidates of minor parties who could get ATL votes.

5.5 Varying the formality requirements

The formality requirements differ for ATL and BTL votes. In particular, BTL votes require at least 6 consecutive preferences in order to be declared formal, whereas ATL votes only require a single preference. This is one reason why the formality rate for BTL is lower once errors are introduced.

We investigated whether changing the formality rules could ameliorate the systematic bias caused by the introduction of errors. Specifically, we varied the number of consecutive preferences required for a formal BTL vote, ranging from 1 (i.e. the same as ATL votes) to 9 (i.e. more stringent than the current rules).

Figure 5 shows the impact of these choices on how often McCulloch was elected instead of McKim. Making the formality requirement less stringent reduced the bias, and once the formality rules were aligned for ATL and BTL votes, the election result remained mostly unchanged even in the presence of errors.

5.6 Truncation of preferences

Other than causing ballots to become informal, errors can result in votes not being counted for certain candidates if the error truncates the preference order. Candidates who obtain more of their votes from later (higher-numbered) preferences should be more affected by such truncation.

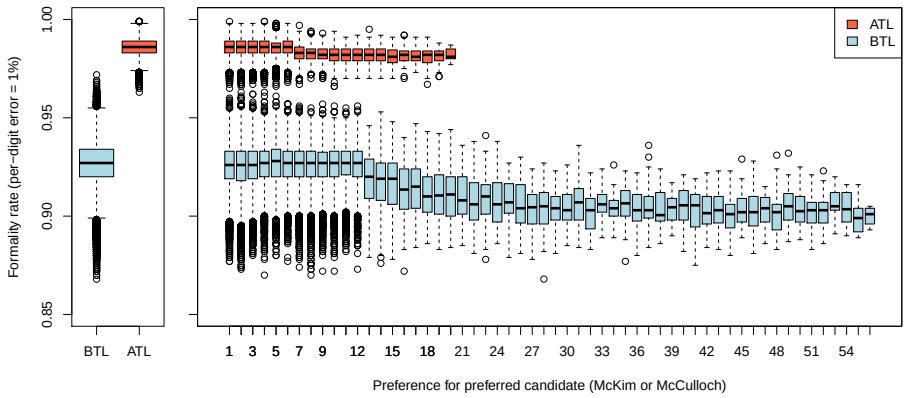


Fig. 4: **Formality rates for votes with random errors injected.** These are split by ATL/BTL (left panel) or by the position of the preferred candidate (right panel). Error rates vary greatly between ATL and BTL votes, but not much between preferences within those categories.

We investigated whether this might be occurring in our simulations. For each ballot, we compared the number of valid preferences before and after simulated errors. There was a clear signal of truncation: ballots that had around 60 valid preferences (which were all BTL) only had on average around 30 valid preferences remaining when the per-digit error was set to 1%. In contrast, ballots that had 10 valid preferences (irrespective of whether they were ATL or BTL) maintained almost 10 valid preferences on average.

While this extent of truncation is stark, it might not necessarily lead to any change in the election outcome because many of the later preferences might not actually be used during the election count.

In the case of the Tasmanian 2016 election, we looked at ballots in favour of each of McKim and McCulloch to see whether they tended to get their votes from earlier or later preferences. Figure 6 shows the distribution of these. Interestingly, we see that McCulloch relies more on later preferences than McKim. Therefore, it is McKim rather than McCulloch that should benefit from any truncation effect. This works in the reverse direction of the formality-induced bias described earlier, however the truncation did not act strongly enough to reverse that bias.

6 Concluding remarks

We are not aware of any previous study of the the effects of random errors in digitization on election outcomes. While there is a considerable body of work on margin of error for polling, there is little study of the effect of errors on elections. Richey [2] examines how ‘errors’ in voting can effect elections, but here the error is that a voter votes for a party that does not represent their best interests.

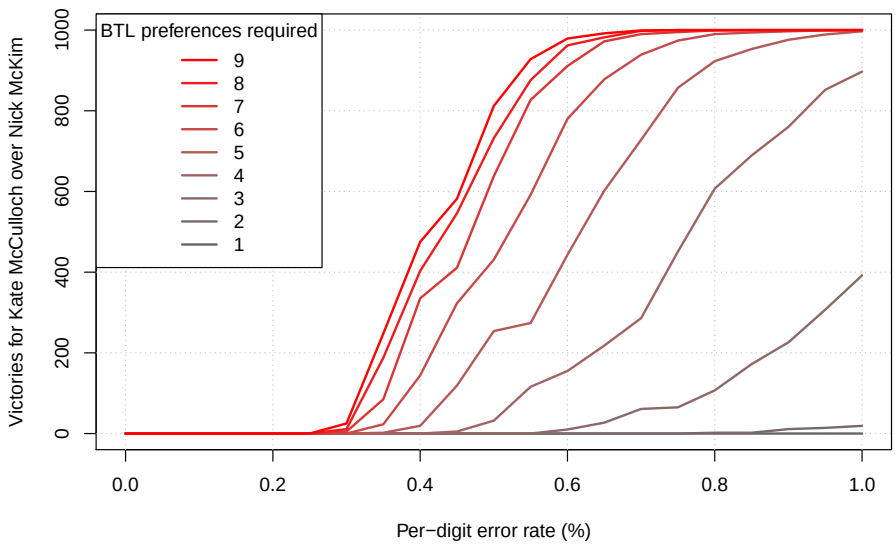


Fig. 5: **The effect of formality rules on election outcomes.** As the number of preferences required for a valid BTL vote increases, so does the rate at which BTL votes are excluded due to random errors. This produces a faster transition from one winning candidate to another as the error rate increases.

The previous section clearly demonstrates that random errors during counting do not necessarily lead to 'random' changes to election outcomes. We were very surprised by the sharp transitions in election results as error rates changed, illustrated in Figure 2. Systematic biases can arise due to interactions with the election rules.

For Australian Senate elections, a key factor is the formality requirements. BTL votes have more stringent requirements, which ends up creating a systematic bias against BTL votes in the presence of random errors. Candidates who rely on BTL votes (e.g. if they are relying on their individual popularity) will be more affected by random errors than those relying on ATL votes (e.g. via membership of their party). Changing the formality requirements to reduce the disparity between ATL and BTL votes also reduces this bias.

Candidates who rely on accumulating later preferences are more affected by random errors than candidates who rely primarily on their first-preference votes. However, this effect was much weaker than the bias induced by differences in formality requirements.

These results raise questions about how formality rules should be specified in order to be fair to candidates with different voting patterns. More relaxed formality rules could be applied which are less likely to have strong differences across different kinds of votes. For example, a BTL vote could be formal if the first 6 most preferred candidates are clear, even if they are not numbered from 1 to 6, e.g. a vote with preferences 1, 2, 4, 5, 6, 7 and no preference 3 still gives a clear ranking of the first 6 candidates.

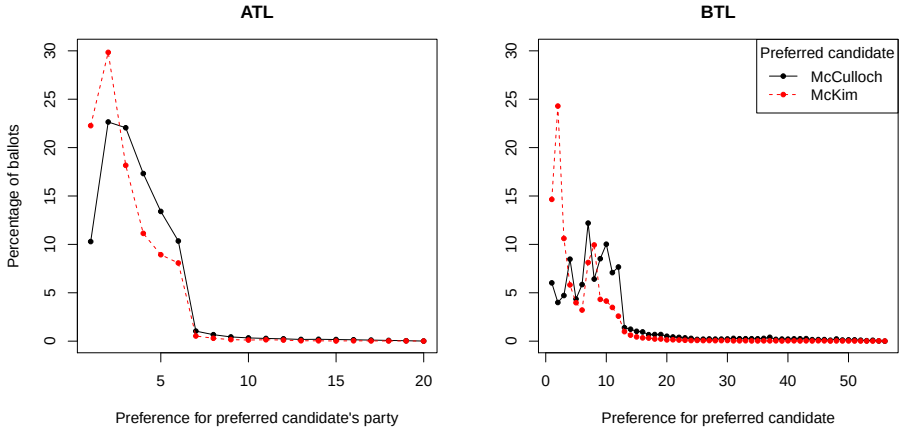


Fig. 6: **Histograms of preference number.** These are shown for a candidate or party depending on whether the votes are above or below the line.

In this paper we consider only Australian Senate elections with their particular ATL/BTL voting mechanism. Two lessons can be taken from this exercise to other forms of voting. First, if there are two or more forms of ballot and the rules for formality are different for these different forms of ballot, then random errors may affect the different forms differently, regardless of whether the voter can choose their form or different voters are assigned to different forms. This is applicable to any kind of election whether plurality voting or ranked voting. Second, considering elections where voters rank candidates with only one form of ballot, e.g. standard STV, Borda, or Condorcet elections, assuming the rules of formality are such that the ballot is truncated when the ranking becomes uninterpretable, then candidates relying on accumulating later preferences will be more affected by random errors than other candidates. But we do not have a real world case that illustrates that truncation errors alone lead to a change in a result.

Acknowledgements. We would like to thank our colleagues who participated in our informal experiment during the 2019 Australian federal election. Thanks also to Philip Stark for very valuable suggestions on improving the paper.

References

1. Dash, K.S., Puhan, N.B., Panda, G.: Unconstrained handwritten digit recognition using perceptual shape primitives. Pattern Analysis and Applications **21**, 413–436 (2018)
2. Richey, S.: Random and systematic error in voting in presidential elections. Political Research Quarterly **66**(3), 645–657 (2013), http://www.jstor.org/stable/23563171
3. Toghi, B., Grover, D.: MNIST dataset classification utilizing k-NN classifier with modified sliding window metric. CoRR **abs/1809.06846** (2018), http://arxiv.org/abs/1809.06846

Post-Quantum Anonymous Veto Networks

Jintai Ding[1][0000−0003−1257−7598], Doug Emery[1], Johannes Müller[2][0000−0003−2134−3099], Peter Y.A. Ryan[2], and Vonn Kee Wong[1]

[1] University of Cincinnati, Ohio, USA
jintai.ding at gmail.com
{ wongvg, emerydg } at mail.uc.edu
[2] SnT, University of Luxembourg, Luxembourg
{ johannes.mueller, peter.ryan } at uni.lu

Abstract. Anonymous veto networks (AV-nets), originally proposed by Hao and Zielinski (2006), are particularly lightweight protocols for evaluating a veto function in a peer-to-peer network such that anonymity of all protocol participants is preserved. Prior to this work, anonymity in all AV-nets from the literature relied on the decisional Diffie-Hellman (DDH) assumption and can thus be broken by (scalable) quantum computers. In order to defend against this threat, we propose two practical and completely lattice-based AV-nets. The first one is secure against passive and the second one is secure against active adversaries. We prove that anonymity of our AV-nets reduces to the ring learning with errors (RLWE) assumption. As such, our AV-nets are the first ones with post-quantum anonymity. We also provide performance benchmarks to demonstrate their practicality.

1 Introduction

In many jury or executive committee votings, certain results are only effective if supported by all members. Such votings, of which there are many instances in the real world, are called *veto votings*. Very recently, for example, the Supreme Court of the United States ruled that guilty verdicts for criminal trials be unanimous.[3] In order to protect each voter's freewill, veto votings are often required to not reveal any sensitive information except for the final result, i.e., whether or not at least one voter vetoed. Such votings are called *anonymous veto votings*.

Solutions for *electronic* anonymous veto protocols have a long history. In fact, David Chaum proposed the first such protocol, named *dining cryptographers network (DC-net)*, more than three decades ago [5, 6]. Since Chaum's original protocol returns the correct result if and only if an odd number of voters decides to veto, modifications of Chaum's protocol have been proposed to solve these and further issues (see, e.g., [12]).

However, DC-nets assume pairwise shared keys among the voters and their complexity is quadratic in the number of voters. In order to overcome these limitations, Hao and Zielinski introduced the concept of *anonymous veto networks*

[3] Ramos v. Louisiana, No. 18-5925, 590 U.S. __ (2020).

(AV-nets) (originally proposed in [14], with some extensions in [1]). In contrast to DC-nets, AV-nets are very lightweight, both regarding the number of rounds, computation, bandwidth and system complexity.

Anonymity of existing AV-nets from the literature relies on the hardness of the decisional Diffie-Hellman (DDH) problem. Since this problem could efficiently be solved by (scalable) future quantum computers, no AV-net with *post-quantum anonymity* has been proposed prior to our work. Unfortunately, as we will explain in Section 2.3, the fact that previous AV-nets are tailored specifically to the DDH problem makes it infeasible to transform them into AV-nets with post-quantum anonymity in a straightforward way.

Our contributions. We present the first completely lattice-based AV-nets. Our protocols are efficient and practically realizable. Anonymity of voters relies on the decisional ring learning with errors (RLWE) assumption. Using the RLWE assumption in our protocol is inspired from [18, 19] in which an RLWE analogue of the Diffie-Hellman key exchange was proposed. Our protocols do not require a central tallying authority; instead the voters themselves securely compute the final result. More precisely, we provide the following contributions:

1. We propose a 2-round lattice-based AV-net that is secure against *passive* (honest-but-curious) adversaries (Section 4). We first precisely describe this protocol (Section 4.1), then show that it produces the correct final result (Section 4.2), and that anonymity/privacy of the voters is guaranteed under RLWE if all but two voters are corrupted by a passive adversary (Section 4.3).
2. We propose a 4-round lattice-based AV-net that is secure against *active* (malicious) adversaries (Section 5). We first precisely the describe this protocol (Section 5.1), then show that the correctness of the final result can publicly be verified (Section 5.2), and that anonymity/privacy of the voters is guaranteed under RLWE if all but two voters are corrupted by an active adversary (Section 5.3).
3. We provide experimental performance benchmarks of our lattice-based AV-nets (Section 6).
4. We discuss the properties of the two lattice-based AV-nets as well as possible alternative approaches (Section 7).

We note that, in the remainder of this paper, we use the expressions "privacy" and "anonymity" interchangeably.

2 AV-Net by Hao and Zielinski

In this section, we first describe the original AV-net proposed by Hao and Zielinski [14] which provides anonymity under the DDH assumption. We then elaborate on why building AV-nets with lattice-based anonymity is challenging and requires careful attention.

2.1 Protocol description

The main idea behind the AV-net protocol by Hao and Zielinski [14] is the following one. The protocol is divided into an offline and an online phase. In the offline phase, the voters collaboratively generate certain related blinding elements, one individual element y_i for each voter V_i. In the subsequent online phase, voters can then decide to either veto or not. If V_i decides not to veto, then she raises y_i (as generated in the offline phase) to a specific integer s_i, and to a random integer r_i, otherwise. After that, all blinded choices are homomorphically aggregated. Furthermore, both in the offline and the online phase, zero-knowledge proofs (ZKPs) of knowledge are integrated to guarantee that voters choose their (otherwise malleable) messages pairwise independently.

The specific structure of the blinding elements $y_1, \ldots, y_m$ generated in the offline phase ensures that the result of the homomorphic aggregation equals 1 if and only if all voters choose "no veto". The technical mechanism behind this concept is based on the following result (details will become clear further below).

Lemma 1. *Let R be a commutative ring. Let $r_1, \ldots r_m$ be elements in R. Then the following equation holds true:*

$$\sum_{i=1}^{m} \sum_{j=1}^{i-1} r_i \cdot r_j = \sum_{i=1}^{m} \sum_{j=i+1}^{m} r_i \cdot r_j$$

Proof. See [14].

Let us now describe the AV-net protocol by Hao and Zielinski [14] with full technical details.

Protocol participants. The AV-net protocol is run among the following participants:

- Voters $\mathsf{V}_1, \ldots, \mathsf{V}_m$.
- Bulletin board B.

We assume that for each voter V_i, there exists a mutually authenticated channel between V_i and the bulletin board B.

Parameters. Let G be finite cyclic group of prime order q with generator g. We assume that the decisional Diffie-Hellman (DDH) assumption holds true in G, i.e., the following two distributions are computationally indistinguishable:

- (g^a, g^b, g^{ab}), where $a, b \xleftarrow{r} \mathbb{Z}_q$.
- (g^a, g^b, g^c), where $a, b, c \xleftarrow{r} \mathbb{Z}_q$.

Offline phase. Each voter V_i runs the following program:

1. $s_i \xleftarrow{r} \mathbb{Z}_q$
2. $h_i \leftarrow g^{s_i}$
3. $\pi_i^1 \leftarrow$ ZKP of knowledge of $\log_g h_i$
4. Publish (π_i^1, h_i)

After all voters have published their h_i's (equipped with valid ZKPs), each voter V_i (locally) computes her individual blinding element y_i as follows:

$$y_i \leftarrow \left(\prod_{j=1}^{i-1} h_j\right) \cdot \left(\prod_{j=i+1}^{m} h_j\right)^{-1}.$$

Online phase. Voter V_i computes her "encrypted" choice as follows:

1. If "no veto", then set $c_i \leftarrow y_i^{s_i}$.
2. If "veto", then choose $r_i \xleftarrow{r} \mathbb{Z}_q$, and set $c_i \leftarrow y_i^{r_i}$.
3. $\pi_i^2 \leftarrow$ ZKP of knowledge of $\log_{y_i} c_i$
4. Publish (π_i^2, c_i)

After all voters have published their c_i's (equipped with valid ZKPs), each voter (locally) computes the final result as follows:

$$\mathsf{res} \leftarrow \begin{cases} \text{no veto} & \text{if } \prod_{i=1}^{m} c_i = 1 \\ \text{veto} & \text{otherwise} \end{cases}.$$

2.2 Correctness and anonymity

We now describe why the AV-net by Hao and Zielinski is correct and provides anonymity under the DDH assumption. We focus on the case of passive adversaries; the ZKPs invoked ensure that the AV-net is also secure against active adversaries (see [14] for details).

Correctness. Let us first assume that all voters choose "no veto". Then, we have that

$$\begin{aligned}\prod_{i=1}^{m} c_i = \prod_{i=1}^{m} y_i^{s_i} &= \prod_{i=1}^{m} \left(\left(\prod_{j<i} h_j\right)\left(\prod_{j>i} h_j\right)^{-1}\right)^{s_i} \\ &= g^{\left(\sum_{i=1}^{m}\sum_{j<i} s_i s_j\right) - \left(\sum_{i=1}^{m}\sum_{j>i} s_i s_j\right)} = g^0 = 1\end{aligned}$$

holds true, where the second but last equality follows from Lemma 1. Conversely, assume that (at least) one voter vetoes, say voter V_l. Then, we have that

$$\prod_{i=1}^{m} c_i = y_l^{r_l} \cdot \prod_{i \neq l}^{m} c_i$$

is distributed uniformly at random in G. Hence, if $|G|$ is sufficiently large, then the probability that this product equals 1 is negligible.

Anonymity. Let V_i be an arbitrary (honest) voter. Assume that at least one further voter V_j is honest, too. Then, the sum $\sum_{j<i} s_j - \sum_{j>i} s_j$ is distributed uniformly at random in $\mathbb{Z}_q$. Hence, if V_i does not veto, then the triple

$$(h_i, y_i, c_i) = (g^{s_i}, g^{(\sum_{j<i} s_j)-(\sum_{j>i} s_j)}, g^{s_i \cdot ((\sum_{j<i} s_j)-(\sum_{j>i} s_j))})$$

is a DDH-triple, and otherwise a random triple

$$(h_i, y_i, c_i) = (g^{s_i}, g^{(\sum_{j<i} s_j)-(\sum_{j>i} s_j)}, g^{s_i \cdot r_i}).$$

Under the assumption that the DDH problem is intractable in G, it is not possible to distinguish between these two distributions.

2.3 Challenges for lattice-based anonymity

As we have seen in Section 2.2, the design of [14] is tailored specifically to reduce anonymity to the DDH-assumption. Therefore, if we want to design an AV-net whose anonymity reduces to a different (e.g., lattice-based) hardness assumption, then we have to adapt all technical details accordingly. This is even more challenging in the case of lattice-based anonymity: controling the noise of lattice-based cryptographic primitives is non-trivial and requires careful attention.

Furthermore, the original AV-net [14] includes ZKPs of knowledge to defend against active adversaries which choose their messages in relation to the honest voters' ones. Even though there exist efficient lattice-based ZKPs in the literature, these ZKPs are tailored to specific lattice-based primitives. Unfortunately, it is not immediately clear how to employ these primitives to construct a lattice-based AV-net. Therefore, we decided to construct an actively secure lattice-based AV-net without ZKPs altogether (Section 5).

3 Cryptographic Primitives

In this section, we introduce the cryptographic primitives that we later employ in our lattice-based veto protocols (Section 4 and 5). Throughout this paper, we use the following parameters and conventions:

- Let n be a power of 2.
- Let R be the cyclotomic ring $\mathbb{Z}[X]/f(X)$ where $f(X) = X^n + 1$.
- Let q be a prime such that $q \equiv 1 \mod 2n$.
- Let R_q be the quotient ring R/qR.
- Let the coefficients of a polynomial in R_q be in the interval $[-\frac{q-1}{2}, \frac{q-1}{2}]$.
- Let $\|\cdot\|$ be the ℓ_2-norm on R_q and $\|\cdot\|_\infty$ be the $\ell_\infty-$norm on R_q.
- Let m be an integer. (This will be the number of voters.)
- Let $\Lambda = \mathbb{Z}^n$.
- Let $\rho_\sigma(\mathbf{x}) = e^{-\pi\|\mathbf{x}\|^2/\sigma^2}$ be the Gaussian function on $\mathbb{R}^n$ with center at the zero vector and the parameter σ.

- Let $\rho_\sigma(\Lambda) = \sum_{\mathbf{x}\in\Lambda} \rho_\sigma(\mathbf{x})$ be the discrete integral of ρ_σ over Λ.
- Let $D_{\Lambda,\sigma}$ be the discrete Gaussian distribution over Λ with center at zero vector and parameter σ. For all $\mathbf{y} \in \Lambda$, we have $D_{\Lambda,\sigma}(\mathbf{y}) = \frac{\rho_\sigma(\mathbf{y})}{\rho_\sigma(\Lambda)}$.
- Let χ be the discrete Gaussian distribution $D_{\mathbb{Z}_q^n,\sigma}$.

The *decisional ring learning with errors (RLWE) problem* is about determining whether a list of polynomial pairs $(a_i, b_i) \in R_q \times R_q$ were generated uniformly at random or were constructed such that a_i is chosen uniformly at random while $b_i = a_i \cdot s + e_i$, where $s \in R_q$ is the secret and $e_i \leftarrow \chi$ is the error.

The parameters are chosen to satisfy the theorem below:

Theorem 1 ([21]). *For n, R, q and β as defined above, there is an efficiently samplable distribution χ over R with $\Pr[\|x\| > \beta : x \leftarrow \chi] \leq \mathsf{negl}(n)$, such that if there exists an efficient algorithm that solves $RLWE^{(m)}_{n,q,\chi}$, then there is an efficient quantum algorithm for solving $n^{2.5}(q/\beta)(nm/\log(nm))^{1/4}$-approximate worst-case SVP for ideal lattices over R.*

We recall some useful lemmas.

Lemma 2 ([17], Lemma 2.5). *For $\sigma > 0$, $r \geq 1/\sqrt{2\pi}$, $\Pr[\|x\| > r\sigma\sqrt{n} : x \leftarrow D_{\mathbb{Z}^n,\sigma}] < (\sqrt{2\pi e r^2} \cdot e^{-\pi r^2})^n$.*

Lemma 3 ([20], Lemma 2). *For $a, b \in R_q$, $\|a \cdot b\|_\infty \leq \|a\| \cdot \|b\|$.*

In addition, we let $\beta = r\sigma\sqrt{n}$ and we need to carefully choose $r \geq 1/\sqrt{2\pi}$ so that choosing x from $D_{\mathbb{Z}^n,\sigma}$ with ℓ_2-norm greater than β has negligible probability according to Lemma 2.

Furthermore, to ensure the correctness of our veto protocol, we require that

$$\frac{q}{4} - 2 \geq m(m-1)\beta^2 + m\beta$$

holds true.

4 Passively Secure Lattice-Based AV-Net

The following AV-net protocol provides privacy in the presence of passive (honest-but-curious) adversaries. In Section 5, we show how to extend the this AV-net such that privacy can be guaranteed even if all but two voters actively deviate from their specified programs.

In what follows, we first describe the passively secure AV-net protocol with full technical details (Section 4.1), then we prove that this protocol is correct (Section 4.2), and eventually elaborate on the privacy it provides (Section 4.3).

4.1 Protocol description

We use the same protocol participants as in the original AV-net (Section 2.1).

Parameters. We briefly recall the main parameters from Section 3 that we use in the passively secure veto protocol. Essentially, all computation is done in the ring R_q. The distribution χ samples elements from R_q such that $\mathrm{RLWE}^{(m)}_{n,q,\chi}$ holds true. Let a be an element from R_q chosen uniformly at random. In what follows, we implicitly assume that all protocol participants take these parameters as input.

Offline phase. If, in the online phase (see below), voter V_i chooses "no veto", then she uses a specific element $y_i \in R_q$ to blind her choice, and a random element otherwise. The elements $y_1, \ldots, y_m$ (for voters $\mathsf{V}_1, \ldots, \mathsf{V}_m$) will have a specific structure such that

- the distribution of blinded "veto" choices is indistuingishable from the uniform distribution over R_q (under RLWE) which itself is the distribution of "no veto" choices, and
- all blinding elements collectively equal out if and only if all voters choose "veto".

In fact, each voter's blinding element y_i is a specific linear combination of elements b_j that are generated by all the other voters V_j ($i \neq j$). More precisely, each voter V_i generates b_i as follows:

1. Choose $s_i, e_i \leftarrow \chi^2$.
2. Set $b_i \leftarrow a \cdot s_i + e_i$.

After all voters have published their b_i's, each voter V_i (locally) computes her individual blinding element y_i as follows:

$$y_i \leftarrow \left(\sum_{j=1}^{i-1} b_j\right) - \left(\sum_{j=i+1}^{m} b_j\right).$$

Online phase. Voter V_i computes her "encrypted" choice as follows:

1. If "no veto", then choose $e'_i \leftarrow \chi$, and set $c_i \leftarrow s_i y_i + e'_i$.
2. If "veto", then choose $r_i \xleftarrow{r} R_q$, and set $c_i \leftarrow r_i$.

After all voters have published their c_i's, each voter (locally) computes the final result as follows:

$$\mathsf{res} \leftarrow \begin{cases} \text{no veto} & \text{if } \| \sum_{i=1}^{m} c_i \|_\infty \leq \dfrac{q}{4} - 2 \\ \text{veto} & \text{otherwise} \end{cases}.$$

4.2 Correctness

In this section, we show that the veto protocol, as defined in Section 4.1, is correct, i.e., it outputs the correct result (with overwhelming probability) if all

participants follow the protocol specification correctly (Theorem 2). To this end, we use the following result which ensures that the error terms introduced (for privacy reasons) do not undermine correctness of the veto protocol except for with negligible probability.

Lemma 4. *The probability that a uniformly chosen random element $r \in R_q$ has max norm less than or equal to $N \geq 1$ is given by*

$$\Pr[\|x\|_\infty \leq N \colon x \leftarrow R_q] = \frac{(2N+1)^n}{q^n}.$$

Theorem 2 (Correctness). *Let P be the veto protocol defined in Section 4.1. Assume that all voters $\mathsf{V}_1, \ldots, \mathsf{V}_m$ (and the bulletin board B) are honest, i.e., run their programs as specified by the protocol. Then, we have that for all runs (of this instance) of P, the following equivalence holds true with overwhelming probability: The final result* res *is "veto" if and only if there exists (at least) one voter V_i who chooses "veto".*

Proof. Let us start with a variant of the veto protocol without error terms, i.e., $e_i, e_i' = 0$ for all voters V_i.[4] Now, if all voters choose "no veto", we have that

$$\begin{aligned}
\sum_{i=1}^{m} c_i = \sum_{i=1}^{m} s_i \cdot y_i &= \sum_{i=1}^{m} s_i \cdot \left(\left(\sum_{j=1}^{i-1} b_j \right) - \left(\sum_{j=i+1}^{m} b_j \right) \right) \\
&= \sum_{i=1}^{m} s_i \cdot \left(a \cdot \left(\left(\sum_{j=1}^{i-1} s_j \right) - \left(\sum_{j=i+1}^{m} s_j \right) \right) \right) \\
&= a \cdot \left(\left(\sum_{i=1}^{m} \sum_{j=1}^{i-1} s_i \cdot s_j \right) - \left(\sum_{i=1}^{m} \sum_{j=i+1}^{m} s_i \cdot s_j \right) \right) = 0
\end{aligned}$$

holds true, where the last equation follows from Lemma 1.

Conversely, if (at least) one voter vetoed, then the sum $\sum_{i=1}^{m} c_i$ is distributed uniformly at random over R_q. Hence, res correctly reflects how voters voted in the veto protocol (without error terms).

Due to space limitations, the proof that the error terms remain sufficiently small, is provided in our technical report [11].

Hence, altogether, we can conclude that (with overwhelming probability) the final result res equals "veto" if and only if at least one voter vetoes. This proves the correctness of the veto protocol defined in Section 4.1.

4.3 Privacy

In this section, we show that the veto protocol, as defined in Section 4.1, provides privacy in the presence of honest-but-curious adversaries. The privacy notion we apply follows [3].

[4] We note that, in this case, the protocol would not guarantee privacy.

Theorem 3 (Privacy). *Assume that $\mathrm{RLWE}^{(m)}_{n,q,\chi}$ holds true. Let A be an arbitrary passive ppt adversary which controls (at most) all but two voters $(\mathsf{V}_i)_{i\in\mathcal{I}_{dis}}$. Let $(\mathsf{V}_i)_{i\in\mathcal{I}_{hon}}$ denote the remaining (uncorrupted) voters. Let $(v_i)_{i\in\mathcal{I}_{hon}}$ and $(v'_i)_{i\in\mathcal{I}_{hon}}$ be two arbitrary vectors of choices that yield the same result* res*. Then, the probability that the adversary A can distinguish between the set of runs in which the honest voters $(\mathsf{V}_i)_{i\in\mathcal{I}_{hon}}$ vote according to $(v_i)_{i\in\mathcal{I}_{hon}}$ or to $(v'_i)_{i\in\mathcal{I}_{hon}}$ is negligible.*

Proof. We distinguish between the following two cases:

1. $(v_i)_{i\in\mathcal{I}_{hon}}$ and $(v'_i)_{i\in\mathcal{I}_{hon}}$ yield the result "no veto".
2. $(v_i)_{i\in\mathcal{I}_{hon}}$ and $(v'_i)_{i\in\mathcal{I}_{hon}}$ yield the result "veto".

In the first case, both $(v_i)_{i\in\mathcal{I}_{hon}}$ and $(v'_i)_{i\in\mathcal{I}_{hon}}$ consist of "no veto" choices only, hence $(v_i)_{i\in\mathcal{I}_{hon}} = (v'_i)_{i\in\mathcal{I}_{hon}}$. In particular, it is impossible to distinguish between runs in which the honest voters vote according to $(v_i)_{i\in\mathcal{I}_{hon}}$ or to $(v'_i)_{i\in\mathcal{I}_{hon}}$.

To prove indistuingishability in the second case, we use the following hybrid argument. To this end, we simulate the protocol as follows: if there exists at least one honest voter who chooses to veto, then *all* honest voters $(\mathsf{V}_i)_{i\in\mathcal{I}_{hon}}$ veto. Under the assumption that $\mathrm{RLWE}^{(m)}_{n,q,\chi}$ holds true, it follows that for any possible set of choices $(\tilde{v}_i)_{i\in\mathcal{I}_{hon}}$ which contains at least one "veto", the simulated protocol is indistuingishable from the original veto protocol in which the honest voters vote according to $(\tilde{v}_i)_{i\in\mathcal{I}_{hon}}$. Due to the symmetry of this argument, we can conclude that no ppt adversary A can distinguish between runs in which the honest voters vote according to $(v_i)_{i\in\mathcal{I}_{hon}}$ or to $(v'_i)_{i\in\mathcal{I}_{hon}}$ if there exist $j,k\in\mathcal{I}_{hon}$ such that $v_j = \text{veto}$ and $v'_k = \text{veto}$.

5 Actively Secure Lattice-Based AV-Net

In this section, we describe how to extend the veto protocol from Section 4 such that it provides privacy and verifiable correctness in the presence of active adversaries.

Let us first explain why the protocol from Section 4 does neither protect privacy nor correctness if (some) voters do not follow their prescribed programs:

- *Privacy*: Assume that we have three voters $\mathsf{V}_1,\mathsf{V}_2,\mathsf{V}_3$, where V_1 and V_2 are honest, and V_3 is malicious and aims to actively break privacy of, say, voter V_1. Now, V_3 waits until V_2 has published b_2 and then simply publishes $b_3 \leftarrow -b_2$. By this, we have that $y_1 = 0$. Hence, if V_1 does not veto, it follows that $c_1 = e'_1$ is chosen according to χ, and that $c_1 = r_1$ is chosen uniformly at random otherwise. Therefore, the adversary (controling V_3) knows that (with high probability) V_1 did not veto if $\|c_1\|_\infty < \beta$. This breaks V_1's privacy.

– *Correctness*: Assume that we have two voters $\mathsf{V}_1, \mathsf{V}_2$, where V_1 is honest and decides to veto, and V_2 is malicious and aims to actively cancel out V_1's veto. Now, V_2 waits until V_1 has published c_1 and then simply publishes c_2 such that $\|c_1 + c_2\|_\infty < \frac{q}{4} - 2$. Therefore, the final result is "no veto" even though V_1 had chosen "veto".

At a high level, what both attacks have in common is that the adversary can adaptively choose the corrupted voters outputs depending on the honest voters' ones. In order to eliminate this vulnerability, we employ a lattice-based commitment scheme as described in Section 5.1. We will then demonstrate that the resulting veto protocol in fact provides verifiable correctness (Section 5.2) and privacy (Section 5.3) against malicious adversaries.

5.1 Protocol description

We now explain how the passively secure veto protocol from Section 4 can be extended in order to defend against active adversary that aim to undermine privacy or verifiable correctness. More precisely, we need to ensure that voters choose their messages pairwise independently. To this end, we additionally employ an arbitrary lattice-based commitment scheme $(\mathsf{KeyGen}_{\mathsf{com}}, \mathsf{Com}, \mathsf{Open})$ which is (at least) computationally hiding and (at least) computationally binding under standard lattice hardness assumptions. More concretely, one could, for example, instantiate this generic commitment scheme with the highly efficient lattice-based commitment scheme by Baum et al. [2].

However, we need to be careful since commitment schemes like [2] are malleable. Even though there are generic compilers for transforming malleable commitment schemes into non-malleable ones (see, e.g., [8]), we are not aware of any existing work that analyzes such compilers in a quantum setting. Therefore, we will specify that voters open their commitments exactly in the reverse order according to which they published them. With this simple trick, we can still use malleable commitment schemes (see Section 7 for a discussion).

More precisely, we extend the veto protocol from Section 4 as follows. We refer to Appendix A for the notation related to the generic commitment scheme $(\mathsf{KeyGen}_{\mathsf{com}}, \mathsf{Com}, \mathsf{Open})$.

Parameters (extended). We denote by $\mathsf{prm}_{\mathsf{com}}$ the joint public parameters of the commitment scheme (computed by running $\mathsf{KeyGen}_{\mathsf{com}}$).

Offline phase (extended). Each voter V_i, after having computed b_i, executes the following steps:

3. Compute $(\gamma_i, \rho_i) \leftarrow \mathsf{Com}(\mathsf{prm}_{\mathsf{com}}, b_i)$.[5]
4. Publish γ_i.
5. Wait until all γ_j were published ($j \in \{1, \ldots, m\}$).

[5] In other words, γ_i is the commitment to b_i using randomness ρ_i (see Appendix A).

6. Set $\sigma \leftarrow$ order of published γ_j's (according to their time stamps).
7. Wait until all (b_j, ρ_j) were published for $\sigma(j) > \sigma(i)$.
8. Publish (b_i, ρ_i).
9. Wait until all (b_j, ρ_j) were published for $\sigma(j) < \sigma(i)$.
10. If $\mathsf{Open}(\mathsf{prm}_{\mathsf{com}}, b_j, \gamma_j, \rho_j) = 0$ for some $j \neq i$, then abort.

Online phase (extended). Each voter V_i, after having computed c_i, executes the following steps:

3. Compute $(\gamma_i', \rho_i') \leftarrow \mathsf{Com}(\mathsf{prm}_{\mathsf{com}}, c_i)$.
4. Publish γ_i'.
5. Wait until all γ_j' were published ($j \in \{1, \ldots, m\}$).
6. Set $\sigma' \leftarrow$ order of published γ_j''s (according to their time stamps).
7. Wait until all (c_j, ρ_j') were published for $\sigma'(j) > \sigma'(i)$.
8. Publish (c_i, ρ_i').
9. Wait until all (c_j, ρ_j') were published for $\sigma'(j) < \sigma'(i)$.
10. If $\mathsf{Open}(\mathsf{prm}_{\mathsf{com}}, c_j, \gamma_j', \rho_j') = 0$ for some $j \neq i$, then abort.

5.2 Verifiable correctness

In this section, we show that the veto protocol defined in Section 5.1 is verifiably correct [7] even if an arbitrary adversary actively corrupts (a subset of) voters.

We note that we can restrict our attention to the case that an adversary aims to swap an honest "veto" into "no veto". In fact, if an adversary (controling at least one voter) wants the final result to be "veto", then he can simply let the corrupted voter run her "veto" program.

Theorem 4 (Verifiable correctness). *Let P be the veto protocol defined in Section 5.1. Assume that the bulletin board B is honest. Assume that the commitment scheme is computationally binding and hiding. Then, we have that for all runs (of these instances) of P, the following implication holds true with overwhelming probability: If there exists an honest voter who chooses "veto", then the final result is "veto" (or the protocol aborts prematurely).*

Due to space limitations, the complete proof is provided in our technical report [11].

5.3 Privacy

In this section, we show that the veto protocol, as defined in Section 5.1, provides privacy in the presence of malicious adversaries.

Theorem 5 (Privacy). *Assume that $RLWE^{(m)}_{n,q,\chi}$ holds true. Assume that the commitment scheme is computationally binding and hiding. Let A be an arbitrary malicious ppt adversary which controls (at most) all but two voters $(\mathsf{V}_i)_{i \in \mathcal{I}_{dis}}$. Let $(\mathsf{V}_i)_{i \in \mathcal{I}_{hon}}$ denote the remaining (uncorrupted) voters. Let $(v_i)_{i \in \mathcal{I}_{hon}}$ and $(v_i')_{i \in \mathcal{I}_{hon}}$ be two arbitrary vectors of choices that yield the same result* res. *Then,*

the probability that the adversary A can distinguish between the set of runs in which the honest voters $(\mathsf{V}_i)_{i\in\mathcal{I}_{hon}}$ *vote according to* $(v_i)_{i\in\mathcal{I}_{hon}}$ *or to* $(v'_i)_{i\in\mathcal{I}_{hon}}$ *is negligible.*

Due to space limitations, the complete proof is provided in our technical report [11].

6 Experimental results

We have implemented the passively secure AV-net described in Section 4. Since the commitment scheme that is additionally required in the actively secure AV-net (Section 5) is generic and independent of the rest of the protocol, any efficient lattice-based commitment scheme can be chosen (e.g., [2]).

Our implementation uses C++ language and NTL library. We run 10,000 times experiments using the parameters (the same as in [20]) $n = 512, \sigma = 4.19, q = 120833$ on a computer with Intel Core i7-6500U CPU @ 2.50 GHz, running Cygwin version 3.1.5, g++ compiler version 9.3.0. Then we evaluate average runtime for discrete Gaussian sampling based on [22] (TimeDGS), polynomial multiplication (TimePoly), and vote tallying (TimeVeto) respectively. We show the experimental results with two decimal precision in Table 1.

Table 1: Runtime (millisecond) of our implementation.

m	TimeDGS	TimePoly	TimeVeto
3	0.42	0.89	0.24
10	2.44	8.44	6.94
15	5.60	23.49	16.51
20	8.69	39.44	24.35

The optimizer used was -O2. GCC basically performs almost all the supported optimizations that do not involve a space-speed tradeoff. This option is to benefit the compilation time and performance of the generated code. -O2 flags the compiler mainly to inline functions when able. -O3 adds some flags for loop unrolling and tree distribution and -Ofast disregards standards compliance and adds a couple extra flags like -ffast-math.

We just tested this code and it also works with -O3 as well as -Ofast, but at $m = 3$ and $10,000$ runs, it does not appear to have any noticeable impact on the execution time of the code. We also tried several values for m and experimentally, no error showed up when $m = 100$ but errors start to show up when m is approximately 125.

7 Discussion

In this section, we elaborate on the properties of the AV-nets proposed and analyzed above.

Post-quantum anonymity. We have proven that the 2-round AV-net (Section 4.1) and the 4-round AV-net (Section 5.1) guarantee anonymity under the decisional RLWE assumption in the presence of arbitrary passive or active adversaries, respectively. The decisional RLWE assumption is a well-studied lattice-based hardness assumptions and commonly believed to be intractable even by quantum algorithms. Since anonymity of previous AV-nets [1, 14] relies on the DDH-assumption, our AV-nets are the first ones with *post-quantum* anonymity.

Observe that, both the two AV-nets proposed in this work as well as the previous one by Hao and Zielinski [14] have the following property: if there is a single voter who vetoes, then this voter knows that she is the only one who vetoed.

Robustness. It is obvious that if just a single voter does not participate in the online phase of our AV-net(s), then the complete protocol needs to restart again. Therefore, similarly to previous AV-nets [1, 14], our protocols have a low level of robustness, too. Typically, in order to increase robustness, protocols for secure computation employ threshold schemes: if at least t out of n parties participate, then the protocol terminates successfully. On the downside, however, threshold schemes lead to stronger trust assumptions for anonymity/privacy. In the case of (our) AV-nets, where we merely require that two voters are honest for anonymity, introducing a threshold structure would impair this mild trust assumption.

We note that in our actively secure protocol, opening the commitments in reverse order puts some burden on the underlying infrastructure, more precisely on the bulletin board. In fact, it is a non-trivial challenge in practice to guarantee verifiable time-stamps. One possible solution to this problem is to employ a distributed ledger technology (DLT).

Round complexity. Previous AV-nets [1, 14] require 2 rounds of interaction, both in the presence of passive and active adversaries. In contrast to that, our actively secure AV-net requires 4 rounds of interaction. The reason for this are the different techniques to make the voters' intrinsically homomorphic outputs *non-malleable*. While [1, 14] employ ZKPs for this purpose, it is not immediately clear how to efficently do this in the lattice-based setting. Therefore, we decided to add two further rounds of interaction in which the voters first commit to their outputs before revealing them. Since there are a number of highly efficient lattice-based commitment schemes (see, e.g., [2]), we argue that our variant is a reasonable trade-off.

Alternative approaches. AV-nets can be regarded as specific instances of secure boardroom voting or, more generally, secure multi-party computation (MPC) protocols. We elaborate on this in what follows.

There are numerous efficient MPC protocols in the literature that could be used for securely evaluating veto functions, in particular with post-quantum privacy (see, e.g., [9]). Typically, employing such generic MPC protocols is advantageous for *complex* result functions. However, generic MPC protocols are less well-suited for the specific case of veto protocols, where the result function is simply Boolean OR.

In a boardroom voting protocol, the voters themselves tally the ballots, without having to rely on a trusted set of talliers or election authorities. Several such protocols have been proposed so far (see, e.g., [13, 15]). However, these protocols employ specific ZKPs, and therefore, as explained above, transforming them into a lattice-based setting undermines efficiency. Furthermore, we note that if we applied one of these boardroom voting protocols to evaluate the veto function, then the final result would reveal how many voters actually vetoed. In contrast to that, in an AV-net, the final result merely reveals whether or not at least one voter vetoed (without revealing the number of vetoing voters). Hence, AV-nets are *tally-hiding* [16] and thus provide an essentially perfect privacy level.

We note that existing verifiable post-quantum secure e-voting systems [4, 10] would not be (immediately) useful for our purposes as well. The reason is that they are neither tally-hiding nor designed for peer-to-peer elections.

8 Conclusion

We proposed the first AV-nets with post-quantum anonymity. The first variant of our protocol requires 2 rounds of interaction and is passively secure, whereas the second one requires 4 rounds of interaction and is actively secure. Anonymity of our AV-net reduces to the decisional ring learning with errors (RLWE) assumption.

Acknowledgements

We thank the anonymous reviewers for their constructive feedback. We also thank Peter B. Rønne for helpful remarks. Peter Y.A. Ryan and Johannes Müller acknowledge support from the Luxembourg National Research Fund (FNR) and the Research Council of Norway for the joint INTER project SURCVS (Number 11747298). Jintai Ding, Doug Emery, and Vonn Kee Wong would like to thank the US Air Force and NSF for partial support. Jintai Ding would also like to thank University of Luxembourg for partial support.

Bibliography

[1] Samiran Bag, Muhammad Ajmal Azad, and Feng Hao. PriVeto: A Fully Private Two-Round Veto Protocol. *IET Information Security*, 13(4):311–320, 2019.

[2] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More Efficient Commitments from Structured Lattice Assumptions. In *SCN 2018, Proceedings*, volume 11035 of *LNCS*, pages 368–385. Springer, 2018.

[3] David Bernhard, Véronique Cortier, David Galindo, Olivier Pereira, and Bogdan Warinschi. SoK: A Comprehensive Analysis of Game-Based Ballot Privacy Definitions. In *2015 IEEE S&P*, pages 499–516, 2015.

[4] Xavier Boyen, Thomas Haines, and Johannes Müller. A Verifiable and Practical Lattice-Based Decryption Mix Net with External Auditing. In *ESORICS 2020*. To appear.

[5] David Chaum. Security Without Identification: Transaction Systems to Make Big Brother Obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.

[6] David Chaum. The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. *J. Cryptology*, 1(1):65–75, 1988.

[7] Véronique Cortier, David Galindo, Ralf Küsters, Johannes Müller, and Tomasz Truderung. SoK: Verifiability Notions for E-Voting Protocols. In *2016 IEEE S&P*, pages 779–798, 2016.

[8] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Non-Interactive and Non-Malleable Commitment. In *ACM STOC, 1998*, pages 141–150. ACM, 1998.

[9] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty Computation from Somewhat Homomorphic Encryption. In *CRYPTO 2012. Proceedings*, volume 7417 of *LNCS*, pages 643–662. Springer, 2012.

[10] Rafaël del Pino, Vadim Lyubashevsky, Gregory Neven, and Gregor Seiler. Practical Quantum-Safe Voting from Lattices. In *Proceedings of the 2017 ACM CCS*, pages 1565–1581, 2017.

[11] Jintai Ding, Doug Emery, Johannes Müller, Peter Y.A. Ryan, and Vonn Kee Wong. Post-Quantum Anonymous Veto Networks. *IACR Cryptology ePrint Archive*, 2020:1023, 2020.

[12] Philippe Golle and Ari Juels. Dining Cryptographers Revisited. In *EUROCRYPT 2004, Proceedings*, volume 3027 of *LNCS*, pages 456–473. Springer, 2004.

[13] Jens Groth. Efficient Maximal Privacy in Boardroom Voting and Anonymous Broadcast. In *FC 2004. Revised Papers*, volume 3110 of *LNCS*, pages 90–104. Springer, 2004.

[14] Feng Hao and Piotr Zielinski. A 2-Round Anonymous Veto Protocol. In *Security Protocols, 14th International Workshop, 2006, Revised Selected Papers*, volume 5087 of *LNCS*, pages 202–211. Springer, 2006.

[15] Aggelos Kiayias and Moti Yung. Self-tallying Elections and Perfect Ballot Secrecy. In *PKC 2002, Proceedings*, volume 2274 of *LNCS*, pages 141–158. Springer, 2002.
[16] Ralf Küsters, Juliad Liedtke, Johannes Müller, Daniel Rausch, and Andreas Vogt. Ordinos: A Verifiable Tally-Hiding E-Voting System. In *IEEE EuroS&P 2020. To appear*, 2020.
[17] Noah Stephens-Davidowitz. Discrete Gaussian sampling reduces to CVP and SVP. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1748-1764. Society for Industrial and Applied Mathematics, 2016.
[18] Jintai Ding, Xiang Xie, Xiaodong Lin. A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem. In *IACR Cryptology ePrint Archive*, Report 2012/688, 2012.
[19] Jintai Ding, Tsuyoshi Takagi, Xinwei Gao, and Yuntao Wang. One Sample Ring-LWE with Rounding and Its Application to Key Exchange. In *Applied Cryptography and Network Security*, pages 323–343, Springer, 2019.
[20] Jintai Ding, Tsuyoshi Takagi, Xinwei Gao, and Yuntao Wang. Ding Key Exchange. Technical report, National Institute of Standards and Technology, 2017. available at `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions`.
[21] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On Ideal Lattices and Learning with Errors over Rings. In *EUROCRYPT 2010. Proceedings*, volume 6110 of *LNCS*, pages 1–23. Springer, 2010.
[22] Chris Peikert. An Efficient and Parallel Gaussian Sampler for Lattices. In Advances in Cryptology CRYPTO 2010, 30th Annual Cryptology Conference, pages 80-97, 2014.

A Commitment Schemes

A *commitment scheme* is a tuple of algorithms $(\mathsf{KeyGen}_{\mathsf{com}}, \mathsf{Com}, \mathsf{Open})$ where:

- $\mathsf{KeyGen}_{\mathsf{com}}$ is a ppt algorithm which takes 1^ℓ and outputs the public parameters $\mathsf{prm}_{\mathsf{com}}$, containing a definition of the *message space* $M_{\mathsf{com}} = M_{\mathsf{com}}^\ell$, the *commitment space* $C_{\mathsf{com}} = C_{\mathsf{com}}^\ell$, and the *opening space* $R = R^\ell$.
- Com is a ppt algorithm which takes $\mathsf{prm}_{\mathsf{com}}, m \in M_{\mathsf{com}}$ and outputs values $c \in C_{\mathsf{com}}$ and $r \in R$.
- Open is a deterministic algorithm polynomial-time algorithm which takes $\mathsf{prm}_{\mathsf{com}}, m \in M_{\mathsf{com}}, c \in C_{\mathsf{com}}, r \in R$ and outputs a bit $b \in \{0, 1\}$.

Impacts and failures of internet voting

Enhancing Self-determination and Capacity-Building: Online Voting in the Indigenous Communities of Canada, Australia and the Unites States

Maximilian Hee[1]

[1] Scytl Secure Electronic Voting, S.A., 08008 Barcelona, Spain
maximilian.hee@scytl.com

Abstract. Although Indigenous communities in many European settler-states have experienced acknowledgment of their status and been increasingly granted the right to administer their own affairs, imposed western-style systems of governance have often proved unviable in their context. However, in some cases communities have utilized these policies as an impetus to regain agency over their lives and the land that they inhabit. In Canada, First Nations have been increasingly drawn to the use of digital technologies to strengthen community connectedness and improve political participation. In particular, internet voting has been utilized to mitigate the challenges of collective decision making that many communities are facing. As Canadian cases have shown that internet voting was able to positively impact the degree of self-determination and capacity building, the question arises as to what extent the deployment of internet voting in the Indigenous communities of other jurisdictions can yield similar results. A cross-comparison of Indigenous self-governance in Canada, Australia and the United States of America shows that Indigenous communities are often facing similar obstacles to effective governance that can be over-come by new means of political participation. The identification of the three underlying factors of self-governance framework, political participation, and social geography serve as an analytical tool that guides the cross-comparison. The resulting analysis demonstrates that, albeit significant similarities between the Indigenous populations, self-governance and the deployment of online voting therein is impacted by differing legislation and socio-political factors.

Keywords: self-governance, online voting, Indigenous communities, Australia, Canada, United States of America

1 Introduction

Since the beginning of European colonization, the Indigenous peoples of the Americas, Australia and New Zealand have been subject to policies of displacement and enforced assimilation that often resulted in cultural extrusion and genocide. Although many have experienced acknowledgment of their status over the course of the 20th century, most legal frameworks established in the respective settler states have practically reaffirmed colonial sovereignty over Indigenous communities. This applies especially to financial and territorial reimbursements paid to Indigenous people but also pertains to early

policies of self-determination, which by imposing western-style systems of administration often proved unviable in the Indigenous context. While it is arguable that the adoption of European forms of governance presented a continuation of historic wrongs, in some cases communities have utilized these policies as an impetus to push "into a paradigm of negotiation which assumes the political or legal authority of Indigenous communities to represent their members and to control resources" (Ford & Rowse, 2012, p. 3). For instance, Ford & Rowse (2012) argue that the neoliberal withdrawal has enabled Indigenous communities to mediate between resource extracting corporations, thereby creating opportunities for enhanced capacity-building.

Within the last decade, Canada and its sizable Indigenous population have continually provided examples of such developments where progressive legislation in combination with new means of collective decision-making are enabling Indigenous communities to overcome previous obstacles to successful self-determination and capacity-building. In particular, the cases of the Whitefish River Nation (WFRN) and the Wasauksing First Nation (WFN) and their deployment of online voting (OLV) for the ratification of the new matrimonial real property law (MRP) and the 2017 Land Code vote, respectively, have showcased the potential benefits of digital voting technology for strengthening political participation and modernizing Indigenous self-governance (Gabel et al., 2016a; Budd et al, 2019). While the WFRN and WFN experiences only present a small sample size of Indigenous existence and the Canadian legal framework is distinct from other settler states, many communities in other jurisdiction face similar challenges of being increasingly targeted by corporate resource extraction and diminishing administrative capacities. Therefore, the question arises as to what extent the deployment of OLV in the Indigenous communities of Australia and the United States of America (USA) can yield similar effects on communal capacity-building and self-determination. The question of a potential applicability to these jurisdictions suggests itself not only because of their shared characteristic of being colonial settler-states, but also because all three have a common law legal system and a similar federal framework. Additionally, the choice of comparing Canada, the USA and Australia and not including additional jurisdictions with sizable Indigenous populations, such as New Zealand[1], Mexico or various South- and Latin-American countries, was made because, in contrast to other settler-states, OLV has been trialed and used in multiple local and regional elections in all three of the selected countries. Although public opinion and acceptance of OLV in the three jurisdictions differ from one another, it is feasible that the respective experiences facilitate the adoption of OLV for the purpose of Indigenous self-governance. Hence, the focus of the article is on OLV deployment in self-governance, rather than Indigenous participation in the federal system of the respective countries.

To answer the aforementioned research question, in section 2 the author will first re-examine the cases of the WFRN and WFN and the respective case studies by Gabel et

[1] While the Indigenous population in New Zealand has no officially recognized self-government framework and are instead given special representation in the federal system, Local Maori communities have already utilized OLV in their local iwi elections. For instance, Te Korowai o Ngāruahine Trust, which is the post-settlement governance entity for Ngāruahine iwi, offered OLV next to traditional voting channels for their 2020 Board and Trustee Elections (Te Korowai o Ngāruahine Trust, 2020)

al. (2016a) and Budd et al. (2019). In doing so, the underlying factors and circumstances that were able to produce the positive outcomes in regards to self-determination and capacity building are going to be identified. In particular, it will be shown that the assessment of the Canadian experiences highlights the need to compare the three Indigenous communities across the three underlying factors, namely: the legal framework of Indigenous self-governance, the political participation in Indigenous communities, and the social geography of the respective populations. Thereafter, the USA (in section 3.1) and Australian (and 3.2) legal frameworks, as well as the contemporary situation of their Indigenous populations, will be examined and analyzed on their potential to yield similar results through the employment of OLV. Eventually, the last two sections of the paper will provide a summary of the findings (section 4) as well as concluding points of references for future research (section 5).

2 Indigenous Existence in Canada

For most parts of the 19th and 20th century, Canadian Indigenous affairs have been governed through the *Indian Act* of 1876, a piece of legislation that has for the time of its enactment sustained the colonial and paternalistic character of Indigenous-state relations. Although the *Indian Act* of today entails certain aspects of self-governance, such as the right to elect community chiefs and councils and the authority to pass by-laws, it only allows for a limited form of local governance with little regard to the respective circumstances of individual communities (Abele & Prince, 2006). Principally, the Act covers to main aspects of Indigenous-state relations. First, it determines the legal status of Indigenous individuals, which is defined through descendants. It was only in 1985 that amendments finally terminated previous regulations that fostered assimilation through not granting the native title to descendants of mixed Indigenous/non-Indigenous couples[2]. Secondly, the Act defines the rights, obligations, and functions of Indigenous communities. Moreover, the Acts describes the way in which communities can be created and governed. Generally, self-governance is only partially envisaged, as Indigenous governments remain accountable to the Crown-Indigenous Relations and Northern Development Canada and Indigenous Services Canada and the designated reserve lands cannot legally be owned by a community or its members (Flanagan et al., 2010).

However, since the end of the 20th century alternative paths to self-governance have come into existence. Based on section 35 of the 1982 Constitution Act, the Inherent Rights Policy of 1995 has led to the launch of self-government negotiations between individual First Nations and the federal government. Contrasting previous legislation, self-government agreements bestow communities with law-making authority in a broad range of matters from governance and socio-economic development to health and education (Alcantara & Davidson, 2015). In addition, communities can enter established agreements or initiate their own negotiations and agreements. However, in order to opt into self-governance, the negotiated agreement needs to be approved by community

[2] The *Indian Act* and *First Nations Election Act* specifically pertain to First Nations, and not Metis or Inuit communities, which are other Indigenous populations in Canada.

vote which needs to entail an approval of at least 25% of the respective community´s electorate (Goodman & Pammet, 2014, p. 215). Although more than 30 communities have already made the transition into self-government agreements and several others are in the process of negotiation, most of Canada's 617 First Nations are still governed under the framework of the *Indian Act* (Goodman & Pammet, 2014).

In addition to the *Indian Act*, the *First Nations Elections Act* came into force in 2015 as a result of negotiations between the Government and First Nation leaders from across Canada. The aim of the Act was to provide First Nations with a stronger, more effective framework of governance by, among other things, introducing longer office terms and allowing for advance polling. However, adoption of the *First Nations Elections Act* are optional, and have to initiated by First Nation Council resolution. The current form of both Acts still requires communities to achieve relatively high participation rates, which in the light of a geographically dispersed population and generally low political participation can present a significant obstacle to effective decision-making, as they have often done so in the past (Gabel at al., 2016b). This has meant the path to more effective governance is rendered continuously complicated for communities lacking administrative capabilities. Nonetheless, it needs to be noted that in 2018 amendments to the *Framework Agreement on First Nation Land Management* were passed that allow First Nations to set their own participation thresholds or choose a simple majority for the passage of Land Code Agreements, thereby providing an opt out for the land management sections of the *Indian Act* and the *First Nations Elections Act.*

While there are several factors at play that prevent Indigenous communities from entering self-government agreements, it is arguable that low political participation rates prevent willing communities from achieving the required 25% approval rate to opt into the *First Nations Elections Act.* Moreover, low participation rates in Indigenous affairs are generally hampering collective decision-making and self-governing capacities even when needed agreement and self-administering frameworks are already given. While low participation rates are often attributed to general distrust of the Indigenous population towards state and federal institutions, it is argued that a lack of community connectedness stemming from a growing off-reserve population is likely to be most decisive (Alport & Hill, 2006). While OLV arguably has the potential to alleviate some of the described obstacles of decision-making in Indigenous communities, it can only do so indirectly and under certain circumstances. Both the *Indian Act* and *First Nations Elections Act* regulations only allow for the use of postal and stationary paper polling and outlaw the use of OLV for referenda and election of representatives. However, the deployment of OLV is not prohibited for community polls and ratification votes such as the ones discussed in this section (Budd et al., 2019, p. 211).

Both the WFRN an WFN serve as exemplary experiences for the circumstances that many Canadian Indigenous communities face in their pursuit of self-governance. At the time at which the research had been conducted by Gabel et al. (2016a) and Budd et al. (2019), off-reserve population of the WFRN as well as that of the WFN made up for a considered share of the total population, of up to one third, with ca. 400 of the WFRN's 1,200 (equivalent to 33%) and 369 of the WFN's 1,090 member (34%) residing outside of their respective reserves (Gabel et al., 2016a, p. 4; Budd et al, 2019, p. 215). Further, at the time of writing, both communities are governed under the framework of the

Indian Act and both had, prior to the employment of OLV, struggled with low participation rates (Gabel et al., 2016a; Budd et al, 2019). However, it shall be noted that with the WFN`s adoption of the Land Code, the First Nation is now only partially governed under the *Indian Act*, as the Land Code present an opt-out of the *Indian Act`s* land management provisions.

In the case of the WFRN, OLV technology was employed for the ratification of the new Matrimonial Real Property Law (MRP). The law itself presented an important piece of legislation for the property rights of Indigenous women and the self-governance capacities of the WFRN more generally, as it was intended to replace persisting inequalities in matrimonial law of the *Indian Act.* OLV was provided both as the channel for early voting and as an additional channel to paper ballots on election day. In the case of the WFN, OLV technology was used for the vote on the passing of Land codes, an integral part of the First Nations Land Management Act that is sought to replace sections of the *Indian Act* regarding the management of reserve lands. In this case, OLV was solely offered as an early voting method, additionally to mail-in ballots in the ratification vote of the WFN's newly drafted land code. As such, they present an important contribution to Indigenous governance and capacity building as they allow communities to regain control over their lands and resources.

As for all ratification administered under the *Indian Act*, both communities were required to achieve a quorum of 25% approval rate, which in both instances was successfully reached. However, uptake of internet voting in the WFRN was significantly differing from that in the WFN: while votes cast via internet in the WFN land code ratification accounted for 30% of the votes, only 12% of the votes cast in the WFRN's ratification of the new MRP were cast via internet (Gabel et al., 2016 p. 9; Budd et al, 2019, p. 216). Nonetheless, as in both cases the quorums for minimum participation were only barely met, it is arguable that the vote cast via internet played an important part in preventing a failure of the ratification votes. Although it is feasible that those who have voted online would have utilized another voting channel if OLV had not been made available, it nevertheless can be assumed that OLV technology facilitated the success of the ratification votes.

Table 1. Comparison of the WFN's and WFRN's ratification votes

	WFN´s Land Code Ratification	WFRN´s MRPL
Population Size	1,090	1,200
Size of off-reserve population	369 (34%)	400 (33%)
Mandated quorum	25% approval	25% approval
Use of OLV	As early voting method	As early voting method and on election day
Participation rate	26%	27%
Proportion that voted via OLV	30% of all votes cast	12% of all votes cast

Moreover, Gabel et al. (2016) and Budd et al. (2019) hold that the employment of OLV improved community connectedness and facilitated political participation more generally as it enabled the involvement of members who are living off-reserve or change residency frequently. In addition, community leaders were reported to find OLV a cost-effective way of keeping off-reserve member engaged and informed. Besides the direct benefits of OLV, Gabel at all (2016) noted that the use of OLV indirectly led to the advancement of self-governance capacity as it successfully contributed to the passage of the respective pieces of legislation. Most importantly, however, the adoption of OLV presented an empowering process in and of itself that ultimately fosters community autonomy (Gabel et al., 2016, p. 222). Considering the fact, that such advantages were made evident by the use in ratification votes only, it seems feasible that extending the use of OLV to elections of representatives and referenda will prove beneficial to communities as well. Although many of the communities governed under the *Indian Act* and the *First Nations Elections Act* are still excluded from such deployments of OLV, Goodman & Pammett (2014) point out that already more than half of 617 First Nations in Canada are governed under the self-government agreements and custom and community election codes that make a wide-ranging use of OLV possible.

3 Indigenous Self-determination in Australia and the United States

Having revisited Indigenous OLV experiences in Canada, three underlying factors can be identified as having played a significant role in enabling the positive effects stemming from the employment of OLV technology. First, without an existing legal framework that acknowledges the inherent right to self-governance, Indigenous self-governance would not exist nor could it be positively impacted by OLV. Second, the social geography of communities that are often widely dispersed with growing off-reserve populations creates a need for community connectedness that can be address by OLV technology. Third, low rates of political participation paired with high participation requirements, which are aggravated by the social geographies, create a hinderance to collective decision-making that can be overcome by the employment of OLV.

In order to identify the extent to which the employment of OLV can yield similar effects in Australia and the USA, the situation of their respective Indigenous population is going to be analyzed in the following section. Hereby, the three underlying factors identified in the previous paragraph will serve as guidance for the analysis

3.1 Indigenous Self-Determination in Australia

Although Australia as a post-colonial settler state exhibits similar characteristics to those of Canada, the evolution of Australia Indigenous affairs is differing in significant ways to that of its Canadian counterpart. Acknowledging the historic wrongs committed against Australia's Indigenous population, the Australian State has taken various measures to compensate Indigenous people for the mistreatment they have experience over the past centuries. While granting Aboriginal and Torrie Strait Islander People a

native title that is accompanied by Land rights and other forms of cultural protection, the Federal state does not view the Indigenous population as a political separate entity and thus fails to grant Indigenous peoples the right for self-governance (Vivian et al, 2017). Moreover, after officially distancing themselves from early policies of assimilation, the path chosen by the federal administration was that of providing Indigenous groups with channels of political representation within the white mainstream society rather than establishing independent Indigenous system outside of it. Over the course of the second half of the 21st century, these channels of representation took on various forms of differing competences, with most of them having been discontinued by following administration. Examples of such bodies include the Department of Aboriginal Affairs (1972-1990), the National Aboriginal Consultative Committee (NAAC) (1972-1985), and the most recent Aboriginal and Torres Strait Islander Commission (ATSIC) (19902005) (Perkins, 2008). While the Department of Aboriginal Affairs was more of a public service with Indigenous employees than a representational body, the NAAC as well as the ATSIC consisted of Indigenous representatives elected by Aboriginals in the 36 regions of Australia to (Patterson et al., 2017).

As the ATSIC was eventually abolished and merged with the Department of Families, Community Services and Indigenous Affairs, Indigenous communities started to seek out other mechanisms of self-determination. In South Australia, for example, the Nation of the Ngarrindjeri people has made numerous political efforts that resulted in constructive and beneficial relationships between their people and state and regional governments (Vivian et al., 2017, p. 217). Additionally, the Gunditjmara People of Victoria are using democratic mechanisms to attend to their peoples` needs and have negotiated several agreements with the state government to advance their self-determination in regards to cultural heritage, land and resource use (Vivian et al., 2017, p. 217). Most notably, however, the amicable attitude of the state of Victoria towards its Indigenous population resulted in the creation of the Victoria`s First People Assembly (VFPA). Although the VFPA is not responsible for the negotiation of treaties, its main objective is the creation of a treaty negotiation framework as well as rules and processes by which a treaty can be agreed in Victoria (VFPA, 2019).

Despite the regional character of Indigenous politics in Australia, it can be argued that approaches like those followed by the VFPA ultimately contribute to the advancement of self-determination and Indigenous governance. However, even if new channels of representation are being created, the general lack of political participation among Australia`s First People presents a challenge of similar magnitude. Ever since the extension of the franchise to Indigenous Australian in the 1960s, voter mobilization has been difficult to achieve. This is partly due to socio-economic reasons, but similar to their Canadian counterpart, many Aboriginal Australians are wary of participating in the Anglo-Australian political system and view doing so as a continuation of institutional assimilation (Hunt et al., 2008). For instance, in 2020 only 76% of Indigenous Australians were registered to vote, in comparison to 96% of the general population (AEC, 2020.) Moreover, it is arguable that a general distrust of the Indigenous population led to the low participation rate in the VFPA election, where only 2,000 of the eligible 30,000 Indigenous voters cast their vote (Towell, 2019).

Despite growing number of Indigenous Australians, the issue of underrepresentation is aggravated by the socio-geographical characteristics of the Indigenous population, which is widely dispersed Australia`s territory and thus varies greatly from state to state (ABS, 2018). While the in the Norther Territory Indigenous population present about 20% of the regional population, they only make up for less than one percent of Victoria´s population. Additionally, the migratory pattern further complicates political organization and collective decision-making. For example, between 2011-2016 45% of Aboriginal and Torres Strait Islander people moved their residency, with many of them moving from one state or territory to another (ABS, 2016). Although there is a general trend of migration to urban areas, almost 20% of the Indigenous population continues to live in remote areas (ibid).

3.2 Indigenous Self-Determination in The United States

The development of Indigenous-state relation in the USA is similar to that in Canada, and thus differs significantly form the Australian approach to Indigenous and settler coexistence. While the first centuries after the colonization of the territory that now comprises the USA was generally characterized by hostility towards the Indigenous population, there was a gradual move away from cultural suppression and assimilation towards recognition and self-determination[3]. Although the passage of the Indian Reorganization Act in 1934 foresaw an extension of Indigenous governance and a strengthening of Indigenous communities, the succeeding administrations terminated most special relationship and agreements between communities and the federal government and implemented assimilationist policies such as mandatory boarding schools and other forms of governmental paternalism (Strommer & Osborne, 2014). It was only after the increased activism for civil rights in the 1960s that the concepts of Indigenous sovereignty and self-determination began to characterize Indigenous-state relationship. Thus, the passage of the Indian Self-determination and Education Assistance Act (ISDEAA) in 1975 is widely regarded as the key-legislation that laid the foundation for self-determination and the state to state character that coined Indigenous affairs ever since.

Most importantly, the ISDEAA acknowledges the status of Indigenous peoples as First Nations and their inherent right for sovereignty. Moreover, the Act grants communities the right "to assume the responsibility, and associated funding, to carry out programs, functions, services and activities that the United States government would otherwise be obliged to provide to Indians and Alaska Natives" (Strommer & Osbourne, 2014, p. 4). As a consequence, First Nations of today are legally authorized to administer their own healthcare and social services, determine matters of education, religion and infrastructure. Additionally, Indigenous governments are given the authority to administer the use of Indigenous lands and the extraction of resources through third parties, which significantly contributes to the economic capacity of Indigenous

[3] It shall be noted that the circumstances and legal matters described in section 3.2 only pertain to Indigenous population in the contiguous USA, and no to the Indigenous peoples in Alaska, Hawaii and Samoa. Affairs regarding Alaska, Hawaii and Samoa Indigenous peoples are each governed by separate legislation.

communities and thus furthers their economic and ultimately political autonomy. Nonetheless, although recognized communities retain their political sovereignty, their status is similar to that of federal states as they receive financial assistance from the federal government and are bound by and subject to some aspects of federal law (Johnson & Hamilton, 1994).

In comparison to their Canadian counterpart, In the USA Indigenous communities enjoy sovereignty in a much wider array of policy areas. Many communities have established their own state organs and political bodies that are not limited to resource and land-management, but also include Indigenous courts, law enforcement agencies and first responders (Johnson & Hamilton, 1994). However, probably the most distinctive feature of Indigenous governance in the USA is that in contrast to Canadian First Nations, collective decision-making within the communities is not bound to electoral codes or federally mandates quorums. Stubben et al. (2005) note that although a majority of communities are still governing themselves through European-style of democratic decision-making their understanding of politics is still heavily influenced by traditional practices of direct democracy and unanimous decision making

As of today, there are 6,8 million Indigenous people living in the USA, with 566 federally recognized communities presiding over the lives of their members in 35 federal states (US Census, 2020). While most of the communities have a population of fewer than 10,000, some nations such as the Navajo People or the Cherokee have more than 200,000 members (Navajo Nation, 2020; Cherokee Nation, 2020). The most populous communities have established electoral commissions and regularly hold elections for position such as chief, deputy chief as well as regional and local councils. For instance, the elections of the Cherokee Nation are held on a specific day and conducted through walk-in polling stations that are on reserve or at specifically assigned polling stations across the country (Cherokee Nation, 2020).

Despite the relatively well-established systems and self-governance, achieving adequate political representation remains a challenge for many Indigenous nations Despite the widespread use of information and communication technology, many struggle to maintain sufficient community connectedness as over 75% of the total Indigenous population is living outside of jurisdictional boundaries of their nations (Milke, 2020).

4 Cross-Comparison of Canadian, Australian and US-American Self-Governance of Indigenous Communities

Having explored the state of Indigenous existence and the varying degrees of self-determination in the Australian and US-American settler state, the following last section of the paper will compare their respective features to the situation of Canada's First Nations in order to come to conclusions about the potential applicability of OLV in the Australian and USA contexts. Hereby, the focus will be on the preconditions identified in section 2, so as to guide the cross comparison of the three jurisdictions. More specifically, the goal of the comparison is to determine the degree to which Australian and US-American Indigenous peoples exhibit the characteristics that facilitated the positive effects of OLV for Canadian First Nations.

4.1 Legal Frameworks of Indigenous Self-Governance

First, the legal frameworks and policy approaches towards the respective Indigenous populations of the three settler states shall be compared. In Canada, although Indigenous peoples are officially recognized and granted the right to administer their own affairs, federal legislation directed at First Nations is still limiting the self-determination and capacity-building of many communities. This is mainly due to the remnants of colonial legislation, such as the *Indian Act*, that exhibit paternalistic and assimilationist characteristics. While the self-governance of many communities remains to be regulated under the *Indian Act*, there has been a gradual move towards the adaption of legislation that mitigates and compensates for the systemic injustices of the existing framework. Moreover, past and on-going negotiations have resulted in the drafting of new self-government agreements that present an opt-out of the *Indian Act* and provide communities with the opportunity to create self-governance structures in accordance with their own values and principles. The cases of the WFRN and WFN can be regarded as first-hand experiences of this advancement in self-governance and capacity building.

Similar to Canada, the US Government's policy approach towards Indigenous people is characterized by recognition of the Indigenous status and acknowledgement of the right for self-governance. The US-American legal framework grants Indigenous communities many of the rights and competences that are also given to Canadian First Nations under the *Indian Act*. Moreover, the degree of autonomy granted through the ISDEAA exceeds that of Canadian communities. Most notably, Indigenous self-governance in the USA is not limited by federally imposed election codes, which arguably facilitates collective decision-making and therefore allows for a higher degree of self-determination.

On the other hand, Australia's legislation and policy approach towards its Indigenous population stands in stark contrast to that of Canada and the United States. Although the Australian state acknowledges the native title and grants Australian First People cultural protection and settlement rights, there is no legal framework that grants Indigenous communities the right to govern themselves as distinct political entities. Instead of presenting a channel for self-determination, Indigenous representation in the form of regional councils solely presents a channel of co-determination within the existing settler society.

4.2 Political Participation

Next, the political participation and the general acceptance of imposed self-governance structure shall be compared. In Canada, political participation of the Indigenous population in Indigenous as well as federal affairs is differing widely across the individual communities, but in most instances turnout rates are far below the non-Indigenous average. While low participation rates present a significant challenge to effective decision-making in First Nation governance, self-determination is additionally impeded by inadequately high election quorums mandate through federally imposed election codes. While in Australia there is generally no nation-wide framework for Indigenous self-governance and hence no mandated election quotes that would hamper collective

decision-making, local efforts of community organization struggle to attract the attention of the Indigenous population and further suffer from low participation rates. In this regard the VFPA is paradigmatic, as during its first general elections the participation rate did not exceed 7% of the eligible voters (VFPA, 2019).

The situation among Indigenous people in the USA, on the other hand, seems to be differing from that of Australia's first people and Canada`s First Nations. While the participation of Indigenous citizens in state and federal election is generally below the average of other non-Indigenous groups, the research conducted over the process of writing this essay has not produces any relevant indication that self-governing communities are suffering from comparable lack of participation. Moreover, as Indigenous people in the USA are granted the right to determine their own election codes, collective-decision making is not bound to externally imposed participation quorums that complicate effective self-governance.

4.3 Social Geography

Lastly, the social geography of Indigenous communities shall be compared. In the case of Canada, high shares of the Indigenous population residing off-reserve as well as a high seasonal mobility of community members aggravated collective decision-making and therefore provided the circumstances in which the employment of OLV proved to be profitable. While in the WFRN and WFN around two thirds of the population was still residing on Indigenous lands, in the whole of Canada only 44% of all First Nation members are still living on their designated reserve lands (Milke, 2006).

The analysis of Australian and USA communities has shown a similar situation of community connectedness. In Australia, the vast majority of the Indigenous population is dispersed over the urban areas of the various federal states, with only 20% remaining in the remote areas traditionally inhabited by Australian first people (ABS, 2016). Additionally, Australia`s Indigenous population is highly mobile, with 45% having changed their place of residency in the period between 2011 and 2016 (ABS, 2016). Under these conditions, it is arguably obvious that community connectedness is difficult to sustain and collective action and decision-making are challenging undertakings for Indigenous communities. Moreover, similar to Canada, the vastness of the Australia territory and great distances between individual rural and urban settlements further complicate such matters.

Indigenous communities in the USA as well have not been spared from urbanization and the dissemination of their population over the territory of the federal states. With 75% of Indigenous residing out of their respective jurisdictions, community connectedness is challenged in a similar way to that of Canadian and Australian communities. However, it can be argued that despite the geographical vastness of the US territory, a more densely developed infrastructure as well as measures of remote voting are able to mitigate the effects of population dispersion more effectively than in Canada or Australia.

Table 2. Comparison of the preconditions in Canada, The USA and Australia

	Canada	USA	Australia

Legal Framework	Allows for comprehensive self-governance	Allows for comprehensive self-governance	Does not allow for any form of self-governance
Social Geography	Highly dispersed; Significant off-reserve population	Highly dispersed; Significant off-reserve population	Highly dispersed
Polit. Participation	Low participation rates	No information	Low participation rates

5 Conclusion

Having evaluated and compared Indigenous self-governance in the United States, Australia and Canada, the following section summarizes the findings in order to come to a conclusion about the applicability of OLV technology in the self-governance of US-American and Australian Indigenous communities. More specifically, the objective of the preceding analysis was to determine the extent to which an employment of OLV in Australia and the USA could yield similar positive results on self-determination and capacity building. As the author has determined, the three underlying factors of the social geography of Indigenous communities, low political participation and existing legislation of self-governance to be most decisive in providing the necessary circumstances under which Indigenous communities could benefit from the employment of OLV technology, the following conclusion is based on the existence of these factors in Australia and the USA.

The analysis of the state of Australia's Indigenous population has shown that local Indigenous communities exhibit similar social geographies. Indigenous communities in Australia are often widely dispersed over the different federal states with a sizable share living in urban metropolitan areas, which hampers community connectedness and aggravates collective decision-making. Additionally, political participation of the Indigenous population is often significantly below that of the average settler population, while turnout rates for local council election, such as the pioneering Victoria`s First People Assembly fail to exceed 10%. However, the lack of a legal framework that grants Australia`s First people the right to govern themselves, presents a considerable obstacle to Indigenous self-determination and the advancement thereof. Although the existence of low community connectedness and low participation rate speak for the utilization of OLV technology, the lack of a designated self-governance structure in which such technology could be employed, call the potential of OLV for Australia's First Peoples into question. However, it is perceivable that once a legal framework of self-governance has been implemented, Australia´s Indigenous communities could benefit from the use of OLV. In such a case, Australia`s geographical make-up and the demographics of its Indigenous population would render traditional voting via polling stations and mail-in ballots burdensome for small communities with limited resources.

In the United States, the social geography of the Indigenous population is similar to that of Canada's First Nations and Australia´s First People. Only 25% percent of the Indigenous people in the USA still reside on the reserve land of their respective Nations,

which present a challenge to collective decision-making. While some of the more populace communities have remote voting channels in place to accommodate for their off-reserve electorate, it is perceivable that for smaller communities, elections present a considered administrative and financial burden. Although research into Indigenous communities on US territory has not revealed any issues with low participation rates, it is arguable that an employment of OLV technology could yield benefits for their degree of self-determination. More specifically, since US-American legislation on Indigenous self-governance is similarly comprehensive as the Canadian framework, it is perceivable that the utilization of OLV technology could lead to more effective decision-making that ultimately contributes to the advancement of self-determination and capacity building. Although it can be argued that OLV`s effect on self-determination will not be as far-reaching as for Canada's First Nations, as there is no need to achieve participation thresholds in order to extend Indigenous autonomy, OLV can nevertheless facilitate the participation of Indigenous individuals residing off community lands in the USA.

Finally, the legitimacy of Indigenous institutions in all three jurisdictions, be they federally recognized or only of regional character, could benefit from an increased political participation and interest of their respective populations. While previous studies have shown that OLV only leads to moderate increases in turn-out rates (Goodman & Stokes, 2018), there has been no research on the participation rates of electorates with significant shares of remote voters. The question whether or not the deployment of OLV can led to noticeable increases of participation rates is still lacking sufficient data backing and hence needs to be addressed in future research. As OLV is thought to make the voting process easier and more comfortable it is conceivable that OLV could lead to increase of participation among the sizable off-reserve population of Indigenous peoples. Further, taking on a pioneering role in the adaption of OLV and modernizing collective decision-making might be able to provide an identity-establishing and empowering process which ultimately benefits the self-determination and capacity building of Indigenous communities. However, it shall be noted that potentially positive effects of OLV, would predominantly impact Indigenous communities that were able to adopt western-style governance system. Hence, it seems unlikely that the employment of OLV has a similar effect in Communities for which western-style systems of governance have proven unviable in the past.

Nevertheless, a potential employment of OLV in an Indigenous context is dependent on a multitude of factors of which only a few have been explored in this paper. Among others, it still needs to be clarified to which extent participation of electorates with high shares of remote voters, such as Indigenous communities or migrant countries, is impacted by OLV. On a more general note, it needs to be stressed that issues and challenges experienced by Indigenous communities in Canada, the USA and Australia also exist in other settler-states where Indigenous peoples are striving for recognition and self-determination. Most notably, Indigenous peoples in New Zealand, Greenland, Peru as wells Columbia, have been granted different forms of self-governance and political representation and thus the potential benefits of OLV for those communities should be included in deliberations for future research.

Acknowledgments. This work has received funding from the European Commission under the auspices of PROMETHEUS Project, Horizon 2020 Research and Innovation action (Grant Agreement No. 780701).

References

1. Abele, F., & Prince, M. J. (2006). Four pathways to Aboriginal self-government in Canada. American Review of Canadian Studies, 36(4), 568-595.
2. Alcantara, C., & Davidson, A. (2015). Negotiating aboriginal self-government agreements in canada: an analysis of the inuvialuit experience. Canadian Journal of Political Science/Revue canadienne de science politique, 48(3), 553-575.
3. Alport, K., & Hill, L. (2006, September). Political exclusion and electronic conduits to civic (re-) engagement in Australia. In Australasian Political Studies Association Conference, University of Newcastle, Newcastle, NSW.
4. Australian Bureau of Statistics (2016). Estimates and Projections, Aboriginal and Torres Strait Islander Australians, 2006 to 2031. Retrieved June 1, 2020, from https://www.abs.gov.au/AUSSTATS/abs@.nsf/Latestproducts/3238.0Main%20Features602006%20to%202031?opendocument&tabname=Summary&prodno=3238.0&issue=2006%20to%202031&num=&view=
5. Australian Electoral Commission (2019). Indigenous Enrolment Rate. Retrieved June 1, 2020, from https://www.aec.gov.au/Enrolling_to_vote/Enrolment_stats/performance/Indigenous-enrolment-rate.htm
6. Budd, B., Gabel, C., & Goodman, N. (2019). Online Voting in a First Nation in Canada: Implications for Participation and Governance. In International Joint Conference on Electronic Voting (pp. 50-66). Springer, Cham.
7. Cherokee Nation (2020). Osiyo. Retrieved June 1, 2020, from https://www.cherokee.org/
8. *First Nations Elections Act*, S.C. 2014, c. 5. (2020). Retrieved August 2, 2020, from https://guides.douglascollege.ca/APA-6/legal
9. Ford, L. (2012). Locating Indigenous self-determination in the margins of settler sovereignty: an introduction. In Between Indigenous and settler governance (pp. 13-23). Routledge.
10. Ford, L., & Rowse, T. (Eds.). (2012). Between Indigenous and settler governance. Routledge
11. Gabel, C., Goodman, N., Bird, K., & Budd, B. (2016). Indigenous adoption of internet voting: a case study of Whitefish River First Nation. International Indigenous Policy Journal, 7(3).
12. Gabel, C., Bird, K., Goodman, N. J., & Budd, B. (2016). The impact of digital technology on First Nations participation and governance. The Canadian Journal of Native Studies, 36(2), 107.Flanagan, T., Alcantara, C., & Le Dressay, A. (2010). Beyond the Indian Act: Restoring Aboriginal Property Rights. McGill-Queen's Press-MQUP.
13. Goodman, N. J., & Pammett, J. H. (2014). The patchwork of internet voting in Canada. In 2014 6th International Conference on Electronic Voting: Verifying the Vote (EVOTE) (pp. 1-6). IEEE.
14. Goodman, N., & Stokes, L. C. (2018). Reducing the cost of voting: an evaluation of internet voting's effect on turnout. British Journal of Political Science, 1-13.
15. Hunt, J., Sanders, W., Garling, S., & Smith, D. (2008). Contested governance: culture, power and institutions in Indigenous Australia (p. 351). ANU Press.
16. Johnson, T. M., & Hamilton, J. (1994). Self-Governance for Indian Tribes: from paternalism to empowerment. Conn. L. Rev., 27, 1251.

17. Milke, M. (n.d.) Increasing number of Aboriginals choose not to live on reserves. Retrieved June 1, 2020, from https://www.fraserinstitute.org/article/increasing-number-aboriginalschoose-not-live-reserves
18. Milke, M. (n.d.). For many Aboriginals live is better off-reserve. Retrieved June 1, 2020, from https://www.fraserinstitute.org/article/many-aboriginals-life-better-reserve
19. Navajo Nation (2013). Navajo Population Profile 2010 U.S Census. Retrieved June 1, 2020, from https://www.nec.navajo-nsn.gov/Portals/0/Reports/NN2010PopulationProfile.pdf
20. Patterson, M., Macq, B. A., & Harvard, M. P. A. Commonwealth Machinery of Government in Aboriginal and Torres Strait Islander Affairs.
21. Perkins, R. (Ed.). (2008). First Australians: an illustrated history. The Miegunyah Press
22. Strommer, G. D., & Osborne, S. D. (2014). The history, status, and future of tribal selfgovernance under the Indian Self-Determination and Education Assistance Act. American Indian Law Review, 1-75.
23. Stubben, J., Ulrich, C. W., & McGrath, A. (2005). Native Americans and Government Policy. Nature, 415, 68-70.
24. Te Korowai o Ngāruahine Trust. (2020). Election 2020. Retrieved August 2, 2020 from https://ngaruahine.iwi.nz/election-2020/
25. Towell, N. (2019). Historic vote but only 7 per cent turned out for aboriginal poll. Retrieved June 1, 2020, from https://www.theage.com.au/politics/victoria/historic-vote-but-only-7per-cent-turned-out-for-aboriginal-poll-20191110-p5395o.html
26. U.S. Census Bureau (2010). The American Indian and Alaska Native Population: 2010. Retrieved June 1, 2020, from https://www.census.gov/history/pdf/c2010br-10.pdf
27. Victoria´s First People Assembly (2019). About the Assembly. Retrieved June 1, 2020, from https://www.firstpeoplesvic.org/about/the-assembly/
28. Vivian, A., Jorgensen, M., Reilly, A., Mcmillan, M., Mcrae, C., & Mcminn, J. (2017). Indigenous self-government in the Australian federation. AILR, 20, 215.

Tripped at the Finishing Line: The Åland Islands Internet Voting Project

David Duenas-Cid[1 2] [0000-0002-0451-4514], Iuliia Krivonosova[1] [0000-0001-7246-1373], Radu Serrano[1] [0000-0003-2183-0313], Marlon Freire[3] [0000-0003-4600-0746], Robert Krimmer[1] [0000-0002-0873-539X]

[1] Tallinn University of Technology, Ragnar Nurkse Department of Innovation and Governance, DigiGovLab, Akadeemia tee 3, 12618 Tallinn, Estonia
{david.duenas, iuliia.krivonosova, radu.serrano, robert.krimmer}@taltech.ee
[2] Kozminski University, Jagiellonska 57/59, 03-301 Warsaw, Poland
dduenas@kozminski.edu.pl
[3] Faculty of Engineering of University of Porto, Porto, Portugal
marlonfreirephd@gmail.com

Abstract. The Åland Islands spent years preparing an internet voting system, to be implemented for the first time in October 2019 for Parliamentary Elections. Despite this, the project was canceled the evening before the expected release date. In this paper, we explore the causes of this failure using a two-pronged approach including Information System failure perspectives and the approach to e-voting Mirabilis, focusing on organizational elements which provoked the decision not to use the system.

Keywords: Åland Islands, Internet Voting, System Failure, Organizations, Convenience Voting

1 Introduction: Three contextual questions

The Åland Islands were expected to introduce an internet voting system (IVS) during their last Parliamentary elections (October 2019), for expatriate voters, with the expectation to extend use of the same system to Municipal elections too and to all possible voters on the next possible occasion. Unexpectedly, internet voting was cancelled the day before it should have started. This paper explores this case approaching it from an Information System (IS) failure framework [18, 20], describing how interactions between the different stakeholders involved are a central element for understanding the final decision, and the e-voting Mirabilis frame, focusing on the organizational elements which provoked the decision to not use the system.

1.1 What are the Åland Islands and how does their electoral system operate?

The Åland Islands are a Swedish speaking autonomous region of Finland comprising around sixty inhabitable islands and around six thousand small rocky islands not suitable for human habitation or settlement. The archipelago is situated in the opening to the Gulf of Bothnia, bordering south-western Finland and central-eastern Sweden and is inhabited by 29,789 citizens, 11,743 of them living in the capital, Mariehamn. The autonomy of the Åland Islands was affirmed in 1921 by the League of Nations, through which Finland would protect and guarantee the continuation of the culture, language and traditions of the archipelago, and the Ålandic Government would have a say in foreigners acquiring franchise and land in the isles [4]. Similarly, the autonomy of Åland was reaffirmed by the treaty for admitting Finland into the European Union. Amongst other elements of self-government, the Åland Islands have their own Parliament (Lagting) and Government (Landskapsregering), elected in their own independent elections.

The uniqueness of Åland's status translates to implementation of its elections, relating to both the archipelago and Finland. The Åland administration is in charge of organizing Parliamentary and Municipal elections, and uses the electoral system of proportional representation, in which voters cast votes for a particular candidate, instead of for a party. Votes are transferred into seats using the D'Hondt method. Participation in elections is determined by acquiring the Right of Domicile in Åland, or after having been an inhabitant of any Ålandic municipality for one year prior to Election Day (the latter only applies for municipal elections). Legislation regulating these elections is covered in the Election Act for Åland [1], adopted by their Parliament in January 2019, on the occasion of introducing internet voting.

1.2 Why were the Åland Islands attempting to use internet voting?[1]

As the head of election administration, Casper Wrede describes [21], the idea to implement this voting channel in the Åland Islands was following the general worldwide trend and popularity of internet voting in the late 1990's, but the initial debate and research which produced the recommendation not to introduce the system until voter integrity and identification issues had been resolved. The idea of postponing introduction of a remote voting system in the islands was reinforced by the Finnish failure in their attempt to use electronic voting machines in 2008 local elections. Using internet voting was again introduced to political debating chambers after discussions on the reform of the electoral system in 2014 where, amongst other proposals, the suggestion was voiced to start introducing internet voting as an additional advance voting channel, only applicable for people living outside the Åland Islands. The introduction of internet voting was expected to be facilitated in two steps: 1) in 2019, only for expatriate, overseas voters in Parliamentary Elections; and 2) in 2023, based on the results of the 2019 experience, internet voting would become available for all voters [21]. Three main elements are mentioned as key factors triggering implementation of internet voting: convenience, turnout, and international projection.

Given the geographic location of the Åland Islands, it has been a long term goal of electoral authorities [19] to make voting more convenient for remote voters, as well as a traditional element considered as a driver for internet voting. The logic is based on two assumptions that 1) a general demand for convenience voting channels exists among the population; and 2) trust has been established towards remote voting channels, implemented in an uncontrolled environment. The Åland Islands have a legacy of convenience and remote voting channels being available to the population, since even before 2019 they were already offering, a number of voting channels consisting of 1) early voting at general voting locations not linked to the voter's place of residence, meaning that a voter could vote at any early voting polling station across the Ålands during an 11-day period; 2) early voting at care institutions; 3) Election Day voting; and 4) Postal voting for those who "are out of the country or are ill/handicapped and unable to vote in any other way" [2].

Advance voting channels are quite popular for the population and currently are used by around 1/3 of all voters who cast a vote (35% in 2019 and 2014 EU Parliament Elections)[3]. Said differently, Postal voting was not able to gain popularity due to the cumbersome procedure. During 2015 elections to the Legislative Assembly, around 150 people voted by post, constituting only 0.7% of all eligible voters [3], with about 10% of postal ballots arriving too late to be counted for the elections. Besides Postal voting, no other voting channels are available to voters residing overseas, outside of the islands.

---

1 For a more detailed development of this point, see our previous work on the preparation of Åland's internet voting project [5]

2 As described in the leaflet produced by the government of Åland to explain how Elections function to citizens: "Election on Åland, 18 October 2015".

3 Statistics and Research Åland, URL: https://www.asub.ax/sv/statistik/valet-europaparlamentet-2019

Åland does not have any embassies, representative agencies, or consulates and, as a result, voters do not have the option to vote in foreign missions. It is no coincidence that expatriates – 'absentee, overseas' voters - constituted a target group for initial use of internet voting.

The introduction of internet voting was also connected to projecting Åland to the outside world. In recent years, the Government of Åland provided IT-services for the public sector and contributed to overall digitization of the islands in various ways, through the public company ÅDA[4]. Both the development of internet voting and digitization of the islands are elements for creating a digital narrative of Ålandic identity and creating a positive image to promote the islands as a place where innovation thrives, and to highlight the positive impacts of their self-government.

In contrast, the reduced costs and time required are not amongst primary reasons for introducing internet voting. Cost savings were highlighted as a potential advantage for the long term [2, 3], under the assumption that a realistic assessment of cost-efficiency would only be possible once the system had been consolidated and the number of users increased. Regarding time savings, another dimension which is often highlighted as a potential positive outcome of using internet voting, the small size of the electorate would limit the potential impact of using the system in this regards.

1.3 Why are we writing this paper?

Discussions on the convenience of introducing internet voting to the Åland Islands were held for more than 20 years, intensifying during the last months of preparatory work. The first use of internet voting seemed to be ready for 'go live' on October 2019 but, at the very last minute and after the system had been set up, the use of internet voting was cancelled hours before elections opened. Our initial goal with this research was to approach the Ålandic case in order to observe their initial use of internet voting and conduct a cost-efficiency calculation of multichannel elections as we had already done for the case in Estonia [9, 10]. The fact that elections were cancelled when our team was already in-place and on site and we had already conducted extensive preparatory work (analysis of electoral law, preliminary interviews, initial study visit) made us direct our gaze towards analyzing the reasons for failure. We had the rare and unexpected opportunity to directly observe management of an electoral crisis and to interview the relevant actors. Our aim is to pinpoint the different elements which may have contributed to this final decision and try to extract lessons to be applied by other electoral managers and for implementing voting technologies. Failures help unveil processes which would remain hidden when assertions are made for systems that are successful [14], in this particular case, the complexity of electoral management and technological innovation and the interaction of different stakeholders.

To do this, we will propose and use a framework describing the Information System (IS) failure and interactions between the different stakeholders involved, relying on interviews conducted during our study visits to the islands.

---

[4] Åland Digital Agenda, see: www.ada.ax/

2 Stakeholders and Models of failure

Several studies targeted the issue of Information Systems (IS) failures [5, 6, 8, 12, 16, 22] over the last few years, and some proposed explanatory frameworks described the concept of IS failure and tackling the determinants for successful implementation [18, 20]. Definitions of an IS failure are generally in line with the two categories Ewusi-Mensah described [8]: either the system fails due to inability to perform to users' levels of expectations or due to the inability of producers to produce a fully-functional, working system for users. Sauer [18] considers the definition of an IS system failure as a system abandonment due to stakeholder dissatisfaction.

Sauer [18] developed an explanatory framework describing IS failure based on three key elements: 1) Supporters, 2) Project Organization and 3) IS. In it, he creates a triangle of dependencies between these three elements and there must be interaction between them to prevent eventual failure occurring. In his analysis, failure is presented as the outcome of the interplay between context, innovation process and support. Flaws occur if the context is inadequately addressed in the innovation process, and, if flaws should accumulate, the system loses support and faces risk of failure. Sauer also highlights the importance of system supporters and their perceptions regarding the system itself, rather than solely focusing on technological characteristics of the IS. In his interactive framework, the IS serves the supporters, while they in turn support the project's organization, and this last component innovates the system. According to Sauer's way of thinking, failure is seen as total abandonment of a system, which occurs when this triangle of dependencies breaks down. The role of Project Organization is seen as a middleman between stakeholders and the IS. What is more, the role of project organization is not limited to this: it also serves as "a mediator" between context, system and stakeholders.

Toots [20] iterated and adapted Sauer's model in order to develop an analytical framework for contextualizing and explaining factors which influence system failure for e-participation. The framework proposed by Toots consists of four key elements, focusing on: a) Innovation Process; b) Contextual Factors; c) Processes with contextual factors interacting with innovation process and stakeholders and; d) Project Organization, where they have the power to change influential contextual factors or if it can, to align the system to the context. The sub-elements of context include technology, organizational variables, and politics. In both frameworks mentioned above from Sauer and Toots, the elements complement one another, creating an interactive triangle of dependencies which allows us to understand the reasons for failure in exchanges occurring between different elements.

The Supporters in Sauer's model can be also viewed as stakeholders in Toots' model, but Toots includes a differentiation between "Project Organization" and "Stakeholders", based on the following logic: *stakeholders need the project organization to develop IS according to their interests* (p. 548). Therefore, Project Organization is viewed as a middleman between stakeholders and the IS, but the role is not limited solely to this, serving also as "a mediator" between context, system and stakeholders.

Even if Toots' efforts bring the causes for e-participation IS failure closer to the case we are analyzing, her model does not apply in full for understanding reasons for the Åland Islands' failure. Of the four key assumptions presented, only two of them are indicative for our case:

"1. Implementation of an e-participation system may be regarded as an innovation process characterized by uncertainty and susceptibility to changes in the context;

2. While contextual factors and changes are not the immediate cause of failure, context may constitute an important trigger for failure."

However, even these assumptions do not apply fully in our case, because Toots, following Macintosh's [13] definition of e-participation, explicitly distinguishes *e-participation from other e-democracy instruments such as e-voting* (p. 546). Ålands' IVS is a type of e-voting and thus could not fully benefit from applying a framework designed for e-participation, even if it is an excellent fulcrum for developing a new iteration of the model.

Some of the arrangements proposed for Toots' model relate to the role stakeholders play and the fact that the technology was never used. One of Toots' arguments is that if using an e-government system is not satisfactory for those who must use it, they will abandon its use and condemn the system to failure. In the case under analysis, the IVS was never used by stakeholders, so their impact is minor. On the contrary, the role of Project Organization and the Context in which the IVS is framed play a more relevant role, since the unequal discourses collected from Election Managers and Vendors highlight the existence of a difference in criteria towards the system. Also, some of the difficulties highlighted for developing IVS relate to adapting to the context, either legal or technological, of the Ålandic environment.

Taking one step forward, for iteration and for adapting Toots' framework to the case of the Åland Islands, we can detect different elements proposed in the framework mentioned: 1) Project Organization existed and managed creation, development and implementation of the system (here, also, a difference to Toots' model, since the role of Project Organization was not to innovate an IS which already existed, but to implement a brand new one); 2) the IS was in-place but never used; 3) the Supporters never accessed the system, but they could track developments through the media and further discard the system; 4) external contextual factors might have facilitated failure of implementation, such as the Data Protection Authority arriving late or integration of the IVS in the Finnish e-Government environment. Failure, in our case is transposed to being the decision to not proceed with internet voting, even with the system in-place, giving more relevance to the interaction between the different elements than to the IS itself.

Since some of the elements included in the frameworks proposed by Toots and by Sauer cannot be included in the same manner as has just been described, their models need to be iterated and adapted to the conditions of the case study. For this reason, we refer to the conceptual model analyzing e-voting implementation – the E-voting Mirabilis [11]. Including this allows enlarging the context in which the IVS is implemented. It focuses on four macro dimensions influencing application of ICT in elections:

- technological dimension;
- legal dimension;
- political dimension;
- social dimension.

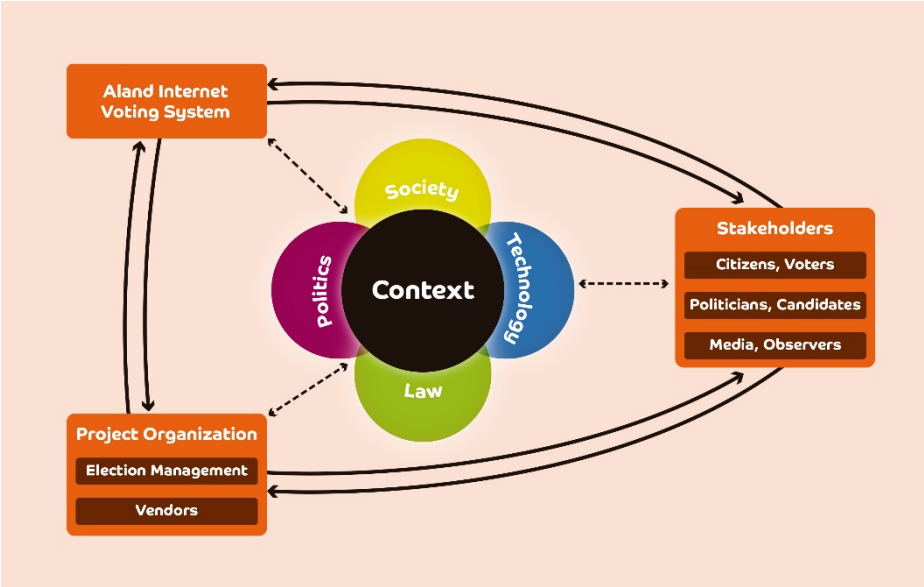
For the technological dimension, we consider what supporting infrastructure for internet voting was already in place (in particular, voter register and voter identification). For the legal dimension, we trace how the legal framework has been amended to adjust for internet voting, and whether it covers such aspects as secure processing of voters' personal data. For the political dimension, we analyze what groups of voters' internet voting was supposed to enfranchise, how the IVS was evaluated, and what was the overall political discussion on its introduction. The social dimension focuses on citizens' understanding and level of trust in IVS.

The E-voting Mirabilis is also helpful for stakeholder categorization, distinguishing between Voters, Politicians, Election managers, Vendors, and Media representatives and election monitors or observers. Combined with Toots' model, distinguishing between stakeholders and project organization, categorization should look like this:

- Stakeholders: Voters; Politicians; Media representatives and election observers;
- Project organization: Vendors; Election managers, Project managers.

Therefore, our theoretical framework builds on the conceptual model of the 'E-voting Mirabilis' [11] and an adaptation of the information system failure framework by Toots [20]. Based on these, we propose and use the "Mirabilis of internet voting System (IVS) failure". Toots' 'e-Participation System' was replaced by the IVS, and inside it we find Krimmer's e-voting components. All around, the 'contextual factors' (Toots) or 'four main macro dimensions' (Krimmer) *that explain the areas that influence e-voting deployment* [11]. Afterwards, Krimmer's five stakeholder groups which help to apply ICT to the electoral process, are grouped as either a 'Stakeholder' or 'Project Organization', according to Toots' framework and to their direct involvement in implementation of internet voting. Relationships between IVS, Project Organization and Stakeholders have remained similar (with some minor changes) to Toots' original diagram.

Fig. 1. Mirabilis of IVS failure.



In the context of the Åland Islands, project organization will be represented by the vendor (Scytl) and the organization responsible for the IVS procurement (ADA) and project management (Electoral Management Body). The rest of the actors will fit into the category Stakeholders: voters, government, election administration, parties, Data Protection Authority, and others. Stakeholders send requirements of IVS to project organization and provide them with the resources to fulfill those requirements. The IVS produced should satisfy stakeholders, otherwise, they will not use it. In other words, the IVS produced should meet the expectations of key stakeholders. In the context of the Åland Islands, this first and foremost concerns the stakeholders responsible for the decision on whether to start using internet voting. Already at the stage of modelling, we can observe that there is a possible mismatch between stakeholders' requirements formulated to project organization at the start of IVS development, and expectations which the final IVS should satisfy.

In this conceptual model, the context plays the key role: it shapes the demands of stakeholders, thus affecting the requirements they will send to project organization; it constrains or defines what is possible for project organization to fulfil the requirements; and the final IVS should serve the context.

3 Methodology

Data collection for developing this case study took place between March and December 2019. During this period, we conducted two visits to Mariehamn in teams of two researchers: 9-16 June and 14-22 October. Most of the interviews and observations included in this research were carried out during these visits to Åland, although we had

completed some preparatory interviews with the Ålandic Electoral Management Body (EMB) before the first visit, and arranged some digitally mediated interviews after the second visit. A total of 20 semi-structured interviews were conducted with EMB, ADA, Scytl, Central Committee for Elections, Data Protection Authority, local politicians, and voters. Many interviews had more than one respondent and some interviewees were contacted at different times. In all, a total of 20 people were finally interviewed, and the interviews were anonymized (see **Fehler! Verweisquelle konnte nicht gefunden werden.**). Data was analyzed using NVIVO qualitative data analysis software following a multi-stage inductive approach consisting of identifying a set of core themes during transcription (including, amongst others, 1) the electoral process, 2) government, 3) introduction of internet voting, 4) cancellation of internet voting and 5) voting organization) and the further coding of interviews based on the above themes. This inductive method was aligned with re-focusing of the research plan described below, allowing us to include the information collected in a context of crisis and relate our conclusions to the literature on Information Systems failure.

Table 1. List of interviewees, anonymized.[5]

Occupation	Date
Head of election administration	March, 2019
Head of IT-unit at Ålands Landskapsregering	June, 2019
System administrator at Ålands Landskapsregering	June, 2019
Legal Director, Government Offices, Unit for Legal and International Affairs	June, 2019
CEO of Åda Ab	June, 2019
Project Manager at Åda Ab	June, 2019
Data Inspector	June, 2019
Minister	June, 2019
Minister	June, 2019
Head of election administration (II)	June, 2019
Voter	October, 2019
Voter	October, 2019
Head of election administration (III)	October, 2019
Data Inspector (II)	October, 2019
Head of IT-unit at Ålands landskapsregering (II)	October, 2019
CEO at Åda Ab (II)	November, 2019
Worker at Åda Ab	November, 2019
Worker at Scytl	November, 2019
Worker at Scytl	November, 2019
Worker at Scytl	November, 2019

[5] The numbers in brackets refer to the number of times the person was interviewed

The case of the Åland Islands was selected due to the fact that they intended to implement internet voting for the first time and it represented a good comparison to research already conducted by the research team. The size of the country and administration allowed swift, effective communication and privileged access to data. Also, it would have covered a relatively unexplored dimension of electoral analysis, the costs of initial implementation of voting channels and their evolution over time.

We must point out here that the methodological plan was reframed during the research, due to cancellation of the IVS. Whilst applying the methodology for calculating costs, the initial plan followed on from previous research [3, 4] and research mentioned in a previous publication on the same case [5]. Cancelling implementation of internet voting took place during the research team's second visit to the Åland Islands, at a time at which the analysis of electoral law and modelling of the electoral processes had already been completed, as well as several interviews for understanding and describing the electoral system, its management and the costs involved. The fact that the research team was on-site during the cancellation, allowed them to observe and conduct interviews about management of the crisis, which were followed by a second round of interviews with the key stakeholders. Hence, this publication is the result of refocusing our research goals, given the opportunity to gather information on a critical case study relating to management of an electoral crisis due to cancellation of a voting channel. As a result of this, the interview design was modified (*the contents of the questionnaire)* in the course of the data collection process, paying special attention to integrating the different steps of data collection in the final analysis of the data.

The value of the data collected is derived from the opportunity and the uniqueness of the situation but, at the same time, it may involve some limitations given that it was not possible to plan such a methodological reconfiguration in advance. Amongst the strengths of our data collection process: 1) we developed a deep analysis of the electoral system prior to cancellation, and so were able to rapidly identify the key stakeholders to interview and the key processes to direct our attention to; 2) the presence of our research team on the ground allowed us to gather first impressions and reflections after cancellation and to experience the moment of cancellation on-site: direct observation of events provides us some interpretative clues which it would not be possible to gather through other data collection methods [7]. Amongst the limitations: we could not access some information on grounds of secrecy and confidentiality; the sources which, according to some discourses, could shed light on legitimacy of their claims.

4 Data Analysis

The context surrounding the Åland IVS looked promising for implementation of the new voting channel. At a socio-political level, no objections were raised against the system, the media did not pay much attention to implementation of the voting channel and no political party openly opposed it. There were more concerns about lowering the age of voters to 16 years of age for example, a reform discussed simultaneously to introduction of internet voting.

The overall political discussion on internet voting was fairly positive. Stakeholder evaluation varies from feeling *fairly optimistic* (I-1) to endorsements: *I always thought that this is a good thing, this is something we need to do* (I-13). The Parliament also has not seen much of the debate on internet voting, besides *some discussion on the security issues* (but) *in general, all parties in Åland responded positively to this voting channel* (I-13). Media outlets in the Åland Islands were not interested in internet voting, until almost right before voting started: *here is not big interest because everybody's focused on the transformation of the municipalities* (I-13), *I think, as a journalist, the interest in the elections will awaken in the end of August, when the campaign starts* (I-13)

This smooth political development crystalized in the decision that, during the first binding trial during the 2019 Parliamentary elections only expatriates (*overseas, absentee voters*) were eligible to vote via the Internet, *most of [the expats] are young people, they are studying or have been studying and stay for some years after studying* (I-3). This decision was considered as a clear improvement of voting conditions for expat voters (*a very strong urge from the younger generation to have a simplified voting procedure, possibly electronic* – I-5*)* since they could avoid the problems associated with using postal ballots to cast their votes (*last election 10% of our postal votes came back too late to count* – I-5).

As a result of which, *the whole new electoral act passed unanimously* (I-3). The legal dimension, in accordance with Krimmer [11], regulates how the electoral code can be changed in order to permit votes cast by electronic means and to provide the level of accountability required to the voter and should further: 1) provide the voter with the ability to see how personal data are processed; 2) include the principle of proportionality when handling personal data; and 3) serve as a guiding indicator. The Election Act for Åland, issued on May 2019, consists of 15 chapters and 122 individual sections (or articles), and defines all voting channels including postal voting, advance voting, Election Day voting and contains *new provisions on internet voting* (I-5). The legal dimension was further bolstered by the 'Registerbeskrivning'[6] or Privacy Policy (2019) which describes processing of personal data in connection with implementation of the Parliamentary and Municipal elections in Åland, including a description of the personal data required, its use during various stages of the election process, and the entities responsible which may interact with it, either directly or indirectly.

In order to specifically implement internet voting, the government *decided quite early [for] the procurement process*, that they *should buy a service, not the system* and that they *need[ed] someone else to run* it (I-10). To this end, the law and the procurement requirements were written in "parallel". As confirmed by an interviewee, this was *not ideal, perhaps theoretically. But in practice, it was quite good because we could adjust the wording and the law, according to what we experience, what is possible and how things should be* (I-10). This procurement process was run by ADA, resulting in a bicephalous organizational structure from the side of the government: ADA for managing the contract and the Electoral Management Body for management of elections, both interacting with the vendor.

---

[6] Available at: https://www.val.ax/sites/default/files/attachments/subject/behandling-av-personuppgifter.pdf Last accessed 15 June 2020

The development of IVS was accompanied by audits and evaluations. The checks and balances are prescribed by law: *the government […] should check and to have a third party to check everything, all the processes. So, we will also have somebody to check when the election takes place that everything is [OK]* (I-4). However, in June 2019, the independent body which would check and review the i-voting system had not yet been defined. The notions of who this independent body could potentially be were still vague: *It could perhaps be some authority from the Finnish state government, but it must be independent from the vendor and from the government… (…) it could also be some representatives from the Finnish authorities. Could be representatives from Estonia, for example. I mean, experts on internet voting, would be possible. Or it could be some audit company like KPMG, or whatever* (I-9).

At some point during development of the IVS, the Data Protection Authority of Åland became interested in auditing the process [17], for the following reasons: *Well, the biggest reason is because this is a new project, that has not been done before. And also, since this is a democratically critical process, pertaining to a lot of sensitive personal information or other special categories of personal information as in political opinions… since that kind of data is being processed […] That is the kind of processes that the data protection authorities should be auditing to make sure that they're safe (I-17)*. The arrival of the Data Protection Authority brought a new along with it player to the table; since it was not possible to conduct the audit on their own, it was necessary to outsource this to an external consultant for *auditing the security documentation sent by [the vendor]. And to see if they fulfilled the safety requirements* (I-17). The main findings of the audit, were that the Data Protection Impact Analysis (DPIA) has not been completed[7].

From a technological perspective, the IVS used the digital infrastructure provided by Finnish government – e-ID systems (e-ID Cards and Mobile-ID) – and private institutions (e-Banking), and consisted of main elements such as an e-ballot box, a list of voters and candidates, voter identification and authentication as well as vote verification.

During the development process of the IVS, a number of deficiencies were detected with the e-Identification system: in relation to integration *during the first pilot we found errors in the Suomi.fi implementation. So when I cast a vote, I was not successfully logged out from the authentication* (…) *And then they have corrected one mistake in Suomi.fi identification but there was still one loop, one error more.* (I-19); *In June already. And then in July again and in August, again* (I-15). Discovery of these problems was motivation for outsourcing a penetration test to an external vendor who dealt directly with the vendor in charge of IVS. The interaction between both vendors presented some problems in relation to accessibility to the source code of the voting system, since the vendor in charge of the penetration test was allowed access to the code but in the premises of the IVS provider, in a different country, and this option was not accepted

[7] For further details on the General Data Protection Regulation in the Alandic elections, see the work of Rodríguez-Pérez [17].

and delayed the auditing process[8]: *The argument that they were unable to access the source code for me is not a valid argument (…) they were invited… but even if they decided to not to come, this particular issue has been tested* (I-20).

According to the vendor's position, the problems detected challenged the development of the system: *during such integration, [or] maybe during any sort of customization or development, when you test, you find things, with the objective to correct them, fix them (*I-20*); The main challenge here is that, since we are not (…) Finnish, we don't have Finnish ID, so we have few test credentials that we can use in our tests to automate them (…) the personnel both from ADA and the government (were) very helpful as well in providing (them) to us (*I-20*)*. Problems were resolved according to their position, and the system was in place and ready to run during the elections as expected: *this issue with the verification of the digital signature. It was corrected, and was said that was corrected (by the vendor).*

The report from the vendor in charge of the penetration test was finished very late on (*we got the report from the security company very late, so it was not so much time to evaluate that and also to have a meeting with them and to discuss about – I-19*) and, even if the problems might have been solved, *we have not run the pilot from start to end (…), never ran it from beginning to end in a test environment (...), it doesn't feel right to do it (run the elections)* (I-19). The result was, cancellation of using internet voting at the very last moment.

5 Discussion and Conclusions

In the complex environment of electoral management, many factors can tip the scales towards failure if these are not perfectly aligned. In the case analyzed, even if there was a long process of preparation, training and a well-documented Electoral Management Body with members and experienced vendors, their joint efforts did not match up to initial expectations and the IVSs could not be implemented. It is not our role (nor our aim) to blame anyone for this outcome, but to understand the process in order to gain some useful knowledge and experience for others who aim to implement similar systems.

As we described, the context in which the IVS was to be implemented appeared to be quite friendly, accommodating, and welcoming: positive political discussions, lack of external agents discussing the suitability of the decision taken. The law was approved on time, as was the procurement process too. The problem, then, relied on the process of adjusting the IVS and the interaction between the members of the project organization, particularly with relation to timing. The accumulation of delays in some deliveries, responses and interactions, combined with organizing pilots during the summer period (in June and in August) reduced the time available for resolving problems detected (problems of integrating IVS into the Finnish e-ID system). Developing two Penetration Tests in a relatively short period of time and the presumed problems of collecting

[8] In this regard, it is worth noting that it was not possible to interview the vendor in charge of the penetration test due to a disclosure agreement. The views collected in this research might be distorted due to this issue.

data for the audits delayed the responses until a time when they were already redundant and no longer required. The Data Protection Authority's appearance late in June, and creating a new parallel legal and document audit probably superimposed a new layer of complexity onto implementing the system. Even if problems could have been resolved, as the vendor in charge of the IVS states, the authorities 'confidence in reliability of the system had already been damaged and the decision to cancel the elections could seem reasonable for those who were legally qualified to make it. Paraphrasing the idea expressed by Oostven and Van den Besselaar [15], *a voting system is only as good as the Administration* ("public" in the original version) *believes it to be.*

The key takeaway we can extract from this case is the relevant role which organization of the overall process plays in successful implementation. In the case under analysis, time management appears to be the main limiting factor for effective resolution of problems identified. We believe that with better time-management, four critical factors could have been managed more effectively: 1) the vendor could have resolved the problems detected in a timely manner, 2) project organizers would have had time to make sure these issues were resolved, 3) the final version of the system could have been tested, and hence, 4) the system could have been operated securely in real time. In addition to this, other factors, that without time constrictions could have had an irrelevant impact, in the case analyzed played an important role. Firstly, the bicephalous structure followed for project management divided the knowledge available on the side of project organizers, that is the technical knowledge separate from contract management and adding to the complexity of the process. Due to this fact, the process was slowed down at critical moments when a more directed management structure could have forced the vendor to react more swiftly in order to solve problems encountered. Secondly, the unexpected problems encountered related to the integration of the Finnish e-Identity system and their late resolution, damaged the trustability of the IVS. A faster detection and a smooth resolution of these problems could have walked the process to a different ending.

In contrast to the case proposed by Toots[20] in which the e-participation system failed due to a lack of a meaningful connection with stakeholders, in the case of the Åland Islands, failure originated on the side of interaction between project organization and the IVS itself, showing, in the end, the relevance of the organizational factor for creating, developing and implementing technological innovations.

6 References

1. Åland Culture Foundation: International Treaties and Documents Concerning Åland 1856 – 2009, http://www.kulturstiftelsen.ax/traktater/eng_fr/ram_right-enfr.htm.
2. Arbetsgruppen för Internetröstning: Rösta per Internet?, Mariehamn (2001).
3. Arbetsgruppen för översyn av vallagstiftningen: Slutrapport, Mariehamn (2015).
4. ÅSUB - Statistics and Research Åland: Åland in Figures, Mariehamn (2019).
5. Bartis, E., Mitev, N.: A multiple narrative approach to information systems failure: A successful system that failed. Eur. J. Inf. Syst. (2008). https://doi.org/10.1057/ejis.2008.3.
6. Beynon-Davies, P.: Information systems 'failure': The case of the London ambulance service's computer aided despatch project. Eur. J. Inf. Syst. (1995). https://doi.org/10.1057/ejis.1995.20.
7. DeWalt, K., DeWalt, B.: Participant Observation: A Guide for Fieldworkers. Altamira Press, Plymouth (2011).
8. Ewusi-Mensah, K.: Software development failures: anatomy of abandoned projects. The MIT Press, Boston (2003).
9. Krimmer, R. et al.: How much does an e-vote cost? Compared Costs per Vote in Multichannel Elections in Estonia. In: Krimmer, R. et al. (eds.) Electronic Voting. Third International Joint Conference, E-Vote-ID 2018. pp. 117–132 Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-030-00419-4.
10. Krimmer, R. et al.: New methodology for calculating cost-efficiency of different ways of voting: is internet voting cheaper? Public Money Manag. 0, 0, 1–10 (2020). https://doi.org/10.1080/09540962.2020.1732027.
11. Krimmer, R.: The evolution of e-voting: why voting technology is used and how it affects democracy. TUT Press, Tallinn (2012).
12. Lyytinen, K., Robey, D.: Learning failure in information systems development. Inf. Syst. J. (1999). https://doi.org/10.1046/j.1365-2575.1999.00051.x.
13. Macintosh, A.: Characterizing e-participation in policy-making. In: Proceedings of the Hawaii International Conference on System Sciences. (2004). https://doi.org/10.1109/hicss.2004.1265300.
14. Mitev, N.: Are social constructivist approaches critical? The case of IS failure. In: Howcroft, D. and Trauth, E. (eds.) Handbook of Critical Information Systems Research: Theory and Application. pp. 70–103 Edward Elgar Publishing, Cheltenham, UK (2005).
15. Oostveen, A.-M., Van den Besselaar, P.: Security as belief User's perceptions on the security of electronic voting systems. Electron. Voting Eur. Technol. Law, Polit. Soc. 47, May 2014, 73–82 (2004).
16. Poulymenakou, A., Holmes, A.: A contingency framework for the investigation of information systems failure. Eur. J. Inf. Syst. (1996). https://doi.org/10.1057/ejis.1996.10.
17. Rodríguez-Pérez, A.: My vote, my (personal) data: remote electronic voting

and the General Data Protection Regulation. In: Krimmer, R. et al. (eds.) Electronic Voting. Fifth International Joint Conference, E-Vote-ID 2020. Springer Cham, Cham (2020).
18. Sauer, C.: Why information systems fail: A case study approach. Alfred Waller Ltd. Publishers, Oxfordshire (1993).
19. Szwed, K.: Głosowanie elektroniczne na Wyspach Alandzkich – idea bez pokrycia czy realny scenariusz? PRZEGLĄD PRAWA Konst. 4, 50, 13–32 (2019).
20. Toots, M.: Why E-participation systems fail: The case of Estonia's Osale.ee. Gov. Inf. Q. Preprint, (2019). https://doi.org/10.1016/J.GIQ.2019.02.002.
21. Wrede, C.: E-voting in a Small Scale – the Case of Åland. In: Krimmer, R. et al. (eds.) The International Conference on Electronic Voting. E-Vote-ID 2016. pp. 109–115 TUT Press, Bregenz (2016).
22. Yeo, K.T.: Critical failure factors in information system projects. Int. J. Proj. Manag. (2002). https://doi.org/10.1016/S0263-7863(01)00075-8.

Internet Voting and Expatriate Voter Turnout

Micha Germann[1][0000−0002−5217−3240]

University of Bath, Bath BA2 7AY, UK m.germann@bath.ac.uk

Abstract. This short paper reports the results of ongoing research into the effect of remote internet voting on electoral turnout among Swiss citizens who live abroad. Preliminary results show that internet voting increases registered expatriate voter turnout by around 5 percentage points compared to mail-only voting. This suggests that internet voting is an effective method to increase turnout among citizens abroad.

Keywords: Internet voting · Turnout · Citizens abroad.

1 Introduction

In response to increasing geographical mobility, most democracies have extended voting rights to citizens who live outside of the state territory [1]. However, electoral turnout among citizens abroad is often very low [5]. In this short paper, I report first results from ongoing research into the potential of remote internet voting to increase expatriate voter turnout.

2 Case and Research Design

I examine the case of Switzerland. Between 2008 and 2019, a total of 15 Swiss cantons trialed internet voting for expatriates. In many of the 15 cantons, the trials extended over several years and covered a large number of electoral events [3]. Internet voting was generally popular among Swiss expatriates. In most of the trials, 50% or more of all votes were cast online [6]. All other votes were cast by mail, the only alternative voting option available to the Swiss abroad.

There are two key challenges with the identification of the causal effect of internet voting on turnout among the Swiss abroad. First, voter turnout is a function of many factors other than internet voting, and little data is available on, for example, the socio-demographic profile expatriate voter populations that could be used for statistical controlling. Second, data on expatriate voter turnout is available only for some but not other cantons and expatriate voter turnout can only be measured in terms of *registered* expatriate voters. The latter constitutes a key concern because internet voting may affect the registration probability, which would give rise to sample selection bias.

To simultaneously minimize the risk of bias due to confounders and endogenous sample selection, I choose to focus the empirical analysis not on the *introduction* of internet voting, but on its *continued provision (or not) after a prolonged period of prior availability*. This strategy minimizes the risk of sample selection bias, principally because internet voting is most likely to affect the

registration probability when it is first introduced. For causal identification, I exploit the circumstance that internet voting was suspended in several cantons in August 2015 due to the discovery of security issues with one of the internet voting systems in use at the time, the Consortium system. The suspension enables me to estimate the causal effect of internet voting on expatriate voter turnout using difference-in-differences estimation, which by design rules out many potential confounders [4, 2].

3 Preliminary Results and Conclusion

The sample consists of 8 cantons.[1] All 8 cantons had started to trial expatriate internet voting between 2008 and 2010, but are only observed starting in 2013 and until and including early 2019. 4 of the 8 cantons were affected by the suspension of internet voting in 2015, during which expatriates could vote only by mail. Internet voting resumed within 1 to 3 years in these cantons. The dependent variable is registered expatriate voter turnout in federal referendums and elections. There were a total of 23 federal electoral events during the period studied. The total number of observations is 184.

The causal effect is estimated using two-way fixed effects regression with standard errors clustered by canton to account for serial correlation. Two-way fixed effects regression generalizes the classic difference-in-difference estimator for two time periods to multiple time periods. I find that turnout among registered expatriate voters decreased by an estimated 5.2 percentage points as a result of the temporary suspension of internet voting ($p < 0.000$). Additional analyses suggest that pre- and post-suspension trends in registered expatriate voter turnout were close to identical in treated and control cases, thus supporting the parallel trends assumption. Overall, the preliminary findings of this study suggest that internet voting can markedly increase turnout among citizens abroad.

References

1. Caramani, D., Grotz, F.: Beyond Citizenship and Residence? Democratization **22**(5), 799–819 (2015)
2. Germann, M.: Making Votes Count with Internet Voting. Political Behavior (in press)
3. Germann, M., Serdült, U.: Internet Voting for Expatriates: The Swiss Case. eJournal of eDemocracy & Open Government **6**(2), 197–215 (2014)
4. Germann, M., Serdült, U.: Internet Voting and Turnout: Evidence from Switzerland. Electoral Studies **47**, 1–12 (2017)
5. Lafleur, J.M.: The Enfranchisement of Citizens Abroad: Variations and Explanations. Democratization **22**(5), 840–860 (2015)
6. Serdült, U., Germann, M., Mendez, F., Portenier, A., Wellig, C.: Fifteen Years of Internet Voting in Switzerland: History, Governance and Use. In: Téran, L., Meier, A. (eds.) ICEDEG 2015, pp. 149–156. IEEE, New York, NY (2015)

[1] Aargau, Basel-City, Fribourg, Geneva, Lucerne, Neuchâtel, St. Gallen, and Thurgau.

Voting Technology Developments in Estonia and France and COVID-19 Pandemic impacts in Ukraine

Planning the next steps for Estonian Internet voting

Sven Heiberg[1], Kristjan Krips[2,4], and Jan Willemson[2,3]

[1] Smartmatic-Cybernetica Centre of Excellence for Internet Voting
Soola 3, 51004, Tartu, Estonia
sven@ivotingcentre.ee
[2] Cybernetica AS, Mäealuse 2/1, 12618, Tallinn, Estonia
{kristjan.krips,jan.willemson}@cyber.ee
[3] STACC, Narva mnt 20, 51009, Tartu, Estonia
[4] Institute of Computer Science, University of Tartu,
Narva mnt 18, 51009, Tartu, Estonia

Abstract. This paper considers the current state of Estonian Internet voting, identifies its shortcomings with respect to the present-day threat landscape, and discusses possible mitigation measures. It turns out that the area requiring the most attention and introduction of new measures is electronic identity. We also propose and analyse an update to the current Estonian individual vote verification protocol allowing to use PC as a verification device in case voting would move to mobile platforms.

1 Introduction

Casting a vote via Internet (i-voting) has been an option in Estonia since 2005. In 2019 Parliamentary elections, about 44% of all the votes were cast via this medium[5]. The system has been a subject of debates and research scrutiny since the beginning of deployment.

The first full security study was composed by a group of Estonian researchers in 2003, and later updated in 2010 [3]. In 2011, several potential problems (e.g. an invalid vote and proof-of-concept vote manipulation malware) surfaced in practice [6]. To counter them, individual verification option was added to Estonian Internet voting in 2013 [9]. In 2014, Springall *et al.* published a study pointing out the need for better verifiability of system-level properties [14]. As a result, in 2017, a completely re-designed IVXV protocol was deployed in Estonia [7].

In 2019, the debate about Internet voting security intensified again in Estonia after a new political coalition was formed. The Minister of Foreign Trade and Information Technology called together a committee that produced a list of open action items to potentially work on[6].

One of the ideas listed was to introduce the option of casting votes from mobile devices. Since this would be quite a significant change in the current

[5] https://rk2019.valimised.ee/en/participation/participation.html
[6] https://www.mkm.ee/sites/default/files/content-editors/e-valimiste_tooruhma_koondaruanne_12.12.2019_0.pdf, in Estonian

Estonian i-voting infrastructure, a separate analysis effort was initiated by the State Information System Authority and State Electoral Office.

The current paper builds on the initial findings gathered during the analysis[7]. Even though the original focus of the study was on mobile voting, it turns out that most of the issues and recommendations are actually more general and hold for the PC-based voting as well. In Section 2, we will first cover the general electronic identity and OS level threats. Section 3 discusses a possible change that introducing mobile voting may bring along for verification. In Section 4, we list and categorise existing and newly proposed mitigation measures. Finally, Section 5 presents the conclusions and sets directions for future work.

2 General risks

2.1 Threat actors

We start our study by identifying the main classes of threat actors.

- **Civil hacktivist seeking publicity.** Such an attacker is not necessarily malicious, but can cause unintended problems as side effects of his activities.
- **Single candidate trying to get more votes.** Such an attacker acting alone has limited resources, and his attacks are not likely to scale too much.
- **Political party trying to increase the number of seats.** Such an attacker has medium level of resources. It may have significant organisational capability, enabling certain attacks (e.g. coercion) to scale quite well.
- **Organization that aims at influencing policy decisions.** Such an attacker may have financial or ideological motives. This category includes large national or international enterprises, and their methods range anywhere from media campaigning and lobbying to direct bribery.
- **Foreign state-level actor interested in gaining more control over the country.** Such an attacker may have significant resources and access to rare technical capabilities (like zero-day attacks against common OS-es).

2.2 eID level risks

The Estonian Internet voting scheme relies heavily on the electronic identity (eID) infrastructure. There are currently three main eID solutions in use in Estonia.

- **ID-card**, first launched in 2002, was historically the first one and is still in wide use. The latest generation of ID-cards also possesses Near Field Communication (NFC) functionality which provides an option of using it in the context of m-voting as well.

[7] `https://www.valimised.ee/sites/default/files/uploads/eng/2020_m-voting-report.pdf`

- **Mobile-ID (mID)**, first launched in 2007, relies on the mobile phone Subscriber Identity Module (SIM) card as the key storage and cryptographic coprocessor.
- **Smart-ID (sID)**, first launched in 2016, is a software-only solution making use of a specific cryptographic scheme [4] where the signature key is split between the mobile device and server.

Right now, only ID-card and mID are used for i-voting.

Regarding security aspects, we consider the user's personal computing environment to be the weakest point in the e-ID ecosystem (see Section 2.3). All the above e-ID solutions use OS input-output mechanisms to display confirmation codes, enter PINs, etc. While ID-card is theoretically also usable with a PIN-firewalled smartcard reader featuring a separate PIN-pad, such readers are not widely available on the market and hardly anyone is using them in Estonia. If an attacker is able to monitor PIN entry of some legitimate session, he will later be able to enter the same PINs in the session of his choosing.

The most serious implication of this threat is an attacker submitting a vote using a compromised e-ID environment without the voter noticing. This is a problem both in the scenario when the attacker changes the originally submitted vote by re-voting, and also when the voter did not intend to vote at all (which is her legal right in Estonia). To complete such an attack, the attacker would need to implement his own voting client. This is feasible as the protocol description is public, even though not always sufficiently detailed [11].

There are a few aspects of user behaviour that contribute to this problem.

- General low level of digital hygiene, e.g. installing software from untrustworthy locations, carelessly opening email attachments, failure to keep the OS updated, etc. Such failures are often required as presumptions for attackers to launch malware-related attacks. Raising digital hygiene awareness is one key measure in raising the security level of every kind of digital services, including i-voting.
- Usage scenarios where ID-card is left attached to the working terminal for extended periods of time, e.g. as a login token. Even though short periods of legitimate ID-card usage might already be sufficient to implement an attack, the login token scenario has more problems. Namely, it is typically implemented at an OS level by leaving the card's authentication environment open. As a result, applications (including malicious ones) do not need to have access to PIN1 in order to perform authentication.

In general, one of the core problems seems to be that e-ID tokens (be it an ID-card or an mID SIM) are getting too intimately connected to the computing platforms and OSes. On one hand, this connection is convenient for the users, but at the same time it increases the attack surfaces and time windows. Whether the corresponding risk level still remains acceptable depends on the application scenario and threat actor we consider.

In case of electronic voting (both PC and mobile platform based), the integrity risks become significant when the attacks start to scale easily. We estimate that out of the threat actors listed in Section 2.1, high-resource state level attackers have the capacity to attack mobile platforms in a sufficiently scalable manner.

There are several possible mitigation measures to both prevent and detect unauthorised use of voter's e-ID. We will describe and discuss these measures in Section 4.3.

2.3 OS level risks

It is very hard to rationally estimate security level of an operating system or a particular version of it. There are several folk beliefs either based on common knowledge ("A newer version of OS should have less vulnerabilities") or some sort of personal view ("iOS is more secure than Android"), but these beliefs are quite hard to quantify.

Concerning the more updated versions having less vulnerabilities we may look at published vulnerability reports[8]. However, even one critical zero-day flaw may be sufficient for a state-level attacker to implement an attack, so the number of unpatched vulnerabilities is not necessarily a good measure of security.

Claims about the comparative security level of specific OSes (say, Android vs iOS or Linux vs Windows) are even more questionable. In case of open development models (Android, Linux) the attackers have easier time of discovering weaknesses, but at the same time public disclosures also speed up patching. For example, the potential bounties paid out for a fresh Android zero-click exploits are even higher that those of iOS[9]. This may be interpreted as an indication that such Android exploits are more rare. However, as argued by Ross Anderson, open and closed development models produce software of roughly comparable security level in the long run [2].

One way how such argumentation could be backed up is by comparing the number of exploits for open source and closed source software. There is a recent study by RAND Corporation, analysing a rare dataset of exploits based on zero-day vulnerabilities [1]. The dataset contained 74 exploits for open source software and 123 exploits for closed source software. The analysis showed that the survival probability for both classes of exploits was roughly the same, with the average life expectancy of an exploit for closed source software being 6.93 years and 6.51 years for open source software.

Acquiring superuser credentials In general, malware has two ways of getting root access to a device. It can either escalate privileges by using an exploit, or abuse the access that an unsuspecting user provides. On a PC, users may choose to run software with root user permissions, but doing the same in Android or iOS is not so easy. While root access gives more freedom to the user, it also breaks

[8] See e.g. `https://www.cvedetails.com/`

[9] `https://zerodium.com/program.html`

the security model of the underlying platform and makes it easier to attack the device. Thus, some vendors are trying to prevent the user from getting root access. E.g. with each new release of iOS, Apple has taken more serious steps to prevent users from getting root access (called *jailbreaking* in iOS community). At the same time, Apple is also working to decrease the motivation of jailbreaking in the first place (e.g. by increasing configurability of the official iOS). As a result, the iOS jailbreaking community has recently decreased[10].

Android rooting, on the other hand, is still happening a lot. It can be classified into hard rooting and soft rooting [15]. The former is done by flashing the device with an executable having root permissions, while the latter is based on exploiting vulnerabilities. Malware applications typically abuse the method from the second category. While there are plenty of vulnerabilities for Android, recent studies show that developing a universal exploit is not common due to the fragmentation of hardware and software [12]. Thus, root exploits are usually tailored either for specific devices, models or operating system versions [5]. However, public sources do not reveal information about zero day vulnerabilities that are stored by governmental entities. The report [1] by RAND corporation revealed that the median lifetime of an exploit based on a zero-day exploit is 5.07 years. Given the long lifetime of the exploits, it is likely that the arsenal of stored exploits is quite large.

3 Verification

One of the problems that arises when Estonia would introduce voting on mobile devices is losing mobile devices as an independent verification platform. Independence of the voting and verification platforms is important for the verification to fulfil its primary goal of detecting whether a vote has been manipulated by a potentially malicious (e.g. malware-infected) voting device [9].

In principle, there are two possible solutions to this problem.

1. Retain verification from the mobile device as the only option, hoping that voters will be using different devices for voting and verification.
2. Allow verifying mobile votes from a PC-based verification app (possibly also allowing verification with mobile devices in parallel).

The first option has the benefit of making use of workflows and apps that the voters are already accustomed to. On the other hand, many voters could perceive the need to grab for yet another mobile device as a superfluous action that gives them little to no added value. Some voters could try to trick the system and verify the vote with the voting device (say, using mirrors to relay

[10] It is hard to find reliable statistics about the actual usage of jailbreaking, but there are several recent posts written by the developers expressing their rapid decline of motivation to continue working on the respective applications, see e.g. `https://www.idownloadblog.com/2019/10/26/coolstar-sileo-development-suspended/` and `https://old.reddit.com/r/jailbreak/comments/7iu0sx/discussion_can_we_please_find_someone_to_help/dr2m6nx/`.

the QR code to the camera, or perhaps finding some esoteric apps that fulfil the same purpose). Behaviour of the voters in this scenario is hard to predict at this point; it would require conducting a dedicated user study.

In order to consider the second option above, we propose using PC-based verification to be used in conjunction with mobile voting.

The current verification scheme (see also [9]) is displayed in Figure 1.

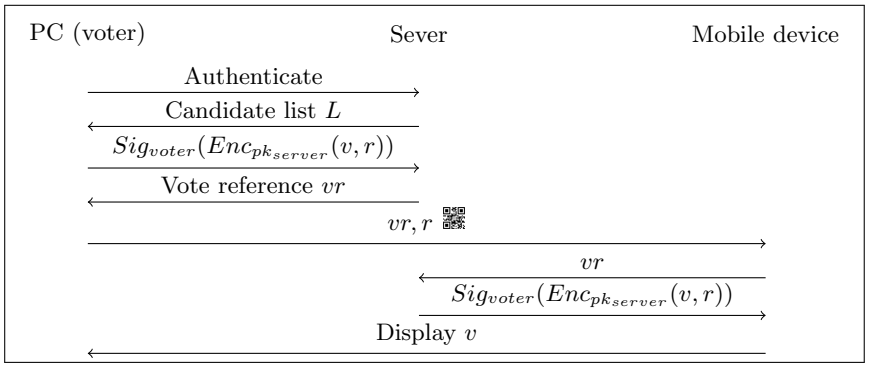


Fig. 1. Present Estonian voting and vote verification protocol

Note that the communication between the voter/PC and the mobile device is close range and optical. After the voting is over, the voting application displays a QR-code containing vote reference vr and encryption randomness r. The voter uses her mobile device to capture and decode the QR-code, downloads the corresponding encrypted vote from the server and decrypts it with the help of r (the latter operation being straightforward for the ElGamal encryption that the IVXV system currently uses). The vote is displayed on the mobile device screen for the user to inspect, again in close range and visually.

In case of voting with the mobile device, it would in principle be possible to display the QR-code on the mobile device screen and capture it with PC. However, not every PC has a camera, so we can not take this design path. Also, capturing the QR-code with the mobile device from the PC screen is a workflow familiar to the users, so we would like to retain it.

Of course, since the PC in the mobile voting scenario does not know vr and r, we have to change the content of the QR-code. Our proposal is to let the PC generate a one-time cryptographic (say, symmetric) key k and display it on screen as a QR-code. The mobile device will then capture and decode it, and use it to encrypt vr and r. The cryptogram will be sent to the PC that will decrypt its content and run the rest of the verification protocol in the familiar manner.

The resulting voting and verification scheme is is displayed in Figure 2

There are two main differences between the protocols presented in Figures 1 and 2.

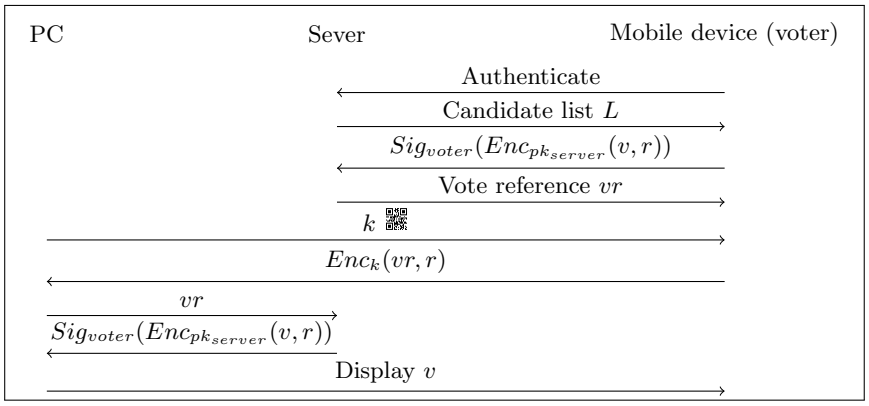


Fig. 2. Proposed Estonian voting and vote verification protocol

First, there is an extra cryptographic key k that aims at protecting vote secrecy by protecting confidentiality of the communication between the mobile device and verification PC. However, the voter has no assurance about the origin of the key – it may have been generated by an adversary in an attempt to breach the verification protocol.

Note that we are considering here the scenario where the verification PC is under the adversarial control. If besides the verification device either the server side or voting device would be malicious as well, we can not obtain meaningful security guarantees for the voter. Thus, it only makes sense to study the situation when the verification PC alone is malicious, but the voting device and server are honest.

Under such a scenario, there are two main kinds of attacks that the attacker can mount.

- Breaching privacy of the vote. This is an inherent risk present with any kind of verification that has to be accepted. This is similar to the present Estonian vote verification.
- Manipulating the verification process (manipulating the keys, delaying messages, etc.) leaving the voter with an impression that she voted for someone else. Note that under the attack model where only the verification PC is malicious, the vote was cast and recorded correctly. Thus, the adversary's activities will efficiently cause voter confusion, mistrust and general havoc. This is also what an attacker can do in the present scheme by manipulating the mobile verification device. There are standard measures designed for such a user experience (essentially, helpdesk will recommend the voter to use different voting and verification devices, and try again).

Thus we conclude that malicious manipulation of the verification device (and the key k along the way) does not make the situation worse compared to the present Estonian vote verification protocol.

The second difference between the two protocols is that there is an extra attack capability potentially gained by the adversary when he only manages to breach the voting device. Unlike the protocol in Figure 1, the protocol in Figure 2 is *active* in the sense that the voting device has to participate in initiating the verification process. Thus, the attacker could dynamically decide which voters to attack depending on whether they start with the verification process or not. For example, malware can delay delivery of the ballot and wait to see if the voting application is closed right after the vote has been cast via the user interface. In such a case it is unlikely that the voter verified the vote and thus malware can drop the vote without the voter noticing it. This kind of an attack could be prevented by introducing a feedback mechanism which notifies the voter once a vote has been successfully cast. This mitigation measure is discussed in Section 4.3.

4 Mitigation measures

In this Section, we are going to elaborate on possible mitigation measures for the risks listed in Section 2. Table 1 summarises the measures and classifies them according to their aim.

4.1 Awareness measures

Increase digital hygiene It is important to raise the general awareness level of digital hygiene. For example, it would have a significant positive impact if many citizens would regularly update their software to patch existing vulnerabilities. While such action is necessary, it won't be possible to educate every voter. In addition, state level actors are able to bypass antivirus software and have access to exploits built on top of zero day vulnerabilities [1].

Promote verification Currently, the rate of verifiers is about 4-5%[11], but the more there are, the smaller attacks we are able to detect [9]. In case individual verification would be more widespread, it would also act more as a preventive measure. When an attacker wants to change the election outcome, the attack should be executed silently. Thus, widespread individual verification can reveal if votes get dropped or changed by malware, and thereby deter such attacks from attackers who have to prevent detection. However, the current vote verification system is not able to detect malware that casts a re-vote which overwrites voter's original choice. The following mitigation measures also address the issue of preventing such malware from succeeding. As a possible new detection mechanism, establishing a feedback channel can also be considered (see Section 4.3).

[11] `https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia`

Table 1. Classification of mitigation measures based on their effect to i-voting.

	Prevention	**Detection**	**Recovery**
Increase digital hygiene	●	●	
Promote verification	◐[1]	●	
Introduce a feedback channel	◐[2]	●	
Do not support legacy mobile operating systems	◐[3]		
Obfuscation	◐[4]		
Add freshness notification to vote verification	◐[1]	●	
Prevent ID-card from being in the reader when not used	●		
Promote the usage of PIN-pad based ID-card readers	●		
Require both ID-card signature and mobile-ID/Smart-ID signature	●		
Analyse i-voting logs		●[5]	
Allow to re-vote on during i-voting period	●[6]		●[7]
Allow to re-vote on paper on election Sunday	●[6]		●[7]
Postpone i-voting			●[8]
Fall back to paper voting after a large scale attack			●[8]

● = measure is effective ◐ = measure is partially effective

[1] In case individual verification is widespread, the motivation for some types of attacks falls.
[2] A feedback channel may stop an attacker who wants to invisibly interfere.
[3] An attacker is able to run his own voting client on legacy operating systems.
[4] Client side restrictions can be bypassed if adversary has full control over the voting device.
[5] This is a system-wide measure to detect anomalies.
[6] The option of the voter re-voting limits the coercer's capability to ensure that coercion was successful.
[7] This is an individual recovery measure for voters who were coerced.
[8] This is a system-wide measure to recover from a malfunction or from an attack.

Prevent ID-card from being in the reader when not used Discourage the scenarios where it is required to leave the ID-card in the reader for extended periods of time, and practices where the card's authentication environment is left open on the OS level. In case voter's device is infected with malware and the voter is not using a PIN-pad-based ID-card reader, the malware could re-vote and thus overwrite the voter's initial choice.

Promote the usage of PIN-pad based ID-card readers Target e-ID solutions with better separated authentication factors. E.g. on the regular PC platforms, make use of PIN-pad-equipped ID-card readers. Without such a reader, malware could issue a re-vote right after the voter has voted as the ID-card is

still in the reader. Currently individual verification would not detect such an attack. For usage with mobile devices as terminals, NFC cards with integrated displays and PIN-pads could be utilised. In case individual verification would provide some integrity guarantees as described in Section 4.3, the NFC based vote signing could be a step forward. While the majority of smartphone users rely on Smart-ID and mobile-ID for daily interactions, the NFC based vote casting could offer a way to prevent malware from re-voting by more strongly separating the e-ID token and the main computing platform.

4.2 Existing measures

Analyse i-voting logs Log analysis can reveal anomalies which can be used to identify attacks. For example, it is possible to monitor when and how many times people vote, which e-ID tools and OSes they use, whether and when they verify their votes, etc. [8].

Allow to re-vote during i-voting period Allowing the voter to overwrite her vote by casting a new i-vote is a measure designed to prevent coercion. The rationale is that if the coercer knows that the voter can easily change her vote, his motivation to coerce (say, to pay for a vote) decreases. It is also possible to go to the polling station during the advance voting period to vote on paper. To enable this, the i-voting period currently ends two hours before the advance paper voting period.

Allow to re-vote on paper on election Sunday If the voter was coerced in the end of the i-voting and she was unable to attend the polling station during the extra two hours of advance voting period, there was no way to cast a re-vote up to 2019. However, this will change in 2021 when the i-voters will have the option to re-vote on paper during the election Sunday as well.

Postpone i-voting This is a legal measure that can be executed when a large scale attack is detected.

Fall back to paper voting after a large scale attack This is a legal measure that allows to cancel i-voting in case a large scale attack is detected that can not be mitigated by other means. That way the voters can be asked to vote on paper during the election Sunday. This is also one of the reasons why i-voting should be limited to the advance voting period.

4.3 Newly proposed measures

Introduce a feedback channel A feedback channel (say, an SMS or email) can be used to notify the voters about their act of voting. This measure would be useful in multiple scenarios. For example, the voter would be able to detect

re-voting malware or malware that drops votes based on a prediction on whether the voter is going to verify the vote. In the latter case, the voter could detect vote dropping attacks even without using the verification system, which would make it difficult for an attacker to avoid detection. This is relevant e.g. when considering the proposed verification scheme for m-voting discussed in Section 3. Similarly to individual verification, the feedback channel is mainly a measure to detect interference. However, as a side effect, it can also deter an attacker in the fear that the attack to be revealed. Again, similar to individual verification, we can hope that this deterrence will also act as an efficient prevention measure.

Introducing a feedback has actually been considered before in Estonia, and the main reason why it has not been implemented this far is the fear of making coercion attacks (e.g. vote buying/selling) easier. Thus, before taking a decision on whether to introduce such a measure or not, a wider analysis including also legal aspects should be conducted.

However, from the technical point of view we make the following observations about the potential coercion-enabling risk.

- Even if the coercer observes a voter during the voting session and demands to see her feedback channel (say, mailbox) during this session, the voter can still re-vote later.
- We assume that it is hard for the coercer to maintain physical access at many victims at the same time (most importantly, during the last minutes of the voting period). However, it is possible to demand virtual presence, say, in the form of e-mailbox passwords. To counter this threat, we can use an email redirection service that the voter can privately configure. In Estonia, there is the official @eesti.ee email redirection service that can be used for this purpose. Every citizen has an official government-supplied email address of the form `personalcode@eesti.ee` and is expected to redirect the emails from there to his/her personal email account.
- If the coercer is trying to get a control of all the digital channels of a voter, there must be sufficient evidence of this attempt so that the voter can turn to the law enforcement. However, the main rationale behind making use of the @eesti.ee redirection service is to lower the coercer's incentive to control the voter's main mailbox, since this gives the coercer no guarantee of detecting a revote.
- If the coercer is willing to go as far as ceasing all the e-ID means from the voter in an attempt of blocking her option of logging onto the @eesti.ee redirection service, he can use the same approach to block the voter's revoting ability already with the present system. However, the voter is still able to cast a paper ballot in case she has access to a passport, driver's licence or any other valid ID. Thus, from this point of view, introducing the notification feedback channel does not open significant new attack vectors.

Of course, in order for the feedback channel to be an efficient measure, care has to be taken in implementation. For example, it should be difficult for a piece of malware operating in the user's voting environment to block the feedback

channel. If mobile voting would be introduced, we have to take into account that people would probably vote and read SMSes from the same device. This would render SMS as a potential feedback channel weaker since malware operating on the mobile device could cast a vote without the user knowing, and also block the SMS that notifies the voter about the vote being cast on her behalf.

A possible drawback of the feedback channel measure is also the possibility for an attacker to generate havoc by sending out a lot of fake notifications. A possible countermeasure would be to include a statement signed with a key of the election organiser. In any case, also the legal impacts to voting freedom need to be assessed before such a measure can be implemented.

Add freshness notification to vote verification Estonian i-voting system gives voters the option to use individual verification. This means that the voters can check whether their vote reached the voting system. The existing implementation allows to verify the vote during a limited time window, which has historically been set between half an hour and an hour. Thus, after casting a vote, the voter has up to an hour to take a smartphone with a verification application and check whether her ballot reached the voting system. It is important to note that the voter is not able to check whether the ballot that reached the voting system will be counted in the tally as such an ability would also make vote selling easier.

The current verification system is optimised for being coercion resistant and thus verification does not reveal if a re-vote has been cast. Now, imagine what could happen when a voting device would be infected and controlled by malware. As noted in Section 2.2, malware can use voter's e-ID if it is directly connected to the infected device, by recording and re-using the PIN codes. The voter is physically not sufficiently fast to remove the ID-card from the card reader to prevent malware from accessing it (which can be done in a fraction of a second). Verifying the previous vote would still succeed with the current set-up.

However, the existing individual vote verification mechanism can be easily extended so that it would also provide a partial integrity check. The verification system could notify the voter during verification whether the given vote was overwritten or not. If the voter performs this verification after she has removed the ID-card from the possibly malicious device and does not use it any longer during the i-voting period, the voter can be sure that malware has not abused access to the ID-card. The verification time window is short and is probably not suitable for re-voting in case the initial vote was given under coercion. The coerced voter can re-vote later after the verification time-window has passed as then the coercer can not check whether the coerced vote was overwritten. In case coercion takes place during the last hour of the i-voting period, the coerced voter can fall back to casting a re-vote on paper (see Section 4.2).

Until ID-card's NFC interface is not used for other activities on a mobile device (nor over a regular smart card reader), the voter can be sure that malware does not have access to the ID-card. This measure only works when the voter

is careful and when malware can not rely on mobile e-ID solutions (i.e. mID or sID) to cast a (re)vote.

Require both ID-card signature and mobile-ID/Smart-ID signature The idea is to force the vote casting to depend on two independent devices. The vote should be accepted only if the timestamps of both signatures are within a certain time-limit. This measure would lower usability of electronic voting, but it may be an acceptable trade-off with increased resistance against malware attacks.

4.4 Other possible measures

Do not support legacy mobile operating systems It is possible to try to restrict the official voting client so that it would run only on up-to-date operating systems. However, the effectiveness of this measure depends on the capabilities and attack goals of the attacker.

The problem is that a really determined and resourceful attacker can develop a voting client also for an old and vulnerable platform where he can potentially run it without the user knowledge. This is doable as the voting protocol is open even though not always documented the best way [11]. If an e-ID utility is also accessible without the user knowledge, the attacker can mount an attack against vote integrity. Efficient measures against this threat include increasing the general level of digital hygiene and establishing a feedback channel as described above.

However, not supporting legacy OSes by the official voting client has a positive effect on vote privacy. If the voter only has access to the voting client on an up-to-date OS, it will be harder for an attacker to develop and deploy malware that would attempt to, say, read the user's screen during the voting session.

Obfuscation Obfuscation and malware detection measures only work against some attackers. State level actors and researchers have the capability to reverse engineer the voting application to detect which measures are used. Once the measures are known, they can be bypassed, assuming that the attacker has root access to the device. A good example of bypassing obfuscation and malware detection measures is given by Specter *et al.* in case of Voatz [13].

5 Conclusions and future work

In this paper we reviewed the current state of Estonian Internet voting, identified its shortcomings with respect to the present-day threat landscape, and discussed possible mitigation measures. Even though the original motivation of the research was the question about feasibility and the associated risks of mobile voting, the conclusions are more general and hold for PC-based i-voting as well.

The most serious attack vectors against Estonian Internet voting system include malicious unauthorised use of e-ID devices (ID-card, mobile-ID). With such an access, the attacker can cast a re-vote and thereby overwrite the choice of the voter. One of the strongest measures suggested against such a threat is end-to-end (E2E) verifiability that would allow every voter to verify that her vote has been correctly counted in the final tally. Unfortunately, such a strong notion of verifiability potentially conflicts with voter privacy and coercion-resistance.

For example, the (to-date the most comprehensive) report by Kiniry *et al.* studies a number of proposed E2E voting schemes and concludes that "No usable E2E-VIV protocol in existing scientific literature has receipt freedom when the voting computer is untrusted." [10]. Currently, the Estonian Internet voting scheme does not provide full E2E verifiability, but instead balances the verifiability and coercion-resistance requirements using a combination of individual verification [9], server-side auditability [7] and an option of re-voting. However, the search for a better balance is on-going and the question of introducing some form of E2E verifiability without increasing the coercibility level of the protocol too much is one of the main directions of future research.

There are still residual risks that E2E verifiability does not address. For example, if a citizen never intended to vote, but due to hostile take-over of her e-ID, the attacker manages to submit a vote on her behalf, the voter would not learn about this fact even if there is strong E2E verifiability in place. Thus, we propose to add an independent notification channel. The question which channel is the optimal one (also considering the implications on coercion-resistance) is still open and needs future study. This includes the need for additional legal analysis on such a measure.

We have also made two other new recommendations – adding freshness notification to the individual verification protocol, and requiring several independent e-ID tools to submit a valid vote. These recommendations also require further analysis from the coercibility and usability points of view, respectively.

In conclusion – any voting protocol suite is a complex set of mechanisms balancing between conflicting requirements. Improving one component may actually decrease the overall security level of the whole system. Thus, before implementing any of the above-mentioned measures, a holistic study of the whole suite needs to be conducted. This will be general direction of our future research steps.

Acknowledgements. This paper has been supported by the Estonian Personal Research Grant number 920 and European Regional Development Fund through the grant number EU48684.

References

1. Ablon, L., Bogart, A.: Zero Days, Thousands of Nights: The Life and Times of Zero-Day Vulnerabilities and Their Exploits. RAND Corporation (2017)

2. Anderson, R.: Open and closed systems are equivalent (that is, in an ideal world). In: Perspectives on free and open source software. MIT Press, Cambridge, MA (2005), https://www.cl.cam.ac.uk/~rja14/Papers/toulousebook.pdf
3. Ansper, A., Buldas, A., Jürgenson, A., Oruaas, M., Priisalu, J., Raiend, K., Veldre, A., Willemson, J., Virunurm, K.: E-voting concept security: analysis and measures (2010), https://www.valimised.ee/sites/default/files/uploads/eng/E-voting_concept_security_analysis_and_measures_2010.pdf
4. Buldas, A., Kalu, A., Laud, P., Oruaas, M.: Server-Supported RSA Signatures for Mobile Devices. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10492, pp. 315–333. Springer (2017)
5. Gasparis, I., Qian, Z., Song, C., Krishnamurthy, S.V.: Detecting android root exploits by learning from root providers. In: 26th USENIX Security Symposium (USENIX Security 17). pp. 1129–1144. USENIX Association, Vancouver, BC (Aug 2017), https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/gasparis
6. Heiberg, S., Laud, P., Willemson, J.: The Application of I-Voting for Estonian Parliamentary Elections of 2011. In: Kiayias, A., Lipmaa, H. (eds.) E-Voting and Identity - Third International Conference, VoteID 2011, Tallinn, Estonia, September 28-30, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7187, pp. 208–223. Springer (2011). https://doi.org/10.1007/978-3-642-32747-6_13, https://doi.org/10.1007/978-3-642-32747-6_13
7. Heiberg, S., Martens, T., Vinkel, P., Willemson, J.: Improving the Verifiability of the Estonian Internet Voting Scheme. In: Krimmer, R., Volkamer, M., Barrat, J., Benaloh, J., Goodman, N.J., Ryan, P.Y.A., Teague, V. (eds.) Electronic Voting - First International Joint Conference, E-Vote-ID 2016, Bregenz, Austria, October 18-21, 2016, Proceedings. Lecture Notes in Computer Science, vol. 10141, pp. 92–107. Springer (2016). https://doi.org/10.1007/978-3-319-52240-1_6, https://doi.org/10.1007/978-3-319-52240-1_6
8. Heiberg, S., Parsovs, A., Willemson, J.: Log Analysis of Estonian Internet Voting 2013-2014. In: Haenni, R., Koenig, R.E., Wikström, D. (eds.) E-Voting and Identity - 5th International Conference, VoteID 2015, Bern, Switzerland, September 2-4, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9269, pp. 19–34. Springer (2015). https://doi.org/10.1007/978-3-319-22270-7_2, https://doi.org/10.1007/978-3-319-22270-7_2
9. Heiberg, S., Willemson, J.: Verifiable internet voting in Estonia. In: Krimmer, R., Volkamer, M. (eds.) 6th International Conference on Electronic Voting: Verifying the Vote, EVOTE 2014, Lochau / Bregenz, Austria, October 29-31, 2014. pp. 1–8. IEEE (2014). https://doi.org/10.1109/EVOTE.2014.7001135, https://doi.org/10.1109/EVOTE.2014.7001135
10. Kiniry, J., Zimmerman, D., Wagner, D., Robinson, P., Foltzer, A., Morina, S.: The future of voting: end-to-end verifiable Internet voting (2015), U.S. Vote Foundation, https://www.usvotefoundation.org/E2E-VIV
11. Krips, K., Farzaliyev, V., Willemson, J.: Developing a Personal Voting Machine for the Estonian Internet Voting System (2020), submitted
12. Meng, H., Thing, V.L., Cheng, Y., Dai, Z., Zhang, L.: A survey of Android exploits in the wild. Computers & Security **76**, 71–91 (2018). https://doi.org/https://doi.org/10.1016/j.cose.2018.02.019, http://www.sciencedirect.com/science/article/pii/S0167404818301664

13. Specter, M.A., Koppel, J., Weitzner, D.J.: The Ballot is Busted Before the Blockchain: A Security Analysis of Voatz, the First Internet Voting Application Used in U.S. Federal Elections (2020), https://internetpolicy.mit.edu/wp-content/uploads/2020/02/SecurityAnalysisOfVoatz_Public.pdf
14. Springall, D., Finkenauer, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., Halderman, J.A.: Security Analysis of the Estonian Internet Voting System. In: Ahn, G., Yung, M., Li, N. (eds.) Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014. pp. 703–715. ACM (2014). https://doi.org/10.1145/2660267.2660315, https://doi.org/10.1145/2660267.2660315
15. Zhang, H., She, D., Qian, Z.: Android root and its providers: A double-edged sword. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. p. 1093–1104. CCS '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2810103.2813714, https://doi.org/10.1145/2810103.2813714

Some Things you may Want to Know about Electronic Voting in France

Chantal Enguehard [1], Camille Noûs [2]

[1] Laboratoire des Sciences du Numérique de Nantes,
2, rue de la Houssinière, BP 92208, 44322 Nantes Cedex 03, France
chantal.enguehard@univ-nantes.fr

[2] Laboratoire Cogitamus
camille.nous@cogitamus.fr

Abstract

In France, electronic voting machines are used in approximately 2 % of voting stations while the voters of the rest of the country still vote with ballot papers. We decided to focus on this small proportion to check whether the electronic voting machines in use in France could pose a problem to voting efficiency and reliability. We compared the accuracy of the voting process by checking whether the number of votes to signatures were equal in each polling station. We found that the gaps between votes and signatures were, on average, four to six times higher when electronic voting machines were used by comparison with polling stations where people votes with ballot papers. We discuss some hypotheses to explain this gap.

1 Voting process

In France, for political elections, the election process takes place entirely in a polling station. It usually concerns one election at a time. There are two ways to cast a vote:
– with a ballot paper: voters have the choice between several paper ballots, each ballot naming a candidate (figure 1). A voter takes several ballot papers naming different candidates. In a voting booth he or she selects one ballot to vote for a candidate, puts it in an envelope and then (outside the voting booth) slips this envelope into a transparent ballot box. If the voter does not want to choose one of the candidates, he or she can vote blank by slipping an envelope containing none of the proposed ballot papers.

Figure 1: Ballot Papers of the two candidates of the French presidential election in 2007

– with a voting machine: a voter chooses a candidate or the blank choice on the voting machine, the voting machine shows a message with the name of the choice, then the voter confirms his or her choice.
We did not consider expatriate French citizens because they may vote remotely (an option that is not allowed in France) and there are no voting machines in their polling stations.

Voters do not decide how they vote. The voting method (ballot papers or voting machine) is mandatory for all the voters of a polling station.
For the two ways of casting a voting, the process to ensure that a person votes only once is identical: each voter has to sign (with a permanent ink pen) a signing sheet made of paper, which completes the voting process.

2 Voting machines used in France

2.1 A brief history

Voting machines have been authorized in France since 1969 [16]. At that time, they were pure mechanical engines. They were subsequently imposed in the *communes*[1] where the government had suspicions of fraud. This first generation of voting machines were withdrawn little by little due to high costs and flaws related to these machines [18].
Around 2000, Direct Register Electronic (DRE) voting machines were introduced in *communes* that were allowed by the interior ministry to use electronic voting. The only criteria was a population greater than 3,500 persons. There is no national register of the *communes* that are allowed to use voting machine. However, we determined that at least 145 *communes* obtained this authorization [11].

These second generation voting machines are computers that are not connected to any network (except electricity).

2.2 Authorized voting machines

The interior ministry authorized different types of voting machines issued from three companies:
– **Nedap**: 2.07 F model [2], ESF1 model [4] and ESF1 (HW 1.06/2.01 – FW 4.02) model [5];
– Election Systems and Software: **iVotronic** model [6];
– **Indra**: "Point & Vote" model [1], "Point & Vote plus" model [3].

These agreements were delivered according to the compliance with technical regulations [12]. The verification of the compliance with these regulations must be done by some agencies accredited according to the European Cooperation for Accreditation and recognized by the Interior Ministry. These technical regulations list 114 requirements.

1 A *commune* can be a huge town such as Paris, or a small village. This is the first stage of the electoral process where election data are collected and then sent to the Interior Ministry.

However the accreditation reports are not public. At the end of its observation mission of the 2007 presidential election [14], the Organization for Security and Co-operation in Europe (OSCE) noted that "*transparency should be improved in order to enhance confidence in the electronic voting method, including through certification, auditing*". This observation mission also noted that the Nedap voting machines have been agreed although they did not comply with all the requirements:
"*However, as a consequence of a complaint in Vaucresson, the Ministry of Interior released an extract from the Bureau Veritas certification report on the NEDAP machines in a court proceeding. This extract contained the assessment of the NEDAP machines on a few of the 114 points. The extract indicated that the NEDAP machines did not fully comply with some criteria but that the discrepancies were minor. This raised concerns that the certification companies have too much discretion in determining the acceptable amount of variance in meeting each certification criteria and in determining whether some criteria are relevant at all.*"

In addition, the 2007 Presidential election has been associated to controversy, highlighted by the media over the 100,000 person-strong petition demanding a return to paper-based elections.

These circumstances caused the French government to form a working group on voting machines during the autumn 2007. Following the recommendation of this working group, the government froze the list of *communes* where electronic voting could be used. Since 2008, the government has been producing a new circular for each election (the first, being [13] with some recommendations about voting machines, as the secure storage of the voting machines since their reception[2], or the prohibition of technical operations without any person from the municipal staff (there is no precision about any technical skills that would be required to understand and control the technical operations)

Questioned by several deputies [7, 8] and senators [19, 20] the government expressed many times the necessity to enhance the legal and technical framework of the voting machines. Nevertheless, since 2008, the situation has been staying unchanged.

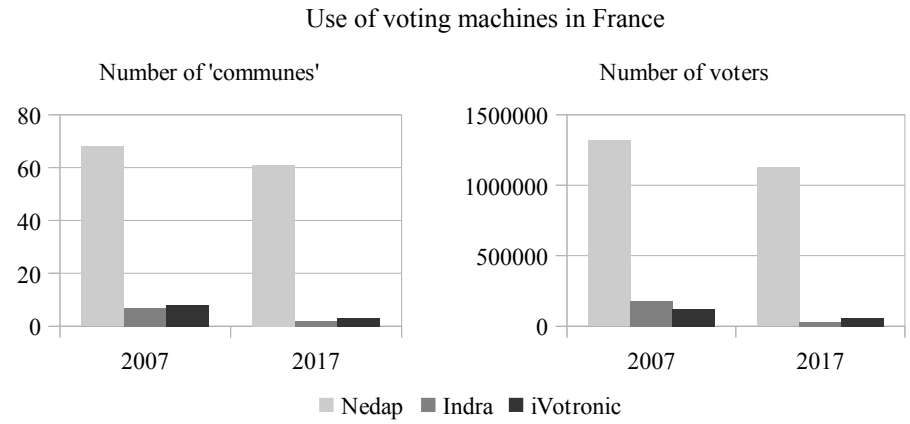
2.1 Utilization

The peak of use of these electronic voting machines was reached during the French presidential election in 2007 with 83 *communes* and more than 1,5 million voters using voting machines: 7 *communes* were equipped with Indra voting machines (180,000 voters), 8 *communes* with iVotronic voting machines (120,000 voters) while Nedap voting machines were used in 68 *communes* (1.3 million voters).
Following the 2007 controversy, 17 *communes* chose to stop the use of voting machines: 5 ones using Indra voting machines, 4 ones using iVotronic using machines, the 7 others using Nedap voting machines.

2 Because almost all the voting machines have been received several months or years before the first circular, this recommendation of a secure storage **since reception** can not be respected. Actually, it is not an obstacle because these circulars are not legally binding.

This evolution strengthened the dominance of Nedap voting machines in France with 92% of the electronic polling station (and 93% of the voters casting votes on a voting machine) (figure 2).

Figure 2: Use of voting machines in France in 2007 ans 2017



Nevertheless the rest of the voters carried on using paper ballots to vote. This partition of the voters into two sets according to the voting method (with voting machines or paper ballots) has allowed us to make a comparison between these two types of voting methods.

3 Objectives

Our main objective is to estimate whether voting machines are more or less precise than ballot papers for collecting votes.

3.1 Difficulties

It is particularly difficult to observe[3] electronic voting. Voters express their choice by clicking on a mouse, touching a screen or pushing a button. This physical movement, which is a force, is transformed into an electric signal which is encoded as information. This information will be transformed several times and finally aggregated to obtain electoral results. In addition, because the vote is secret (a vote is secret when nobody can know what an elector has voted), an observer should not be able to see an entire string of transformations. Therefore electoral results cannot be independently verified [17]. As these transformations occur at an electronic level, they cannot be directly observed (i.e. by the human eye). In addition, it is not allowed to

3 In this context « to observe » should be understood as the collection and study of data when voting occurs according to the OSCE manual [15].

carry out a forensic audit of voting machines [9] in France because of commercial and industrial privacy laws.

3.2 Votes and Signatures

To check whether a voter has voted, there is a signing sheet that each voter must sign. This rule applies to both paper and electronic voting. We therefore decided to use the difference parameter between the number of votes and the number of signatures to define a measure of accuracy of a polling station[4]. Theoretically, in a polling station, the number of signatures should be equal to the number of votes. Nevertheless, there can be differences between votes and signatures (either more or fewer signatures than number of votes). These differences are slight (around 1 difference per 1000 votes), nevertheless, they can be measured. We have therefore defined the K error rate as the number of differences between the number of votes and signatures per 1000 votes.

$$K = |\text{number of votes} - \text{number of signatures}| \,/\, \text{number of votes} * 1{,}000 \qquad (1)$$

This measure underestimates the loss or excess of votes or signatures because, for instance, it is possible for there to be both an excess vote and an excess signature in the same polling station. In such a case the gap between signatures and votes will be null. The only way to prove the existence of such compensation is by collecting the remarks written by the voting officials on the register.

Here is an example collected from the electoral register of the polling station 8 in Nevers (second round of the presidential election in 2017): "*Voter X signed without hitting 'validate' after having voted*"[5]. At this polling station, we therefore expect to get one vote less than signatures, but actually, the numbers of votes and signatures were equal (780 for the both). An explanation of this situation could be that another voter voted twice electronically.

The K rate is easily generalized to a set of polling station:

Let n be the number of polling station

Let V_i be the number of votes counted in the polling station i

Let S_i be the number of signatures counted in the polling station i

$$K = \frac{\sum_{i=1,n} | V_i - S_i |}{\sum_{i=1,n} V_i} * 1{,}000 \qquad (2)$$

4 In France, there is always only one ballot box or one voting machine in a polling station.

5 extract from polling station 8, Nevers, electoral register 6 May 2017. « L'électeur X a signé sans valider son vote »

4 Methods

The voting data were collected directly from the *mairies* (town halls) from various sources: official websites, photocopies of registers of the polling stations or tables sent by the *mairies* in answer to our request for voting data.
For each election round, we defined two sets of *communes*. The first set is referred to as SEV (Set of *communes* with Electronic Voting). The second set is referred to as SBP (Set of *communes* with Ballot Papers). Both sets were chosen for their compatibility in number of inhabitants per *communes* so as to avoid large *communes* being compared to small villages. In addition, we chose to count SBP *communes* that were situated in the same departmental region as SEV *communes*. These two sets were compared to each other according to the K error rate factor.
The data we collected was issued from 250 to 400 *communes*, (depending on the responses we received). After 2007, the data relates to almost all the *communes* where voting machines were used and until 40 % of the *communes* that met the criterion of belonging to the SBP set.

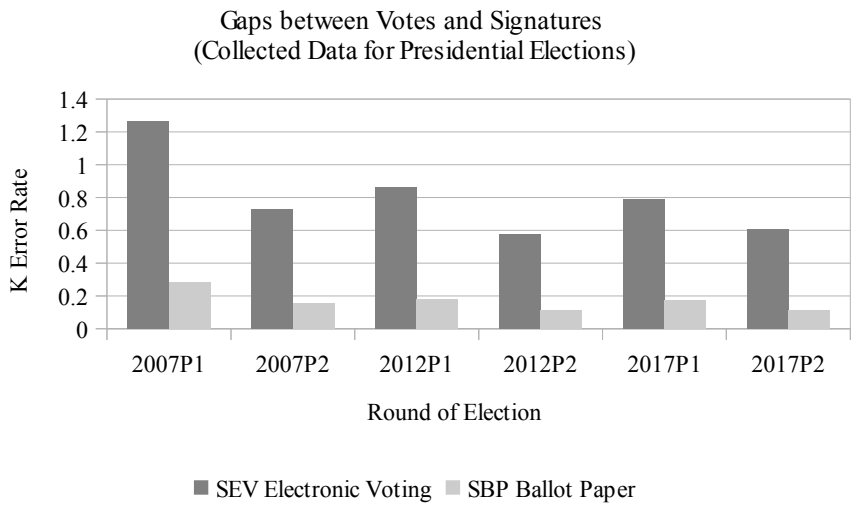
5 Results

We present the results obtained during the national elections only: presidential (table 1 and figure 3) and legislative (table 2 and figure 4) elections (to elect the representatives at the National Assembly) in 2007, 2012 and 2017. Presidential elections occur in two rounds while some legislative elections elect a winner in the first round.
For each set of data, the K rate was established according to the formula (2).

Table 1: Collected data and K rate for presidential elections in 2007, 2012 and 2017

		2007		**2012**		**2017**	
		Round 1	Round 2	Round 1	Round 2	Round 1	Round 2
SEV Electronic Voting	Number of votes	922,937	876,691	917,856	939,141	1,020,006	960,368
	Differences between votes and signatures	1167	638	790	542	802	583
	K error rate	**1.26**	**0.73**	**0,86**	**0.58**	**0.79**	**0.61**
SBP Ballot Papers	Number of votes	2,106,234	2,079,629	2,977,610	3,064,284	3,997,741	3,739,382
	Differences between votes and signatures	596	316	529	347	684	418
	K error rate	**0.28**	**0.15**	**0.18**	**0.11**	**0.17**	**0.11**

Figure 3: K rate for presidential elections in 2007, 2012 and 2017[6]



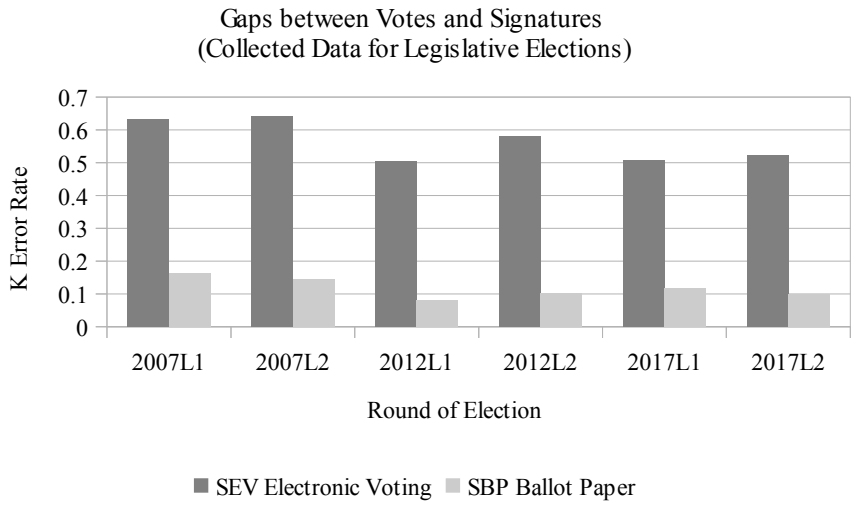
We observe that, for the French presidential elections between 2007 to 2017, the K error rate of the polling stations of SEV (electronic voting) is 4.4 times (second round of 2007 election) to 5.4 times (second round of 2017 election) higher than the K error rate of the polling stations of SPB (ballot papers).

Table 2: Collected data and K rate for legislative elections in 2007, 2012 and 2017

		2007		**2012**		**2017**	
		Round 1	Round 2	Round 1	Round 2	Round 1	Round 2
SEV Electronic Voting	Number of votes	494,964	327,373	660,482	584,195	633,407	536,410
	Differences between votes and signatures	313	210	333	339	321	281
	K error rate	**0.63**	**0.64**	**0.50**	**0.58**	**0.51**	**0.52**
SBP Ballot Papers	Number of votes	1,363,289	1,116,509	2,073,632	1,797,743	2,432,711	2,099,601
	Differences between votes and signatures	222	161	170	185	284	205
	K error rate	**0.16**	**0.14**	**0.08**	**0.10**	**0.12**	**0.10**

6 2007P1 is the first round of the 2007 presidential election, 2007P2 is the second round of the same, etc.

Figure 4: K rate for legislative elections in 2007, 2012 and 2017[7]



Our observations on legislative elections in France are similar to those on presidential elections : between 2007 to 2017, the K error rate of the polling stations of SEV (electronic voting) is 3.9 times (first round of 2007 elections) to 6.1 times (first round of 2012 elections) higher than the K error rate of the polling stations of SPB (ballot papers).

In addition, we note that, in both ballot paper and electronic voting methods, the gap between votes and signatures is more often than not an excess of votes (from 60% to 77%), than an excess of signatures. That is to say, there are more votes than signatures. This balance between vote excess and signature excess has been stable throughout the elections we have studied.
During the 2007-2017 period, we also investigated local and European elections, representing 14 election rounds and found similar results [10].

6 Analysis

Indisputably, the use of electronic voting machines jeopardizes the accuracy of the voting process in the sense that it increases difference between votes and signatures. This observation is quite puzzling and we have examined different explanations for this phenomenon.
When voting machines were first used, we considered the hypothesis that the discrepancy between votes and signatures at polling stations with voting machines were due to the novelty of this voting process. Voters, but also officials, may have found the new voting machines a challenge. But the discrepancy between votes and signatures continued as the years went by.

7 2007L1 is the first round of the 2007 legislative election, 2007L2 is the second round of the same, etc.

We looked for correlations between specific circumstances and the occurrence of differences between votes and signatures. Was the gap between votes and signatures higher when there were many voters, or many candidates, or many votes by proxy? None of these tracks gave significant results.

Our last investigation concerns an explanation expressed by an official: in polling stations with ballot papers, an excess in votes may have been reduced by the retraction of some blank votes in order to balance out the number of votes and signatures. This illegal manipulation of votes cannot be achieved so easily when voting machines are used. We studied this possibility by suppressing excess votes issued from the SEV sets. First it appeared that this attempt was not sufficient enough to explain the discrepancy we measured: even with no more votes in excess, the K error rate of the SEV sets were still 1,3 to 1,8 higher than those of the SBP sets. In addition we have observed in the initial data a *quasi* regular distribution of the number of votes with no matching signatures (excess votes) compared to the number of signatures with no matching vote (missing votes): on average there are two excess votes for each missing vote (figures 5 and 6). Suppressing all the excess votes would disturb this proportion since the excess votes would completely vanish (0%). Thus, the hypothetical removal of blank votes in polling stations with ballot papers does not explain the difference between votes and signatures when using voting machines.

Figure 5: Proportion of excess votes during three presidential elections

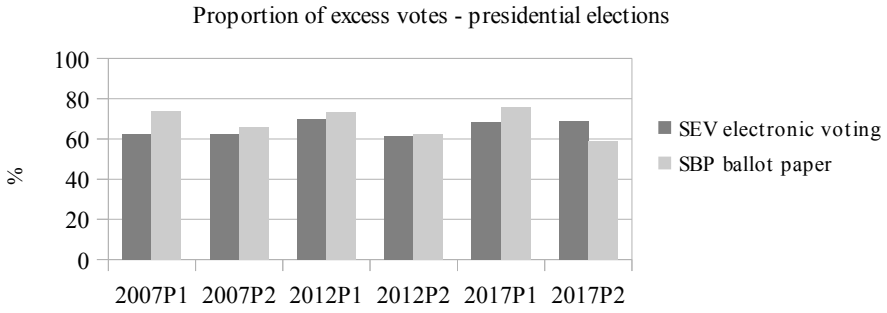
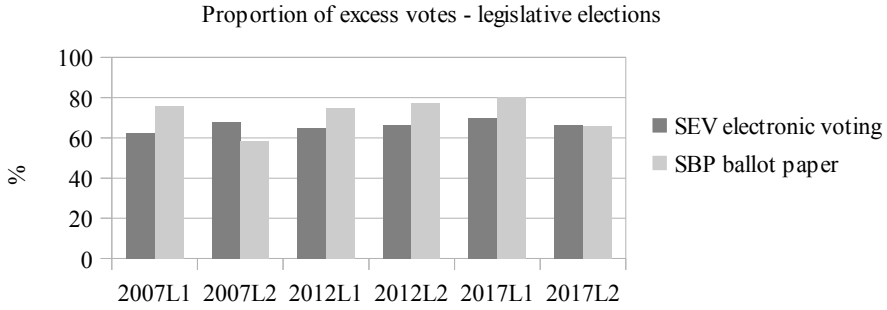


Figure 6: Proportion of excess votes during three legislative elections



Finally we examined this discrepancy with the help of the remarks written in the polling station minutes where there was a gap between votes and signatures. Sometimes the official reported that some voters had voted several times, some had not cast a vote, and some forgot to sign the signing sheets. In most cases, they could not explain the gap between votes and signatures. We cannot exclude that the voting machines were the cause of some of these gaps. All software handles errors and exceptions and some treatments may nullify some votes, or even destroy votes or create votes. Even soft error could occur [22], as the spontaneous inversion of the value of a bit of a voting machine in 2003 in Belgium [21].

8 Conclusion

Among politicians, there is perhaps a common belief that voting machines cannot make mistakes and would improve the collection of votes. The results we obtained and presented in this paper show that accuracy did not improve when voting machines were used in France.

These surprising results were obtained due to the following conditions

– some voters used voting machines while others cast votes with ballot papers;

– the collection of signatures was independent of the vote casting process;

– the possibility to collect a large amount of detailed data.

It would be interesting to repeat this study in other countries where these conditions could be met.

A DRE voting machine is not a neutral technical object, it is a computer which processes information. Because of the secrecy of the vote, such information processing cannot be tracked openly. Conversely, ballot papers and a transparent ballot boxes are neutral technical objects because they don't transform the ballot papers into something else. If the ballot box is watched consistently, the ballot papers that are counted are exactly the same as those which had been collected. In addition, the counting process can take place publicly under the eyes of the general public and, by the way, strengthen the confidence of the voters in the electoral results.

References

1. Arrêté du 7 mai 2004 portant agrément d'un modèle de machine à voter. INTA0400343A. (2004).
2. Arrêté du 27 décembre 2004 prorogeant l'agrément d'un modèle de machine à voter. INTX0407923A. (2004).
3. Arrêté du 20 avril 2005 portant agrément d'un modèle de machine à voter. INTX0508365A. (2005).
4. Arrêté du 26 octobre 2005 modifiant l'arrêté du 8 mars 2005 portant agrément d'un modèle de machine à voter. INTA0500753A. (2005).
5. Arrêté du 12 avril 2007 portant agrément d'une machine à voter. INTA0750387A. (2007).
6. Arrêté du 15 février 2008 portant agrément d'une machine à voter. IOCA0804110A. (2008).
7. Assemblée Nationale. Question écrite n°52045 de Dominique Le Mèner au ministère de l'intérieur, 13ème législature, JO (16 juin 2009, page 5763), JO (11 août 2009, page 7943). (2009).
8. Assemblée Nationale. Question écrite n°52045 de François de Rugy au ministère de l'intérieur, 13ème législature, JO (16 février 2010, page 1586), JO (9 juin 2010, page 7340). (2010).
9. Bishop, Matt. Peisert, Sean. Hoke, Candice. Graff, Mark. Jefferson, David.: E-Voting and Forensics: Prying Open the Black Box. EVT/WOTE'09, Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, Montreal, Canada, August 10-11 (2009).
10. Enguehard, Chantal. Graton, Jean-Didier.: Machines à voter et élections politiques en France : étude quantitative de la précision des bureaux de vote. Cahiers Droit Sciences et Technologie, n°4, p.159-198, Presses Universitaires d'Aix-Marseille. (2014).
11. Enguehard, Chantal.: Communes ayant obtenu l'autorisation d'utiliser des machines à voter en France (2004-2013). Observatoire du Vote. (2013).
12. Ministère de l'intérieur, de la sécurité intérieure et des libertés locales: Règlement technique fixant les conditions d'agrément des machines à voter. Annexe à l'arrêté du 17 novembre 2003. INTX0306924A. 17 novembre (2003).
13. Ministère de l'intérieur, de l'outre-mer et des collectivités territoriales: Utilisation des machines à voter à l'occasion des élections municipales et cantonales des 9 et 16 mars 2008. INTA0800023C, 1er février (2008).
14. Organization for Security and Co-operation in Europe / Office for Democratic Institutions and Human Right: France Presidential Election 22 April and 6 May 2007 OSCE/ODIHR Election Assessment Mission Report". 4 October (2007).
15. Organization for Security and Co-operation in Europe / Office for Democratic Institutions and Human Right: Election Observation Book. sixth edition. (2010). ISBN 978-92-9234-778-9.
16. République Française. Loi n°69-419 du 10 mai 1969 modifiant certaines dispositions du code électoral. Journal Officiel de la République Française. 11 mai (1969).
17. Rivest, Ronald L.: Requiring Software Independence in VVSG 2007: STS Recommendations for the TGDC. National Institute of Standards and Technology. (2006).
18. Sénat de France: Réponse du ministère de l'Intérieur à la question écrite n° 20558 de M. Georges Gruillot. Journal Officiel Sénat, page 971. 27 mars (1997).
19. Sénat de France: Réponse du ministère de l'Intérieur à la question orale n° 0107S de Mme Agnès Canayer. Journal Officiel Sénat, page 3464. 9 novembre (2017).
20. Sénat de France: Réponse du ministère de l'Intérieur à la question écrite n° 03181 de M. Bernard Bonne. Journal Officiel Sénat, page 512. 8 février (2018).

21. Sénat et chambre des représentants de Belgique: Rapport concernant les élections du 18 mai 2003. numéro 3-7/1 (Sénat) Doc 51 0001/2 (Chambre). (2004).
22. Ziegler, J. F. Lanford, W. A.: Effect of Cosmic Rays on Computer Memories. Science. 206 (4420): 776–788. November, 16. (1979).

Acknowledgements

Warm thanks to my friends Alexandra Reynolds and John Johnson who kindly reviewed this article.

Cyberattacks, Foreign Interference and Digital Infrastructure Robustness: How to Conduct Secure Elections in the Transatlantic Community Amid the Coronavirus Pandemic

David Levine[1], Beata Martin-Rozumilowicz[2]

[1]Alliance for Securing Democracy Election Integrity
[2] International Foundation for Electoral Systems

Abstract. Considering several key elections this autumn, this paper seeks to explore the connection between recent cyberattacks, various foreign influence campaigns and states' digital infrastructure robustness during the COVID-19 pandemic. The pandemic has introduced greater complexity to an already challenging task of conducting secure, democratic elections. This paper seeks to help democracies conduct more secure elections during these unprecedented times, examining the challenges many countries face with securing elections, including those posed by foreign influence prior to the advent of the pandemic. Then it looks at how the pandemic has made securing elections even more difficult by examining how some election officials' responses to the coronavirus have create new vulnerabilities in election infrastructure. Finally, it provides possible solutions to address the election security threats that have been exacerbated by this crisis.

Keywords: Coronavirus, Elections, Cyberattacks, Robustness

1 Introduction

With several key upcoming elections across the transatlantic region, this paper seeks to explore the connection between recent cyberattack, various foreign influence campaigns and states' digital infrastructure robustness during the COVID-19 pandemic. The coronavirus pandemic has introduced an additional layer of complexity into an already challenging task of conducting secure, democratic elections. Prior to the pandemic, many democracies were working to try and secure their elections from foreign adversaries, often with limited budgets. These challenges have only grown more acute as a result of the pandemic.

Since COVID-19 arrived, much attention has, correctly, been focused on how to administer elections in a manner that reduces the likelihood of contracting the virus. However, after reviewing many elections held in Europe and the United States ("transatlantic region"), including several during the pandemic, we believe that more can and should be done to secure them, particularly since both the foreign interference

and pandemic threats show no signs of dissipating. While this paper seeks to help democracies, particularly those in the transatlantic region, conduct more secure elections during these unprecedented times, it is not tailored to any one specific country or election. Instead, it examines challenges many countries have faced with securing elections, including those posed by foreign influence prior to the advent of the pandemic.

Then it looks at how the pandemic has made securing elections even more difficult by examining how some election officials' responses to the coronavirus have create new vulnerabilities in election infrastructure. Finally, it provides possible solutions to address the election security threats that have been exacerbated by this crisis. The integrity of future elections held during COVID-19 could go a long way towards bolstering or undermining citizens' trust in democratic elections.

2 Challenges to Secure Elections pre-COVID-19, including Foreign Interference

The new difficulties of the pandemic have not displaced the challenges that election officials throughout the transatlantic community faced before the coronavirus arrived, including the threats posed by malicious foreign actors. While many of these assets, such as online voter registration systems, electronic pollbooks, electronic voting devices, and election night reporting websites, were initially deployed with the aim of making elections easier to participate in and administer, some have also introduced additional points of vulnerability for malicious attacks that need to be identified, mitigated and managed.

Ukraine's 2014 presidential election is a case in point. Here, a three-pronged attack was launched on the eve of the presidential election against the Central Election Commission (CEC) website, which helps broadly disseminate the election results. Hackers infiltrated CEC computers and deleted key files, rendering the tabulation system inoperable; breached the CEC's computer's network infrastructure and released many of the Commission's emails and other documents onto the Internet; and installed a "virus" covertly on CEC computers that nearly resulted in a fringe candidate, Dmytro Yarosh, being portrayed as the winner. Instead, the attack was caught and mitigated before the results were publicly presented, but not before Channel One in Moscow broadcast false results with a faked CEC webpage purporting Yarosh had won the election. Even though election night reporting provide unofficial results, the public can perceive them as official, which is why providing assurance to the public that the election night reporting data is accurate and protected is so critical to the public's confidence in elections.

The United States 2016 presidential election further underscored the importance of securing elections from foreign interference. Beginning in 2014, Russia began attacking the United States in an effort to influence the 2016 election, and more broadly undermine the integrity of U.S. elections and American confidence in democracy. It made efforts to influence the election through disinformation using social media and other tactics; conducted cyber intrusion operations against entities, employees and

volunteers affiliated with a presidential candidate's campaign as well as both convention committees; and targeted U.S. election systems, conducting cyberattacks against private technology firms that make election software, as well as the election infrastructure at the state and local level. While there is no evidence that Russian actors altered vote totals in the 2016 election, it targeted many states' voter registration systems and public election websites, and was in position to delete or change voter data in at least one state.

While France's 2017 presidential election is a success story in how to counter foreign electoral interference pre-pandemic, it also illustrated many of the steps a state must take to successfully defend its elections from a foreign adversary. For example, France's National Cybersecurity Agency (ANSSI) offered to meet with and educate all campaign staff on the risks of cyberattacks and disinformation early in the election cycle, even holding an open workshop on cybersecurity in October 2016. In December 2016, the minister of defense announced the creation of a cyber command agency composed of 2,600 cyber experts. Shortly after President Macron's political movement En Marche! announced that it was the target of an orchestrated attack, in February 2019 the Ministry of Foreign Affairs, at the behest of the head of ANSSI, announced cessation of electronic voting for citizens abroad because of the high risk of cyberattacks.

3 The Difficulty in Securing Elections under COVID-19

The election infrastructure is comprised of physical, cyber and human assets, all of which are susceptible to intentional and unintentional threats. Physical assets are things such as ballots, voting locations and storage facilities that support or provide protection for election activities. Cyber assets are hardware and software such as voter registration systems, election-night reporting websites and electronic voting equipment. Human assets are personnel with unique training, experience, knowledge, skills, and authorities, whose absence could hinder election activities. They include election officials, information technology and security staff, election equipment vendor employees, and temporary staff such as poll workers. Since the onset of the COVID-pandemic, securing each of the above assets has become increasingly difficult.

3.1 Human Assets

As of August 19, 2020, there had been 21,989,366 confirmed cases of COVID-19, including 775,893 deaths reported to the World Health Organization, and these numbers not only affect society at-large, but elections as well. For example, in many countries, such as the United States, the people who have traditionally administered elections at polling places are often older workers who are more susceptible to COVID-19. These workers often help verify a voter's eligibility, assist the voter with casting a ballot, and protect the voted ballots from any untoward behavior.

While many countries have used an array of measures to try and limit the risk of spreading COVID-19 during in-person voting, there is a still significant concern that

large numbers of pollworkers who have historically helped conduct elections will not do so again until the virus is brought under control. With the United States continuing to see high COVID-19 case numbers and coronavirus infections again rising in Europe, it is important that every democracy in the transatlantic community recruit and train a surplus of poll workers so that it can adequately service voters during an election, even if many poll workers drop out on short notice.

Such a concern is not merely theoretical. For example, the government of Alabama recently issued an emergency proclamation to help municipalities "that are struggling to find election workers due to COVID-19." Although there is no 'silver bullet' for finding extra workers, a few ideas that could be helpful include raising the renumeration for people that serve as poll workers on Election Day during the pandemic; lowering the age requirement to serve as a pollworker and allowing polling station workers to serve in places other than their own locality. Consideration could also be given to targeting certain organizations, such as businesses, social organizations, and sports teams, who might be more civic-minded and willing to pitch in.

3.2 Cyber Assets

COVID-19 has not only made it more difficult to protect election workers, but election operations as well. For example, many election officials have worked from home or away from their traditional work sites during the outbreak, often using networks that lack the firewalls[1] of their traditional sites and are more exposed to cybersecurity threats. This added challenge creates new targets for those interested in conducting disrupting cyber-attacks on elections infrastructure. Good cybersecurity practices for remote environments are therefore critical.

It is important that election officials review the technology their offices are using while working away from their traditional worksites, such as videoconferencing and chat services. They should evaluate the technology against their own policies and their country's cybersecurity standards, and seek assistance as needed from other security experts to make their offices as secure as possible.

As part of these efforts, election officials need to update their devices regularly. This includes consistently installing updates on, or 'patching' devices[2] that are used at home, including laptops, tablets, phones and home routers. Operating systems, browsers and other applications used by election personnel should also be patched. If the IT department approves, auto-updates should be enabled. Such steps help address identified vulnerabilities, which may allow bad cyber actors unauthorized access to information systems or networks.

Election workers should also know how to avoid phishing attacks, rogue Wi-Fi hot spots, and other malicious activity. If necessary, election officials should seek out organizations in their countries to provide ongoing training and assessments on these

[1] A part of a computer system or network which is designed to block unauthorized access while permitting outward communication.

[2] Patching is the process of applying available updates to an operating system, website, software, hardware or plugin.

threats. This will help ensure that they stay abreast of the most significant threats and know how to respond in the event of an attack. If possible, election officials should also adopt two-factor authentication. Requiring this for all log-ons is an important way of reducing unauthorized access to sensitive infrastructure.

While not all of the above measures may be applicable to any country right now, it is imperative that election officials throughout the transatlantic community prepare for the possibility of remote work and social distancing in the run up to elections for as long as the pandemic is around. That way, in the event that one or more election officials is infected with the coronavirus, preparation for any given election can continue unabated. In that same vein, it is imperative that election managers cross-train their staff on different functions and consider how staff from other agencies could provide assistance on short notice.

3.3 Physical Assets

In addition to cyber and human assets, securing physical election assets amid the pandemic has also become a greater challenge. One example of this is voting equipment. In response to COVID-19, more voting by mail is occurring throughout the world, including within the transatlantic community. Some countries such as the United States, South Korea, Poland, and France have expanded who is eligible to vote by mail during the pandemic. Other countries such as Australia have encouraged voters to vote by mail, while some like Germany and Switzerland have resorted to conducting certain elections solely by mail ballot. Although it is not a certainty, an increase in voted mail-in ballots could result in official election results not being known until later than is customary due the process of receiving, processing, verifying and counting such ballots.

Authoritarian regimes such as Russia and Iran have already tried to used coronavirus precautions and resulting delays in the voting process as evidence of election malfeasance. One way to counter such disinformation efforts is to try and improve the speed and accuracy of vote counting by getting additional equipment to help tabulate mail-in ballots. As a number of studies have shown, using machines, such as the ballot optical scanners used in South Korea and much of the United States, to initially tabulate the results can be faster and more accurate than hand-counting, while at the same time offering the possibility of a voter verified paper trail. Some election officials are also using barcode scanners to more quickly process inbound mail ballots, envelope openers to more quickly open inbound ballots, significant verification software to more quickly and accurately verify the voter returning the ballot; additional hardware (computers, monitors and scanners) to support the adjudicating of signatures; and a ballot monitoring camera to provide transparency to the public about the operation. Processing, storing and counting large numbers of mail ballots using some of the above equipment requires a lot of space; even more due to the pandemic.

3.4 Elections during the Pandemic Cost More

The adjustments made to human, cyber and physical assets in response to COVID-19 not only require careful planning and execution, but more money. In the past, changes to voting procedures that impacted election security could often be done gradually to accommodate voters, candidates, election workers, government budgets, and other factors. Now, many countries are being forced to keep up with the evolution of the pandemic just to ensure that their elections are safe. For example, New Zealand is planning to spend $19 million to fund additional staff and safety measures for its October 2020 parliamentary elections - a cost of around $6.20 for each expected voter. To put that in additional context, if the U.S. were to match New Zealand's investment for its 2020 presidential election, it would need to spend some $750 million more based on its 2016 turnout.

Countries are implementing a range of measures to help ensure that voting during COVID-19 can be done safely. Many are purchasing materials such as personal protective equipment, protective screens, and sanitation supplies to protect voters, election workers and others who visit elections offices or polling places. A number are modifying work places and polling locations to ensure social distancing, whether that's putting markings on floors, printing additional signage, or having additional people on hand to remind the public of these changes. If the facilities can't sufficiently accommodate social distancing, election officials are often seeking out additional facilities to safely accommodate voters and their workers. And after such changes are implemented, many election officials are notifying voters and the public of them through mailings, newspaper and television ads, digital ads, and other means.

Working to address each of the aforementioned considerations in a relatively short period of time is critical, but it is already straining some election officials' budgets. In the United States, the coronavirus pandemic has drastically changed voting behavior. Millions more voters are requesting mail ballots, far more than expected prior to the virus, and the costs associated with this are significant. For example, in Macon-Bibb County, Georgia the elections board indicated that it was already short of cash, with an August runoff and the November general election still to come. A flood of absentee ballot requests increased election expenses and the county's budget has shrunk as Covid-19 has slashed tax revenues. For other election jurisdictions that are similarly situated, this makes financing the administration of future elections, let alone securing them, a major challenge.

In the Republic of Georgia, even before the pandemic on 28 October 2019, the country experienced one of the most extensive cyberattacks witnessed to date, bringing down some 2,000 website and two television stations.[3] Then in March 2020, just as the pandemic was starting to take hold, another data breach connected with voter's data took place. Sources reported that that "voter information for more than 4.9 million Georgians, including deceased citizens, had been published on a hacking forum…" Although it was eventually determined that the data was not that of the Geor-

[3] The U.S. later went public, blaming the GRU for this attack.

gian CEC, the event underscored the continued seriousness of this issue, despite the rising threat of and resources required by the pandemic.

4 EMB Responses and New Vulnerabilities in Election Infrastructure

As election officials rush to modify their election systems to account for COVID-19, they must build infrastructure that can handle the strains of large, challenging elections while remaining secure. Otherwise, their responses to the coronavirus could create the new vulnerabilities in the election infrastructure that underpins their elections. If election infrastructure is expanded or changed, security and resiliency measures should be part of their design, not introduced after the fact.

Poland's governing 'Law and Justice' party initially proposed conducting its 10 May presidential election as with full postal voting for the first time, on the grounds that the pandemic didn't constitute an emergency and that the situation could worsen in the autumn, even though the country was under lockdown at that point to limit infections during the coronavirus pandemic. In preparation for this scenario, the Polish Post (Poczta Polska) requested personal data of Polish citizens via email without any additional protection or password. While email is convenient for sharing information, it has limited security protections and should not be used for sending sensitive information, such as personal data. Email can be viewed or tampered with at multiple places in the transmission process and is often used in cyber-attacks on organizations. As a result, some Polish local authorities voiced their concerns about potential privacy violations and refused to provide such data. Putting the security of its citizens personal information at risk in this manner could have made it vulnerable to a hack and leak operation like those done to the Democratic National Committee and Emmanuel Macron's presidential campaign during the 2016 US presidential campaign and 2017 French presidential campaign, respectively.

To print the ballots for the full postal election, the Polish national government awarded a contract to a firm that could not ensure the security of the ballots. A few days later, copies of the ballots were leaked, and an angry presidential candidate demonstrated how easy it would to copy them and submit multiple votes. Vendors often build and maintain much of a State's election infrastructure, by doing things such as printing ballots, creating election websites and maintaining voter registration databases. Such roles can make them targets for adversaries. It is, therefore, imperative that such vendors follow cybersecurity good practices, have processes for reporting cyber incidents, conduct background checks and other security measures for personnel, and maintain supply chain integrity, among other things. Fortunately, a few days before this election was to set take place, an agreement was reached to delay the election and it was subsequently rescheduled for on 28 June with voting in polling stations under health protection measures.

The Ukrainian case is also exemplary. The presidential administration had announced intentions to introduce full-scale Internet voting by the time of the next elections (nationwide local elections have now been scheduled for 25 October 2020). This

is in a country that had not conducted any previous pilot projects or had not introduced any technology in their elections other than the website results page discussed above. Such a move would present clear risks to the integrity of the election process, although it was partially posited as solving health concerns raised by COVID-19. It presents a stark example of vulnerabilities that can be created when EMBs introduce quick responses to the crisis.

5 Possible Solutions to Addressing Election Security Threats

As countries move to hold key elections across the transatlantic region in the coming months, there are a number of things they can potentially do to secure their elections against the threats above:

Ensure your voter registration databases are secure during the pandemic. Due to the pandemic, many election officials and their staff have been working from home, and there's a greater risk for people who are teleworking to become victims of a cyberattack, like a spear phishing campaign. In that vein, if election officials remotely access election infrastructure such as the voter registration database, it is imperative that they do this in a secure a manner as possible. Voter registration systems are often critical and interconnected components of states' election infrastructures, and as the 2016 U.S presidential election demonstrated, foreign adversaries are capable of targeting and infiltrating them. Therefore, for those countries that utilize similar voter registration databases, there are a number of steps they should try and take to ensure they are more resilient. Those include requiring multi-factor authentication and passwords that are consistent with international cybersecurity standards; monitoring all voter registration database login attempts and backing up their databases on a regular basis.

Use paper-based voting methods. Some have argued that a fully digital voting process will protect election workers that might otherwise contract coronavirus from tabulating the paper ballots votes, but a recent study cited by the United States Center for Disease Control and Prevention asserts that the virus can survive on paper or cardboard for only 24 hours. As a result, voted ballots sent through the mail are unlikely to carry the virus and many voted ballots that could be subject to a subsequent recount or audit will have a small risk of transmission as well. Paper-based voting system are also the most secure. Paper ballots can be verified more easily by most voters, secured more effectively by most poll workers, and reviewed / audited more accurately after an election. In the event that any election-related infrastructure, such as electronic voting machines (e.g., ballot scanners) or election night reporting websites, are breached by bad actors or experience technical glitches, paper ballots can be used to verify the election outcome and thereby ensure public confidence in the election.

Ensure that the election process, including procedures modified in response to COVID-19, are observable to the public. In response to the pandemic, many democracies are adjusting their voting procedures, such as expanding opportunities to

vote before Election Day, which impacts the security of their elections. These changes should not only be shared in a timely and proactive manner with the electorate, but observable as well. If the public can see that the adjusted procedures for conducting an election are beyond reproach, it makes it much difficult for foreign adversaries or other bad actors to either interfere with the modified election infrastructure or create doubt about the adjusted election procedures among large segments of the electorate.

Implement robust post-election audits to validate the results of elections conducted amid the pandemic. Due to the uncertainty around the pandemic, many election officials have been forced to make significant changes to their elections in short periods of time. Such changes have included quickly scaling up vote by mail operations, expanding early voting opportunities, consolidating Election Day polling places, and recruiting scores of new workers, any of which could create more vulnerabilities. One way to mitigate any mistakes that could arise from such changes is to conduct robust post-election audits. As places such as the state of Colorado have shown, reviewing statistically significant samples of voted paper ballots to verify the winner of the cost helps to ensure that any issues with the tabulation of the election results are caught and corrected. The gold standard is the "risk-limiting audit" (RLA), which uses statistics to determine how many ballots must be audited following the results of an election to verify that the outcome is correct. That said, RLAs can take a good deal of time, expertise and resources to plan and implement, and many election authorities could find it easier to first conduct a smaller, more traditional audit of a certain percentage of ballots before trying RLAs. While such audits may not be as full proof, they are certainly better than no audit at all.

Ensure that the election night reporting system data is accurate and protected. The attack on Ukraine's 2014 presidential election underscored the importance ensuring that election night reporting system data is accurate and protected. Before COVID-19, allowing public observation of the actual tabulation of results was a great way to retain credibility in the face of such attacks. However, because of COVID-19 and the need to socially distance to reduce the risk of contracting the virus, it could be harder to observe in-person the tabulation of results. Election officials will therefore need to develop other resiliency measures to deploy in the event of such attacks, whether that is establishing redundant election night reporting sites to be made available in the event the main site is attacked, live-streaming the results tabulation in a secure manner, or developing a comprehensive public outreach communications plan in the event of similar attacks.

Give voters as many secure choices as possible to cast their ballots. Elections that offer only in-person voting on a single day are higher risk for COVID-19 spread because there will likely be bigger crowds and longer wait times. Such elections are also higher risk from a security perspective because any issues that arise are harder to detect, investigate and/or recover from in a timely manner that ensures all who wish to vote can successfully do so. Depending on the county and its election, this could mean offering longer voting periods (more days and/or hours), more opportunities to vote by mail, or opportunities to vote outside of traditional polling stations. For example, in Sweden, changes in the legislation ahead of its 2004 vote meant that in addition to voting by mail, eligible voters could vote ahead of Election Day in places

such as libraries, senior citizen homes, and even shopping malls. This expanded access to the ballot while ensuring that the integrity of the vote was protected as well.

Communicate widely and proactively accurate information about elections to the public through an explicit communication strategy, including information about the pandemic and how the election is being secured in response to the virus. That will help ensure confidence in the election process and reduce the likelihood that bad actors, including foreign adversaries, can amplify mis- and disinformation in a manner that successfully undermines confidence in a state's election and democratic processes more broadly.

Work to ensure that all government agencies involved in the administration and securing of a country's elections are accessible, flexible, open and supportive of one another. Clear timely communication between different government agencies is critical to better identifying and responding to election cyber threats. This has been noted in countries like Bosnia and Herzegovina, Georgia and Ukraine, where only limited coordination currently exist and more must be done to promote better cybersecurity practices throughout government and society at large.

Ensure that all individuals involved in the administration of elections know what to do in the face of cyber threats from foreign adversaries. This includes getting cyber training, having good cyber hygiene, and saying something when you see something. Ensuring the cybersecurity of elections is a common responsibility. Anyone who has access to an elections system, no matter how minor, bears some responsibility for the cybersecurity and integrity of the election. As this paper is the latest to note, 'security through obscurity' is no longer a viable option. Instead, training of election management bodies and their partners should be done on a consistent, ongoing basis by security experts with knowledge in the field, many of whom can be found at key academic institutions, think tanks, and other private sector and civil society organizations.

In Ukraine, in relation to Internet voting proposal discussed above, the international community quickly stepped in to try to countenance a more nuanced approach and introduce emerging good practices in this space. In particular, IFES advocated and undertook a feasibility study into the question and made a series of recommendations to analyze the problem that the Ukrainian authorities were trying to solve, examining key parameters of long, medium and short-terms costs, the possible impact on turnout, efficiency, end-to-end verifiability, result audit possibilities, security concerns, and especially the key questions of trust and transparency. This was later further developed into a global white paper. On this basis, a group of key international experts was also brought together in June 2020 to discuss these issues in a Global Online Seminar on Internet Voting; more than 500 people registered to examine the risks and benefits of such an approach.

In Bosnia and Herzegovina, North Macedonia, and Georgia IFES has undertaken innovative cybersecurity and elections assessments, which has resulted in concrete programming being developed and later funded by international donors, which include risk mapping and mitigation strategies, building better institutional communication channels and more robust technology infrastructure, and developing and deploy-

ing technology and cyber-hygiene trainings. The efficacy is already being proven and in the most recent COVID-19 election on 15 July 2020 in N. Macedonia, these efforts helped thwart a large-scale cyberattack from bringing down the CEC's results system.

These are but some possible solutions to addressing election security threats that can be developed and deployed. But they should focus in the first place in analysis (both of feasibility and risk), concrete and prioritized recommendations, and then concrete implementation in improving system robustness, but equally important human understanding and behavior.

6 Conclusions

Cybersecurity and foreign influence threats already presented a serious risk to democratic elections prior to the advent of the COVID-19 crisis. The pandemic has further complicated this situation, by forcing many countries to quickly adjust some of their traditional voting processes. In some cases, countries have deployed novel technology solutions shortly before an election, making proper planning and resource allocation, both human and financial, more challenging.

As countries' authorities and election management bodies make changes to their election processes in response to COVID-19, they must carefully consider the election security risks such changes introduce, while ensuring that elections carried out during the pandemic are accessible, secure, and legitimate. Doing otherwise risks making elections more vulnerable to adversaries and undermining public confidence in the democratic process.

Legal aspects and evaluation of Internet Voting experiences

Secure Online Voting for Legislatures

Aleksander Essex and Nicole Goodman

[1]Department of Electrical and Computer Engineering
Western University, London, ON, Canada
aessex@uwo.ca

[2]Department of Political Science
Brock University, St. Catharines, ON, Canada
ngoodman2@brocku.ca

Abstract. Often discussed in the context of general elections, remote electronic voting has recently become a pressing topic for legislatures. Parliaments and assemblies worldwide face a stark choice between the legislative traditions of in-person debate and voting with new physical distancing requirements brought about by the COVID-19 pandemic. Faced with suspending legislative activity, or a drastically reduced complement of in-person representation, legislatures are naturally exploring remote online voting options. Unlike general elections, legislative divisions[1] are typically *not* secret—they are a matter of public record, which significantly simplifies the detection and recovery from errors or faults. We examine the *why*, *how*, *what*, and *where* of legislative e-voting, with a particular focus on the Canadian context. Analyzing four approaches to remote electronic voting currently in use, we argue voting via video teleconference presents Canada's House of Commons with the most workable solution.

Keywords: E-voting, Online voting, Legislative voting, Legislatures, Elected representatives

1 Introduction

Discussions of electronic voting are often situated in the context of public elections. Electronic voting technologies are used for public elections in countries worldwide, including Armenia, Australia, Canada, Estonia, India, Norway, Switzerland, and the United States. They are also adopted by private organizations such as unions, political parties, and corporate firms for internal votes [13]. Electronic voting in a legislative context has attracted increased interest with the onset of the COVID-19 pandemic, which has seen governments around the world suspend or modify legislative sittings. These actions are unprecedented in many countries, notably parliamentary democracies such as the United Kingdom and Australia, which kept legislatures open during previous crises such as the 1918 Spanish

[1] A vote taking method where members are *divided* into groups supporting, opposing, or abstaining from a motion.

Flu pandemic and both world wars [7, 38]. These extraordinary circumstances have caused governments to rethink how to keep legislative democracy working in emergencies and to explore remote electronic voting as a possible solution for legislative and committee votes.

This article examines remote electronic voting in a legislative context, paying specific attention to Canada, given the country's consideration of remote voting options at the time of writing. Evaluating how countries worldwide respond to legislative voting in a pandemic, we examine why voting remotely online works better in a legislative context than in a public general election. We also consider how legislatures could adopt remote electronic voting and the type of remote electronic voting that will work best in Canada and other parliamentary democracies. Our argument is two-fold. First, we assert that remote electronic voting is more workable in legislatures than in public elections because most votes are a matter of public record. The government could also support the voting system and the accompanying process. Second, we make the case that voting by video teleconference represents the best option for the House of Commons in terms of cybersecurity.

The article proceeds as follows. First, we define and categorize remote voting options and provide background on voting in the legislative context. Next, we give a brief review of the literature and legislative functions. Third, we review how legislatures around the world have tried to balance procedures with physical distancing requirements. Fourth, we outline the key requirements for how to conduct legislative divisions safely online. Fifth, we identify four main ways to hold remote divisions and weigh relative risks and benefits, arguing that video voting is the most workable option. We conclude by discussing implications for legislative democracy.

2 Definitions and Background on Legislative Voting

Definitions. In this article, we use the terms *remote online voting* and *remote electronic voting* to reference several types of voting that both ask ballot questions and receive vote preferences via internet-connected devices. We use these terms interchangeably for stylistic relief to refer to four main subtypes: voting via email, web, mobile app, and video teleconference. Typically, the category of *electronic voting* encompasses a range of technologies such as voting machines, ballot scanners, and internet voting systems [10]. While references to *online or internet voting* are more targeted to the use of the internet and ICTs for the casting, recording, and counting of votes [43], both definitions can include digital voting types that are not remote and require voters to attend a physical location to cast a ballot. This article focuses only on voting approaches that can be done *remotely*.

Under these definitions, voting by email may fall outside the traditional online voting gamut given that voting preferences may not be returned via email. Likewise, voting by video teleconference is different from systems traditionally identified as online voting since ballots may not be recorded directly recorded

by an internet-facing server depending on the specifics of the approach. Web- and mobile app-based methods, by contrast, are typically identified as online voting systems. All four types examined here rely on the internet and ICTs to support the voting process. According to our conception of how they would work in a legislative context, members receive ballot questions and communicate their voting preferences via the internet.

To delineate between these four sub-types, we define them as follows:

- **Email**. A member receives a ballot form electronically via email and submits their vote via email.[2]
- **Web**. Ballots are accessed and cast via a website
- **Mobile App**. Ballots are accessed and cast via an app on a mobile device.
- **Video Teleconference**. Takes place via video by a physical show of hands or voice.[3]

Voice Votes and Divisions. Legislative voting differs from voting in general elections. Each voter in a general election can cast a ballot, and ballots are individually tabulated to produce an objective, numeric total. Legislatures do not follow this explicit model of recording and counting individual vote preferences—at least not initially. Typically a threshold of dissent on a particular question must be met before a formally recorded vote occurs. The frequency of decisions combined with the comparatively small and traditionally in-person nature of legislatures means that it is more efficient to subject questions to an initial voting step called a voice vote, determining whether a formal recorded vote is held.

A *voice vote* involves the leader of the legislature (i.e., speaker, chair) inviting members approving a motion to vocalize their support by calling/shouting out. Next, members opposing the motion are invited to vocalize their dissent. The speaker then makes a subjective judgment as to which group contained more members. If a threshold of members disputes the speaker's determination, the question proceeds to a division.

A *division* individually counts members by their voting intentions. The term derives from physically dividing members into groups: those supporting the question, those opposing it, and those who abstain.

The particular method of the physical division of the legislature varies by country. For example, the UK House of Commons and German Bundestag have separate labeled rooms called division lobbies, which members enter according to their voting preference. The number of occupants in each room is tabulated, and the highest occupancy room decides the outcome. Australia's House of Representatives directs members to move to the chamber's side, reflecting their vote: right side for support, left side for dissent. Other countries like Canada and the US direct members to sit in their regular seats. Those voting in favour are asked

[2] There are variations of this approach that could include receiving the vote via email and submitting by other means such as postal mail or fax as done in the US [42].

[3] Indicating voting intent with a teleconference app's "raise-hand" feature could be regarded as a hybrid of web, application and video modes.

to rise and remain standing until counted. This standing and counting continue for dissenting and abstaining members.

Some legislative voting tasks are done by secret ballot. For example, the speaker of Canada's House of Commons is elected via secret ballot. In this article, however, we focus on non-secret divisions where each member's vote is a matter of public record.

3 Literature

E-voting Literature. The literature has mostly focused on understanding the cybersecurity and social and political effects of electronic voting in the context of public elections [13]. Our review found only a couple of articles addressing electronic voting in legislatures or assemblies. One paper, published in Spanish, examines solving public verifiability of the vote in the context of elections and parliaments [31]. The other article examines the conceptual aspects of electronic voting and the differences between voting elections and collegiate bodies [32].

Aside from these works, contributions examine other shifts toward a virtual legislature such as electronic petitions [24], the concept of e-parliament [27], and the idea of parliament opening up to technology more generally [36]. Since the onset of the pandemic, there has also been a move to explore the concept of a virtual legislature [37]. The lack of literature examining electronic voting in a legislative context could be explained by the fact that legislatures are steeped in tradition and are often slow to embrace technology to ensure maintenance of the institutional heritage. In addition, prior to COVID-19 there was no overarching pressure for legislatures to consider voting remotely except for individual circumstances such as the birth of a baby (i.e., as in the UK).

Despite this lacuna, there is a rich literature addressing legislatures and their functions. We briefly review these functions and, further down, examine the extent to which remote electronic voting enhances or impedes the ability of members to exercise these goals.

Legislative functions. In his review of legislatures, David Docherty [11] points to three functions of modern parliaments: scrutiny, representation, and lawmaking. Such functions are also noted in other work [2]. *Scrutiny* implies holding the government to account to ensure it is meeting citizens' needs, spending judiciously, and acting appropriately. Opportunities for scrutiny include Question Period, debates on legislation, and key items such as the budget and throne speech, committee work, and caucus discussions [11].

Second, members are elected as agents of *representation.* Once elected, there is a general expectation that members will speak up for the voters who elect them; although there is debate regarding whether members have an obligation to represent local issues or put national concerns first [14]. Some members, however, may embrace a free mandate not tied to their constituencies. In addition, representation may look different in 'working' and 'debating' parliaments. The former focuses on committee work established in standing orders (e.g., Nordic

parliaments), while the later emphasizes plenary debates (e.g., Canadian and British parliaments) [3].

A final function of legislatures is *lawmaking*, whereby members pass or defeat legislation. Part of passing legislation, however, involves debate. As Docherty points out, "Parliament is the one forum where public debate *must* be held on legislation," which includes debate in the legislature and discussions at committees [11].

4 Survey of Responses to Legislative Voting During COVID-19

To get a sense of how legislatures have adapted voting during the pandemic, we surveyed legislative responses Table 1. For practical reasons we present several of these cases here in Table 1, which reflects all four currently identified approaches for online voting in legislatures as well as modified in-person sittings.

In-person sittings. Legislatures generally opted to either continue holding sessions with reduced members present or postpone sittings as temporary workarounds to protect members' health and safety. Australia, Canada, Estonia, and the United States are cases where national legislatures have continued to sit with restrictions. On March 13, for example, both Australia and Canada announced sittings would proceed with a reduced number of members and a reduction in sitting days. The scope of legislation during that time focused solely on the discussion and passage of emergency measures [30]. While Australia's House will meet again in August, Canada has postponed regular sittings until October 2020 [15]. In the meantime, Canada's Standing Committee on Procedure and House Affairs (PROC) is investigating remote voting [29].

Other chambers have moved more quickly to implement remote electronic voting. While the context and arrangement of legislative politics differ in each jurisdiction, it is interesting that such a diversity of approaches has been adopted across legislatures.

Email voting. The EU Parliament was one of the first to move forward with a digital solution by adopting email voting. While a limited number of members still attended the House physically, the majority voted via email. Under this approach, members receive a ballot paper to their official EU Parliament email address. They then print, sign, scan, and email their ballots to an internal parliamentary mailbox where they are recorded and counted. Votes must include the member's name and vote selection in a readable format. To be counted as valid, they must be signed and submitted before the close of the vote [23].

Web-based. An example of web-based adoption is the system UK House of Commons staff developed in several weeks. This system's infrastructure already existed as part of the MemberHub platform, which was introduced in 2017 to provide digital service to members and staff [40]. The presence of this groundwork made development easier.

When a vote is called, members receive a notification by SMS or email. This notification specifies whether the vote is a division or party whip vote [20]. The

Country	Legislature	Special Measures Taken	Date (2020)	Permanency of Measures	Scope of Legislation During Crisis	Process of Change
Australia	House of Representatives	In-person voting with restrictions ('pairing' to reduce number of MPs)	March 13	Temporary. In-person sittings with physical distancing resumed May 12	Essential legislation only	Parliament and Senate approved return
UK	House of Commons	Web-based: Initial hybrid model (reduced in-person with video conferencing), later use of MemberHub voting website	April 21	Continuing hybrid proceedings for MPs requiring physical distancing	Remote technology for questions, urgent questions and ministerial statements	Procedure Committee
Canada	House of Commons	In-person voting with significantly reduced (ca. 10%) complement of MPs	March 13	Extended from April 20th and remain in use	Emergency legislation only. Video-conferencing for debates and committees. Studying possibility of remote voting.	Procedure and House Affairs Committee
EU	Parliament	Email hybrid: limited in-person presence with majority of members voting remotely via email	March 20	Continuing until July 31 unless extended	Maintaining core functions (passing laws, plenary sessions, budgetary approvals)	Bureau of the European Parliament
Brazil	Chamber of Deputies	Application-based: First Virtual Plenary session held March 25th using Infoleg mobile application. MPs cast votes remotely from their mobile devices	March 17	Temporary: In-person sittings expected to eventually resume	First virtual plenary on March 25th, ten virtual sessions as of April 9th, 15 pieces of legislation, six urgency motions and one constitutional amendment passed	Chamber of Deputies
Belgium	House of Representatives	Video hybrid: In-person and remote voting via custom application	March 23	Temporary: In-person sittings expected to eventually resume	Emergency legislation relating to COVID-19	House of Representatives

Table 1: Changes to Legislative Voting During the COVID-19 Pandemic

system requires multi-factor authentication and uses Microsoft logins [1, 39]. Once accessed, the member can view the motion text and has the option to select 'Aye' or 'No.' The tabulation of ballots is observed by the Public Bill Office team, and outcomes are provided to the Speaker for announcement in the House. Members have 15 minutes to cast a ballot once receiving the vote notification [41].

Application-based. Like the UK, Brazil's application-based online voting also relied on existing infrastructure. Following the transition to virtual sittings, the Chamber of Deputies moved to vote via the mobile application Infoleg. Originally developed to create transparency for citizens regarding House business, the application was loaded to members' mobile devices and adapted to allow members to register their presence and cast ballots remotely [33].

Members install the system and then register the device with the legislature's internal network. Each registration requires a unique identification. When joining a plenary session, members are presented with a Zoom link, a registration button, and a vote button. Members can cast their ballot via the vote button during the voting period and by entering a personal password to confirm the vote. The system uses two-factor authentication [9].

Video teleconferencing. Finally, Belgium has adopted a hybrid model that relies on video teleconference voting via Zoom for committee votes. Other jurisdictions, such as Canada, Brazil, and the UK, are also using Zoom to facilitate committee meetings or virtual parliament sessions. Voting in the Belgian context has been conducted verbally and also by a show of hands. Voting for plenary sessions will be conducted through a digital voting system recently developed by the legislative IT department [20].

There are also other approaches not explored here. One example is the Isle of Man's use of a chat box in Microsoft Teams for parliamentary voting. To exercise this, members types 'yes' or 'no' in the chatbox. This allowed members to vote quickly and votes to be tallied within 3-4 minutes [12, 37]. Despite differences in approaching legislative voting during the pandemic, one commonality is that many legislatures are relying on video for committees and plenary sessions [37]. Second, legislatures that had IT infrastructure in place beforehand seem to be better positioned to adapt traditional processes. While it could be that certain modes of technology work best in the political and cultural contexts of specific legislatures, in what follows, we argue that voting by video teleconference is the most workable short-term solution for legislative voting.

5 Verifiability Requirements

Remote legislative divisions are workable from a cybersecurity perspective, as they differ from general elections in three crucial ways. First, a member's vote is a matter of public record, making it possible to verify it was correctly recorded and counted. Second, the smaller scale and more frequent nature of divisions make it feasible for the legislature to support its members with cybersecurity infrastructure and technology to ensure the protection of electronic information.

Third, legislatures can provide members training on the procedures necessary to ensure votes are successfully entered into the record [16].

With these elements, we argue that a secure, remote online solution for legislative divisions is viable. However, care must be taken in the design so that faults are detected, reported, and corrected. We use the term *fault* to denote an incorrectly recorded vote. Three broad categories of faults include human-error (misunderstandings, improperly followed procedures), technical failure (software errors, network outages), and malicious intent (hacking).

The non-secret nature of legislative divisions lays the foundation of fault detectability, however, technology and procedures must be developed to ensure three essential security goals reliably:

1. **Detection**. The voting member must check the recorded vote.
2. **Reporting**. The voting member must reliably report faults to the legislature.
3. **Recovery**. The legislature must have a mechanism to recover from and correct the fault.

6 Options for Remote Divisions

We examine four different methods for remote online divisions and compare each according to credential transferability, intent capture accuracy, and attack complexity given privileged access to modify the vote. A summary is shown in Table 2.

Casting Mode	**Typical Authentication Scenario**	**Transferable Credential?**	**Intent Capture Accuracy**	**Modification Complexity w/ Privileged Access**
Email	Password	Yes	Potentially ambiguous	Low
Website	Password	Yes	Unambiguous	Low
Mobile App	Password	Yes	Unambiguous	Low
Video Teleconference	Face and voice	No	Potentially ambiguous	Medium–High

Table 2: Comparison of Remote Voting Modes

6.1 Email-based Divisions

There are two main scenarios involving the question being put to member via emails. In the first case, the vote is written out as text and returned to the legislature via email. The member's voting intent may be ambiguous if, for example,

the voter makes a typo, or the email client performs an auto-correct. In the other scenario, the email contains a link that opens a website or app.

Email-based question delivery and vote return is perhaps the riskiest of these modes for at least two reasons. One is that email offers no inherent message integrity. Although closed communities such as legislatures can feasibly overcome this shortcoming with a secure email infrastructure such as S/MIME, our examination of our email exchanges with Canadian members of parliament revealed the use of DomainKeys Identified Mail (DKIM) signatures only. While this indicates the email came from Parliament's IT infrastructure, it does not tie the email to the specific member, nor does it prescribe any particular course of action in the event of a signature failure. The second is that email is still a vector for phishing, especially sophisticated spear-phishing efforts, which could be used to misdirect the member to an attacker-controlled return email address or website.

Finally, the complexity of modifying a returned vote is low for an attacker with privileged access (e.g., one with root privileges on the local machine, or a man-in-the-middle network connection). In such a scenario, vote modification would require changing a few characters in the response email or HTTP POST.

6.2 Web-based Divisions

Web-based vote return involves the member signing into a website using their browser. While typically providing a more formal authentication and message integrity mechanism via TLS, the onus is still on the member to recognize when they are misdirected to a fake/credential-harvesting website, are the subject of a TLS-stripping attack [6], or the sign-on page loads resources from an insecure 3rd-party [18].

As noted above, the UK House of Commons recently added a voting feature to their MemberHub website. However, it acknowledged that the "high level of authentication (of traditional voting) is not replicated in the remote voting system over MemberHub," and the "temporary purpose of the system is not sufficient to justify the development and expense" of a more secure solution [34].

MemberHub uses multi-factor authentication, a parliamentary virtual private network (VPN), and sign-on is redirected to a Microsoft login portal. We observed, however, that the MemberHub domain `memberhub.parliament.uk` did not use HSTS headers or HSTS pre-loading, which would be necessary to prevent TLS-stripping.

Tests of the system also experienced problems, with some MPs failing to receive notifications, and two-factor authentication being disabled. Despite a web-based voting approach nominally having a more clear-cut intent capture methodology than email (i.e., enforcing selection of a single explicit option instead of writing a free-form response), in a remote division in on 13 May 2020, several members voted opposite from their intention by mistake [35]. The Deputy Speaker additionally noted that "there is no provision under the current temporary system by which a Member can change their vote once it has been cast."

6.3 Mobile App-based Divisions

Mobile app-based vote return is comparable to web-based return, except the mobile app environment reduces the attack surface in several essential ways. It removes more of the human-element from verifying the security of the network connection. The app can be designed to enforce TLS strictly, and certificate pinning could remove some of the threats posed by a foreign nation-state to the public-key infrastructure.

Also, having a dedicated interface prevents the member from being misdirected to a fake website. Furthermore, designing an app for a single purpose seems to be a better practice than using a general-purpose extensible web-browser, whose behavior can, by design, be arbitrarily modified with extensions (cf. [5]).

6.4 Video-based Divisions

Video teleconferencing allows members to express their voting intent verbally or visually (e.g., physically raise a hand). In our conversations with members of Canada's PROC, one of the concerns members and witnesses expressed was the possibility of a member delegating their vote to someone (e.g., a staffer or party whip). This is possible in each of the other three other modes, and instances of credential delegation have been reported in online general elections (such as in the 2018 Ontario municipal election) [5]. Casting a vote over video teleconference, by contrast, offers a non-transferable credential: the member's face and voice.

Expressing intent, however, may potentially lead to the most significant opportunity for ambiguity. For example, the city council of Sarnia, Ontario passed a bylaw by mistake when one of the councilors voted "disagree." However, the first syllable dropped due to a glitchy network connection, causing the council to understand his intent as "agree" [22], and tipping the vote count in favor of the motion. Fortunately, city staff caught the error after the fact, and the decision was reversed.

The aeronautical industry may provide some guidance in this area. For example, to qualify for a license to be a restricted radio operator, pilots must learn speech transmission, phonetic alphabets, reserved procedural words and phrases, and other special techniques to ensure clarity.

Finally, unlike the other modes of vote return, an attacker with privileged access would face comparatively greater technical complexity to modify a vote response, involving selectively modifying and/or replaying video.

6.5 Threats to Video-based Divisions

In this section we explore the primary attack vectors for divisions held by video teleconference. An attack tree based on the attacker goal of modifying a member's vote is shown in Figure 1. Broadly speaking, the attack vectors identified involve either modifying the video feed, or disrupting it.

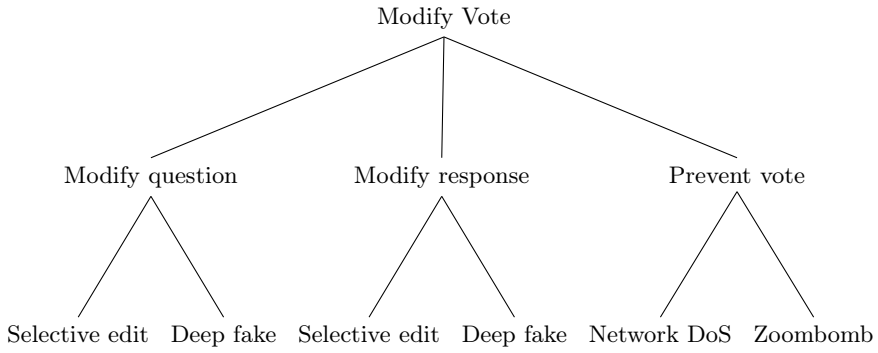


Fig. 1: Attack Tree for Remote Video Teleconferencing-based Divisions

Modifying Video. To alter a member's vote, one main attack strategy would be to modify the video feed; either the video feed of the question being put to the member or the member's video feed responding with their vote. Modifications could broadly take two forms. One is selective edits, either selectively replaying, inserting, or deleting portions of the video stream.

An example of a *replay* attack could involve selectively replaying earlier video. For example, if a member voted "agree" to a question earlier in the day, that video segment could be replayed as the legislature counts those in favour. An earlier video segment of the member sitting silently listening could be replayed as the legislature counts those who disagree. An example of an *insertion* attack might be possible if the question is put to the member as a text-based image. An attacker could insert/overlay an image of a different question. From the view of the legislature/speaker, the video of the member appears genuine. An example of a *deletion* attack could follow the approach of the Sarnia council vote and selectively truncate syllables (e.g., deleting "dis" from "disagree").

Finally, our conversation with Canada's PROC touched on concerns over video voting's eventual vulnerability to deepfakes. However, it was agreed that this methodology was likely not a realistic threat at this time.

Preventing the Vote. Another strategy to alter voting outcomes would be to prevent the question from being put to the members, or limiting their ability to respond. One traditional approach is a network-based denial-of-service attack; however, a far more plausible approach would be to degrade internet access of select members in rural ridings. Canada's PROC committee raised concern over the consistency of internet connectivity throughout the country and acknowledged that "internet connectivity and speed vary throughout the country. This lack of connectivity would especially be the case for members representing rural or remote areas, who could potentially face internet-related challenges," [29].

On the other end of the deniability spectrum is the phenomenon of "Zoombombing" in which unauthorized participants disrupt video voting. When we testified to the Commons Procedure committee in June 2020, we were provided

an 11-digit Zoom meeting ID and a 6-digit password representing a maximum upper bound of 56-bits of entropy. Meeting IDs, however, are not always correctly conceptualized as secrets. For example, the UK government was criticized for its use of Zoom after the Prime Minister tweeted a meeting ID [19]. Furthermore, a recent study demonstrated the ability to guess random Zoom meeting IDs [8], and another found a vulnerability in the waiting room feature that revealed the video-feed decryption key to non-admitted users [25]. Worse, they found these keys passed through servers in China, even when all meeting participants were outside of China [26].

7 Legislative Implications

Scrutiny. The introduction of remote online voting in the legislature has implications for legislative democracy. For one, while establishing a secure online voting system looks after the voting portion of legislative business, it does not address motion introduction or debate. This raises questions about the extent to which members can fulfill their scrutiny function in a situation where they are not face to face to oppose and question one another [2, 11]. Specifically, concerns have been raised that virtual proceedings mean debates are "silted" and "scripted," limiting the ability to debate legislation adequately [28].

That said, scholars have called into question the extent to which scrutiny has been adequately fulfilled in recent years given that there are often fewer sitting days than in the past and that members do not meet as frequently to debate. In addition, committees have been criticized for being underutilized and not reaching out to witnesses enough, affecting participation and responsiveness [11]. As noted above, some countries are employing technology to fill in the gaps beyond voting by facilitating committees via Zoom (e.g., Belgium) [20] and hosting virtual chamber debates and plenary sittings (e.g., European Union Parliament) [37]. Albeit, these are temporary and not likely long-term solutions to address debate and accountability in a virtual legislature.

Lawmaking and Representation. The functions of lawmaking and representation are perhaps better fulfilled with the support of remote electronic voting since it allows for necessary laws to pass with enhanced legislative voice while in-person sittings are modified. The passage of legislation with a handful of members significantly limits representation, inclusion, and participation [11]. This situation continues to occur in Canada. On March 23, for example, 32 of 338 members attended a session to pass emergency measures. Similarly, on May 26, 51 members voted 28-23 to pass a controversial motion suspending regular sittings until October [16, 15]. Such few members making important decisions in a time of crisis pose its own issues for democracy.

While remote electronic voting allows legislatures to fulfill lawmaking functions and representation as it pertains to the passage of bills. Although cybersecurity is solvable for legislative divisions, this solution addresses only the legislative business's voting component. The scrutiny function depends on the

participation and inclusion of members to represent their constituents in votes and in debate. In this way, while we present a solution for legislative divisions during a pandemic or other emergency, addressing the dimension of accountability with remote means is more challenging and is an area scholars and practitioners can continue to reflect upon to determine if it can be replicated virtually.

Accessibility. Beyond the COVID-19 pandemic, remote electronic voting might bring long-term benefits to members' voting accessibility. For example, it could be useful in future cases of national crises such as a second or third wave of COVID-19, the onset of other viruses, or in situations of extreme weather brought on by climate change where members may not be able to attend the legislature physically. Having the infrastructure in place to seamlessly adopt a remote voting system would support democracy and ensure governance could continue uninterrupted during uncertain times in the future.

Adoption of these practices could also be helpful for the accessibility of individual members in exceptional circumstances. Such circumstances could include the birth of a child and while nursing an infant [21], in cases of emergency in a particular area where a member is required to stay in their constituency or in severe illness situations. Allowing these types of accommodations is not unlike introducing convenience voting reforms in public elections to make voting easier for groups of electors facing additional voting barriers [17]. However, the implementation of such allowances would have to be carefully thought through and outlined to ensure that members do not take advantage of these permissions in cases of a busy constituency schedule or a bad cold.

To be sure, some legislatures already have proxy voting in place for certain situations, notably following a child's birth. The UK House of Commons, for example, has proxy voting for new parents, while Australia's House of Representatives allows proxy voting for members nursing infants [4]. Remote electronic voting would further enable member accessibility in this regard by allowing them to cast their own vote. This could be a long-term change as part of legislative modernization to enhance the equality of legislative voting.

Conclusion

This article identified four possible ways of voting remotely using technology in the legislature: email, web-based, application-based, and video teleconference voting. We argue that remote electronic voting is solvable for regular, non-anonymous votes because they are a matter of public record, making it easier to verify the votes are correctly cast, recorded, and counted as intended. Legislatures also have the resources to put necessary procedures in place and the technical capacity to support these changes. Of the voting types reviewed, we assert that voting by video teleconference presents the most workable remote electronic voting solution for Canada's House of Commons. The technology is less easy to spoof (compared to changing an email), and voice and face cannot

be shared. Using video also interfaces more closely with the parliamentary tradition of standing during a vote. Voting by video teleconference is a model other legislatures should consider in the pandemic and post-pandemic stages and, in future instances, when remote voting is warranted.

Despite the advantages of video, implementing a voting solution still requires the establishment of a verification loop and other procedures. For example, procedures need to be instituted to allow for the checking and double-checking of votes, reporting, and correction of errors, and to ensure a recount if necessary. Even though the voting component is solvable, issues remain regarding how members can adequately fulfill the scrutiny of legislatures in a virtual chamber. This is a topic for future research.

Future research could also more closely examine the types of remote electronic voting being used and evaluate their functionality and security. Studies could examine how remote online voting affects legislative democracy, particularly how these changes impact motion introduction and debate. In addition, it could examine how complementary technical solutions can support members to carry out these crucial objectives during uncertain times. Finally, studies could consider how members respond to particular electronic voting methods and the extent to which they can identify cognitive biases and report ballot errors.

The COVID-19 pandemic has made clear that remote electronic voting may be needed in legislatures during times of emergency. Developing the proper approach for elected members to vote remotely online will not only promote the inclusion of legislative voice and enhance representation, but it will also ensure that votes are carried out as safely as possible, promoting democratic integrity. Building the capacity and infrastructure today will enhance government's ability to govern in a crisis tomorrow.

Acknowledgments. Both authors contributed equally to this work. Special thanks to Richard Ackerman for many helpful discussions. Thanks to Joe Abley, Valere Gaspard, Gary O'Brien and the members of the Canadian House of Commons Standing Committee on Procedure and House Affairs (PROC) for their helpful insight and feedback. We also extend our sincere thanks to Amanda Tieber for research assistance.

References

[1] Akerman, R. Remote voting in the UK House of Commons: Remote Divisions become reality, Paper Vote Canada 2 Blog. Published on May 12, 2020. Available: `https://papervotecanada2.wordpress.com/2020/05/12/remote-voting-in-the-uk-house-of-commons-remote-divisions-become-reality/`

[2] Atkinson, M. M., Thomas, P. G. Studying the Canadian Parliament. Legislative Studies Quarterly, 423-451, 1993

[3] Arter, D. Scandinavian Politics Today. Manchester University Press; 1999

[4] BBC News. Commons approves proxy voting trial for new parents, BBC News. Published on January 29, 2019. Available: https://www.bbc.com/news/uk-politics-47027143

[5] Cardillo, A., Akinyokun, N., Essex, A. Online Voting in Ontario Municipal Elections: A Conflict of Legal Principles and Technology? International Joint Conference on Electronic Voting (E-Vote-ID) 2019. Lecture Notes in Computer Science, vol 11759, 2019.

[6] Cardillo, A., Essex, A. The Threat of SSL/TLS Stripping to Online Elections. International Joint Conference on Electronic Voting (E-Vote-ID). LNCS vol. 11143, pp. 35-50, 2018.

[7] Carr, K. Parliament sat during world war two and Spanish flu, Morrison should not be cancelling it for coronavirus, The Guardian. Published on April 3, 2020. Available: https://www.theguardian.com/commentisfree/2020/apr/03/parliament-sat-during-world-war-two-and-spanish-flu-morrison-should-not-be-cancelling-it-for-coronavirus

[8] Chailytko, A. Zoom-Zoom: We Are Watching You. Check Point Research. Published on: Jan 28, 2020. Available online: https://research.checkpoint.com/2020/zoom-zoom-we-are-watching-you/

[9] Chamber of Deputies, Brazil. Virtual Plenary Strategy and Architecture, Directorate of Innovation and Information Technology, 2020. Available: https://www.ipu.org/file/9013/download

[10] Council of Europe. Council of Europe Adopts New Recommendation on Standards for E-Voting. Electoral Assistance Newsroom. Published on June 14, 2017. Available: https://www.coe.int/en/web/electoral-assistance/-/council-of-europe-adopts-new-recommendation-on-standards-for-e-voting

[11] Docherty, D.C. Legislatures. UBC Press; 2011

[12] Electoral Reform Society. Isle of Man: World's oldest parliament goes online. Published on April 6, 2020. Available: https://www.electoral-reform.org.uk/isle-of-man-worlds-oldest-parliament-goes-online/

[13] Essex A., Goodman N. Protecting Electoral Integrity in the Digital Age: Developing E-Voting Regulations in Canada. Election Law Journal: Rules, Politics, and Policy. 2020

[14] Eulau, H. Changing views of representation. The politics of representation continuities in theory and research, 31-53, 1978

[15] Global News. Normal House of Commons sittings to be waived another 4 months amid coronavirus, Global News, May 26, 2020. Available: https://globalnews.ca/news/6987857/coronavirus-canada-parliament-voting/

[16] Goodman N., Essex, A. Online voting entirely possible for MPs during times of crisis, Policy Options, March 25, 2020. Available: https://policyoptions.irpp.org/magazines/march-2020/online-voting-entirely-possible-for-mps-during-times-of-crisis/

[17] Gronke, P., Galanes-Rosenbaum, E., Miller, P. A., Toffey, D. Convenience voting. Annu. Rev. Polit. Sci., 11, 437-455, 2008

[18] Halderman J.A., Teague V. The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election. E-Voting and Identity (Vote-ID), Lecture Notes in Computer Science, vol 9269, 2015.

[19] Hamilton, I. Researchers found and bought more than 500,000 Zoom passwords on the dark web for less than a cent each, Business Insider. Published on April 14, 2020. Available https://www.businessinsider.com/500000-zoom-accounts-sale-dark-web-2020-4

[20] Inter-Parliamentary Union. Country compilation of parliamentary responses to the pandemic. Published on March 25, 2020. https://www.ipu.org/country-compilation-parliamentary-responses-pandemic

[21] Johnston, R. How e-voting could close Canada's political gender gap, The Conversation, April 28, 2020. Available: https://theconversation.com/how-e-voting-could-close-canadas-political-gender-gap-136163

[22] Kula, T. Council Decision Changed Amid Technical Glitch. The Sarnia Observer. Published on June 4, 2020. Available: https://www.theobserver.ca/news/local-news/council-decision-changed-amid-technical-glitch

[23] Library of Congress. European Union: Parliament Temporarily Allows Remote Participation to Avoid Spreading COVID-19. Published on April 21, 2020. Available: https://www.loc.gov/law/foreign-news/article/european-union-parliament-temporarily-allows-remote-participation-to-avoid-spreading-covid-19/

[24] Lindner, R., Riehm, U. Broadening participation through e-petitions? An empirical study of petitions to the German parliament. Policy & Internet, 3(1), 1-23, 2011

[25] Marczak, B., Scott-Railton, J. Zoom's Waiting Room Vulnerability. Citizen Lab, University of Toronto. Published on April 8, 2020. Available: https://citizenlab.ca/2020/04/zooms-waiting-room-vulnerability/

[26] Marczak, B., Scott-Railton, J. Move Fast and Roll Your Own Crypto: A Quick Look at the Confidentiality of Zoom Meetings. Citizen Lab tech report. Citizen Lab, University of Toronto. Published on April 3, 2020. Available: https://citizenlab.ca/2020/04/move-fast-roll-your-own-crypto-a-quick-look-at-the-confidentiality-of-zoom-meetings/

[27] Missingham, R. E-parliament: Opening the door. Government Information Quarterly, 28(3), 426-434, 2011

[28] Parliamentary Business. Minister questioned on the continuation of hybrid proceedings after recess, Parliament.UK, May 20, 2020. Available: https://www.parliament.uk/business/news/2020/may/ministers-questioned-on-continuation-of-hybrid-proceedings-after-recess/

[29] Parliamentary Duties and the COVID-19 Pandemic. Report of the Standing Committee on Procedure and House Affairs. Canada House of Commons, May 2020. Available: https://www.ourcommons.ca/DocumentViewer/en/43-1/PROC/report-5/

[30] Parliament of Australia. Australian COVID-19 response management arrangements: a quick guide. Published on April 28, 2020. Avail-

able: https://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/pubs/rp/rp1920/Quick_Guides/AustralianCovid-19ResponseManagement#_Toc38973757

[31] Dalmau, Rubén Martínez. Aspectos diferenciales del uso del voto electrónico en los procesos electorales y en los órganos colegiados. Corts: Anuario de derecho parlamentario 25, 229-245, 2011

[32] i Esteve, Jordi Barrat. Vot electrònic i òrgans col· legiats: El cas de les Corts Valencianes. Corts: Anuario de derecho parlamentario 21, 125-138, 2009

[33] Peixoto, T. Virtual Parliaments in Times of Coronavirus: Flattening the Authoritarian Curve?, DemocracySpot Blog. Published on April 21, 2020. Available: https://democracyspot.net/2020/04/21/virtual-parliaments-in-times-of-coronavirus-flattening-the-authoritarian-curve/

[34] Procedure under coronavirus restrictions: remote voting in divisions. Procedure Committee, UK House of Commons. HC 335, May 2020.

[35] Remote Division result: New Clause 2, UK House of Commons Hansard, vol. 676, 13 May 2020.

[36] Reynolds, W. "Open Parliament": More Than Data. Canadian Parliamentary Review, 42(3), 33, 2019

[37] Samara Centre for Democracy. Towards a Virtual Parliament: Design choices and democratic values. Published on May 1, 2020. Available: https://www.samaracanada.com/democracy-monitor/towards-a-virtual-parliament

[38] Stokel-Walker, C. What happens if coronavirus forces us to close parliament? WIRED. Published on May 12, 2020. Available: https://www.wired.co.uk/article/parilament-uk-coronavirus

[39] Stokel-Walker, C. Inside the troubled, glitchy birth of parliament's online voting app. WIRED. Published on April 23, 2020. Available: https://www.wired.co.uk/article/virtual-parliament-voting

[40] Stutely, M., Barnes, T. MemberHub: changing the way MPs ask questions. Parliament.UK Blog. Published on November 27, 2017. Available: https://pds.blog.parliament.uk/2017/11/27/memberhub-changing-the-way-mps-ask-questions/

[41] Stutely, M. MPs make history with remote voting – the story of how it happened, Parliament.UK Blog. Published on May 14, 2020. Available: https://pds.blog.parliament.uk/2020/05/14/mps-make-history-with-remote-voting-the-story/

[42] Thompson, J. R. Email Voting in Indiana Elections. E-Vote-ID 98, 2018

[43] Wolf, Peter, Rushdi Nackerdien, and Domenico Tuccinardi. Introducing electronic voting: essential considerations. International Institute for Democracy and Electoral Assistance (International IDEA), 2011

E-Voting System evaluation based on the Council of Europe recommendations: *n*Votes

David Yeregui Marcos del Blanco[1][0000-0001-7702-6602] David Duenas-Cid[2,3] [0000-0002-0451-4514] and Héctor Aláiz Moretón[0000-0001-6572-1261]

[1] University of Leon, Campus de Vegazana, s/n, 24071 León
dmarcb01@estudiantes.unileon.es
hector.moreton@unileon.es
[2] Tallinn University of Technology, Akadeemia tee 3, 12618 Tallinn, Estonia
[3] Kozminski University, Jagiellonska 57/59, 03-301 Warsaw, Poland
david.duenas@taltech.ee / dduenas@kozminski.edu.pl

Abstract. E-voting implantation has been facing important challenges in recent years. Several incidents, together with a lack of evaluation methodologies social and cultural customs hinder a broader application. In this work, the authors aim to contribute to a safer introduction of e-voting tools by applying a practical evaluation framework strongly based on the security requirements issued by the Council of Europe (CoE) in 2017 to *n*votes, a system that has been utilized to cast over 2 million votes over the last 6 years.
The ultimate goal of the analysis is not to judge from a rigid, "infallible" but to contribute to a gradual and secure implementation of e-voting solutions in the democratic processes. The authors believe it can constitute a useful source of information for election officials, researchers and voters.

Keywords: e-democracy, e-voting, system evaluation, nvotes.

1 Introduction

Since the first implementation of remote electronic voting in the 90s [4], the process of dissemination of internet voting did not meet the initial and promised expectations. Several countries experimented with the possibility of adding internet voting systems to their elections[1], but it just turned into a reality in a reduced number of them: Estonia, Canada, Australia, Switzerland or Norway, amongst others. The Estonian case is the most prominent success story, using Internet Voting uninterruptedly since 2005 in all elections [1] an reaching high levels of acceptation [2] and cost efficiency [3, 4].

The dissemination of internet voting technologies is challenged by a complex set of factors that affect different layers of administration, law, society and technology [5] and that should be achieved in a constant dialogue between themselves: dealing with complexity in electoral management, reforming electoral laws, ensuring transparency,

[1] For a better understanding, see International IDEA's database on use of ICT in Elections: https://www.idea.int/data-tools/data/icts-elections (last accessed 4 June 2020)

neutrality and participation and ensuring secure and risk-free technological apparatus. The latter factor, has been constantly labelled as an important element not only for the correct functioning of the internet voting and its integration in the electoral systems, but also as an element projecting trust in the society where the system is being implemented [6,7,8].

Pursuing the same goal, the creation of trust as a key element for the adoption of internet voting systems, the Council of Europe (CoE) proposes a set of recommendations to guide the process of implementation of electronic remote voting systems [9]. The CM/Rec(2017)5 updates the previous Recommendations from 2004 and integrates lessons learned from previous experiences and developments in the electoral field to create a useful and up-to-date document. Specifically, proposes a set of Principles, Standards and Requirements that every electronic voting system should fulfil for the development of elections and for reinforcing the democratic principles that are the common heritage of its member states [10]: Elections should be Universal, Equal, Free and Secret, should meet a set of regulatory and organizational requirements, should be transparent and allow observation and should be accountable, and should use reliable and secure systems.

In view of the aforementioned list, this paper presents an analysis on how the system *nVotes* fits within the CoE requirements. The ultimate goal of the authors is not to judge from a rigid *immovable* or *infallible* point of view for the sake of pin pointing shortcomings, but to establish a comprehensive, multi-faceted evaluation in order to improve the knowledge and security level in the deployment of e-voting systems

2 Related Works

The research work of Bräunlich, Grimm and Richter in 2013 [111] is considered one of the most relevant to date. The authors presented the first interdisciplinary collaboration which has transformed legal requirements into technical criteria. Specifically, they established thirty Technical Design Goals (TDG), using the KORA methodology (*Konkretis-ierung Rechtlicher Inforderungen, Concretization of Legal Requirements*) [12]. This methodology had been used previously for mobile devices amongst others.

Neumann combined the previous methodology of Bräunlich, Grimm and Richter with the *Common Criteria for IT-Security Evaluation* [13] and established sixteen technical requirements to relate the legal criteria to Bräunlich's TDGs.

While Neumann's work [14] has critically contributed to constructing a very valuable framework, it still had room for improvement from a practical standpoint:

On the one hand, the security evaluation framework is aimed at schemes rather than entire systems, with the author himself coming across an example of a structural flaw that would not be identified using his evaluation scheme: *"for instance, the Vote Forwarding Server and the Vote Storage Server of the Estonian Internet voting scheme are developed and maintained by the same vendor"* [14, p. 135].

Additionally, the security evaluation assumes that the voters will use the authentication tools sufficiently. Unfortunately, the tendency of the voters is not to verify: for

instance, one of the largest electoral e-voting initiatives which took place in New South Wales in 2015, showed that only 1.7% of 283.669 votes were verified [15].

Furthermore, Neumann's framework is based on probabilistic attack strategies through Monte-Carlo simulations [14]. While represeting an interesting approach indeed, it is less useful for a practical evaluation standpoint. As a result, the author concludes: "we therefore recommend to incorporate the security evaluation framework into a larger decision-support system for elections officials" [14, p. 138].

Following with the above recommendation, a decision-support system was proposed by Marcos, et al. as a practical evaluation framework [16]. It is in accordance with the guidelines from the 2017 Council of Europe's *("Guidelines on the implementation of the provisions of Recommendation CM/Rec(2017)5 on standards for e-voting")* [17] and deals with the five key principles of a democratic election (universal, free, equal, direct and secret) detailed in the same document.

3 Evaluation Methodology

As previously stated, while Neumann's work set out an irrefutable improvement, it constitutes a scheme evaluation tool with probabilistic proofs as its core with Monte-Carlo simulations rather than a practical evaluation framework tool for election officials and other stakeholders involved in the democratic processes.

In 2018, Panizo et al. proposed an extended evaluation approach [19] in the context of the Spanish Constitution [18] and the CoE's e-voting recommendations [17]:

1. Defining an homogeneous series of e-voting requirements with the KORA methodology [12] as its basis, together with the CC and ISO 27001-IT Grundschutz guideline [13], their assimilation by Simic-Draws et al. [20], the Guidelines of the Council of Europe [17] and Neumann's methodology [14].
2. Formal conformity between point 1 and Bräunlich's TDG's [11], as in Figure 1.
3. Consultation with more than 30 international experts in e-voting (Research and Industry Experts or RIE, selected using the snowball [21] and judgement [22] sampling methodologies) to review the evaluation framework and add weighting factors.
4. Formal definition of the practical evaluation framework, including two sine-qua-non requirements (E2Ev and Coercion Resistance) and 41 evaluation items.

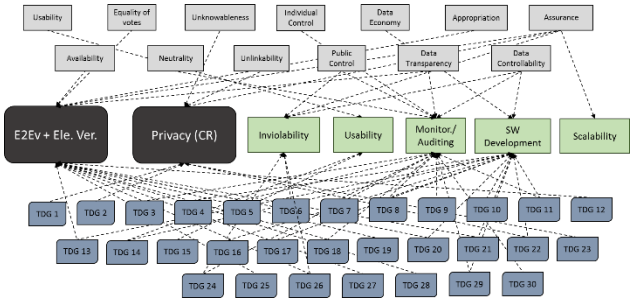


Fig. 1. Integration of Panizo [19] and Bräunlich [11]

The work in [16] established for the first time a correlation between the end to end verifiability (E2Ev) and coercion resistance (CR) to the legal requirements for a democratic process and the Council of Europe: *"The five key principles of electoral law are: universal, equal, free, direct and secret suffrage and they are at the root of democracy"* (article 68 of the Spanish Constitution [18]).

Specifically, Marcos et al. Set out the equivalence of the aforementioned five key principles, into a formal authentication of the E2Ev the universal, free, equal and direct properties and its coercion resistance for the secrecy prerequisite (based on the findings by Hirt and Sako on the matter in [46]).

The methodology presented to this point is solid from a legal point of view but still lacks the technical and practical approach necessary for a complete evaluation.

In order to solve the shortcomings, five practical requisites were introduced, partially based on the research by Benaloh, Rivest, Ryan and Volkamer [23], [24]. Subsequently, the requisites were codified, refined and subdivided into 73 specific items by means of a partial application of Zissis and Lekkas [25] and New Zealand's Department of Internal Affair's Communication on e-voting [26] [2].

As a final step, e-voting RIEs from Canada, France, Norway, Switzerland, Germany and Spain among other countries were consulted to assign a weighting factors.

The following Figure 2 visually represents the complete evaluation methodology:

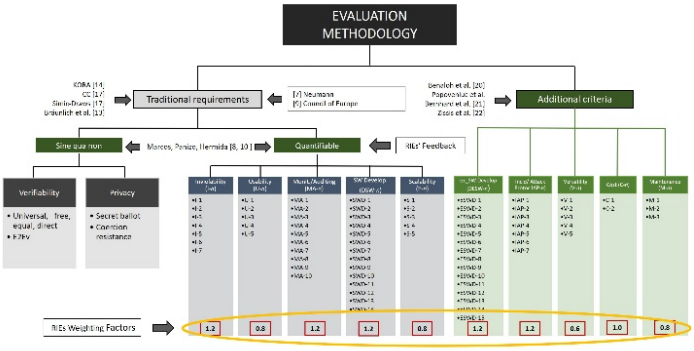


Fig. 2. Complete evaluation framework [16]

The sine-qua-non requirements (end-to-end verifiability and coercion resistance, representing the five compulsory principles of a democratic election), which evaluation is not a numerical value related to performance but instead in terms of "holds" (○) or "does not hold" (×). There is a third possibility, when the property "stands under determined, credible assumptions" (Δ).

The second quantifiable and additional criteria, totaling 10 requirements, are evaluated from 0 to 10. In order to obtain the numerical evaluation for each criterion, the corresponding measurable sub-items are evaluated with three possible outcomes: non-compliant (×), partially compliant (Δ) and compliant (○).

[2] For a complete explanation of the previous process, please refer to the original work in [6], [8].

Due to space constraints, the evaluation framework design, implementation and constituent requirements has been simplified. For a full explanation, the reader can refer to Dr. Marcos' PhD thesis which originated the methodology [27].

It is relevant to mention that this practical evaluation methodology has also been applied to Helios Voting and published by the IEEE [19].

4 nVotes Analysis

4.1 Introduction

nVotes [28] is a remote e-voting system developed by the Spanish company Agora Voting SL in 2014. Its roots trace back to 2009 and the Internet Party, although the developing team has since then dropped any political affiliation and nVotes is currently an apolitical project.

Until 2017, nVotes was known as Agora Voting and under such moniker it was one of the 18 European start-ups to be accepted in the Impact Accelerator project, and awarded with 100,000 EUR [29].

According to their website, nVotes has been used to cast over 2 million votes for over 150 clients, including Public Administrations like the Barcelona Provincial Council, Madrid City Council; Political Parties like Podemos, Ahora Madrid and Barcelona en Comú, as well as Education Institutions like UNED University in Spain.

4.2 Main characteristics

As previously mentioned, the methodology presented in Section 3 has been already applied to other relevant e-voting tools, including Helios Voting [19] or iVote by Scytl [30]; in both cases with numerous bibliography and research resources available:

- Helios Voting is a very well-known open source e-voting system [31], which has been used as blueprint for several variations and improvements such as Helios KTV [32] or Belenios [33].
- Scytl is probably the most widely used e-voting system at a global level, including numerous legally-binding elections and pilots for a total of over 100,000 processes managed and more than 200 employees. The information available ranges from research papers to Government reports and corporate presentations.

In the case of of nVotes, the available bibliography is much more limited due to the fact that they are neither a research standard tool, nor a global company. In order to complement the publicly available information, the authors of this document got in touch with nVotes and they key people have always been open and supporting in providing all the available information and answers to the questions raised.

Additionally, the authors were provided with two documents named *"Technical Overview"* and *"Client Action Protocol"*, which have been extremely useful for conducting the analysis. They are at the reader's disposal upon request to the authors since they have not been published before.

nVotes scheme components and cryptographic primitives. According to the information included in the *"Technical Overview"* and complemented with a Q/A with nVotes technical team, the key elements are:

- Registry: The registration database programmed in Python. It includes the SMS service platform Esendex [34], server certificate with TSL support, Cloudfare [35] and Fail2ban [36] for protection against DDoS attacks and hardware redundancy 1+1.
- Virtual Polling Station: TLS server validation, *cast-or-audit* voting javascript (similar to that of Helios Voting [31]), random number generator (not specified), HMAC client authentication, Election Manager with Scala REST API, Postgresql database and similar to the Registry case, Cloudfare and Fail2ban DDoS protection.
- Electoral Authority: HTTP distributed queue, TLS client/server authentication, mixnet library *Verificatum* [37] and tabulation library OpenSTV [38].
- Election Verificator: a Python/Java

With regards to the main cryptographic primitives, they are the following:

- El Gamal Homomorphic Encryption [39]
- Pedersen Threshold Distributed Key Generator [40]
- *Verificatum* verifiable mixnet [37]
- Fiat-Shamir heuristic to convert Zero Knowledge Proofs into Non-Iteractive Zero Knowledge Proofs [41]
- Schnorr Signature [42] to make the ElGamal Encryption IND-CCA2.

nVotes voting sequence. As presented in the *"Technical Review"* and *"Client Action Protocol"* documents, the voting procedure is as follows:

1. Authorities distributedly generate the Election's Public Key with Pedersen [40].
2. Eve (voter) access the Registry site and provides the required personal information, including a security code which has been sent independently by SMS
3. The Registry system compares the information provided with the census. If it is correct, Eve is forwarded to the Virtual Polling Station.
4. Eve fills her vote, encrypts it and sends it. Alternatively, she can audit it but in such case, the cast vote is no longer valid and will not be tallied. This *cast-or-audit* approach is also implemented in Helios Voting [31].
5. Once the vote casting period ends, the authorities jointly proceed with the mix and decryption of the ballots
6. The decrypted votes are tallied
7. The election results are published, together with the tally results, the vote's ciphertexts as well as the mixnet and decryption Zero Knowledge Proofs.
8. Voters and third parties can download and execute the election verificator

Once nVotes has been introduced, together with its associated scheme components, cryptographic primitives and voting process, the practical evaluation methodology for e-voting systems [16] can be applied.

The analysis is intended to be a sort of a guideline, which introduces strengths and potential weaknesses in order to establish a safe range if utilization and to offer directions as to how to improve the voting system.

4.3 End to End Verifiability

Unfortunately, there is no formal, universal definition for end-to-end verifiability (E2Ev). Additionally, symbolic analysis of security protocols still find associative and commutative operators are out of reach. It is then not possible to analyze a homomorphic property [43] such as:

$$enc(pk; v_1)*enc(pk; v_2)=enc(pk; v_1+v_2) \quad (1)$$

and therefore, a case by case analysis has to be conducted for each system.

Currently, probably the most widely accepted definition of E2Ev is the one by Benaloh et al. in [23] and is comprised of the properties: “Cast as intended”, “Recorded as cast” and “Tallied as recorded”.

For the first and second items, nVotes presents a similar approach to that of Helios Voting: the voter can audit her vote until she is convinced that it is trustable. Once cast, she receives a hash of the encrypted vote, which she can check on public bulletin board. Finally, for the tallied as recorded condition, ElGamal together with *Verificatum* mixnet [37] and Schnorr [42] are implemented.

Consequently, on the question of nVotes being E2Ev or not and similar to the analysis in [18] for Helios Voting, it can be considered end to end verifiable assuming that:

- The cast and audit mechanism is used by a large enough number of voters so that ballot alteration will not go unnoticed.
- The Election Authorities and the Bulletin Board (BB) are honest
- An attack which gains control of the Registry/Ballot is detected.

For the first precondition, Acemyan in [44] and the New South Wales case [15] have shown that voters’ ballot verification percentage is quite low and they should not be responsible of part of the security of an e-voting system.

As for the other two prerequisites, in a perfect scenario nVotes would be compliant but in real elections, both the Election Authorities and/or the BB can illegally introduce votes (ballot stuffing). For public, legally binding elections, it is not acceptable.

To sum up, provided that nVotes implementation is limited to elections with a low risk of corruption such as student government bodies, local clubs, online groups, and other education-related organizations, the pre-assumptions could be acceptable. For other, more demanding types of elections, E2Ev cannot be recommended.

Evaluation: Δ. E2Ev holds if the preconditions set in nVotes’ *Technical Overview* document are accepted and its use is limited to low corruption risk elections.

4.4 Coercion Resistance

Assuming probably the most accepted definition of privacy levels by Juels et al. [45] and the proof by Hirt and Sako [46] that receipt-freeness is not enough for preserving it in electronic elections, the required level is Coercion Resistance. It implies that a voter cannot provide to an attacker any proof of her vote or even whether she voted or nor, even if she is willing to cooperate.

As for nVotes, the voter receives a verification code after casting the ballot, therefore she can prove it to a potential attacker.

Additionally, the Election Administrator of an Election can verify whether an specific person in the census has voted or not, which clearly compromises the privacy.

Evaluation: X. Does not hold.

4.5 Inviolability (I-n)

nVotes' *Technical Overview* document includes an integrity, privacy and availability analysis. The authors include the possibility of "ballot stuffing" if the Election Administrators are corrupt and of DDoS attacks despite implementing specific tools [35], [36].

There have also been questions raised about the census integrity used in consultative referenda [47, 48] and the separation between the tally administrator and the census administrator, which can be the same person and thus lead to potential collusions (I-4).

Safe authentication protocols, tracking tools, Risk Assessment and modularity principles are partially compliant, with room for improvement.

Table 1. Inviolability in nVotes

I-n	Definition	Val
I-1	Software and auxiliary system's protection w/ safe authentication protocols. Access via third-parties/vulnerable-servers not permitted.	Δ
I-2	Action protocols in the event of compromised inviolability.	X
I-3	Tracking tools and offline backup copies available.	Δ
I-4	Distributed control in the critical nodes with division of responsibilities to minimize collusion risks.	X
I-5	Existence of *Risk Assessment* and *Threat Modelling* protocols.	Δ
I-6	Modularity principles to confine potential attacks and coding bugs.	Δ
I-7	Proper updating of items I-1…I-6	Δ

Evaluation: 4/10 points. The inviolability policy presents vulnerabilities which, for private elections (while being very serious), are ultimately up to the organizer whether to take the risk or not. For legally binding public elections, they are not acceptable and nVotes inviolability should be improved before being used in such environment.

4.6 Usability (U-n)

nVotes presents a satisfactory performance in terms of simplicity and clarity in the voting process (U-1, U-3) as well as in intuitiveness and lexicon choice both for the voter and the administrators.

Concerning the aspects to be improved, there is no version adapted to collectives with special needs, the SMS authentication might prove challenging for the elders and the verification codes are too long and "imposing" voters with no technical background. An intermediate usability layer might be advisable. Overall, usability is satisfactory while it could be enhance with some simple, easy to implement changes.

Table 2. Usability in nVotes

U-*n*	Definition	Val
U-1	Simplicity in the authentication, voting and verification	O
U-2	Special attention to vulnerable groups pursuant to the Council of Europe and the United Nations' resolutions on the matter.	X
U-3	Transparency & clarity communicating the voter that the voting process has successfully ended/vote has been received.	O
U-4	Privacy and integrity preference over usability in a compromise.	X
U-5	Intuitive/user-friendly admin interface for setup and management.	O

Evaluation: 6/10 points

4.7 Monitoring/Auditing (MA-n)

This aspect is especially relevant for nVotes due to the possibility of *Ballot Stuffing* if the Administrators are corrupt or collide or due to DDoS attacks.

Probably due to the nature and scope of the elections managed, the Monitoring and Auditing Protocol is based on the Administrators training. According to nVotes' team, a unified protocol including all the auditing activities is currently being generated.

Until then, nVotes generates retrievable logs, and provides information and data in an easily understandable format. Even so, at this point the Monitoring/Auditing Protocol is still largely to be developed and implemented; therefore not satisfactory.

Table 3. Monitoring/Auditing in nVotes

MA-*n*	Definition	Val
MA-1	External, independent and distributed.	X
MA-2	MA protocol from the design phase, to assure a correct development throughout the whole lifecycle of the project.	X
MA-3	*Specific control on Risk Assess* and *Thread Modelling strategies.*	X
MA-4	Generation of periodical, tamper-proof, indelible logs; stored offline in premises guarded by different personnel from other critical nodes.	Δ
MA-5	Implementation from census collecting to post-electoral maintenance.	Δ
MA-6	Well-documented, detailed information in the appropriate format.	Δ
MA-7	Existence of a test bench to verify that the system is working correctly.	X
MA-8	The members of the monitoring/auditing team must be independent from the rest of authorities/administrators involved.	X
MA-9	Auditing protocol for previous attacks and the MA protocol itself.	X
MA-10	In the event of a successful attack, the system will give total priority to the vote/voter's privacy, even calling off the elections.	X

Evaluation: 3/10 points

4.8 Software Development (SWD-n)

nVotes displays an overall solid Software Development (partly because of its open source approach), with a satisfactory performance in usual software engineering practices (SWD-1), FAQ (SWD-4), impartiality (SWD-5), ballot cast termination (SWD-8), compatibility (SWD-9), third party access (SWD-10), and protocolized application (SWD-13).

Regarding the distributed approach (SWD-2), it has been correctly implemented for key generation and encryption/decryption but there is no separation between the census and the bulletin board. If the same person is responsible for both of them, there is an important risk of collusion.

Finally, the primitives are well implemented but some of them have been already been proven flawed and should be reviewed (SWD-11). Additionally, more frequent updates would be preferable (SWD-14).

Table 4. Software Development in nVotes

SWD-*n*	Definition	Val
SWD-1	Usual software engineering requirements in terms of design, implementation and documentation.	O
SWD-2	Distributed approach on critical operations. No authority should have attributions to single-handedly modify critical parameters.	Δ
SWD-3	User-friendly approach. User's guide and administrator's guide well documented and available well in advance.	Δ
SWD-4	Secure and accessible website, with a well-documented FAQ.	O
SWD-5	The voting options must be presented in a totally objective and unbiased way, showing no preference whatsoever.	O
SWD-6	System must not provide the voter with evidence to proof her vote.	X
SWD-7	The system must guarantee the voter's privacy throughout the whole voting process, not being possible to rebuild the vote/voter link.	Δ
SWD-8	The voting process must offer the possibility to be terminated at any time, not saving any information compromising the voter's privacy.	O
SWD-9	SW to be tested in every platform, operational system and browser with a market share ≥ 1%.	O
SWD-10	Software must neither allow for third-party access (incl. social media) nor include links to programs/sites outside the e-voting infrastructure.	O
SWD-11	The cryptographic primitives shall be tested in advance under conditions more demanding than the ones expected during the elections in order to avoid breakdowns and foresee shortages.	Δ
SWD-12	Access to the source code by independent experts to reinforce security. The code developer can demand an NDA to protect its IP.	Δ
SWD-13	Use of protocolized systems/open standards to improve interoperability.	O
SWD-14	Update policy, against new e-voting attacks as they are discovered.	X

Evaluation: 7/10 points

4.9 Scalability (S-n)

nVotes has managed elections up to 150,000 votes in consultative referenda of political parties, although they didn't managed many of the ex_software activities, which were handled by the Party itself.

So far, the system has proved to be scalable to the amount of votes already managed in private elections. The shortcomings related to monitoring, ex-software development and potential collusion request a further in-depth improvement before being considered for introduction in public binding elections.

Table 5. Scalability in nVotes

S-*n*	Definition	Val
S-1	Maximum capacity tests both from a SW and a HW standpoint in environments more demanding than the elections to be managed.	Δ
S-2	Ad-hoc performance tests for the most critical operations (authentication, encryption/decryption, cryptographic primitives, tallying ...).	X
S-3	Existence of test benches more demanding than the actual elections.	X
S-4	Clear indicators and metrics on the max manageable size and complexity from a SW (cryptographic capabilities, number of voters) and ex_SW (infrastructure, costs, logistics, second channels etc.) standpoints.	Δ
S-5	Clear definition of election which can be adequately handled by the *e-voting* system (from consultative referenda to politically binding elections).	Δ

Evaluation: 5.5/10 points

4.10 Ex-Software Development (ESWD-n)

Ex_Software development is intimately related to the increased complexity of public binding elections. The lower the score in this category, the less recommended it is for the analyzed e-voting system to be implemented for such type of elections.

In the case of nVotes, it has been deployed only for private elections and referenda, and therefore has not implemented ESWD1-4, ESWD6-7, and ESWD-10.

The aspects in which the development is satisfactory are: authentication by alternative channels (ESWD-11) and the master initialization protocol (ESWD-12).

As for the communication/problem solving/back up policy (ESWD5, 6, 8, 9, 14, 15), nVotes stated that they offer different levels of services according to the needs and budget of each election. They can even let the client handle most of the activities related to back-up protocols, responsibilities attributions etc.

While that could make sense from a business perspective, the security implications in case of a misuse or a scandal, and the potential impact in the reputation of nVotes, advice against allowing the election organizer to handle such sensitive actions.

Table 6. Ex_Software Development in nVotes

ESWD-*n*	Definition	Val
ESWD-1	Design, development & update of SWD/ESWD protocols in parallel.	N/A
ESWD-2	Safe protocol for credential, permission & responsib. distribution.	N/A
ESWD-3	Automated access control and infrastructure surveillance.	N/A
ESWD-4	Auditing and independent observers' protocol.	X

ESWD-5	Distributed *back-up* protocol.	Δ
ESWD-6	Distribution of attributions and responsibilities throughout the whole ex_sw development to minimize collusion risks.	X
ESWD-7	Availability of complementary, non e-voting systems.	X
ESWD-8	Voters must be informed about the e-voting process in advance, through websites, telephone, information stands…	Δ
ESWD-9	If re-voting is permitted, provide a reinforced information campaign to explain the prevalence of paper ballot.	Δ
ESWD-10	Organize opinion polls on selected cohorts to gather reliable feedback on usability, tendencies and improvements.	X
ESWD-11	Authentication of credential submission by alternative channels.	O
ESWD-12	Master initialization protocol to be executed right before the start of the e-voting period to verify the correct operation/readiness.	O
ESWD-13	Implementation, to the extent possible, of protocolized and standardized systems to improve interoperability.	Δ
ESWD-14	Free assistance phone service available before/during the election.	
ESWD-15	Complete PR strategy to promote e-voting and train voters, including: webinars, stands, demos, open days etc.	Δ

Evaluation: 4/10 points

4.11 Incidents and Attacks Protocol (IAP-n)

Due to the track record of elections managed by nVotes, they do not have a proper protocol in place, presenting only partial compliance in distributed/modular approach and actions taken towards limiting the risk of an attack with the introduction of Cloudfare [35] and Fail2Ban services[36].

In conclusion, nVotes needs to develop a proper Incidents and Attacks Protocol before being used for legally binding, public elections.

Table 7. Incidents and Attacks protocol in nVotes

IAP-*n*	**Definition**	**Val**
IAP-1	*Risk Assessment (RA), Privacy Impact Assessment (PIAS), Penetration Testing (PT), Control Validation Plan (CVP)* and *Control Validation Audit (CVA)* protocols.	Δ
IAP-2	Specific prevention protocols for each cryptographic scheme.	X
IAP-3	All the information shall be kept to the extent possible in the country's National soil.	O
IAP-4	Implementation of protocols and reinforcement operations to minimize the risk of permanent losses of information.	Δ
IAP-5	Reinforced distributed approach to contribute to the absence of critical nodes which undermine the e-voting system's viability.	Δ
IAP-6	Training and awareness campaigns to minimize the risk of voter-driven attacks (*phishing*, social engineering, etc.).	X
IAP-7	Hackers/indep. experts to test and compromise the system beforehand.	X

Evaluation: 4/10 points

4.12 Versatility (V-n)

nVotes can be used by the voter with a standard internet connection, hardware and Operative System. While it works in most of the available browsers and devices, there is no compatibility study available.

Regarding the existence of different versions depending on the type of election (yes/no, 1/N, N/M, order etc.) there are no adapted versions but according to the data in *Verificatum* [37], its performance is satisfactory enough to not require adapted versions. The authors believe that such statement is only partially true and largely depends on the range of the election.

Finally, the score against the WCAG 2.0 standard was good but not brilliant (A).

Table 8. Versatility in nVotes

V-*n*	Definition	Val
V-1	Versions adapted to different election typologies (yes/no, 1/N...).	Δ
V-2	Specific solutions for vulnerable groups (disabilities, illiterates etc.).	X
V-3	The voter shall be able to vote using her personal device, through a standard internet connection without installing any additional SW.	O
V-4	E-voting system tested in browsers/devices w/ a market share ≥ 1%.	Δ
V-5	The interface is WCAG 2.0 AA compliant.	Δ

Evaluation: 5/10 points

4.13 Cost (C-n)

Cost in a sensitive issue for e-voting systems. Most of them are not transparent in their pricing policy. That is understandable to a certain point, but even the cheapest option should offer a sufficient security level.

nVotes used to have a very clear, direct policy with 3 plans with a fix cost of 0.2 EUR per voter plus other associated costs. In its simplest version, it was possible to organize a 1.000 voter election with all the required elements for a little over 1.000 EUR. Currently, the policy has changed and there is no clear indication of the cost for the organization of an election.

While probably still an affordable option, the authors believe that the previous, more transparent approach was better from a user's point of view.

Table 9. Cost in nVotes

C-*n*	Definition	Val
C-1	Transparency and clarity in the cost breakdown.	
C-2	System cost related to quality and performance. Comparison with other e-voting solutions.	

Evaluation: Review (6/10 points)

4.14 Maintenance (M-n)

Both from a software and ex-software perspective. On the software side, nVotes is an open source project and therefore very open and verifiable. It is regularly updated.

Regarding the ex_software aspect, there is not much improvement and it would be very advisable in order to extend the safe utilization range of the system.

As for everlasting privacy and post-quantum security, nVotes team is working on it but there is no expected imminent announcement.

Finally, the maintenance cost is quite limited and performed internally.

Table 20. Maintenance in nVotes

M-*n*	Definition	Val
M-1	Covering both SW and ex_SW aspects. Frequency, thoroughness and existence of security logs to check the maintenance process are also evaluated.	Δ
M-2	Maintenance as *everlasting privacy*.	N/A
M-3	Maintenance cost itself.	Δ

Evaluation: 6.5/10 points

5 Final Results and Conclusion

nVotes [218] is a remote e-voting system developed by the Spanish company Agora Voting SL and active since 2014. It has managed a total of 2 million votes with up to 150.000 votes in the same election.

In order to complement the relatively limited publicly available information for the analysis in this article, they have been diligent and helpful and the authors with like to extend their gratitude for their availability.

The ultimate goal of the analysis is not to judge from a rigid, "infallible" perspective for the sake of it, but to try contribute to a gradual and secure implementation of e-voting solutions in the democratic processes.

The formula and table below summarize the findings and scores of nVotes:

$$\sum_{i=1}^{n} \frac{f_1 \cdot w_1 + \cdots + f_n \cdot w_n}{n} \cdot \frac{n}{t} = \sum_{i=1}^{n} \frac{f_1 \cdot w_1 + \cdots + f_n \cdot w_n}{t} \quad (2)$$

Table 31. Practical Evaltuation Methodology [16] applied to nVotes

Requirement	Code	Weight	nVotes
E2Ev	E2Ev	N.A.	Δ
Coerc. Resistance	CR	N.A.	**X**
Inviolability	(I-*n*)	**1.2**	4 * 1.2 = **4.8**
Usability	(U-*n*)	**0.8**	6 * 0.8 = **4.8**
Monitoring/Audit	(MA-*n*)	**1.2**	3 * 1.2 = **3.6**
Software Devel.	(SWD-*n*)	**1.2**	7 * 1.2 = **8.4**
Scalability	(S-*n*)	**0.8**	5.5 * 0.8 = **4.4**
Ex_Soft. Develop.	(ESWD-*n*)	**1.2**	4 * 1.2 = **4.8**
Incid./AttackProt.	(IAP-*n*)	**1.2**	4 * 1.2 = **4.8**
Versatility	(V-*n*)	**0.6**	5 * 0.6 = **3**
Cost	(C-*n*)	**1.0**	7 * 1.0 = **7**
Maintenance	(M-*n*)	**0.8**	6.5 * 0.8 = **5.2**
TOTAL		**10**	**50.8**

Due to the nature of the elections in which nVotes has been deployed, it is in an intermediate position between Helios Voting and Scytl's iVote systems. nVotes can manage elections with a number of voters that Helios Voting has not been able to proof so far while showing serious shortcomings in legally binding elections, where a strong infrastructure, ex-software policies and monitoring/auditing protocols are a must. Therefore, currently nVotes' safe range of use is that of private elections.

The areas in which nVotes presents a stronger performance are:

- Open source approach, with good software engineering and possibility of review by researchers/academia
- Intuitive, simple and user-friendly interface for both the voter and the administrators.
- Compatibility
- Open standards, modularity
- Support service during the elections

Conversely, the aspects which should be improved include:

- No proper Audit/Monitoring or Incidents/Attacks protocols in place
- Policy for credential, access and permit distribution. Currently allows for collusion to happen between the census administrator and the election administrator
- Ex_software development
- Certain cryptographic primitives implemented are vulnerable [41]
- No version for voters with special needs

Additionally, the election administrator can know whether a voter has voted or not and a voter with a fake ID might be able to authenticate to vote. Even for private elections, it should be an issue to be solved.

In short and considering all the points reviewed in the analysis, the authors estimate that nVotes is currently not ready to be introduced for public, politically binding elections due to the limitations in auditing, monitoring, backup and potential collusion. Its current secure rage is that of private elections, always taking into account the highly recommended distribution of administrative roles.

To conclude, the authors hope that it can contribute, even if modestly, to improve the knowledge and security level in the deployment of e-voting systems, through the comprehensive, multi-faceted results presented. Nonetheless, in order to make the best possible decision, Elections Officials should also consider complementing the information contained in this document with other inputs from different, more atomistic and cryptographically formal analyses.

Acknowledgements. The contribution of Dr. David Duenas-Cid is based upon work supported by the Estonian Research Council grant (PUT 1361 "Internet Voting as Additional Channel for Legally Binding Elections: Challenges to Voting Processes Reengineering", 2017–2020); and by the Polish National Research Center grant (Miniatura 3 - 2019/03/X/HS6/01688 "Zaufanie do technologii w e-administracji: Powtórna analiza nieudanego wdrożenia elektronicznych maszyn do głosowania w Holandii (2006-07)").

References

1. Vinkel, P., Krimmer, R.: The how and why to internet voting an attempt to explain e-stonia. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 178–191 (2017).
2. Solvak, M., Vassil, K.: Could Internet Voting Halt Declining Electoral Turnout? New Evidence That E-Voting Is Habit Forming. Policy and Internet. 10, 1, 4–21 (2018).
3. Krimmer, R. et al.: How much does an e-vote cost? Compared Costs per Vote in Multichannel Elections in Estonia. E-Vote-ID 2018. pp. 117–132 Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-030-00419-4.
4. Krimmer, R. et al.: New methodology for calculating cost-efficiency of different ways of voting: is internet voting cheaper? Public Money Manag. 0, 0, 1–10 (2020).
5. Trechsel, A.H. et al.: Potential and Challenges of E-Voting in the EU Study., Bruss (2016).
6. Gjøsteen, K.: Analysis of an internet voting protocol. IACR Cryptol. ePrint Arch. 1–16 (2010). https://doi.org/10.1007/978-3-642-32747-6_1.
7. Kulyk, O. et al.: Electronic Voting with Fully Distributed Trust and Maximized Flexibility Regarding Ballot Design. EVOTE2014. pp. 139–149 TUT Press, Bregenz (2014).
8. Oostveen, A.-M., Van den Besselaar, P.: Security as belief User's perceptions on the security of electronic voting systems. Electron. Voting Eur. Technol. 47, May 2014, 73–82 (2004).
9. Council of Europe: Recommendation CM/Rec(2017)5 of the Committee of Ministers to member States on standards for e-voting. (2017).
10. Driza Maurer, A.: Updated European standards for e-voting. The Council of Europe recommendation Rec(2017)5 on standards for e-voting. In: Krimmer, R. et al. (eds.) E-Vote-ID 2017. pp. 146–162 Springer, Bregenz (2017).
11. Bräunlich K., Grimm R., Richter P., "Sichere Internetwahlen Ein rechtswissenschaftlich-informatisches Modell." Nomos (2013).
12. Hammer V., Pordesch U., KORA (Konkretisierung Rechtlicher Inforderungen). Betriebliche Telefon und ISDN-Anlagen rechtsgemäss gestaltet. (1993.)
13. "Common Criteria for Information Technology Security Evaluation Part 1 : Introduction and general model July 2009 Revision 3 Final Foreword," *Nist*, vol. 49, no. July, 93 (2009).
14. Neumann S.R., "Evaluation and Improvement of Internet Voting Schemes Based on Legally-Founded Security Requirements,", Technische Universität Darmstadt. (2016).
15. "Electoral Commision New South Gales,". [Online]. Available: http://www.elections.nsw.gov.au/voting/ivote Last accessed 2020/05/12
16. Marcos del Blanco D. Y. , Panizo Alonso L., and Hermida Alonso JA., "The need for Harmonization in the online voting field: Towards an European Standard for edemocracy," E-Vote-ID 2016, Bregenz, Austria, October 18-21, 2016, Proceedings, pp. 339–340 (2016).
17. Standards, "Guidelines on the implementation of the provisions of Recommendation CM/Rec(2017)5 on standards for e-voting," *1289 th Meet. , 14 June 2017 2 . 3 Ad hoc Comm. Expert. Leg. , Oper. Tech. Stand. e- voting (CAHVE)*, no. June, pp. 1–19, (2017)
18. "Constitución Española" pp. 101931–101941, https://www.boe.es/legislacion/documentos/ConstitucionCASTELLANO.pdf Last accessed 2020/05/12
19. Panizo Alonso L., Gasco M., Marcos del Blanco DY, Hermida Alonso JA, Alaiz Moreton H. "E-voting system evaluation based on the Council of Europe recommendations: Helios Voting", IEEE Transactions on Emerging Topics in Computing, (2018).
20. D. Simić-Draws *et al.*, "Holistic and Law Compatible IT Security Evaluation: Integration of Common Criteria, ISO 27001/IT- and KORA," *Int. J. Inf. Secur. Priv.*,7 pp. 16–35, (2013).
21. Goodman, L.: Snowball Sampling. Ann. Math. Stat. 32, 148–170 (1961).
22. Kish, L.: Sample Design in business research. [American Statistical Association,Ltd.].

23. Benaloh, J.D.C., Rivest, R., Ryan et al.: End-to-end verifiability. arXiv e-prints. (2014).
24. Bernhard, D., Neumann, S., Volkamer, M.: Towards a Practical Cryptographic Voting Scheme Based on Malleable Proofs. 4th International Conference, Vote-ID 2013 Proceedings. pp. 176–192. Springer Berlin Heidelberg, Berlin, Heidelberg (2013).
25. Zissis, D., Lekkas, D.: Design, Development, and Use of Secure ElectronicVotingSystems,http://services.igiglobal.com/resolvedoi/resolve.aspx? (2014).
26. Taiwhenua, T.T.: The Department of Internal Affairs - Online voting, https://www.dia.govt.nz/online-voting. Last accessed 2020/05/12
27. Marcos del Blanco DY "Cybersecurity applied to e-democracy: Cryptographic analysis and development of a practical evaluation methodology for remote electronic voting systems and its application to the most relevant solutions" University of Leon, (2018) http://riasc.unileon.es/archivos/documentos/tesis/Tesis_David_Y_Marcos.pdf
28. nVotes Homepage https://nvotes.com/ last accessed 2020/05/14
29. Impact Accelerator, https://www.impact-accelerator.com/ last accessed 2020/05/14
30. Marcos del Blanco DY., Gascó M. "A Protocolized, Comparative Study of Helios Voting and Scytl/iVote" International Conference on eDemocracy & eGovernment (ICEDEG), pp. 31-38. IEEE (2019).
31. Adida, B.: Helios: Web-based Open-audit Voting. In: Proceedings of the 17th Conference on Security Symposium. pp. 335–348. USENIX Association, Berkeley, CA, USA (2008).
32. Kulyk, O., Teague, V., Volkamer, M.: Extending Helios Towards Private Eligibility Verifiability. 5th International Conference, VoteID 2015,Proceedings. pp. 57–73. Springer (2015).
33. Cortier V., Gaudry P., Glondu S. "Belenios: a simple private and verifiable electronic voting system". Foundations of Security, Protocols, and Equational Reasoning, pp.214-238, (2019).
34. Esendex Homepage, https://www.esendex.es/ last accessed 2020/05/14
35. Cloudfare Homepage, https://www.cloudflare.com last accessed 2020/05/15
36. Fail2ban Homepage, https://www.fail2ban.org last accessed 2020/05/14
37. Verificatum Homepage, https://www.verificatum.org/ last accessed 2020/05/14
38. Open STV Homepage, https://www.opavote.com/?openstv=1 last accessed 2020/05/14
39. T. ElGamal. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms". InProceedings of CRYPTO 84, vol 196 of LNCS, pp. 10-18, Springer, (1985).
40. Pedersen T. "A Threshold Cryptosystem without a Trusted Party". Advances in Cryptology – EUROCRYPT'91. D. Davies editor. Springer – Verlag LNCS series, (1991).
41. Fiat A., Shamir A. "How to prove yourself: Practical solutions to identification and signature problems." CRYPTO'86, pp. 186-194. Santa Barbara, USA. (1986).
42. Schnorr. CP "Efficient Identification and Signatures for Smart Cards". CRYPTO '89, vol. 435 de Lecture Notes in Computer Science, pp. 239–252. Springer, (1989).
43. V. Cortier, "Formal Verification of e-Voting: Solutions and Challenges," ACM SIGLOG News, vol. 2, no. 1, pp. 25–34, (2015)
44. Acemyan CZ, Kortum P., et al. "From Error to Error: Why Voters Could not Cast a Ballot and Verify Their Vote with Helios, Pret a Voter and Scantegrity II". Rice University. In The Usenix Journal of Election Technology and Systems, (2015)
45. Juels A., Catalano D., Jakobsson M. Coercion-Resistant Electronic Elections. In: Chaum D. et al. (eds) Lecture Notes in Computer Science, vol 6000. Springer, (2010)
46. Hirt, M., Sako, K.: Efficient Receipt-free Voting Based on Homomorphic Encryption. 19th ICTAC. pp. 539–556. Springer (2000).
47. El Español Homepage, https://www.elespanol.com/espana/20160511/123987880_0.html last accessed 2020/05/15
48. 20 minutos Homepage, https://www.20minutos.es/noticia/2419700/0/podemos-defiende-fiabilidad/sistema-votacion-acusaciones/primarias/ 2020/05/15

My vote, my (personal) data: remote electronic voting and the General Data Protection Regulation

Adrià Rodríguez-Pérez[1 2][0000-0002-5581-1340]

[1] Scytl Secure Electronic Voting, S.A., 08008 Barcelona, Spain
[2] Universitat Rovira i Virgili, 43002 Tarragona, Spain
adria.rodriguez@scytl.com

Abstract. On 19 September 2019, the Data Protection Authority of the Åland Islands (in Finland) published its findings on the data processing audit for the autonomous region's parliamentary election special internet voting procedure. It claimed that there were faults in the documentation provided by the processor, which in turn meant that the election's integrity could not be guaranteed without further precautions from the government of the Åland Islands. Since the European Union's General Data Protection Regulation (GDPR) entered into force in May 2018, it has set new critical requirements for remote electronic voting projects. Yet, to date, no specific guidance nor research has been conducted on the impact of GDPR on remote electronic voting. Tacking stock of two recent internet voting experiences in the Åland Islands and France, this paper aims at identifying and understanding these new requirements. More specifically, based on these two case studies it analyses four different challenges on the processing of personal data in remote electronic voting under the GDPR: the definitions and categories of personal data processed in online voting projects; the separation of duties between data controllers and data processors; the secure processing of (sensitive) personal data, including the use of anonymisation and pseudonymisation techniques; as well as post-election processing of personal data, and possible limits to (universal) verifiability and public access to personal data.

Keywords: Internet voting, data protection law, GDPR

1 Introduction

Since the European Union (EU)'s General Data Protection Regulation (GDPR) entered into force in May 2018, it has set new critical requirements for the processing of personal data in remote electronic voting projects. In some countries where internet voting is widely used, both in public as well as in private elections, data protection authorities have adopted or updated their regulations on i-voting. This is the case, for instance, of the Recommendation on the security of e-voting systems by the French *Commission Nationale de l'Informatique et des Libertés* (CNIL). Yet, this case is rather the exception than the rule. In turn, no specific guidance at the European level has been provided on this matter.

Tacking stock of two recent internet voting experiences in the Åland Islands (an autonomous region in Finland) and France, this paper aims at identifying the nature of these new requirements, to understand how they have been translated into practice, and to comprehend how they have impacted the implementation of i-voting. More specifically, it addresses the four following aspects: (i) the definitions and categories of personal data processed in these two experiences; (ii) the separation of duties between data controllers and data processors; (iii) the secure processing of (sensitive) personal data, including anonymisation and pseudonymisation techniques; and (iv) post-election processing of personal data, including its destruction, as well as possible limits to (universal) verifiability and public access to personal data. To the best of our knowledge, this one is the first academic paper on the topic. Thus, our goal is to identify some critical aspects in the implementation of GDPR's requirements in online voting, rather than to come up with solutions on how to guarantee compliance with its provision.

To do so, we start by providing an overview of the legal framework governing the use of personal data in elections (section 2). First, we analyse the wider, overarching principle of secret suffrage (section 2.1). In the framework of remote electronic voting, it helps us identify the requirement of data minimisation, as well as that of respect with provisions on data protection. We then move to study the main provisions on personal data protection at the European level (section 2.2). More specifically, we study data protection by comparing it to the international right to respect for private life, and then we move to analyse the more recent provisions on European data protection law, with a specific focus on the EU's GDPR, which was adopted in May 2016 and entered into force two years later. This analysis will allow us to argue that the requirements for personal data processing are independent of and complementary to those of secret suffrage. Following (section 3), the actual implementation of the GDPR's provisions in real internet voting projects is studied. We focus on the extent to which the (planned) used of internet voting in the Åland Islands (section 3.1) and France (section 3.2) complied with the provisions of the new EU Regulation. Drawing from these two projects, we have identified the four above-mentioned trends, which we consider specifically relevant when it comes to the processing of personal data in i-voting under the GDPR (section 3.3). After this analysis, the fourth and final section provides the conclusion of the paper, attempts to draw some lessons learned, acknowledged limitations in our study, and outlines potential future research.

2 Beyond secret suffrage: European data protection law

2.1 The right to vote and secret suffrage

Secret suffrage is one of the key principles of the right to free elections. The obligation to guarantee the secrecy of the ballot features in both Article 21(3) of the Universal Declaration on Human Rights (UDHR) as 'secret vote', as well as in Article 25(b) of the International Covenant on Civil and Political Rights (ICCPR) as elections held by 'secret ballot' (International IDEA, 2014: 43). In Europe, the right to free elections is enshrined in Article 3 of the Protocol (no. 1) to the Convention for the Protection of Human Rights and Fundamental Freedoms (ECHR). Article 3 of the Protocol explicitly

recognises that democratic elections are to be held by secret vote or by equivalent free voting procedures. In this sense, "the secrecy of the vote is [considered] an aspect of free suffrage, which aims to shield voters from any pressure that might result from the knowledge of his [sic] choice by third parties and, in fine, to ensure the honesty and sincerity of the vote" (Lécuyer, 2014: 76).

As part of secret suffrage, the Council of Europe's recently updated Recommendation CM/Rec(2017)5 on standards for e-voting specifies that "[p]rovisions on data protection shall be respected" (Council of Europe, 2017a: 20). More specifically, it states that "[t]he e-voting system shall process and store, as long as necessary, only the personal data needed for the conduct of the e-election" (2017a: 20), and that "[t]he e-voting system and any authorised party shall protect authentication data so that unauthorised parties cannot misuse, intercept, modify, or otherwise gain knowledge of this data" (Council of Europe, 2017a: 21). The Guidelines on implementation of the Recommendation also state that "[t]he legal framework should include procedures for the process of data destruction, in particular to align processing, storing and destruction of the data (and equipment) of voting technology with the personal data protection legislation" (Council of Europe, 2017c: 28.d), and that "printing of voter identification data such as polling cards should be reviewed to ensure security of sensitive data" (Council of Europe, 2017c: 48.a).

These standards are related to the requirement of 'data minimisation', which refers to "data necessary for fulfilling legal requirements of the voting process" (Council of Europe, 2017b: 65). Interestingly enough, this provision of the Recommendation's Explanatory Memorandum states that it is "[t]he electoral management body in charge of organising e-voting [who] identifies such data and should be able to explain what are the underlying legal provisions and considerations that render them necessary" (Council of Europe, 2017b: 65). The Explanatory Memorandum concludes that "data minimisation aims at ensuring data protection and is part of vote secrecy" (Council of Europe, 2017b: 65). However, and as we will see now, we should consider personal data protection requirements as protecting a distinct, independent legal asset.

2.2 The rights to respect for private life and to personal data protection

From the right to respect for private life to the right to personal data protection. The right to privacy (article 12 of the UDHR and art. 17 of the ICCPR), also known as the right to respect for private life (article 8 of the ECHR), provides that "everyone has the right to respect for his or her private and family life, home and correspondence." Interference with this right by a public authority is prohibited, except where the interference is in accordance with the law, pursues important and legitimate public interests and is necessary in a democratic society (EU Agency for Fundamental Rights and Council of Europe, 2018: 18). The development of computers and the Internet presented new risks to the right to respect for private life. In response to the need for specific rules governing the collection and use of personal information, a new concept of privacy emerged, known as 'information privacy' or the 'right to informational self-determination' (EU Agency for Fundamental Rights and Council of Europe, 2018: 18).

Data protection in Europe began in the seventies at the national level, and afterwards, data protection instruments were established at the European level: first, in the Council of Europe's Convention for the Protection of Individuals with regard to Automatic Processing of Personal Data (Convention 108), adopted in 1981; and then in the European Union's Directive 95/46/EC on the protection of individuals with regards to the processing of personal data and on the free movement of such data. Over the years, data protection developed into a distinctive value that is not subsumed by the right to respect for private life (EU Agency for Fundamental Rights and Council of Europe, 2018: 19).

While both rights strive to protect similar values (i.e., the autonomy and human dignity of individuals) the two differ in their formulation and scope: while the right to respect for private life consists of a general prohibition on interference, the protection of personal data is viewed as a modern and active right, putting in place a system of checks and balances to protect individuals whenever their personal data are processed. The right to personal data protection thus comes into play whenever personal data are processed. Therefore, it is broader than the right to respect for private life. Any processing operation of personal data is subject to appropriate protection. Data protection concerns all kinds of personal data and data processing, irrespective of the relationship and impact on privacy. Processing of personal data may infringe on the right to private life. However, it is not necessary to demonstrate an infringement on private life for data protection rules to be triggered (EU Agency for Fundamental Rights and Council of Europe, 2018: 20). In our opinion, the same could be argued for personal data protection and secret suffrage: the former cannot be subsumed by this latter principle.

Data protection regulations in the EU. From 1995 until May 2018, the principal EU legal instrument on data protection was the Directive 95/46/EC (EU Agency for Fundamental Rights and Council of Europe, 2018: 29). In 2009, debates on the need to modernise EU data protection rules began, with the Commission launching a public consultation about the future legal framework for the fundamental right to personal data protection. The proposal for the regulation was published by the Commission in January 2012, starting a long legislative process of negotiations between the European Parliament and the Council of the EU. After adoption, the GDPR provided for a two-year transition period. It became fully applicable on 25 May 2018, when the Directive 95/46/EC was repealed (EU Agency for Fundamental Rights and Council of Europe, 2018: 30).

The adoption of GDPR in 2016 modernised EU data protection legislation, making it fit for protecting fundamental rights in the context of the digital age's economic and social challenges. The GDPR preserves and develops the core principles and rights of the data subject provided for in the Directive 95/46/EC. In addition, it has introduced new obligations requiring organisations to implement data protection by design and default, to appoint a Data Protection Officer in certain circumstances, to comply with a new right to data portability, and to comply with the principle of accountability (EU Agency for Fundamental Rights and Council of Europe, 2018: 30). Furthermore, under EU law regulations are directly applicable and there is no need for national implementation. Therefore, the GDPR provides for a single set of data protection rules to the whole EU. Finally, the regulation has comprehensive rules on territorial scope: it

applies both to businesses established in the UE, as well as to controllers and processors not established in the EU that offer goods or services to data subjects in the EU or monitor their behaviour (EU Agency for Fundamental Rights and Council of Europe, 2018: 31).

Ahead of the elections to the European Parliament of 2019, the European Commission released a guidance document on the application of the Union's data protection law in the electoral context. The goal of the document was to "provide clarity to the actors involved in election processes – such as national electoral authorities, political parties, data brokers and analysts [and] highlight the data protection obligations of relevance for elections" (European Commission, 2018: 2). Specifically, the document addressed key obligations for the various actors, the role as data controller or data processor, principles, lawfulness of processing and special conditions for the processing sensitive data, security and accuracy of personal data, and data protection impact assessment, to name just a few examples. Yet, it is worth noticing that the guidance document does not make specific reference to the use of (remote) electronic voting technologies.

3 Remote electronic voting experiences under the GDPR

3.1 The parliamentary elections in the Åland Islands, Finland

In 2014, the Government of the Åland Islands started studying how to amend the Election Act for Åland. Among other issues, they wanted to know whether internet voting could be introduced for the elections to their parliament. Work on a new Election Act for Åland started in 2017. A draft law was approved by the Government in 2018, and the Parliament passed it in January 2019. The law was then signed by the President of Finland by mid-May. Thus, the Election Act for Åland, together with the Act on the Autonomy of Åland, provide the basic electoral framework for the autonomous region. The law provides that "[a]dvance voting via the internet shall be organised in parliamentary elections if a reliable system for electronic voting via the internet is available" (Election Act for Åland, section 78).

The Government of Åland started to work on the procurement of an internet voting system for the 2019 parliamentary elections in 2018. In March, they published a Request for Information. They received answers from five different providers, but they realised that only two providers would meet the requirements of their tender. The tender was published in October 2018 and two offers were received (from the two vendors that they expected that would bid). Scytl Secure Electronic Voting, S.A. (Scytl) was awarded the project. The contract with Scytl was signed in early January 2019.

On 19 June, the Åland Data Protection Authority (DPA) decided to conduct a data protection audit for the 2019 Election Special Internet Voting Procedure (2019a)[1]. The goal was to "identify potential risks with the treatment before the election would take place" (DPA, 2019c). The audit was conducted by TechLaw Sweden AB (TechLaw). While the object of the audit was the Government of the Åland Islands' treatment of i-voters' personal data, "Scytl [the processor] got the questions asked directly from the

[1] All translations from the original reports in Swedish by the author, using an online tool.

Data Inspectorate [as] a practical solution to save time" (DPA, 2019c). The report was concluded on 12 September and the findings were published on the 19 of September, together with another report by the DPA. The DPA criticised, "inter alia, the lack of clarity of contracts between the Government, ÅDA[2] and Scytl, as well as, the issue regarding the personal data of i-voters" (Krimmer et al., 2019: 11). The report also identified faults in the documentation provided by the processor (Scytl), which in turn meant that the election's integrity could not be guaranteed without further precautions from the government of the Åland Islands (DPA, 2019b). On 13 December, the DPA also published a report with comments from Scytl. The purpose of the comment from Scytl was "to find out any misunderstandings that may have arisen regarding their security measures by the reporter employed by the Data Inspectorate" (DPA, 2019b).

3.2 The consular elections in France

Internet voting in France dates back to 2003, with the passing of the first law allowing the use of internet voting for the elections to the Assembly of French Citizens Abroad (*Sénat,* 2014: 38)[3]. Subsequently, the Ministry of Foreign and European Affairs (MEAE) carried out three pilot projects during the 2003, 2006, and 2009 elections (OSCE/ODIHR, 2012b: 9). Nowadays, internet voting is foreseen as an additional voting channel for French voters abroad. They can cast an i-vote for the elections to the National Assembly (the country's directly elected lower house, with 577 seats) and for the election of the Consular Advisers and Delegates. For the elections to the National Assembly, a constitutional amendment of 2008 introduced 11 seats to be elected by voters residing abroad (OSCE/ODIHR, 2012a: 3). In 2012, voters had the possibility to vote online for these seats (*Sénat*, 2014: 37) for the first time (OSCE, 2012a: 1). However, in 2017 this possibility was halted due to "concerns of foreign cyber threats as well as over certain technical issues" (OSCE/ODIHR, 2017: 6). On their side, Consular Advisers and Delegates are based at each embassy with a consular district and at each consular post. They are elected for a six-year period during the month of May, their first elections taking place in 2014 (*Sénat*, 2014: 37). The next elections were scheduled on May 2020. Yet, the MEAE decided to post-pone these elections due to the Covid-19 pandemic. Scytl was also the technology provider for these two elections, having signed a contract with the MEAE for a four-year period in May 2016 (*Sénat*, 2018: 38).

In France, and since internet voting requires the set-up of data files with the citizens enrolled on consular lists (*Sénat*, 2014: 43; 2018: 29), this technology is under the legal supervision of the CNIL. In 2010, the CNIL adopted a Recommendation on the security of e-voting systems (CNIL, 2010). The Recommendation provides "general guidelines regarding minimal privacy, secrecy and security requirements for any internet voting" (OSCE/ODIHR, 2012b: 12). The CNIL prescribes both 'physical' measures (such as access controls to the servers or rules for the clearance of authorized employees), as well as software-related ones (i.e., firewalls) (*Sénat*, 2014: 37). The Recommendation

[2] According to Krimmer et al. (2019: 9): "In Åland, it is not the government itself, but a particular agency, ÅDA, which is acting as the procurement agent being in charge of the procurement process with the Government as the "real" customer."

[3] All translations from the original reports in French by the author.

was updated in 2019, precisely to take stock of the new requirements introduced by the GDPR after it entered into force (CNIL, 2019b). The goal of the update was for it to apply to future developments in internet voting, "with a view to better respect the principles of personal data protection, and to inform data controllers on their choice for an online voting system" (CNIL, 2019a). Furthermore, a General Security Regulatory Framework (RGS) is established by the *Agence nationale de la sécurité des systèmes d'information* (ANSSI) to regulate minimal requirements on "electronic certificates, encryption levels, and authentication mechanisms" (OSCE/ODIHR, 2012b: 12).

3.3 Comparing remote electronic elections under GDPR

In what follows, we provide an overview of the most relevant issues in these two experiences concerning the application of the GDPR. More specifically, we will focus on (i) the definitions and categories of personal data processed; (ii) the separation of duties between data controllers and data processors; (iii) the secure processing of (sensitive) personal data, including the use of anonymisation and pseudonymisation techniques; and (iv) the post-election processing of personal data, including its destruction, as well as possible limits to (universal) verifiability and public access to personal data.

This list of issues is not exhaustive, since these aspects have been identified as relevant in the two experiences studied here. It is likely that additional issues could be raised in different cases, or after the implementation of these two specific projects.

Definition and categories of personal data. According to EU law, data are personal if they relate to an identified or identifiable person, the 'data subject' (EU Agency for Fundamental Rights and Council of Europe, 2018: 83). The GDPR defines personal data as information relating to an identified or identifiable natural person (GDPR, art. 4.1). Any kind of information can be personal data provided that it relates to an identified and identifiable person[4]. Personal data covers information pertaining to the private life of a person, as well as information about their public life (EU Agency for Fundamental Rights and Council of Europe, 2018: 86).

The GDPR stipulates that a natural person is identifiable when he or she "can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, and online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that person" (GDPR, art. 4.1). Yet, according to the Article 29 Data Protection Working Party (Article 29 Working Party), it is also "possible to categorise [a] person on the basis of socio-economic, psychological, philosophical or other criteria and attribute certain decisions to him or her since the individual's contact point (a computer) no longer requires the disclosure of his or her identity in the narrow sense" (2007: 15). Identification, thus, requires elements which describe a person in such a way that he or she is distinguishable from all other persons and recognisable as an individual (EU Agency for Fundamental Rights and Council of Europe, 2018: 89). Establishing the identity of a person may need additional attributes to ensure that a person is not

[4] For the applicability of European data protection law there is no need for actual identification of the data subject: it is sufficient that the person concerned is identifiable.

mistaken for someone else. Sometimes, direct and indirect attributes may have to be combined to identify the individual to whom the information relates. Date and place of birth are often used. In addition, personalised numbers have been introduced in some countries to better distinguish between citizens. Biometric data, such as fingerprints, digital photos or iris scans, location data and online attributes are increasingly used to identify persons in the technological age (EU Agency for Fundamental Rights and Council of Europe, 2018: 90).

Personal data about candidates. Based on the above, it is clear that data about candidates is personal data and thus falls under the scope of the right to personal data protection and of the GDPR. It goes without saying that candidates are to be described in such a way that they are distinguishable from all other persons and recognisable as individuals. How was personal data about candidates processed in these two experiences? In Åland, the online voting process was similar to the paper-based one (Krimmer et al., 2019: 11), where voters do not mark or select a candidate in the ballot but write their number on a blank ballot paper. Likewise, in the Åland's voting platform, voters were not "able to select a candidate by clicking on it in the list of candidates displayed. [Instead, a] voter will need to insert the number of a candidate, exactly like it is done when a voter cast a vote on paper" (Krimmer et al., 2019: 11). On the other hand, in France, the Election Management System service used by the election managers to configure the election (GUES), includes personal data about each candidate. This data includes their name, surname, sex, birth date, phone, e-mail, etc. Similar information is also processed for candidates' substitutes.

Authentication data. Authentication means proving that a certain person possesses a certain identity and/or is authorized to carry out certain activities (EU Agency for Fundamental Rights and Council of Europe, 2018: 83). This is a procedure by which a person is able to prove that they possess a certain identity and/or is authorised to do certain things, such as enter a security area, withdraw money from a banking account or, as in this case: cast an i-vote. Authentication can be achieved by comparing biometric data, such as a photo or fingerprints in a passport, with the data of the person presenting themselves. However, this kind of authentication can only be conducted face-to-face (i.e., when voters cast a paper ballot in polling stations). An alternative for the remote setting is to ask for information which should be known only to the person with a certain identity or authorisation, such as a personal identification number (PIN) or a password. In addition to these, electronic signatures are an instrument especially capable of identifying and authenticating a person in electronic communications (EU Agency for Fundamental Rights and Council of Europe, 2018: 95).

Voter authentication was similar in both the Åland Islands and in France. In Åland, the voters had to go to a website provided by ÅDA and authenticate via BankID (TechLaw, 2019: 9). Upon successful authentication, the voter received a KeyStore with the election public key (to encrypt the vote) and their voter private key (to digitally sign the encrypted vote). The voter is identified internally by the voting platform using a randomly generated pseudonymous (VoterID) "that is used to ensure that a vote has been cast by an eligible voter and that no voter has voted twice" (Scytl, 2019: 24).

According to Scytl (2019: 24), "under no circumstances can Scytl correlate this voter identifier with the real identity of the voter".

In addition to the vote and the voterID, Scytl's voting system also stores the voters' IP addresses (TechLaw, 2019: 8). In a 2011 ruling, the Court of Justice of the EU (CJEU) held that users' IP addresses "are protected personal data because they allow those users to be precisely identified" (CJEU, 2011: para. 51). The CJEU has also considered that a dynamic IP address, which an online media services provider registers when a person accesses a website that the provider has made accessible to the public, constitutes personal date where only a third party (i.e., the internet service provider) has the additional data necessary to identify the person (EU Agency for Fundamental Rights and Council of Europe, 2018: 91). According to Scytl (2019: 24), it is not possible to link the vote or the voter with the IP because they have "no information to correlate IP addresses with the real identity of the voter".

Encrypted and digitally signed electronic ballots. There are special categories of data, so-called 'sensitive data', which require enhanced protection and, therefore, are subject to a special legal regime (EU Agency for Fundamental Rights and Council of Europe, 2018: 83). These are special categories of personal data which, by their nature, may pose a risk to the data subjects when processed and need enhanced protection. Such data are subject to a prohibition principle and there are a limited number of conditions under which such processing is lawful (EU Agency for Fundamental Rights and Council of Europe, 2018: 96). Within the framework of the GDPR, the following categories are considered sensitive data: personal data revealing racial or ethnic origin; political opinions, religious or other beliefs, including philosophical beliefs; trade union membership; genetic data and biometric data processed for the purpose of identifying a person; and, personal data concerning health, sexual life or sexual orientation. Since digital ballots reveal political opinions (they contain the political preferences of voters), they must be considered sensitive data. As a matter of fact, research conducted by Duenas-Cid et al. (2020) concludes that it was precisely the processing of political opinions as a special category of personal data that motivated an audit in the Åland Islands.

In both the Åland Islands (Scytl, 2019: 11) and in France, votes are encrypted and sealed in encrypted envelopes (directly on the voter's computers). The encrypted vote is then digitally signed (also in the voting device). Since votes are digitally signed, only the votes cast (and signed) by eligible voters are verified and stored in the voting server (i.e., the digital ballot box) (Scytl, 2019: 38). In the case of Åland, the system also provided individual verifiability (cast-as-intended and recorded-as-cast verifiability). In practice, it means that after casting their vote, voters could log into the voting service to check that their vote had reached the voting server unaltered (TechLaw, 2019: 8).

Data processing: the role of data controllers and data processors. 'Data processing' concerns any operation performed on personal data. According to the GDPR, "processing of personal data […] shall mean any operation […] such as collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, restriction, erasure or destruction" (art. 4.2).

Whoever determines the means and purposes of processing the personal data of others is a controller under data protection law. If several persons take this decision together, they may be joint controllers. A 'processor' is a natural or legal person that processes the personal data on behalf of a controller. If a processor determines the means and purposes of data processing itself, they become a controller. Any person to whom personal data are disclosed is a 'recipient' (EU Agency for Fundamental Rights and Council of Europe, 2018: 101). Any person other than the data subject, the controller, the processor and persons who are authorised to process personal data under the direct authority of the controller or processor is considered a 'third-party'.

The most important consequence of being a controller or a processor is a legal responsibility for complying with the respective obligations under data protection law. In the private sector, this is usually a natural or legal person. In the public sector, it is usually an authority. There is a significant distinction between a data controller and a data processor: the former is the natural or legal person who determines the purposes and the means of processing, while the latter is the natural or legal person who processes the data on behalf of the controller, following strict instructions. In principle, it is the data controller that must exercise control over the processing and who has responsibility for this, including legal liability (EU Agency for Fundamental Rights and Council of Europe, 2018: 101). Yet, processors also have an obligation to comply with many of the requirements which apply to controllers[5]. Whether a person has the capacity to decide and determine the purpose and means of processing will depend on the factual elements or circumstances of the case.

As has been already seen, according to the Council of Europe's Recommendation it is "[t]he electoral management body in charge of organising e-voting [who] identifies such data and should be able to explain what are the underlying legal provisions and considerations that render them necessary" (Council of Europe, 2017b: 65) In a similar vein, the GDPR clearly states that the processor may only process personal data on instructions from the controller, unless the EU or Member State law requires the processor to do so (art. 29). According to the GDPR, if the power to determine the means of processing is delegated to a processor, the controller must nonetheless be able to exercise an appropriate degree of control over the processor's decisions regarding the means of processing. Overall responsibility lies with the controller, who must supervise the processor to ensure that their decisions comply with data protection law and their instructions (EU Agency for Fundamental Rights and Council of Europe, 2018: 108).

For the sake of clarity and transparency, the details of the relationship between a controller and a processor must be recorded in a written contract (GDPR, art. 28.3 and .9). The contract between the controller and the processor is an essential element of their relationship, and is a legal requirement (GDPR, art. 28.3). It must include, in particular, the subject matter, nature, purpose and duration of the processing, the type of

[5] Under the GDPR, "processors must maintain a record of all categories of processing activities to demonstrate compliance with their obligations under the regulation" (art. 30.2). Processors are also required to implement appropriate technical and organisational measures to ensure the security of processing (art. 32), to appoint a Data Protection Officer (DPO) in certain situations (art. 37), and to notify data breaches to the controller (art. 33.2).

personal data and the categories of data subjects. It should also stipulate the controller's and the processor's obligations and rights, such as requirements regarding confidentiality and security. Having no such contract is an infringement of the controller's obligation to provide written documentation of mutual responsibilities, and could lead to sanctions (EU Agency for Fundamental Rights and Council of Europe, 2018: 109). Yet, in the case of the Åland Islands the DPA criticized, precisely, "the lack of clarity of contracts between the Government, ÅDA and Scytl" (Krimmer et al., 2019: 11). In France, the CNIL's updated Recommendation specifically provides that "the processing of personal data, including the voting systems, must in principle be subject to a data protection impact assessment (PIA) when meet at least two of [several] criteria". Among these, this project seems to include, indeed, at least two of these criteria, i.e.: processing of sensitive data (i.e., political opinions) and large-scale processing of personal data. Thus, such an assessment is required in internet voting in France.

Anonymisation, pseudonymisation and (sensitive) personal data. Data are anonymised if they no longer relate to an identified or identifiable individual (EU Agency for Fundamental Rights and Council of Europe, 2018: 83). Pseudonymisation is a measure by which personal data cannot be attributed to the data subject without additional information, which is kept separately. The 'key' that enables re-identification of the data subjects must be kept separate and secure. Data that have undergone a pseudonymisation process remains personal data (EU Agency for Fundamental Rights and Council of Europe, 2018: 83). The principles and rules of data protection do not apply to anonymised information. However, they do apply to pseudonymised data (EU Agency for Fundamental Rights and Council of Europe, 2018: 83).

The process of anonymising data means that all identifying elements are eliminated from a set of personal data so that the data subject is no longer identifiable (GDPR, Recital 26). In its Opinion 05/2014, the Article 29 Working Party analysed the effectiveness and limits of different anonymisation techniques. It acknowledged the potential value of such techniques, but underlined that certain techniques do not necessarily work in all cases. To find the optimal solution in a given situation, the appropriate process of anonymisation should be decided on a case-by-case basis. Irrespective of the technique used, identification must be prevented, irreversibly. This means that for data to be anonymised, no element may be left in the information which could, by exercising reasonable effort, serve to re-identify the person(s) concerned (GDPR, Recital 26). The risks of re-identification can be assessed by taking into account "the time, effort or resources needed in light of the nature of the data, the context of their use, the available re-identification technologies and related costs" (EU Agency for Fundamental Rights and Council of Europe, 2018: 94). When data have been successfully anonymised, they are no longer personal data and data protection legislation no longer applies. On the other hand, pseudonymisation means that certain attributes (such as name, date of birth, sex, address, or other elements that could lead to identification) are replaced by pseudonym. EU law defined 'pseudonymisation' as 'the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that

the personal data are not attributed to an identified or identifiable natural person' (GDPR, art. 4.5). Contrary to anonymised data, pseudonymised data are still personal data and are therefore subject to data protection legislation. Although pseudonymisation can reduce security risks to the data subjects, it is not exempt from the scope of the GDPR (EU Agency for Fundamental Rights and Council of Europe, 2018: 94). The GDPR recognises various uses of pseudonymisation as an appropriate technical measure for enhancing data protection, and is specifically mentioned for the design and security of its data processing (GDPR, art. 25.1). It is also an appropriate safeguard that could be used to process personal data for purposes other than for which they were initially collected.

Based on these provisions, it is clear that both anonymisation and pseudonymisation techniques were used in these two projects. However, most of the time the data processed is pseudonymised, not anonymised. Since it is always possible to relate the encrypted data to a pseudonymous, which in turn can be related to the actual voter identity[6], it is difficult to argue that no element has been left in the information which could, by exercising reasonable effort, serve to re-identify the person(s) concerned[7]. In Åland, and since multiple voting is supported (Election Act for Åland, Section 61), it is necessary to keep the link between the encrypted vote and the VoterID to cleanse those online votes cast by voters who have cast more than one i-vote, as well as those who have also cast a postal vote or an advanced one in polling stations. In France, it is necessary to prevent i-voters from casting a paper vote in polling stations on election day[8]. In order to prevent a voter from casting a second vote, the voter rolls need to be updated. More specifically, at the end of the internet voting period, a mark is included by the side of the name of those voters who have already voted, i.e.: a list of voters having voted (*liste d'émargement*) is generated. The main implication here is that pseudonymous data remain personal data and must be processed as such.

Yet, it is also possible to talk about anonymised data. In the two projects we can find "both technological and procedural guarantees" (Scytl, 2019: 41) in place to break the link between the vote and the voter's pseudonymous identifier (VoterID). In the case of Åland, during the counting phase a mix-net removes the connection between the identity of the voter and their vote (TechLaw, 2019: 8). According to Scytl (2019: 12), this "cryptographic mixing process shuffles the encrypted votes and re-encrypts them at the same time. In this way, any correlation between the original encrypted votes and the re-encrypted ones is broken". Once mixed, it is no longer possible to link a vote with the identity of the voter who has cast it. In France, on the other hand, homomorphic tallying is used. In homomorphic tallying, the different options (whether selected or not) are encrypted separately, aggregated, and then decrypted anonymously. When the

[6] Which is necessary to "to guarantee that all votes have been cast by eligible voters and that only the appropriate number of remote electronic votes per voter gets counted" (Scytl, 2019: 38).

[7] Recital 26 of the GDPR explicitly includes a scenario where it is foreseeable that further data recipients, other than the immediate data user, may attempt to identify the individuals (EU Agency for Fundamental Rights and Council of Europe, 2018: 91).

[8] Contrary to good practice (Council of Europe, 2017c: 9.b), in France once a voter has cast an i-vote, they cannot cast a second vote in person to cancel it.

voter issues their vote, the voting client generates as many cyphertexts as possible options. Therefore, the encrypted vote is represented as an array of as many individual ciphertexts as possible voting options there are within the ballot. During the counting phase, the digital ballot box is exported from the online component of the voting system and imported in the offline one. In the offline environment, all the ciphertexts from all the votes corresponding to the same voting options are aggregated (multiplied), which allows for the computation of a unique aggregated cyphertext for each option. In both cases, the private key used for decryption is protected by a cryptographic secret-sharing scheme (Shamir) that requires the collaboration of several members of the electoral commission to reconstruct the key before decryption. Thus, to decrypt these results, it is required that a minimum number of their members meet to reconstruct the election private key: i.e., three out of five persons in Åland (Election Act for Åland, Section 61) and four out of the eight members of the Bur*eau de vote électronique* (BVE) in France (Cod*e électoral,* R177-5).

Post-election: the destruction of data, universal verifiability and public access to personal data and. The CNIL's Recommendation (2019a) states that all supporting files of an election (such as copies of the source and executable codes of the programs and the underlying system, voting materials, signature files, results' files, backups) must be kept under seal until the channels and deadlines for litigation are exhausted. This conservation must be ensured under the supervision of the electoral commission under conditions guaranteeing the secrecy of the vote. Obligation must be made to the service provider, if necessary, to transfer all of these media to the person or to the third party named to ensure the conservation of these media. When no contentious action has been taken to exhaust the time limits for appeal, these documents must be destroyed under the supervision of the BVE. This requirement is not new, and already in 2012 various audits were conducted on data destruction in the context of the parliamentary elections (OSCE/ODIHR, 2012b: 13). Along these lines, the Council of Europe's Recommendation also provides, in its Explanatory Memorandum, that "[t]he duration of processing, storing etc. [of personal data] also depends on legal requirements, namely those related to appeals". While these measures may be necessary to ensure the preservation of data protection in the long term, they may prevent the election data from being audited or universally verified[9]. Notwithstanding, the Election Act for Åland (Section 99) requires that "after confirming the result of the election, the ballot papers and a copy of the combined list of candidates or a copy of a list of presidential candidates is placed in a container, which shall be sealed as islaid down by the Ministry of Justice. These are to be kept until the next corresponding elections have been conducted."[10]

Overall, there is a growing realisation of the importance of government transparency for the functioning of a democratic society (EU Agency for Fundamental Rights and

[9] Universal verifiability refers to "tools which allow any interested person to verify that votes are counted as recorded" (Council of Europe, 2017b: 56).

[10] That is so even if an "appeal shall be sent to a competent Provincial Administrative Court within 14 days from the confirmation of the election results" (Election Act for Åland, Section 102).

Council of Europe, 2018: 62). The right to receive information, which forms part of freedom of expression, may come into conflict with the right to data protection if access to documents would reveal other's personal data. Art. 86 of the GDPR clearly provides that personal data in official documents held by public authorities and bodies may be disclosed by the authority or body concerned in accordance with EU or Member State's law to reconcile public access to official documents with the right to data protection (EU Agency for Fundamental Rights and Council of Europe, 2018: 63). Balancing between data protection and access to documents requires a detailed, case-by-case analysis. Neither right can automatically overrule the other. The CJEU has had the chance to interpret the right to access to documents containing personal data in two cases (EU Agency for Fundamental Rights and Council of Europe, 2018: 65). According to these judgements, interference with the right to data protection in the context of access to documents needs a specific and justified reason. Furthermore, according to the principle of storage limitation, data must be kept 'in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed' (GDPR, art. 5.1.e). For internet voting, it seems advisable that this information is kept at least until the next election has taken place (and not, as it is provided in the CNIL's recommendation, until the channels and deadlines for litigation are exhausted). Consequently, data would have to be erased or anonymised if a controller wanted to store them after they were no longer needed and no longer served their initial purpose (EU Agency for Fundamental Rights and Council of Europe, 2018: 63).

4 Conclusion

The entry into force of the EU's GDPR has set new requirements for the implementation of internet voting in Europe. Yet, no general guidance has yet been provided on how it impacts this kind of projects specifically. In this context, we have aimed at identifying some critical aspects in the implementation of GDPR's requirements in online voting, to understand how they have been translated into practice, and to comprehend how they have impacted the implementation of i-voting projects.

Two sorts of conclusions can be inferred from this research. First, the requirements for personal data processing in remote electronic voting projects are independent of secret suffrage and cannot be subsumed by this latter principle. Personal data protection is broader than the principle of secret suffrage since any processing of personal data is subject to appropriate protection. Thus, data that may not fall under the scope of secret suffrage, such as personal data about candidates, is also covered by the GDPR. Second, our account of the internet voting experiences in the Åland Islands and in France has allowed us to identify some critical aspects related to the GDPR in the implementation of internet voting projects, namely: the categories of personal data processed (both about voters and candidates), as well as the processing of special categories of personal data (i.e., the votes, which are personal data that reveal political opinions); aspects related to the role played by data controllers (normally, electoral authorities) and processors (usually, technology vendors and services' providers); the use of pseudonymisation techniques for the processing of 'sensitive data'; and, the post-election processing

of personal data, including its destruction, as well as possible limits to (universal) verifiability and public access to personal data. As we have seen, all these aspects could benefit from more guidance, be it by the national regulator or at the wider EU-level.

Acknowledgments. This work has received funding from the European Commission under the auspices of PROMETHEUS Project, Horizon 2020 Research and Innovation action (Grant Agreement No. 780701).

References

1. Act on the Autonomy of Åland (1991, last amended 2010).
2. Åland Data Protection Authority: DNR T1-2019 (2019). https://www.di.ax/anslagstavla/dnr-t1-2019, last accessed 2020/08/03. We refer to it as (2019a).
3. Åland Data Protection Authority: Resultat och beslut av den beslutade Dataskyddstillsynen gällande personuppgiftsbehandling i Lagtingsvalet, särskilt fokus I-valet Dnr T1-2019 (2019). https://www.di.ax/anslagstavla/dnr-t5-2019, last accessed 2020/08/03. We refer to it as (2019b).
4. Åland Data Protection Authority: Rapport om Säkerhetsåtgärder i E-valet samt svar från Scytl, (2019). https://www.di.ax/anslagstavla/rapport-om-sakerhetsatgarder-e-valet-samt-svar-fran-scytl, last accessed 2020/08/03. We refer to it as (2019c).
5. Article 29 Data Protection Working Party: Opinion 4/2007 on the concept of personal data (2007). https://www.clinicalstudydatarequest.com/Documents/Privacy-European-guidance.pdf, last accessed 2020/08/03.
6. Article 29 Data Protection Working Party: Opinion 05/2014 on Anonymisation Techniques (2014). https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf, last accessed 2020/08/03.
7. CNIL: Délibération n° 2010-371 du 21 octobre 2010 portant adoption d'une recommandation relative à la sécurité des systèmes de vote électronique (2010). https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000023124205&categorieLien=id, last accessed 2020/08/03.
8. CNIL: Sécurité des systèmes de vote par internet : la CNIL actualise sa recommandation de 2010 (2019). https://www.cnil.fr/fr/securite-des-systemes-de-vote-par-internet-la-cnil-actualise-sa-recommandation-de-2010, last accessed 2020/08/03. We refer to it as (2019a).
9. CNIL: Délibération n° 2019-053 du 25 avril 2019 portant adoption d'une recommandation relative à la sécurité des systèmes de vote par correspondance électronique, notamment via Internet (2019). https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000038661239, last accessed 2020/08/03. We refer to it as (2019b).
10. Code électoral, France (last amended 2019).
11. Council of Europe: Recommendation CM/Rec(2017)5 of the Committee of Ministers to member States on standards for e-voting (2017). https://search.coe.int/cm/Pages/result_details.aspx?ObjectId=0900001680726f6f, last accessed 2020/08/03. We refer to it as (2017a).
12. Council of Europe: Explanatory Memorandum to Recommendation CM/Rec(2017)5 of the Committee of Ministers to member States on standards for e-voting (2017). https://search.coe.int/cm/Pages/result_details.aspx?ObjectId=0900001680726c0b, last accessed 2020/08/03. We refer to it as (2017b).

13. Council of Europe: Guidelines on the implementation of the provisions of Recommendation CM/Rec(2017)5 on standards for e-voting (2017). https://search.coe.int/cm/Pages/result_details.aspx?ObjectId=090000168071bc84, last accessed 2020/08/03. We refer to it as (2017c).
14. Duenas-Cid, David, Krivonosova, Iulia, Serrano, Radu, Freire, Marlon, Krimmer, Robert: Tripped at the finish line: the Åland Islands internet voting project. In Krimmer, Robert et al. (eds.), Electronic Voting. Fifth International Joint Conference, E-Vote-ID 2020. Springer International Publishing, Cham (2020).
15. Election Act for Åland (2019).
16. EU Agency for Fundamental Rights and Council of Europe: Handbook on European data protection law - 2018 edition (2018). https://fra.europa.eu/sites/default/files/fra_uploads/fra-coe-edps-2018-handbook-data-protection_en.pdf, last accessed 2020/08/03.
17. European Commission: Free and Fair elections. Guidance Document. Commission guidance on the application of Union data protection law in the electoral context (2018). https://ec.europa.eu/commission/sites/beta-political/files/soteu2018-data-protection-law-electoral-guidance-638_en.pdf, last accessed 2020/08/03.
18. International Covenant on Civil and Political Rights (1966).
19. International IDEA: International Obligations for Elections. Guidelines for Legal Frameworks (2014). https://www.idea.int/sites/default/files/publications/international-obligations-for-elections.pdf, last accessed 2020/08/03.
20. Krimmer, Robert, Duenas-Cid, David, Krivonosova, Iulia, Serrano, Radu, Freire, Marlon, Wrede, Casper: Nordic Pioneers: facing the first use of Internet Voting in the Åland Islands (Parliamentary Elections 2019) (2019). https://doi.org/10.31235/osf.io/5zr2e, last accessed 2020/08/03.
21. Lécuyer, Yannick. Le droit a des élections libres. Strasbourg, Council of Europe (2014).
22. OSCE/ODIHR: Republic of France Parliamentary Elections, 10 and 17 June 2012. Needs Assessment Mission Report (2012). https://www.osce.org/files/f/documents/7/5/90763.pdf, last accessed 2020/08/03. We refer to it as (2012a).
23. OSCE/ODIHR: Republic of France Parliamentary Elections, 10 and 17 June 2012. Election Assessment Mission Final Report (2012). https://www.osce.org/files/f/documents/7/7/93621.pdf, last accessed 2020/08/03. We refer to it as (2012a).
24. OSCE/ODIHR: France Presidential and Parliamentary Elections, 2017. Needs Assessment Mission Report (2017). https://www.osce.org/files/f/documents/0/8/311081.pdf, last accessed 2020/08/03.
25. Protocol (no. 1) to the Convention for the Protection of Human Rights and Fundamental Freedoms (European Convention on Human Rights, ECHR) (1952).
26. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation, GDPR) (2016).
27. Sénat: Rapport d'information fait au nom de la commission de lois constitutionelles, de législation, du suffrage universel, du Règlement et d'administration générale (1) sur le vote électronique (2014). https://www.senat.fr/rap/r13-445/r13-4451.pdf, last accessed 2020/08/03.
28. Sénat: Rapport d'information fait au nom de la commission de lois constitutionelles, de législation, du suffrage universel, du Règlement et d'administration générale (1) sur le vote électronique (2018). http://www.senat.fr/rap/r18-073/r18-0731.pdf, last accessed 2020/08/03.

29. Scytl Secure Electronic Voting, S.A.: Åland's I-voting Project. Clarification of the Audit Report by the Åland Data Protection Authority (2019). https://www.di.ax/sites/default/files/attachment/pinboard-message/data_protection_audit_clarifications_v3.0.pdf, last accessed 2020/08/03.
30. TechLaw Sweden AB: Granskning av säkerhetsåtgärder hos Scytl (2019). https://www.di.ax/sites/default/files/attachment/pinboard-message/rapport-aland-scytl-190916_0.pdf, last accessed 2020/08/03.
31. Universal Declaration on Human Rights (1948).

Best Practices and Usable Coercion Resistance

CHVote: Sixteen Best Practices and Lessons Learned

Rolf Haenni, Eric Dubuis, Reto E. Koenig, and Philipp Locher

Bern University of Applied Sciences, CH-2501 Biel, Switzerland
{rolf.haenni,eric.dubuis,reto.koenig,philipp.locher}@bfh.ch

Abstract. The authors of this paper had the opportunity to closely accompany the CHVote project of the State of Geneva during more than two years and to continue the project after its abrupt stop in 2018. This paper is an experience report from this collaboration and the subsequent project continuation. It describes the lessons learned from this project and proposes some best practices relative to sixteen different topics. The goal of the paper is to share this experience with the community.

1 Introduction

Developing a verifiable Internet voting system is a delicate task. While conducting elections over the Internet seems intuitively like a simple matter of counting votes submitted by voters, it actually defines a unique combination of difficult security and privacy problems. As a response to these problems, numerous cryptographic protocols have been proposed to guarantee different combinations of often conflicting security properties. While many aspects of the general problem are solved today in theory, it turned out that transforming them into reliable practical systems is a completely different challenge. In fact, not many projects have been successful so far. In the Switzerland, which played a pioneering role in the early days of Internet voting, three completely untransparent systems were in used for pilot elections with a limited number of voters over more than a decade. They were all black-box system with no verifiability. One of them was the CHVote system from the State of Geneva.

1.1 Project Context

As a response to the third report on *Vote électronique* by the Swiss Federal Council in 2013 and the new requirements of the Swiss Federal Chancellery [1,16], the State of Geneva invited leading scientific researchers and security experts to contribute to the development of their second-generation system *CHVote 2.0*. In this context, a collaboration contract between the State of Geneva and the Bern University of Applied Sciences was signed in 2016. The main goal of this collaboration was the specification of a cryptographic voting protocol that satisfies the new requirements to the best possible degree. The main output of this project is the *CHVote System Specification* document [9], which is publicly available

at the *Cryptology ePrint Archive* since April 2017. In the course of the project, updated document versions have been released in regular intervals.

In November 2018, the council of the State of Geneva announced an abrupt stop of the CHVote 2.0 project due to financial reasons.[1] This implied that with the release of Version 2.1 of the specification document in January 2019, the collaboration between the State of Geneva and the Bern University of Applied Sciences came to an end. In June 2019, the State of Geneva released all the public material that have been created during the CHVote 2.0 project, including the Java source code.[2] The implemented cryptographic protocol corresponds to Version 1.4.1 of the specification document.

To continue the CHVote project independently of the support from the State of Geneva, a new funding from *eGovernment Switzerland* has been acquired by the Bern University of Applied Sciences in August 2019. The main goal of this project was to release a final stable version of the specification document and to update the cryptographic core of the protocol based on the code released by the State of Geneva. As a first project deliverable, the current Version 3.0 of the specification document has been released in December 2019 [9]. At the time of writing this paper, the developed Java code is not yet complete. Since the project is in its final stage, the code is expected to be released soon under a non-proprietary license.[3] The general purpose of the project is to make the achievements available to others for pursuing it further.

1.2 Goals and Paper Overview

This paper presents a retrospective view of the CHVote project over the last four years. The paper is divided into three sections. The two main sections describe our experience and lessons learned from our work related to the specification document and the development of corresponding software, respectively, and the final section discusses some general aspects of the project. The whole paper contains our proposal for best practices on sixteen different topics. We present these topics project in chronological order. While we think that they all have played an important role for the success of our project, we do not claim that the given list is complete or that all points are directly applicable to all similar projects.

Nevertheless, we believe that our experience is worth to be shared with the community, who may struggle with similar problems in other e-voting projects. Sharing our experience with the community is therefore the general goal of this paper. As such, it should been seen as an experience report, which may be helpful in other projects as a guideline for achieving the required quality level in a shorter amount of time. Some of the proposed best practices may even set a certain minimal quality benchmark for e-voting projects in general.

[1] For further details about the reasons for abandoning the project, we refer to the State Council's press statement at https://www.ge.ch/document/12832/telecharger.
[2] See https://chvote2.gitlab.io
[3] See https://gitlab.com/chvote3

2 Specification

Item 1: Modeling the Electoral Systems

Democracies around the world use very different electoral systems to determine how elections and referendums are conducted. A major challenge in the design of CHVote was to cover the variety of electoral systems that exist in the Swiss context. On a single election day, democratic decisions are sometimes taken simultaneously on federal, cantonal, and communal issues, with election laws that differ from canton to canton. To cope with this complexity, we managed to map all electoral systems into a concise and coherent *electoral model* that is applicable to all possible situations. The core of this model is an *election event*, which consists of several independent *k-out-of-n elections*, in which voters can choose exactly k different candidates from a candidate list of size n. An election event is therefore defined by two vectors of such values k and n.

With this simple model, we were able to cover all electoral systems from the Swiss context with their specific properties, exceptions, and subtleties.[4] Elections of the Swiss National Council turned out to be the most complicated use case, but by splitting them into two independent elections, one 1-out-of-n_p party election and one cumulative k-out-of-n_c candidate election, they fit nicely into the general model [9, Section 2.3.2]. By reducing this complexity to essentially two public election parameters and by instantiating them to past election events in all regions of our country, we managed to determine upper limits $k_{\max} = 150$ and $n_{\max} = 1500$ for the overall problem size.

Defining a general electoral model and keeping it as simple and coherent as possible turned out to be a really important abstraction layer, which allowed us to design the cryptographic protocol independently of the variety of election use cases. The above-mentioned estimation of the maximal problem size defined important cornerstones for judging the suitability of cryptographic techniques and for anticipating potential performance bottlenecks. Therefore, we recommend to carefully design a suitable model of the electoral system as early as possible in projects like this.

Item 2: Modeling the Electorate

For a given election event in the given context of the CHVote project, an additional complication is the possibility that voters may not be eligible in all elections. This can happen for two reasons. First, since cantons are in charge of organizing elections, it may happen that elections are held simultaneously in different communes of a given canton, possibly in conjunction with cantonal and federal elections. In such cases, voters are equally eligible for federal and cantonal issues, but not for communal issues. Second, since non-Swiss citizens are allowed to vote in some canton and communes, they may be part of the electorate for cantonal or communal issues, but not for federal issues.

[4] We only had to admit one exception from the general model to allow write-in candidates in some cantons.

To map all possible cases of restricted eligibility into a general model, we introduced in CHVote the concept an *eligibility matrix*, which defines for a given electorate the eligibility of each voter in each election. By connecting this matrix with the two vectors from the general election event model, we can derive for each voter the number of admissible choices in each election. To ensure the correctness of an election outcome, it is absolutely critical for all involved parties to know these values at all times. This includes auditors performing the verification process in the aftermath of an election. The eligibility matrix is therefore a third fundamental public election parameter. Without taking it as additional input, the verification of an election result can not produce a conclusive outcome.

Item 3: Cryptographic Building Blocks

Given the central role of the cryptographic building blocks in a voting protocol, we recommend describing them in the beginning of the specification document. This lays the grounds for the whole document, for example by introducing respective terms and formal notations. By describing the building block next to each other, ambiguities and conflicts in the formal notations can be eliminated in a systematic manner. Given the overall complexity of the CHVote protocol, finding a coherent set of mathematical symbols and using them consistently throughout the whole document was a ongoing challenge during the project. Providing the highest possible degree of disambiguation improves greatly the document's overall readability.

Another important aspect of describing the cryptographic building blocks is to select from the large amount of related literature exactly what is needed for the protocol. Everything can be instantiated to the specific use case and underspecified technical details can be defined to the maximal possible degree. Examples of such technical details are the encoding methods between integers, strings, and byte arrays, or the method of computing hash values of multiple inputs. Another example of an often underspecified building block is the Fiat-Shamir transformation, which is widely applied for constructing non-interactive zero-knowledge protocols [6]. The significance of doing these things right is well documented [4, 15]. A separate chapter on these topics helps to present all important cryptographic aspects in a concise form.

Item 4: Cryptographic Parameters

The collection of cryptographic building blocks defines a list of cryptographic parameters for the protocol. This list of parameters is an important input for every participating party. In CHVote, it consists of a total of twenty parameters, which themselves depend on four top-level security parameters [9, Section 6.3.1 and Table 6.1]. In theory, proper parameterization is fundamental for defining the protocol's security properties in the computationally bounded adversary model, and in practice, proper parameterization provides the necessary flexibility for adjusting the system's actual security to the desired strength. Given its central role in the security model, we recommend making the cryptographic parameters as clear and visible as possible to everyone.

For building an even more solid basis for an actual CHVote implementation, explicit values are specified for all cryptographic parameters. We introduced four different security levels [9, Section 11]. Level 0, which provides only 16 bits of security, has been included for testing purposes. Corresponding mathematical groups are large enough for hosting small elections, but small enough to avoid expensive computations during the tests. Providing a particular security level for testing turned out to be very useful for the software development process. Levels 1, 2, and 3 correspond to current NIST key length recommendations for 80 bits (legacy), 112 bits, and 128 bits of security, respectively [2]. All group parameters are determined deterministically, for example by deriving them from the binary representation of Euler's number. Applying such deterministic procedures demonstrates that the parameters are free from hidden backdoors.

Item 5: Parties and Communication

Parties participating in a cryptographic protocol are usually regarded as atomic entities with distinct, responsibilities, abilities, goals, and attributed tasks. In the design of the protocol, it is important for the parties and their communication abilities to match reality as closely as possible. In CHVote, we decided to consider the voters and their voting devices as two separate types of parties with very different abilities. This distinction turned out to be useful for multiple purposes. First, it enables a more accurate adversary model, because attacks against humans and machines are very different in nature. Second, by including the tasks of the human voters in the abstract protocol description, it provides an accurate model for simulating human voters in a testing environment.

If a voting protocol depends on fully trusted parties, particular care must be applied in the design of their responsibilities and tasks. The *election administrator* and the *printing authority* fall into this category in CHVote. In both cases, we placed great emphasis on limiting their responsibilities to their main role in the protocol. The printing authority, for example, only applies a deterministic algorithm to assemble the inputs from multiple election authorities. The resulting voting cards, which are then printed and sent to the voters, are the only output of this procedure. The procedure itself can be executed in a controlled offline environment. After terminating this task, the printing authority is no longer involved in the protocol, i.e., all its resources can be freed for other tasks. In the aftermath of an election, the voting cards of all participating voters can be reconstructed from the publicly available information. In this way, possible frauds or failures by a corrupt printing authority can be detected. It also means that the printing authority does not need to protect any long-term secrecy.

The definition of the parties in the abstract protocol model includes a description of their communication abilities. Properties of corresponding communication channels need to be specified, again in close accordance with a possible real-world setting. In CHVote, several authenticated and one confidential communication channel are needed to meet to protocol's security requirements [9, Figure 6.1]. This implies the existence of a public-key infrastructure (PKI), which needs to be precisely specified as part of the communication model. To minimize the size of

the PKI and the resulting key management overhead, we recommend keeping the number of participating parties (except the voters) as small as possible. Ideally, this PKI can be mapped one-to-one into an implementation of the system.

Item 6: Protocol Structure and Communication Diagrams

A precise and comprehensive description of the voting protocol is the most fundamental system design output. To cope with the overall complexity, we divided the CHVote protocol into three phases and a total of ten sub-phases. We drew protocol diagrams for each of these sub-phases. A portion of one of these diagrams is shown in Figure 1. Each diagram shows the involved parties, the relevant elements of the acquired knowledge, the messages exchanged between the parties, and all conducted computations. The description of the computations involves calls to algorithms, which are given in a separate section (see Item 11). To optimally connect these diagrams with the remaining parts of the document, we strictly applied our consistent set of mathematical notations and symbols (see Item 3). Keeping these diagrams up-to-date and ensuring their correctness and completeness was a constant challenge during the protocol design. Given their fundamental role in the whole system design, we recommend spending sufficient effort to achieve the best possible result. We see the communication diagrams of the protocol as the core of the system's master plan, which does not permit any lack of clarity or unanswered questions.

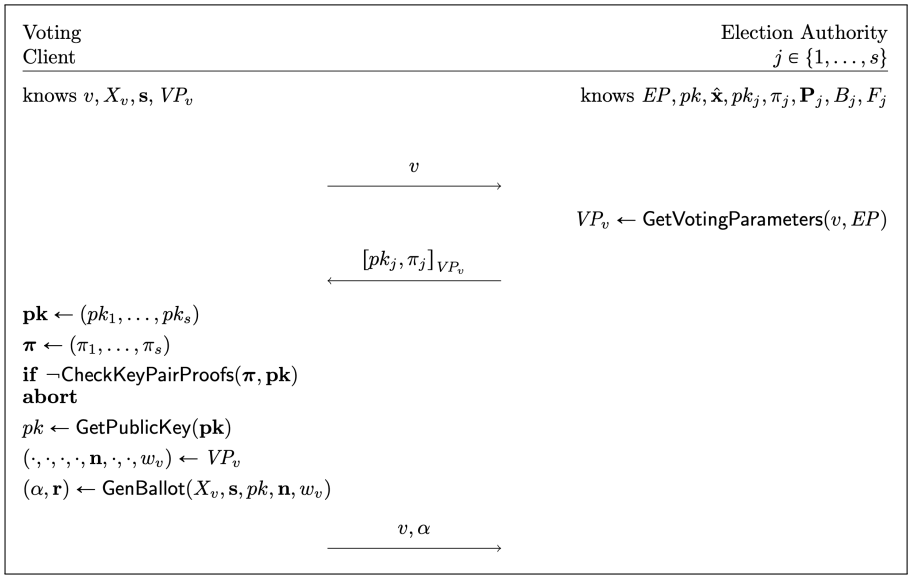


Fig. 1: Exemplary communication diagram: vote casting sub-phase (first part).

Item 7: Pseudo-Code Algorithms

To push the given amount of technical details to the limit, we decided in an early stage of the CHVote project to provide a full set of pseudo-code algorithms for

every computational task in the protocol [9, Section 8]. The current version of the protocol consists of a total of 79 algorithms and sub-algorithms for very different purposes, including primitives for converting basic data types, for computing hash values of complex mathematical objects, or for generating digital signatures. A large portion of the algorithms deals with the core of the CHVote protocol, which realizes a method for transferring verification codes obliviously to the voters in a distributed manner [8]. Other algorithms describe the verifiable mix-net and the distributed decryption process [10, 12]. By maintaining the consistent set of mathematical symbols and notation, this section of the specification document is smoothly integrated into the big picture of the cryptographic protocol. A tremendous amount of initial work, re-factoring, and housekeeping was necessary to reach the stability of the current set of algorithms. Like in regular code, we applied certain pseudo-code style guides to achieve a maximally consistent result. In Figure 2, the algorithm for generating a ballot is given as an example.

Algorithm: $\mathsf{GenBallot}(X, \mathbf{s}, pk, \mathbf{n}, w)$
Input: Voting code $X \in A_X^{\ell_X}$
Selection $\mathbf{s} = (s_1, \ldots, s_k)$, $1 \leqslant s_1 < \cdots < s_k \leqslant n$
Encryption key $pk \in \mathbb{G}_q$
Number of candidates $\mathbf{n} = (n_1, \ldots, n_t)$, $n_j \in \mathbb{N}^+$, $n = \sum_{j=1}^{t} n_j$
Counting circle $w \in \mathbb{N}^+$
$x \leftarrow \mathsf{ToInteger}(X, A_x)$ // see Alg. 4.8
$\hat{x} \leftarrow \hat{g}^x \bmod \hat{p}$
$\mathbf{p} \leftarrow \mathsf{GetPrimes}(n + w)$ // $\mathbf{p} = (p_0, \ldots, p_{n+w})$, see Alg. 8.1
$\mathbf{m} \leftarrow \mathsf{GetEncodedSelections}(\mathbf{s}, \mathbf{p})$ // $\mathbf{m} = (m_1, \ldots, m_k)$, see Alg. 8.24
$m \leftarrow \prod_{j=1}^{k} m_j$
if $p_{n+w} \cdot m \geqslant p$ **then**
  return $\perp$ // $\mathbf{s}$, $\mathbf{n}$, and w are incompatible with p
$(\mathbf{a}, \mathbf{r}) \leftarrow \mathsf{GenQuery}(\mathbf{m}, pk)$ // $\mathbf{a} = (a_1, \ldots, a_k)$, $\mathbf{r} = (r_1, \ldots, r_k)$, see Alg. 8.25
$r \leftarrow \sum_{j=1}^{k} r_j \bmod q$
$\pi \leftarrow \mathsf{GenBallotProof}(x, m, r, \hat{x}, \mathbf{a}, pk)$ // see Alg. 8.26
$\alpha \leftarrow (\hat{x}, \mathbf{a}, \pi)$
return $(\alpha, \mathbf{r})$ // $\alpha \in \mathbb{G}_{\hat{q}} \times (\mathbb{G}_q^2)^k \times (\mathbb{Z}_{2^\tau} \times (\mathbb{Z}_{\hat{q}} \times \mathbb{G}_q \times \mathbb{Z}_q))$, $\mathbf{r} \in \mathbb{Z}_q^k$

Fig. 2: Exemplary pseudo-code algorithm: ballot generation.

To the best of our knowledge, enhancing the specification document of an e-voting system with a complete set of pseudo-code algorithms was a novelty in 2017—and still is today. Our experience with this approach is very positive in almost every respect. First, it added an additional layer to the protocol design, which created an entirely new perspective. Viewing the protocol from this perspective allowed us to recognize certain problems in the protocol design at an early stage. Without detecting them by challenging the protocol from the pseudo-code perspective, they would have come up later during code development.

Another positive effect of releasing pseudo-code algorithms in an early version of the specification document was the possibility of giving third parties the oppor-

tunity to inspect, analyze, or even implement the algorithms (see Item 15). Within a few months, we received feedback from two different implementation projects in different programming languages—from the CHVote developers in Geneva and from students of ours [13, 14]. This feedback was useful for further improving the quality of the specification document, but more importantly, it demonstrated that we managed to considerably reduce the complexity of developing the core tasks of the protocol in a suitable programming language. Our students, for example, who had only little experience in developing cryptographic applications, managed to fully implement all protocol algorithms from scratch in less than four months time. The resulting code from these projects also demonstrated how to almost entirely eliminate the error-prone gap between code and specification. This gap is a typical problem in comparable projects, especially when it comes to check the correctness of the code by external auditors. Without such a gap, auditors can enforce the focus of their inspection to software-development issues. In the light of these remarks, we learned in this project that providing pseudo-code algorithms defines an ideal interface between cryptographers and software developers.

Item 8: Usability and Performance

During the design of the CHVote protocol, we realized that parts of the overall complexity can be left unspecified without affecting the protocol's security properties. We separated some issues that only affect the usability or the performance of the system from the core protocol and discussed them in separate sections.[5] The general idea is to identify aspects that *can* be implemented in a real system or in a certain way, but with no obligation to do so. The benefit of separating them from the core protocol is a higher degree of decoupling in the specification document, which permits discussing corresponding aspects independently of each other. An example of such an aspect is the strict usage of unspecified alphabets for all the codes delivered or displayed to the voters [9, Section 11.1]. Since the actual choice of the alphabets only affects usability (not security), it is something that can be discussed from a pure usability perspective. The situation is similar for various performance improvements, which are optional for an actual implementation. By studying them in a more general context and by publishing the results, our work generated valuable side-products [10, 11].

3 Implementation

Item 9: Mathematical Library

The languages of mathematicians and computer scientists are fairly similar in many respects, but there are also some fundamental differences. One such difference comes from the stateless nature of most mathematical objects, which is very different from mutable data structures in imperative or object-oriented programming languages such as Java. Other differences stem from established

[5] The performance section of the specification document is currently under construction. It will be included in one of the next releases.

conventions. One example of such a convention is the index notation for referring to the elements of a list, vector, or matrix, which usually starts from from 1 in mathematics and from 0 in programming. If a complex cryptographic protocol needs to be translated into programming code, this difference makes the translation process error-prone.

To minimize in our CHVote implementation the difference between specification and code, we introduced a Java library for some additional immutable mathematical objects. The core classes of this library are `Vector`, `Matrix`, `Set`, `ByteArray`, `Alphabet`, and `Tuple` (with sub-classes `Pair`, `Triple`, ...). All of them are strictly generic and immutable. Applying generics in a systematic way greatly improves type-safety, for example in case of complex nested types such as

```
Triple<BigInteger, Vector<String>, Pair<Integer, ByteArray>>.
```

Working with immutable objects has many advantages. They are easier to design, they can always be reused safely, and testing them is much easier [5, Page 80]. `String` and `BigInteger` are examples of given immutable classes in Java. In our mathematical library, we adopted the convention of accessing the elements of a vector of size n with non-zero indices $i \in \{1, \ldots, n\}$, and similarly for matrices and tuples. This delegates the translation between different indexing conventions to theses classes and therefore eliminates the error-proneness of this process. It also creates a one-to-one correspondence between indexing variables in the specification and the code, which is beneficial for the overall code readability.

In our experience of implementing the CHVote protocol, the mathematical library turned out to be a key component for achieving the desired level of code quality in a reasonable amount of time. Given its central role in all parts of the system, we put a lot of effort into performance optimizations, rigorous testing, and documentation. We highly recommend the creation and inclusion of such a library in similar projects.

Item 10: Naming Conventions

Most programming languages have a well-established set of naming conventions. Generally, software developers are advised to "*rarely violate them and never without a very good reason*" [5, Page 289]. Not adhering to the conventions usually lowers the code readability and makes code maintenance unnecessarily complicated, especially if multiple developers are involved. In some situations, deviations from common conventions may even lead to false assumptions and programming errors. In Java, the naming convention for variables, fields, and method parameters is to use a connected sequence of words, with the first letter of each subsequent word capitalized (a.k.a. "camel case"), for example `maxVoterIndex`. Abbreviations such as `max` or single letters such as `i` are allowed, as long as their meaning in the given context remains clear.

In our implementation of the cryptographic protocol, we decided to deviate from general Java naming conventions. To achieve our goal of diminishing the gap between specification and code to the maximal possible degree, we decided to adopt the mathematical symbols from the protocol specification as precisely as possible in the code. This includes defining upper-case variable names in Java such

as `Set<Integer> X` for a set X of integers. In such cases, we prioritized project-internal naming consistency over general Java naming conventions. Tagged, boldface, or Greek variable names are spelled out accordingly, for example $\hat{\alpha}_{ij}$ as `alpha_hat_ij` or $\mathbf{k}'$ as `bold_k_prime`. We strictly applied this pattern throughout all parts of the code. Code that is written in this way may look quite unconventional at first sight, but it turned out to be a key element for making the Java code look almost exactly the same as the pseudo-code. As an example, consider our implementation of the algorithm GenBallot in Figure 3, which closely matches with the pseudo-code from Figure 2.

```java
public class GenBallot extends ch.chvote.algorithms.common.GenBallot {

    public static Pair<Ballot, Vector<BigInteger>>
        run(String X, IntVector bold_s, QuadraticResidue pk, IntVector bold_n, int w, Parameters params) {

        // PREPARATION
        int n = Math.intSum(bold_n);
        Precondition.checkNotNull(X, bold_s, pk, bold_n, params);
        Precondition.check(params.GG_q.contains(pk));
        Precondition.check(IntSet.NN_plus.contains(w));
        Precondition.check(Set.String(params.A_X, params.ell_X).contains(X));
        Precondition.check(Set.IntVector(IntSet.NN_plus).contains(bold_n));
        Precondition.check(Set.IntVector(IntSet.NN_plus(n)).contains(bold_s));
        Precondition.check(bold_s.isSorted());

        // ALGORITHM
        var x = ToInteger.run(X, params.A_X);
        var x_hat = Mod.pow(params.g_hat, x, params.p_hat);
        var bold_p = GetPrimes.run(n + w, params);
        var bold_m = GetEncodedSelections.run(bold_s, bold_p);
        var m = Math.prod(bold_m.map(QuadraticResidue::getValue));
        if (bold_p.getValue(n + w).getValue().multiply(m).compareTo(params.p) >= 0) {
            throw new AlgorithmException(GenBallot.class, AlgorithmException.Type.INCOMPATIBLE_MATRIX);
        }
        var pair = GenQuery.run(bold_m, pk, params);
        var bold_a = pair.getFirst();
        var bold_r = pair.getSecond();
        var r = Mod.sum(bold_r, params.q);
        var pi = GenBallotProof.run(x, Mod.prod(bold_m), r, x_hat, bold_a, pk, params);
        var alpha = new Ballot(x_hat, bold_a, pi);
        return new Pair<>(alpha, bold_r);
    }
```

Fig. 3: Exemplary Java code: ballot generation.

Item 11: Implementation of Pseudo-Code Algorithms

We already discussed our view of the pseudo-code algorithms as an ideal interface between cryptographers specifying the protocol and software developers implementing corresponding code (see Item 7). In such a setting, the implementation of the algorithms inherently defines an important bottom layer of the whole system architecture. To strengthen the overall clarity in our implementation of the algorithms, we decided the create separate utility class for all top-level algorithms. Each of them contains exactly one static method `run(<args>)`, which implements the algorithm (plus static nested classes for all sub-algorithms), for example `GenBallot.run(<args>)` for the algorithm GenBallot. This way of structuring

the algorithm module establishes direct links to the specification document. These links are clearly visible by inspecting the project's package structure. A section of this package structure is shown in Figure 4.

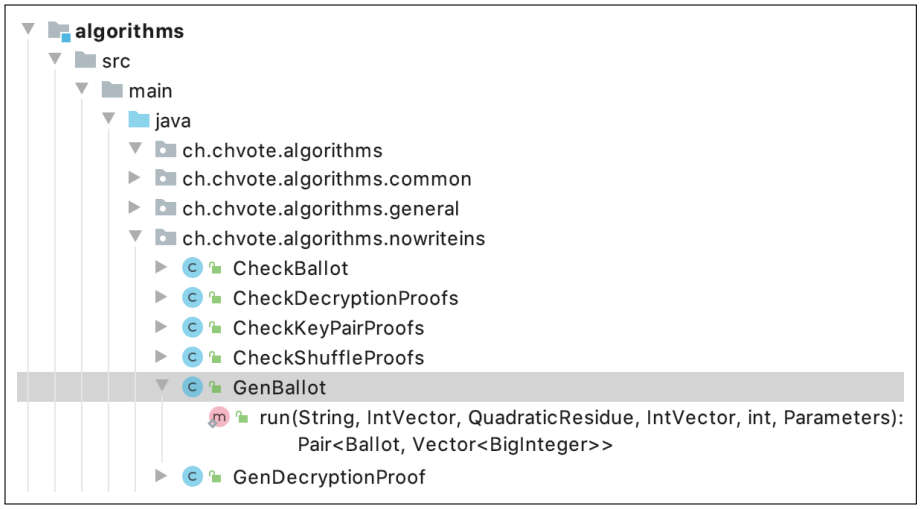


Fig. 4: Package structure of static utility classes for top-level algorithms.

Given the central role of the protocol algorithms for the whole system, we put extra care and effort into developing this part of the code. To obtain the best possible code consistency, we defined a set of project-internal coding style guidelines and applied them strictly to all algorithms. Each algorithm went through an internal reviewing and testing process over multiple rounds, which involved different persons according to the *Four Eyes Principle*. The result is a consistent set of Java methods that are perfectly aligned with the pseudo-code algorithms from the specification. The example shown in Figures 2 and 3 demonstrates how precisely the algorithms have been translated into code.

We see perfect alignment between specification and code as a quality criterion of highest priority. This implies that even the smallest change in either the specification or the code needs to be updated immediately on the other side. The general idea here is to view them as *the same thing*. This view enables third-party auditors that are familiar with the naming conventions and coding style guidelines to check the translation from specification to programming code at minimal costs. We believe that auditing the implementation of the algorithms remains a diligent (but mostly routine) piece of work, which does not necessarily require the involvement of cryptographic experts.

Item 12: Parameter Validity Checks

An important aspect of the proposed way of implementing the protocol algorithms is the introduction of systematic validity checks of all input parameters. These checks complement the built-in type safety obtained from strictly using the generic mathematical library (see Item 9). The domains of all input parameters

are specified in the pseudo-code algorithms, for example $X \in A_X^{\ell_X}$ in GenBallot for a string of characters from the alphabet A_X of length ℓ_X, which translates into the following line of Java code (see Figure 3, Line 36):

```
Set.Strings(params.A_X, params.ell_X).contains(X)
```

Provided that these checks are sufficiently strong for detecting all possibilities of invalid parameters—or invalid combinations of parameters—of a given algorithm, they ensure that the algorithm always outputs a meaningful result. In case of a failed check, it is clear that something else must have gone wrong, for example that a message with a corrupt content has been received or that some stored data has been modified. Every failed check therefore indicates some deviation from a normal protocol run. This is the reason for implementing them in a systematic way for all top-level algorithms (sub-algorithms do not require such checks).

To minimize the overhead of performing these checks each time an algorithm is called, we managed to entirely eliminate expensive computations such as modular exponentiations. To efficiently perform membership tests $x \in \mathbb{G}_q$ for the set $\mathbb{G}_q \subset \mathbb{Z}_p^*$ of quadratic residue modulo a safe prime $p = 2q + 1$, we implemented the *membership witness* method proposed in [10]. The corresponding class `QuadraticResidue`, which realizes this test with a single modular multiplication, is part of our mathematical library. In Figure 3, the parameter `pk` is of that type, and its membership test is conducted in Line 38.

Item 13: Implementation of Protocol Parties

To implement the protocol based on the algorithms, we designed a software component for every involved party. These components share some code for various common tasks, but otherwise they are largely independent. For the design of each party, we derived a state diagram from the protocol description in the specification document. This diagram defines the party's behavior during a protocol run. Typically, receiving a message of certain type triggers the party to perform a transition into the next state. The transition itself consist of computations and messages to be sent to other parties. The computations, which we call *tasks*, can be implemented by calling corresponding protocol algorithms.

The left-hand side of Figure 5 shows the UML state diagram of the printing authority (printer), which consists of two states SP1 and SP2 and one error state EP1. In SP1, the printer expects messages of type MAP1 and MEP1. If all messages are received, the transition into SP2 (or EP1) is triggered. This involves computing task TP1 and sending two types of messages MPV1 and MAX1. The error state EP1 is reached in case of an exception of type AE (algorithm exception) or TE (task exception). This diagram represents the printer's view of the printing sub-phase [9, Protocol 7.2], which is the only sub-phase in which the printer is active. Similar state diagrams exist for all other parties and sub-phases. We defined further naming conventions and strictly applied them to all tasks and message types.

Modeling the parties using the (extended) state machine formalism turned out to be the ideal approach for structuring the parties' implementations in the most natural way. It also allowed us to apply the *state pattern*, one of the well-known

"Gang of Four" design patterns [7, Page 305]. This made our implementation very transparent from a general software-engineering perspective. The right-hand side of Figure 5 shows a section of the package structure, which illustrates for example that the party class `Printer` depends on three state classes `SP1`, `SP2`, and `EP1`, and one task class `TE1`. Every other party is implemented in exactly this way. Every task and every message type is connected to one of the sub-phase diagrams in the protocol specification, and vice versa.

Using the state pattern, we achieve close correspondence between specification and code also on the abstraction layer representing the parties. Again, we see the code and the specification related to the parties as essentially *the same things*, which means that the slightest change on one side needs to be updated immediately on the other side. In this way, we tried to achieve a similar level of structural clarity and code quality as for the algorithm implementation. The state pattern was also useful for establishing the flexibility of running multiple election events simultaneously (possibly using different protocol versions).

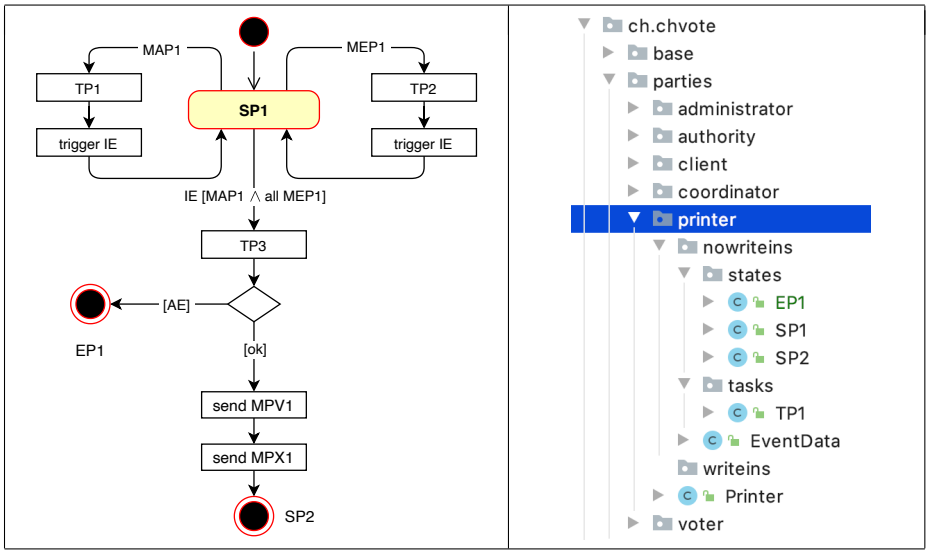


Fig. 5: State diagram of the printer (left) vs. package structure of party classes (right).

Item 14: Cryptographically Relevant Code

Providing code for all algorithms and all parties concludes the implementation of the cryptographically relevant part of the protocol. This is where flaws in the code can cause critical errors or vulnerabilities. Generally, we recommend structuring the software design into *cryptographically relevant* and *cryptographically irrelevant* components and to link them over suitable interfaces. Our current implementation of the CHVote protocol is limited to the cryptographically relevant part of the system, but we provide the required interfaces, for example for connecting our code to concrete high-performance messaging and persistence services.

For testing purposes, we only implemented these interfaces in a rudimentary way, but this turned out to be sufficient for simulating even the most complex election use case from top to bottom. Such a simulation can be conducted on a single machine using any common development environment, i.e., no complex installation of a distributed test environment over multiple servers is required. This is an efficient environment for running all sorts of functional tests with a clear focus on the cryptographic protocol. With almost no communication overhead, it is also ideal for analyzing and optimizing the overall protocol performance. A precondition for establishing a complete test run is the implementation of all protocol parties, including the (human) voters. Even if corresponding code will obviously not be included in a real-world deployment of the system, we see it as an indispensable component of our implementation.

Given its central role in the overall security of the system, we tried to make the cryptographically relevant part of the code accessible to the broadest possible audience. For that, we decided to avoid dependencies to complex third-party libraries or software frameworks as far as possible. We only admitted two dependencies to the widely used native GMP library for efficient computations with large numbers and to the Spock framework for enabling data-driven tests. Both libraries are almost entirely invisible in our implementation, i.e., there is no need to familiarize reviewers with these technologies (except for reviewing the tests). Generally, we see complex frameworks based on annotation, reflection, or injection mechanisms as unsuited for developing cryptographically relevant code. They are great for implementing enterprise software components at minimal costs, but they often tend to obscure the general program flow. This reduces the overall code readability and makes static code analysis more difficult.

4 Project Management

Item 15: Transparency

We started this project from the beginning with the mindset of maximal transparency. At an early stage of the project in 2017, we published the first version of our specification document [9]. At that time, we had already published a peer-reviewed paper describing the cryptographic core of the protocol [8]. The feedback that we received, mostly from members of the e-voting community, was very useful for improving the protocol and its security properties. The most important feedback came from Tomasz Truderung on April 19, 2017, who found a subtle but serious flaw in the construction of our protocol. This flaw had been overlooked by the reviewers of the published paper. After a few weeks, we were able fix the problem to a full extent and update the protocol accordingly. In the meantime, the success of the entire project was at stake.

We recall this anecdote here for making two important points. First, releasing specification documents of an e-voting project usually launches a public examination process in the community. The outcome of this process is sometimes unpredictable, but the received feedback has the potential of greatly improving the quality of the protocol. At the time of writing this document, we have not yet

released the source code for public examination, but we expect a similar amount of interest and feedback from the community. Second, a cryptographic protocol without formal security definitions and rigorous proofs provides not a sufficiently solid foundations for building a system. In CHVote, a different group of academics was contracted by the State of Geneva to perform this task. The outcome of this sister project was released in 2018 [3]. The high quality of their work leads one to suppose that the above-mentioned flaw would have been detected in their analysis. Unfortunately, their report has not yet been updated to the current version of the protocol.

In this project, our mindset of maximal transparency always allowed us to openly discuss all aspects of our work with many different people, including students of ours who developed various prototypes [13, 14]. This created a permanent challenge for the cryptographic protocol, which forced us to constantly question our design decisions and improve our technical solutions. We conclude that releasing all cryptographically relevant documents as a matter of principle was fundamental for the success of the project. More generally, we see it as an important trust-establishing measure.

Item 16: Verifier

The last point we want to mention in this paper is an important aspect for a verifiable e-voting system. Unfortunately, we were not yet able to cover it in this project. It's about specifying the verification software—sometimes called *the verifier*—for the proposed protocol. In the original project setting of the State of Geneva, it was planned to outsource the specification and development of the verifier to a third-party institution. To establish a certain degree of independence between the protocol and the verifier, this decision of the project owners was perfectly understandable. We never questioned this decision, but it prevented us from paying enough attention to this important topic. When the project was dropped in fall 2018, the outsourced verifier project had started, but it was not yet very advanced. This finally led to the current situation, where the specification and the implementation of the e-voting protocol are both very advanced, but almost nothing is available for the verifier. Even though, the e-voting protocol describes how to verify certain cryptographic aspects, but that is not to be confused with the complete verification of the whole voting process.

We believe that in projects like this, it's best to let the specification of the protocol and the verifier go hand in hand, and to apply the same level of preciseness and completeness to both of them. We see the verifier as the ultimate way of challenging the protocol run, both in the abstract setting of the specification document and in the concrete setting of executing the code on real machines. So far, this challenge is missing in our project.

5 Conclusion

In software development, best practices are available in many areas. They are very useful for developers to avoid bad design decisions and typical programming

mistakes. This certainly also holds for developing an e-voting system, but the delicacy of implementing a cryptographic protocol makes the situation a bit more complicated. We therefore believe that the e-voting community should come up with its own set of best practices and define respective minimal standards. This paper makes a first step into this directions based on our experience from the CHVote project. Among the discussed sixteen topics, we believe that the advice of providing all algorithmic details in pseudo-code is the most important one, together with structuring the source code into a cryptographically relevant and a cryptographically irrelevant part.

References

1. *Verordnung der Bundeskanzlei über die elektronische Stimmabgabe (VEleS) vom 13. Dezember 2013 (Stand 1. Juli 2018).* Die Schweizerische Bundeskanzlei (BK), 2018.
2. E. Barker. Recommendation for key management. NIST Special Publication 800-57, Part 1, Rev. 5, NIST, 2020.
3. D. Bernhard, V. Cortier, P. Gaudry, M. Turuani, B. Warinschi. Verifiability analysis of CHVote. *IACR Cryptology ePrint Archive*, 2018/1052, 2018.
4. D. Bernhard, O. Pereira, B. Warinschi. How not to prove yourself: Pitfalls of the Fiat-Shamir heuristic and applications to Helios. *ASIACRYPT'12, 18th International Conference on the Theory and Application of Cryptology and Information Security*, LNCS 7658, pages 626–643, Beijing, China, 2012.
5. J. Bloch. *Effective Java.* Addison-Wesley, 3rd edition, 2018.
6. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. *CRYPTO'86, 6th International Cryptology Conference on Advances in Cryptology*, LNCS 263, pages 186–194, Santa Barbara, USA, 1986.
7. E. Gamma, R. Helm, R. Johnson, J. Vlissides. *Design Patterns – Elements of Reusable Object-Oriented Software.* Addison-Wesley, 1994.
8. R. Haenni, R. E. Koenig, E. Dubuis. Cast-as-intended verification in electronic elections based on oblivious transfer. *E-Vote-ID'16, 1st International Joint Conference on Electronic Voting*, LNCS 10141, pages 277–296, Bregenz, Austria, 2016.
9. R. Haenni, R. E. Koenig, P. Locher, E. Dubuis. CHVote system specification – version 3.0. *IACR Cryptology ePrint Archive*, 2017/325, 2020.
10. R. Haenni and P. Locher. Performance of shuffling: Taking it to the limits. *Voting'20, FC 2020 International Workshops*, Kota Kinabalu, Malaysia, 2020.
11. R. Haenni, P. Locher, N. Gailly. Improving the performance of cryptographic voting protocols. *Voting'19, FC 2019 International Workshops*, LNCS 11599, pages 272–288, Frigate Bay, St. Kitts and Nevis, 2019.
12. R. Haenni, P. Locher, R. E. Koenig, E. Dubuis. Pseudo-code algorithms for verifiable re-encryption mix-nets. *Voting'17, FC 2017 International Workshops*,, LNCS 10323, pages 370–384, Silema, Malta, 2017.
13. K. Häni and Y. Denzer. CHVote prototype in Python. Project report, Bern University of Applied Sciences, Biel, Switzerland, 2017.
14. K. Häni and Y. Denzer. Visualizing Geneva's next generation e-voting system. Bachelor thesis, Bern University of Applied Sciences, Biel, Switzerland, 2018.
15. S. J. Lewis, O. Pereira, V. Teague. How not to prove your election outcome. Technical report, 2019.
16. U. Maurer and C. Casanova. Bericht des Bundesrates zu Vote électronique. 3. Bericht, Schweizerischer Bundesrat, 2013.

Human Factors in Coercion Resistant Internet Voting – A Review of Existing Solutions and Open Challenges

Oksana Kulyk[1] and Stephan Neumann[2]

[1] IT University of Copenhagen, Denmark, okku@itu.dk
[2] stephanneumann@tutamail.com

Abstract. While Internet voting has a potential of improving the democratic processes, it introduces new challenges to the security of the election, such as the possibility of voter coercion due to voting in uncontrolled environments. Cryptographic research has resulted in a number of proposals for protecting against such coercion with the help of counter-strategies that can be used by the voter to convince the coercer that they obeyed their instructions while secretly voting for another voting option. So far, these proposals have been theoretical, and their usability in terms of ability of the voter to apply the counter-strategies in practice has not been thoroughly investigated. We conducted a literature review to identify the available counter-strategies and assumptions on voters' capabilities. We evaluated the identified assumptions and conclude a number of usability issues. We provide recommendations on further research directions and practical considerations in designing coercion resistant voting systems are provided.

1 Introduction

With the ongoing digitalization of society, Internet voting has often been discussed as a way to facilitate democratic processes. These discussions are furthermore more prominent in 2020 given the ongoing pandemic, as many argue, making remote voting a necessary option to protect the population. Several countries, e.g. Estonia and Switzerland, introduced Internet voting as an additional voting channel in order to improve convenience for the voters and support voters who would otherwise be unable to get to a polling station. However, introducing technology in electoral processes also introduced new risks, in particular, risks connected with security and privacy. One of these risks is the possibility of voter coercion, stemming from the fact that the voting occurs in an uncontrolled environment where voter privacy and, correspondingly, the secrecy of the ballot is no longer guaranteed. An adversary who is physically present next to the voter while they cast their vote – for example, a household member or a supervisor at work – would be able to ensure that the voter casts the vote they are instructed to cast. Even a remote coercer could instruct the voter to reveal which voting option they voted for, for example, by requesting the voter to prepare and send a recording of the voting procedure.

In order to prevent such attacks, a number of works in the e-voting community focused on developing schemes for the voting systems with the so-called coercion resistance property, see e.g. [1, 5, 28]. A scheme that is coercion resistant aims to protect voters' privacy even if the adversary can actively communicate with the voter and coerce them to reveal secret information or to behave in a certain way. A related concept is receipt-freeness, which specifically focuses on preventing the voter from creating a receipt that would prove to the adversary how they voted. One of the ways the schemes satisfy coercion resistance and/or receipt-freeness is via the so-called counter-strategy. The idea is, that the voter pretends to follow the coercer's instructions, while secretly following a different procedure that allows them to vote for their preferred voting option. The counter-strategy succeeds if the coercer is not able to tell whether it has been applied, or whether the voter has voted as instructed.

While the underlying cryptographic mechanisms of the proposed schemes can guarantee the success of a counter-strategy under the defined security model, it is still crucial to ensure that the voter is capable of performing them correctly. Usability therefore becomes a fundamental issue. While a number of works have investigated usability and other human factors in e-voting (see e.g. [37, 46]), only a few have considered the actions required by the voter to ensure coercion resistance from the usability point of view [42, 43]. These studies have pointed that the counter-strategies proposed by investigated systems rely on complicated concepts not understandable by the voter and on complex actions required from the voter. As these studies focused on the evaluation of a specific voting scheme and its implementation, no systematic investigation on the available counter-strategies from a variety of systems has been done yet. The general practical issues of coercion resistant voting systems are studied by Krips and Willemson [33], however, their work does not focus on human factors of such systems.

This paper describes the results of a conducted literature review to identify the counter-strategies available in voting literature on the topic of coercion resistance. We study the assumptions regarding the voter capabilities in applying these counter-strategies from the human factors point of view. We identify a number of challenges in designing coercion resistant systems and provide recommendations on addressing these challenges and future work directions.

2 Methodology

In order to identify the existing counter-strategies a search using keywords "coercion resistance voting" and "receipt-freeness voting" in SpringerLink, IEEE, ACM and USENIX proceedings databases has been conducted. The search was limited to papers in computer science written in English language that are in open access from the authors' institution. Additionally, a search using the same keywords was performed in Google Scholar. From the search results, the papers that propose an Internet voting scheme satisfying some variant of receipt-freeness and/or coercion resistance were identified.

Note, we do not include proposals for polling-place voting, as these assume a controlled environment. We furthermore do not consider the proposals that rely on security mechanisms other than counter-strategies (e.g. rerandomisation of a vote by a voting system component [13] or relying on a tamper-resistant device that does not reveal the encryption randomness to the voter [8]), since these do not protect against an attacker that is either physically present or demands a recording of the voting procedure from the voter. Furthermore, we exclude the papers that focus on improving one specific part of the procedure towards providing better protection against coercion (e.g. such as the individual verification in the original proposal in Selene [49], or preventing disclosure from published tally results in ShuffleSum [10]) without considering other steps of the election procedure such as actual vote casting.

3 Results

A total of 51 papers were identified, containing the proposals that can be classified into the following categories: fake credentials, deniable vote updating, vote masking and code voting. Table 1 provides an overview of the number of proposals in each category. We explain the counter-strategies in each category in more detail, considering the following coercion scenario. The adversary wants to coerce the voter to cast a vote for Eve, while the voter attempts to cast a vote for Alice instead[3]. In our description, we focus on human factors challenges and assumptions of the counter-strategies, referring to the work by Krips and Willemson [33] for an overview of more technical assumptions or coercion resistant systems.

Counter-strategy	**Papers**	**Total**
Fake credentials	[2–6, 12, 15, 16, 18, 19, 22, 24, 27–29, 32, 41, 43, 47, 48, 52–55, 57, 58, 61, 65, 66]	29
Deniable vote updating	[11, 14, 20, 25, 34, 36, 38, 39, 44, 45, 56]	11
Masking	[7, 17, 26, 30, 31, 50, 59, 62–64]	11

Table 1. Classification of scientific papers into counter strategies against voter coercion.

3.1 Fake credentials

By far the most popular counter-strategy relies on the existence of so-called fake credentials. The idea behind the counter-strategy is as follows. Given a space of available credentials $\mathcal{C}$, the voter is provided with a unique and secret credential

[3] Note, while other possible combinations of goals for both adversary and the voter exist (for example, the voter might want to avoid voting for Eve without necessarily casting a ballot for another candidate, or adversary might want to force the voter to abstain instead of voting for a specific candidate), these are only briefly discussed and are not in the focus of this paper.

$\hat{c} \in \mathcal{C}$ during voter registration, so that the credential is distributed before the election via an untappable channel. When voting, the voter uses the credential $\hat{c}$ to authenticate themselves to the voting system. If the voter is coerced, instead of authenticating themselves with their real credential, they generate and use a so-called fake credential $c' \neq \hat{c}$. The fake credential is indistinguishable from the real one by the adversary, and is accepted by the voting system without outputting an authentication error. The votes submitted with the fake credential are, however, are excluded from tallying. The voter then can cast a valid vote for their preferred candidate when they are not being observed by the adversary. In case the voter only wants to prevent voting for an adversarial candidate, no further actions are required.

While some of the described schemes do not specify how the credentials are stored, providing only a description of the protocol without going into practical implementation aspects (e.g. [28]), others rely on storing various values such as cryptographic secret keys on a tamper-resistant trusted device (see e.g. [43, 47]. The purpose of this device is to ensure, that neither an adversary nor the voter themselves can get access to the information stored on it.

Human factors assumptions The success of the fake credential counter-strategy depends on how secure these credentials are managed. This results in the assumptions on voters' behaviour as described below.

Inputting real credentials. The first assumption is crucial, first and foremost, for the case when no coercion occurs and the voter simply wants to cast a vote for their preferred candidate. In that case, they have to enter their real credential into the system. They, however, would not be provided with any feedback from the system, whether the credential is actually correct – after all, a potential coercer who observes the voting would otherwise be able to tell whether the voter obeys the adversary's instructions or applies a counter-strategy. This assumption is especially crucial in systems where any credential $c \in \mathcal{C}$ is admissible by the system and treated as fake as long as $c' \neq \hat{c}$ – in such a case, any typo or other mistake in entering the credential will result in casting an invalid vote, without the voter knowing it.

One approach to facilitate this assumption relies on the so-called panic passwords [15]. The idea is to use a separate type of credential that would allow the voter to signal being coerced. Thus, each voter is assigned a set $I \subset \mathcal{C}$ of admissible credentials, of which $\hat{c} \in I$ is the only real credential that allows casting a valid ballot. Whenever the voter authenticates themselves using any value $c \in \mathcal{C}/I$, the system outputs an authentication error. If the voter uses $c' \in I$, $c' \neq \hat{c}$, the system treats c' as a fake credential and the voter as coerced, and accepts c' without outputting any error. Using such an approach it is crucial to define I in such a way that makes it unlikely that the voter mistakenly enters another credential $c' \in I$ instead of c. The authors of [15] propose to define I as any passphrase that consists of a given number of dictionary words. Such a system is likely to protect against typos (especially if one excludes dictionary

words that differ from each other by a single letter, and hence, prone to being mixed up due to typos). However, it will not protect voters who do not remember the passphrase exactly, for example, not being sure about the order of the words in a passphrase. One can furthermore argue that panic passwords introduce further usability issues: as such, it is crucial to ensure that the voters understand the concept of panic passwords, namely, that out of many admissible passwords available to them, only one can be used for casting a valid ballot. Finally, if the voters are expected to generate their passwords themselves, it should be taken into account that humans often find it difficult to come up with passwords that are secure enough.

Generating fake credentials. A related assumption is required to ensure that the coerced voters are capable of applying a counter-strategy without alerting the adversary. This assumption might be easier to fulfill if the system accepts any credential $c' \in \mathcal{C}$ as a fake credential without outputting a warning. Still, the voters need to be explained how and when they should do it. As with panic passwords, generating a convincing fake credential would also get more complicated if the voters are required to understand the rules of how panic passwords are constructed and generate one accordingly.

3.2 Deniable vote updating

Another method of resisting coercion is the so-called deniable vote updating. The idea is simple: while the voter might be coerced to cast a particular vote in presence of an adversary, they can cast another vote, overwriting their previous one, when the adversary is gone. This method, in particular, is relied upon in real-world Internet voting in Estonia and was deployed in the Norwegian Internet voting system between 2011 and 2013. The coercion resistance property, in particular, is achieved due to deniability of vote updating – the adversary should be unable to tell whether the voter has cast another vote, even if the voter would try to prove that they did not do it. This deniability is achieved either via restricting access to the election information, or via cryptographic solutions that enable deniability while also publishing the cast ballots for verifiability. As opposed to fake credentials-based systems, where the voting credentials are generated and distributed as a part of the voting system and specifically designed to be coercion resistant, systems based on deniable vote updating assume that an existing infrastructure is used for authenticating the voters. Such an infrastructure can be implemented via tamper-resistant trusted hardware tokens, such as smart cards in Estonia. Forwarding those types of authentication material could have severe impact to voters beyond the voting process, which lowers the risk of forwarding voting materials.

A variant of deniable vote updating is a so-called flexible vote updating. As opposed to simple vote updating that follows the last-vote-counts policy, the final ballot that is included in the tally is calculated as a function of all the ballots cast by the voter in the election, expressed by a function $F(v_1, ..., v_n)$. One example of such function is the proposal in [11,36], which sets $F(v_1, ..., v_k) = \sum_{i=1}^{k} v_i$. In this

way, the system ensures protection against last-minute attacks that might occur if the adversary demands that the voter casts their vote during the very last minute of the voting phase of the election, either observing the voter while they do so (including remote observation or recordings of the voting procedure provided by the voter), or checking the public election information for the ballots posted by the voter. In that case, if the voter is coerced to cast a vote for v_{Eve} using the system with flexible vote updating, they cast a ballot for $v' = v_{Alice} - v_{Eve}$ beforehand, so that their final ballot is computed as $v' + v_{Eve} = v_{Alice}$.

Human factors assumptions An advantage of the deniable vote updating strategy is its initial simplicity: if the voter is not coerced, the vote casting process is no different from simpler voting systems that do not ensure coercion resistance. Even in case of coercion, the concept of voting again in order to overwrite the vote cast under coercion would most probably fit into the mental models of the voters. The simple vote updating strategy therefore only relies on one assumption:

Make sure to vote after (or before) coercion. As opposed to fake credentials approach, the deniable vote updating strategy requires the voter to take additional action in order to make sure that the adversarial vote will not be counted. Thus, in addition to ensuring that the voter has such a possibility by being free from adversarial observation, the voter should also keep in mind that they need to go through the voting process again at some point. More complexity, however, is introduced if the flexible vote updating is used. Namely, the following assumptions becomes of crucial importance:

Remember all the votes cast in the election. At the moment of casting their vote, the voter should keep track of all the votes cast in the election, including votes that they might be coerced to cast in the future.

Calculate values to cast. The voter should be able to calculate the value they should cast in order to get their preferred vote to be counted; that is, given $v_1, ..., v_{k-1}$ as the votes cast in the election, the voter should be able to calculate v_k so that $F(v_1, ..., v_k) = v_{Alice}$.

Input v_k. Once the value v_k is calculated, the voter has to input it without making any errors.

Similar to the fake credentials strategy, the system would not be able to output all the previously cast votes on voter's request or provide any feedback on the resulting value $F(v_1, ..., v_k)$ upon casting v_k without violating coercion resistance. Note, that the consequences in making a mistake in inputting v_k are even more severe than in the fake credential counter-strategy when voting in absence of coercion. While failing to input a correct credential can only in casting an invalid ballot that will not be counted, choosing a wrong value v_k can in worst case result in a final ballot $v = F(v_1, ..., v_k)$ that will be counted as a valid vote for one of the candidates in the election other than Alice.

Note also, that much of this complexity can be hidden behind the user interface of the voting client; as such, the system with flexible vote updating can be modified into the system with simple vote updating, if the voting client stores all the votes $v_1, ..., v_{k-1}$ cast so far, and casts a value v_k so that $F(v_1, ..., v_k) = v_{Alice}$ if the voter inputs "Alice" as their choice in the user interface. Such a modification, however, will make the system vulnerable to last-minute coercion. A possible solution would be to let the voter choose between simple and flexible vote updating in an election, by offering to download two different voting clients; this, however, would require further computer literacy from the voter, as well as the ability to understand the difference between the offered choices.

3.3 Masking

As opposed to fake credentials and deniable vote updating counter-strategies that are aimed at nullifying the vote cast in presence of an adversary (with a possibility to change it to a vote for the voter's preferred candidate), masking enables the voter to cast their preferred vote for Alice while letting the adversary think that the same cast vote is a vote for Eve. The idea is, that before the election, the system commits to a secret masking value $b \in \mathcal{B}$ and shares it with the voter. When casting the vote, the voter utilises a function $M : \mathcal{B} \times \mathcal{V} \to \mathcal{V}$ to submit a masked ballot $v_M = M(b, v_{Alice})$, from which the value $v_{Alice} = M^{-1}(v_m, b)$ will be extracted by the voting system. A voter who is coerced would, correspondingly, cast the same masked ballot v_M and provide the coercer with a fake masking value b' selected such as $v_M = M(b', v_{Eve})$. Different variants of masking strategy have been proposed, such as using $\mathbb{Z}_n$ as a set of possible votes v_m and using a one time pad $b \in \mathbb{Z}_n$ with $M(v, b) := v + b$, using permutation π of candidate list $v_1, ...v_L$, with $b = (\pi(1), ..., \pi(L))$ and $M(v_i, b) := \pi(i)$ or using a code list $b = x_1, ..., x_L$ with a unique code assigned to each one of the candidates $v_1, ..., v_L$ and $M(v_i, b) := x_i$ (the so-called code voting).

Human factor assumptions The main assumption crucial for the masking counter-strategy is the voter being able to calculate the value v_m that results in a vote for an intended candidate (i.e. so that $M^{-1}(v_m, b) = v_{Alice}$). This results in the following assumptions:

Recalling b. While the voter does not have to manually input the masking value during vote casting, they are expected to recall it correctly in order to perform the calculation of $M(v_{Alice}, b)$.

Calculating $M(v_{Alice}, b)$. Even if the voter remembers b, they are still expected to calculate the masked ballot that corresponds to their intended vote b.

Input v_m. Finally, once the value $v_m = M(v_{Alice}, b)$ is calculated, the voter has to input it without making any errors.

Similar to the fake credential counter-strategy, the voting system would not output any feedback regarding $M^{-1}(v_m, b)$ for a cast v_m. Similar to the deniable

vote updating strategy, failing to cast a correct masked ballot, due to mistakes either in recalling b or in calculating or inputting v_m can in worst case result in a vote being cast for another candidate that will be counted in the tally.

As one way to mitigate this assumption, the scheme in [7] proposes to use a mobile app that receives and outputs the value b from the voting system as the voter starts the voting process. As discussed above, such an approach requires a trusted mobile device and does not protect against a physically present coercer. Another solution is the code voting approach that uses paper code sheets containing printed codes for each candidate, e.g. with x_{Alice} and x_{Eve} corresponding to votes for Alice and Eve respectively. The idea is that the voter reads the code of their chosen candidate during vote casting, without having to recall it from memory. Similar to the app approach, the voter would be vulnerable against physically present adversary. However, assuming that the voter can print fake code sheets by themselves and expects the coercer to force them to vote for Eve, they could switch the codes on the fake sheets, setting $x'_{Alice} = x_{Eve}$ and $x'_{Eve} = x_{Alice}$. Yet another way to ensure that the cast masked ballot is the same vote that the voter intended to cast is the use of so-called return codes. The idea is to assign a code $r_1, ..., r_L$ to each candidate, and provide the return code sheets with codes printed on paper to the voter. After receiving a ballot with a vote for a candidate v_i, the voting system outputs a code r_i to the voter, which they should compare to the code on their return code sheet. While the use of return codes is commonly used to protect against malicious voting device, it can also be used as a help for the voter to ensure that they input the correct masked ballot. In order to avoid coercion, however, the voter would have to fake the return code sheet, assuming a certain level of computer skills.

4 Discussion

Following the description of counter-strategies and their related assumptions, we discuss the human factors related with applying the counter-strategies and make recommendations on designing coercion resistant systems.

4.1 Identified human factors and challenges

As the discussion of different counter-strategies revealed, there is a number of issues related to human factors that need to be addressed for ensuring proper use of coercion resistant voting systems, with some of these issues known from usable security research in other domains (see e.g. [51, 60]). These issues can be clustered as follows.

Unrealistic assumptions The complexity of the proposed counter-strategies is a significant issue that could potentially prevent the voters from applying these counter-strategies correctly. As such, they tend to require capabilities that are difficult or impossible to attain, such as being able to remember long, random-looking credentials or to input them on their first try without any errors.

While these limitations have been acknowledged in previous research, often by the authors of the proposed schemes, the suggested methods to aid the voters in their task either had to rely on additional security assumptions such as trusted hardware, or introduced further complexity for the voters.

Self-efficacy issues Even if the voters are actually capable to apply a counter-strategy, a seeming complexity of the process might still discourage them from it. This leads to lack of self-efficacy: even if the system actually provides ways for the voters to protect themselves against coercion, the voters might still feel helpless and unable to do so. Such lack of self-efficacy has been identified as an issue in other aspects of electronic voting that require actions from the voter that are unfamiliar to them from paper-based voting, such as verifying the integrity of one's cast vote [37]). This issue, however, might be even more crucial for coercion resistance, since the voter is under additional stress from coercion and the consequences of failure are potentially higher. If the voter tries to apply a counter-strategy and fails, they might face repercussions from the adversary. Even in the vote buying scenario, where the voter does not suffer any negative repercussions, but instead does not get his pay from the adversary, the voter might consider it a more rational decision to obey the adversary, if they do not see their vote as valuable enough.

Limited interactive feedback As opposed to voting in general, the system cannot provide feedback on the status of vote casting (e.g. whether the voter is applying a counter-strategy or not). All the explanations and voter instructions have to be provided in a non-interactive form, that is, they should not depend on the actions of the voter and whether they apply the counter-strategy.

Trust and acceptance Even in absence of coercion, the voters have to change their vote casting procedure, often to incorporate non-intuitive elements, such as entering a masked value instead of their vote, having to remember all the previously cast votes when updating, or remembering and distinguishing between different kinds of credentials. If explicit instructions to avoid coercion are provided, the voter might be altered and distrust the system. On the other hand, mentions of increased security of the system might make the voters accept the system more, once they are provided an explanation of the risks that are present in Internet voting and that the system is designed to protect against (see [35, 40] for related studies on the concept of cast-as-intended verifiability).

4.2 Recommendations

Considering the identified human factors and challenges, we propose a set of recommendations for future implementations of coercion resistant voting systems.

Involve the user Involving the user in the development of security-critical systems has been widely recommended in usable security research, including research on usability of electronic voting systems [46]. This is especially relevant when the assumptions on voters' capabilities are inherent in the cryptographic protocol, and any improvements after the system is implemented will most likely come with a change to the security model assumed in the initial scheme. Considering usability from the beginning of the development, including getting iterative feedback for the system prototypes from the users, would therefore help to identify potential issues early. This feedback can furthermore be used to design new counter-strategies that are more aligned with mental models of potential voters.

Provide aids The counter-strategies presented above rely on the voter *remembering certain secrets*, be it their real credential, votes cast previously in the election, or masking value. While the voter could write them down and use as a reference during vote casting, this could be an issue with over-the-shoulder coercion, where the adversary can observe the whole voter environment. For such a scenario, the secrets should be explicitly designed easy to remember (but at the same time, not easy to guess to the adversary). The voter should furthermore be provided with guidelines on how to remember these secrets, e.g. based on memorisation strategies for PINs [23].

In addition to secrets individual to each voter, there is also a need to remember the *steps of counter-strategy*, e.g. the rules of generating panic passwords, or general instructions on how and when the counter-strategy can be applied. A number of counter-strategies furthermore require the voter to perform some calculations, such as generating a panic password according to a set list of rules, or performing mathematical calculations, such as the XOR-function with the masking value or the sum of all the ballots cast within election so far. Moreover, several of these calculations also have to be performed during coercion-free voting. As the system can only provide limited feedback, the voter will not notice if they make a mistake in these calculations and thereby accidentally cast a ballot for a wrong candidate.

In case the secrets such as credentials or masking values are sent to the voter as voting materials, either via email or paper post, the voter should be able to *fake these materials* in case the coercer demands access to them.

As mentioned above, aids to these fundamental components of coercion resistant voting systems cannot be presented in an interactive way to the voter. Furthermore, in a scenario with the physical presence of the adversary, even non-interactive supplementary materials (e.g. paper-based instructions) cannot be used, as the adversary will demand the voter to put them away. We therefore propose that early in the development process, user studies are carried out in order to align voting system specifics and requirements with voter capabilities. The introduction of new voting systems, possibly related to new concepts such as coercion resistance, shall be conducted by incorporating accompanying awareness and education campaigns. One should, however, be careful in ensuring that the

inclusion of this additional information will not overwhelm the voter or make them distrust the system. It shall be emphasized here that previous research has proven that voters tend to accept and manage slightly more complex processes if this results in an increase of voting security [35]. By involving voters early in the process and providing them continuous support throughout the election, we mitigate the risks that come with the limited voter feedback of coercion resistant voting systems.

Do not over-rely on technology The unrealistic assumptions of the coercion resistant schemes show that the problem of coercion in remote voting is unlikely to be solved only by technology. Even if a usable solution is found, it is not guaranteed that the voter is capable of actually applying the solution, especially given the high-stress situation of coercion. When implementing the internet voting system, even the one that is designed to provide coercion resistance protection, one needs to be of the limitations of such protection, and include non-technological measures to prevent voter coercion and vote buying.

Consider implementing means for detecting coercion For some of the counter-strategies, the information available to the election officials might reveal some insights on whether coercion was attempted. This would include presence of votes with invalid credentials (for the fake credentials counter-strategy), unusually frequent vote updating (for deniable vote updating) or invalid ballots (for deniable vote updating or masking). The concept of coercion evidence [21] was designed to provide this feature specifically. Such a feature could be a valuable tool in enabling the coerced voters to signal abuse to the authorities. At the same time, it can lead to false positives, such as voters making mistakes during coercion-free voting e.g. by entering an invalid credential, or malicious voters who misuse the coercion detection mechanisms to undermine the legitimacy of the election and the trust of the electoral system. One way to resolve this would be enabling to track the potential coercion attempts back to the individual voters. In that case, however, potential privacy issues have to be considered.

5 Conclusions

It is difficult to ensure coercion resistance in e-voting systems, as even the solutions that propose cryptographic protocols are hard to implement in a way that the voters are able to used them effectively. This is evidenced e.g. from the real-world applications of Internet voting systems, where it is either assumed that no coercion takes place (i.e. there are other safeguards in society that protect against this), or some form of protection against coercion is implemented at the cost of verifiability (e.g. deniable vote updating in Estonia and Norway). Given the issues outlined in the paper, designing a practical and usable coercion resistant scheme is a challenge.

It, however, has to be noted, that coercion cannot be fully excluded via traditional in polling-place voting as well, including traditional paper ballots. The

possibility of so-called "ballot selfies", which would be even harder to prevent as new devices such as smart watches and other wearables that are capable of recording and are harder to detect are becoming wide-spread. For these reasons, Benaloh in particular argued [9] that the techniques to achieve coercion resistance in Internet voting might be of greater help in preventing coercion than simply relying on safety of voting booths. An important direction of future work is therefore developing solutions for the aforementioned human factor-related challenges, including implementations of existing cryptographic schemes, their evaluation via empirical studies and development of new schemes that allow for counter-strategies more suitable for practical use.

A particular challenge is to integrate the coercion resistant property with verifiability, ensuring that the voters can also verify that their vote has been counted correctly. Such an integration is particularly challenging, as the voter should not be able to use the results of verification to construct a proof of how they voted to the adversary. While a few works consider providing verifiability in coercion resistant voting (see e.g. [49]), further investigation into the investigation of human factors involved in ensuring both of these properties is needed.

An interesting further direction of future work is studying the perception of the voters of risk and benefit trade-offs that come from applying coercion resistant strategies, as well as cross-cultural studies investigating the perceptions of these trade-offs in different societies. We furthermore did not consider other technical issues with implementing coercion resistant systems, such as the need to implement an untappable channel between the voter and the voting server (see [33] for an overview and discussion of such issues), which would have to be considered in relation to the human factors as well.

References

1. Achenbach, D., Kempka, C., Löwe, B., Müller-Quade, J.: Improved coercion-resistant electronic elections through deniable re-voting. JETS: USENIX Journal of Election Technology and Systems **3**(2), 26–45 (2015)
2. Araújo, R., Barki, A., Brunet, S., Traoré, J.: Remote electronic voting can be efficient, verifiable and coercion-resistant. In: International Conference on Financial Cryptography and Data Security. pp. 224–232. Springer (2016)
3. Araujo, R., Foulle, S., Traoré, J.: A practical and secure coercion-resistant scheme for internet voting. In: Towards Trustworthy Elections, pp. 330–342. Springer (2010)
4. Araújo, R., Rajeb, N.B., Robbana, R., Traoré, J., Youssfi, S.: Towards practical and secure coercion-resistant electronic elections. In: International Conference on Cryptology and Network Security. pp. 278–297. Springer (2010)
5. Araújo, R., Traoré, J.: A practical coercion resistant voting scheme revisited. In: International Conference on E-Voting and Identity. pp. 193–209. Springer (2013)
6. Aziz, A.: Coercion-resistant e-voting scheme with blind signatures. In: 2019 Cybersecurity and Cyberforensics Conference (CCC). pp. 143–151. IEEE (2019)
7. Backes, M., Gagné, M., Skoruppa, M.: Using mobile device communication to strengthen e-voting protocols. In: Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society. pp. 237–242. ACM (2013)

8. Basin, D., Radomirovic, S., Schmid, L.: Alethea: A provably secure random sample voting protocol. In: 2018 IEEE 31st Computer Security Foundations Symposium (CSF). pp. 283–297. IEEE (2018)
9. Benaloh, J.: Rethinking voter coercion: The realities imposed by technology. In: Presented as part of the 2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (2013)
10. Benaloh, J., Moran, T., Naish, L., Ramchen, K., Teague, V.: Shuffle-sum: coercion-resistant verifiable tallying for stv voting. IEEE Transactions on Information Forensics and Security **4**(4), 685–698 (2009)
11. Bernhard, D., Kulyk, O., Volkamer, M.: Security proofs for participation privacy, receipt-freeness and ballot privacy for the helios voting scheme. In: Proceedings of the 12th International Conference on Availability, Reliability and Security. p. 1. ACM (2017)
12. Bursuc, S., Grewal, G.S., Ryan, M.D.: Trivitas: Voters directly verifying votes. In: International Conference on E-Voting and Identity. pp. 190–207. Springer (2011)
13. Chaidos, P., Cortier, V., Fuchsbauer, G., Galindo, D.: Beleniosrf: A non-interactive receipt-free electronic voting scheme. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 1614–1625. ACM (2016)
14. Chen, G., Wu, C., Han, W., Chen, X., Lee, H., Kim, K.: A new receipt-free voting scheme based on linkable ring signature for designated verifiers. In: 2008 International Conference on Embedded Software and Systems Symposia. pp. 18–23. IEEE (2008)
15. Clark, J., Hengartner, U.: Selections: Internet voting with over-the-shoulder coercion-resistance. In: International Conference on Financial Cryptography and Data Security. pp. 47–61. Springer (2011)
16. Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: Toward a secure voting system. In: 2008 IEEE Symposium on Security and Privacy (sp 2008). pp. 354–368. IEEE (2008)
17. Dossogne, J., Lafitte, F.: Mental voting booths. In: Nordic Conference on Secure IT Systems. pp. 82–97. Springer (2011)
18. Essex, A., Clark, J., Hengartner, U.: Cobra: Toward concurrent ballot authorization for internet voting. In: EVT/WOTE. p. 3 (2012)
19. George, V., Sebastian, M.: An efficient homomorphic coercion resistant voting scheme using hierarchical binary search tree. In: 2009 WRI World Congress on Computer Science and Information Engineering. vol. 1, pp. 502–507. IEEE (2009)
20. Gjøsteen, K.: The norwegian internet voting protocol. In: International Conference on E-Voting and Identity. pp. 1–18. Springer (2011)
21. Grewal, G.S., Ryan, M.D., Bursuc, S., Ryan, P.Y.: Caveat coercitor: Coercion-evidence in electronic voting. In: 2013 IEEE Symposium on Security and Privacy. pp. 367–381. IEEE (2013)
22. Grontas, P., Pagourtzis, A., Zacharakis, A.: Coercion resistance in a practical secret voting scheme for large scale elections. In: 2017 14th International Symposium on Pervasive Systems, Algorithms and Networks & 2017 11th International Conference on Frontier of Computer Science and Technology & 2017 Third International Symposium of Creative Computing (ISPAN-FCST-ISCC). pp. 514–519. IEEE (2017)
23. Gutmann, A., Renaud, K., Volkamer, M.: Nudging bank account holders towards more secure pin management. International Journal of Internet Technology and Secured Transactions **4**(2), 380–386 (2015)

24. Haghighat, A.T., Dousti, M.S., Jalili, R.: An efficient and provably-secure coercion-resistant e-voting protocol. In: 2013 Eleventh Annual Conference on Privacy, Security and Trust. pp. 161–168. IEEE (2013)
25. Heiberg, S., Martens, T., Vinkel, P., Willemson, J.: Improving the verifiability of the estonian internet voting scheme. In: International Joint Conference on Electronic Voting. pp. 92–107. Springer (2016)
26. Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 539–556. Springer (2000)
27. Iovino, V., Rial, A., Rønne, P.B., Ryan, P.Y.: Using selene to verify your vote in jcj. In: International Conference on Financial Cryptography and Data Security. pp. 385–403. Springer (2017)
28. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Proceedings of the 2005 ACM workshop on Privacy in the electronic society. pp. 61–70. ACM (2005)
29. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Towards Trustworthy Elections, pp. 37–63. Springer (2010)
30. Kiayias, A., Zacharias, T., Zhang, B.: End-to-end verifiable elections in the standard model. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 468–498. Springer (2015)
31. Kim, S., Oh, H.: A new universally verifiable and receipt-free electronic voting scheme using one-way untappable channels. In: Advanced Workshop on Content Computing. pp. 337–345. Springer (2004)
32. Koenig, R., Haenni, R., Fischli, S.: Preventing board flooding attacks in coercion-resistant electronic voting schemes. In: IFIP International Information Security Conference. pp. 116–127. Springer (2011)
33. Krips, K., Willemson, J.: On practical aspects of coercion-resistant remote voting systems. In: International Joint Conference on Electronic Voting. pp. 216–232. Springer (2019)
34. Krzywiecki, Ł., Kutyłowski, M.: Lagrangian e-voting: Verifiability on demand and strong privacy. In: International Conference on Trust and Trustworthy Computing. pp. 109–123. Springer (2010)
35. Kulyk, O., Neumann, S., Budurushi, J., Volkamer, M.: Nothing comes for free: How much usability can you sacrifice for security? IEEE Security & Privacy (2017)
36. Kulyk, O., Teague, V., Volkamer, M.: Extending helios towards private eligibility verifiability. In: International Conference on E-Voting and Identity. pp. 57–73. Springer (2015)
37. Kulyk, O., Volkamer, M.: Usability is not enough: Lessons learned from human factors in security research for verifiability. E-Vote-ID 2018 p. 66 (2018)
38. Kutyłowski, M., Zagórski, F.: Verifiable internet voting solving secure platform problem. In: International Workshop on Security. pp. 199–213. Springer (2007)
39. Locher, P., Haenni, R., Koenig, R.E.: Coercion-resistant internet voting with everlasting privacy. In: International Conference on Financial Cryptography and Data Security. pp. 161–175. Springer (2016)
40. Marky, K., Kulyk, O., Renaud, K., Volkamer, M.: What did i really vote for? on the usability of verifiable e-voting schemes. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. p. 176. ACM (2018)
41. Neji, W., Blibech, K., Rajeb, N.B.: Incoercible fully-remote electronic voting protocol. In: International Conference on Networked Systems. pp. 355–369. Springer (2017)

42. Neto, A.S., Leite, M., Araújo, R., Mota, M.P., Neto, N.C.S., Traoré, J.: Usability considerations for coercion-resistant election systems. In: Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems. p. 40. ACM (2018)
43. Neumann, S., Volkamer, M.: Civitas and the real world: problems and solutions from a practical point of view. In: 2012 Seventh International Conference on Availability, Reliability and Security. pp. 180–185. IEEE (2012)
44. Nguyen, T.A.T., Dang, T.K.: A practical solution against corrupted parties and coercers in electronic voting protocol over the network. In: Information and Communication Technology-EurAsia Conference. pp. 11–20. Springer (2013)
45. Nguyen Thi, A.T., Dang, T.K.: Enhanced security in internet voting protocol using blind signatures and dynamic ballots. In: Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services. pp. 278–281. ACM (2012)
46. Olembo, M.M., Volkamer, M.: E-voting system usability: Lessons for interface design, user studies, and usability criteria. In: Human-Centered System Design for Electronic Governance, pp. 172–201. IGI Global (2013)
47. Patachi, Ş., Schürmann, C.: Eos a universal verifiable and coercion resistant voting protocol. In: International Joint Conference on Electronic Voting. pp. 210–227. Springer (2017)
48. Rønne, P.B., Atashpendar, A., Gjøsteen, K., Ryan, P.Y.: Short paper: Coercion-resistant voting in linear time via fully homomorphic encryption. In: International Conference on Financial Cryptography and Data Security. pp. 289–298. Springer (2019)
49. Ryan, P.Y., Rønne, P.B., Iovino, V.: Selene: Voting with transparent verifiability and coercion-mitigation. In: International Conference on Financial Cryptography and Data Security. pp. 176–192. Springer (2016)
50. Sako, K., Kilian, J.: Receipt-free mix-type voting scheme. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 393–403. Springer (1995)
51. Sasse, M.A., Flechais, I.: Usable security: Why do we need it? how do we get it? O'Reilly (2005)
52. Schläpfer, M., Haenni, R., Koenig, R., Spycher, O.: Efficient vote authorization in coercion-resistant internet voting. In: International Conference on E-Voting and Identity. pp. 71–88. Springer (2011)
53. Shirazi, F., Neumann, S., Ciolacu, I., Volkamer, M.: Robust electronic voting: Introducing robustness in civitas. In: 2011 International Workshop on Requirements Engineering for Electronic Voting Systems. pp. 47–55. IEEE (2011)
54. Smart, M., Ritter, E.: Remote electronic voting with revocable anonymity. In: International Conference on Information Systems Security. pp. 39–54. Springer (2009)
55. Smart, M., Ritter, E.: True trustworthy elections: remote electronic voting using trusted computing. In: International Conference on Autonomic and Trusted Computing. pp. 187–202. Springer (2011)
56. Sodiya, A.S., Onashoga, S., Adelani, D.: A secure e-voting architecture. In: 2011 Eighth International Conference on Information Technology: New Generations. pp. 342–347. IEEE (2011)
57. Souheib, Y., Stephane, D., Riadh, R.: Watermarking in e-voting for large scale election. In: 2012 International Conference on Multimedia Computing and Systems. pp. 130–133. IEEE (2012)

58. Spycher, O., Koenig, R., Haenni, R., Schläpfer, M.: A new approach towards coercion-resistant remote e-voting in linear time. In: International Conference on Financial Cryptography and Data Security. pp. 182–189. Springer (2011)
59. Storer, T., Duncan, I.: Two variations to the mcesg pollsterless e-voting scheme. In: 29th Annual International Computer Software and Applications Conference (COMPSAC'05). vol. 1, pp. 425–430. IEEE (2005)
60. Volkamer, M., Renaud, K., Kulyk, O., Emeröz, S.: A socio-technical investigation into smartphone security. In: International Workshop on Security and Trust Management. pp. 265–273. Springer (2015)
61. Weber, S.G., Araujo, R., Buchmann, J.: On coercion-resistant electronic elections with linear work. In: The Second International Conference on Availability, Reliability and Security (ARES'07). pp. 908–916. IEEE (2007)
62. Wen, R., Buckland, R.: Masked ballot voting for receipt-free online elections. In: International Conference on E-Voting and Identity. pp. 18–36. Springer (2009)
63. Xia, Z., Tong, Z., Xiao, M., Chang, C.C.: Framework for practical and receipt-free remote voting. IET Information Security **12**(4), 326–331 (2018)
64. Yi, X., Okamoto, E.: Practical mobile electronic election. In: 2011 IEEE/SICE International Symposium on System Integration (SII). pp. 1119–1124. IEEE (2011)
65. Zaghloul, E., Li, T., Ren, J.: Anonymous and coercion-resistant distributed electronic voting. In: 2020 International Conference on Computing, Networking and Communications (ICNC). pp. 389–393. IEEE (2020)
66. Zhang, Y.: An open framework for remote electronic elections. In: International Conference on Cryptology and Network Security. pp. 304–316. Springer (2008)

Revisiting Practical and Usable Coercion-Resistant Remote E-Voting *

Ehsan Estaji[1], Thomas Haines[2], Kristian Gjøsteen[2], Peter B. Rønne[1], Peter Y. A. Ryan[1], and Najmeh Soroush[1]

[1] SnT & University of Luxembourg, Luxembourg {firstname.lastname}@uni.lu
[2] Norwegian University of Science and Technology, Trondheim, Norway
{firstname.lastname}@ntnu.no

Abstract. In this paper we revisit the seminal coercion-resistant e-voting protocol by Juels, Catalano and Jakobsson (JCJ) and in particular the attempts to make it usable and practical. In JCJ the user needs to handle cryptographic credentials and be able to fake these in case of coercion. In a series of three papers Neumann et al. analysed the usability of JCJ, and constructed and implemented a practical credential handling system using a smart card which unlock the true credential via a PIN code, respectively fake the credential via faking the PIN. We present several attacks and problems with the security of this protocol, especially an attack on coercion-resistance due to information leakage from the removal of duplicate ballots.
Another problem, already stressed but not solved by Neumann et al, is that PIN typos happen frequently and would invalidate the cast vote without the voter being able to detect this. We construct different protocols which repair these problems. Further, the smart card is a trusted component which can invalidate cast votes without detection and can be removed by a coercer to force abstention, i.e. presenting a single point of failure. Hence we choose to make the protocols hardware-flexible i.e. also allowing the credentials to be store by ordinary means, but still being PIN based and providing PIN error resilience. Finally, one of the protocols has a linear tally complexity to ensure an efficient scheme also with many voters.

Keywords: Electronic voting · JCJ protocol · Human-based error · Usability.

1 Introduction

One of the main threats in remote electronic voting is that they are inherently susceptible coercion-attacks due to the lack of a voting booth. In their seminal paper, Juels, Catalano and Jakobsson [10] gave a formal definition of coercion-resistance and further devised a protocol (JCJ) satisfying this strong security property. To achieve this, JCJ assumes a coercion-free setup phase where the voter get a credential which is essentially a cryptographic key. To cast a valid ballot

* This research were supported by the Luxembourg National Research Fund (FNR).

this key needs to be entered correctly together with the vote. In case of coercion, the voter can simply give a fake random credential to the coercer and even cast a vote together with the coercer using this fake credential – the corresponding vote will be removed in the tally process. The tally process of weeding out the ballots with fake credentials and duplicates, however, suffers from a quadratic complexity problem in the number of voters and cast ballots. Several paper are devoted to reduce the tally complexity in JCJ, see e.g. [18,2,6,20], however, each with their drawbacks. JCJ and similar constructions however also suffer from usability deficits, see also [14]. Especially, the voter intrinsically cannot directly check if a cast ballot is valid and will be counted, see however [8].

Moreover the handling and storing of long credentials is a notorious usability problem, getting even harder with a coercer present. The usability was analysed by Neumann et. al. [16,15,5] and led to a protocol using smart cards for handling voter's credentials. The stored credential is combined with a PIN code to produce the full credential which will be compared with the credential stored by the authorities on the bulletin board. In this paper we revisit this protocol and present several attacks on coercion-resistance and verifiability, but also possible repairs.

Whereas the the smart card provides a solution to the usability problem, it also comes with strong trust assumptions and problems

- The smart card is generally needs to be trusted. A malicous card could e.g. use the wrong credential invalidating the cast ballot without detection, and we cannot let the voter check if the ballot is correct without introducing coercion threats.
- The coercer can take the smart card away from the voter to force abstention.
- It is more expensive, less flexible and harder to update than a purely software solution.
- One of the attacks that we found is that a coercer can use the smart card to cast ballots on his own. This not only endangers coerced voter's real vote, but due to a leak of information in the weeding phase, the coercer can also detect, with non-negligible probability, whether the coerced voter has cast an independent ballot against his instructions.

In this paper we will present protocols that repair, or at least diminishes the attack probability of, the last point by constructing new duplicate removal methods in JCJ. Further, the protocols constructed in this paper are hardware-independent: they could use a smart card, or they can be implemented using combination of a digitally stored cryptographic length key and a PIN only known by the voter. The long credential could be stored in several places – or even hidden via steganography. At ballot casting time the software will take as input the digital key and the password to form the credential submitted with the vote. Depending on the level of coercion, the coerced voter can either fake the long credential or, for stronger levels of coercion, the voter can reveal the the digitally stored credential to the coercer, but fake the PIN. Due to our improved tally, the coercer will not know if he got faked credentails or PINs.

Another major problem with the original construction, already discussed as an open problem in [16], is the high chance of users doing a PIN typo error which will invalidate the vote and remain undetected. Note that naively giving feedback on the correctness of the PIN is not possible for coercion-resistance as it would allow the coercer to check whether he got a fake PIN or not. Instead, we will define a set of allowed PIN errors (e.g. chosen by the election administrator), and we will consider a ballot as valid both if it has a correct PIN or an allowed PIN error, but invalid for other PINs. We construct protocols which at tally time secretly check whether a given PIN is in the set of allowed PINs and will sort out invalid ballots. The protocols can accommodate general PIN error policies, however Wiseman et. al. [22] studied usual errors in PIN entries. Two frequent errors are transposition errors (i.e. entering "2134" instead of "1234") and wrong digit number errors (i.e. entering "1235" instead of "1234"). Correcting for both of these errors is however problematic, as we will see, since the set of independent PINs becomes small.

The outline of paper is as follows. In Section 2 we present attacks and problems of the orignal NV12 scheme. Our improved protocols are presented in Section 3. In Section 4 we make a preliminary analysis of how many independent PINs exist when allowing certain PIN errors. Finally we conclude in Section 5.

2 Analysis of NV12: Attacks and Problems

Neumann et al. [16] carried out a usability analysis of JCJ and proposed a new scheme (NV12) for handling the credentials and vote-casting. In [15] a few modification were made to prevent side-channel attacks and an efficiency analysis was done, and finally [5] presented a prototype implementation and its efficiency.

2.1 The scheme:

In this subsection we give a brief overview of the NV12 scheme, we refer to [15] and the JCJ/Civitas papers [10,4] for more details. The entities participating in the NV12 protocol are: **A supervisor:** who is in charge of running election and declaring election authorities; **The voter:** who intends to cast her vote; **The voter's smart card, reader and computer:** which serves as interface between the voter and the JCJ / Civitas system. The smart card reader has a screen and PIN entry interface; **A registrar:** who administrates the electoral register; **A supervised registration authority and a set of registration tellers:** that provide the voter with her credential; **A set of tabulation tellers:** that are in change of the tallying process; **A set of ballot boxes:** to which voters cast their votes; **A bulletin board,** BB**:** that is used to publish information. The ballot boxes will publish to BB.

The framework of the scheme is as follows

1. **Setup Phase.** This step is the same as JCJ/ Civitas; an election public key, pk, will be computed and published.

2. **Registration Phase.** After offline and online registration phases, the voter's credential divided by the chosen PIN is stored on the smart card alongside with a designated verifier proof.
3. **Voting Phase.** The voting procedure is split into two phases implementing Benaloh challenges to the vote encryption
   - **Challenge:** The smart card commits to an encryption of the vote by displaying $\mathrm{hash}\big(\mathsf{enc}(\mathsf{vote}, \mathsf{pk}, r)\big)$. The voter notes down this hash, and if the encryption is challenged, the smart card releases the randomness r to the voter's computer, and the voter can verify the hash indeed was consistent with the vote choice via a third device. This challenge procedure can be reiterated.
   - **Cast:** When the voter chooses to cast, she then enters the PIN. Now, the ballot of the form $\langle \{\mathsf{CRD}\}_{\mathsf{pk}}, \{vote\}_{\mathsf{pk}}, \sigma, \phi \rangle$ is generated where σ is a zero-knowledge proof (ZKP) of well-formedness of the vote and ϕ is a ZKP of knowledge of both the credential and vote. This is sent anonymously to a ballot box. $\mathrm{hash}(\langle \{\mathsf{CRD}\}_{\mathsf{pk}}, \{vote\}_{\mathsf{pk}}, \sigma, \phi \rangle)$ is displayed and written down by the voter, and can be checked with the stored ballot in the ballot box to ensure stored-as-cast verifiability.
4. **Tallying Phase.** This step is also the same as JCJ/ Civitas.

The important trust assumptions made in [15] are

- For privacy it was assumed:
  - Half of the remote registration tellers and the supervised registration authority are trustworthy.
  - Neither the smart cards nor smart card readers can be corrupted.
  - The adversary is not able to corrupt a threshold set of tabulation tellers.
- For coercion-resistance we further need:
  - There is a point in the voting phase, in which the adversary cannot control the voter.
  - The adversary cannot control the voter's computer.
  - The channel to the ballot boxes is anonymous
- For verifiability it was assumed:
  - The adversary is not able to corrupt smart cards. With the Benaloh challenges implemented this was reduced further to [16]: The adversary cannot control the voting environment and the verification environment at the same time.

2.2 Attacks

We will now present attacks and discuss how to repair these.

Benaloh challenge problem: The first attack is on individual verifiability. The Benaloh challenge is available for the user to challenge whether the encryption of the vote is done honestly. The smart card and reader commits to the hash of the encryption via the screen of the smart card reader. The problem is that this hash is not checked for the cast ballot. Instead, what is checked for the cast ballot is that the hash of the full ballot including the encryption of the credential and ZKPs matches what is received in the ballot box. This means that the smart card can at first encrypt all votes honestly and commit to these. However, when the PIN is entered to cast a ballot, it can encrypt its own vote choice and include this in the ballot without being detected even if the verification environment is honest – this violates the trust assumption above.
Repair: Both the hash of the vote encryption and the full ballot needs to be compared with the values that can be calculated from the ballot received by the ballot box. This however reduces usability as now two hashes needs to be checked by the voter, a task which is not trivial. Particularly, the adversary can precompute hashes that are hard to distinguish for the voter - e.g. matching on the leading part. Another choice is to commit to the full ballot in the Benaloh challenge, however this requires the voter to enter the PIN for each challenge. Since it is a general problem in e-voting that verification checks are too infrequent among real voters, having to enter a PIN for each challenge further undermines the Benaloh challenge security. It might also happen that a voter would then maximally challenge once, and hence an efficient strategy for the adversary would be to cheat after the first challenge.

Brute force attack: The second attack in on coercion-resistance for a coercer demanding access to the smart card, alternatively on verifiability for a local adversary who manages to get access to the smart card undetected. The adversary could here simply try to guess the PIN and cast a vote. This is not detectable by the voter due to anonymity of the vote casting. Unfortunately, the PIN space cannot be scaled since it is upper bounded by the ability of the voter to remember and enter PINs correctly. Hence, the probability of guessing the PIN is not negligible. Further, the probability can be boosted by casting multiple votes. Note also, whereas we can assume that it is in the interest of the voter to use a correct smart card reader, the adversary can use a malicously constructed reader. Thus the ballot casting can be automated and the PIN space can be covered to get a probability of a valid cast vote to be 1. This is not impossible, e.g. according to [5] vote casting took about 13 seconds including network time. The theoretical value with network was around 8 seconds, and the value of modern smart cards should be much lower. However, even with the 2014 timings, the creation of the ballots (without sending) could be done in 22 hours. Note that whether the ballot is counted in the end will depend on the vote update policy, and when the voter is casting her own vote, however, here the adversary is free to optimise his strategy, e.g. try to cast last.
Repair: The smart card could demand that a certain time has to pass between each ballot cast. This time can however not be too long, otherwise a coercer

might detect it or utilise it for a forced abstention. Thus this repair can only lower the probability for casting a ballot with correct PIN.

Leaky duplicate removal: This is an attack on coercion-resistance, but can also be an attack on verifiability to boost the attack above. In the simplest form the coercer uses the smart card to cast a vote with some trial PIN. The coercer wants to determine if this trial PIN is a correct PIN. According to the protocol the voter will cast her true vote using the correct PIN at some secret point during the voting phase. However, in the tally phase credentials are weeded using plaintext equivalence tests (PETs) of the encrypted credentials directly on the submitted ballots.[3] If the coercer now sees an equivalence with his submitted trial ballot, he can guess that it was the voter casting the other ballot, and probably with the correct PIN. Thus he has determined the correct PIN and that the voter defied his instructions in one go. To boost the attack he can simply try several PINs.[4] In standard JCJ such an attack would not work since the submitted trial credential would have the same probability of being identical to the coerced voter's credential as for it to be identical to any other voter's credential, and further the probability would be negligible.

A local adversary getting access to the smart card could also follow this strategy to try to know the PIN and cast valid votes. This might actually be detected by the voter if he checks the weeding on BB and sees a duplicate of his own vote (note this was also mentioned in [17]), but in the protocol the voter is not instructed to do this. Thus the PIN is not really protecting against unauthorized use of the smart card.

Repair: It is actually surprisingly hard to make a tally protocol which does not leak information to prevent this attack. The original JCJ protocol relies on the fact that guessing the real full credential can only happen with negligible chance. A first repair could be to mix the ballots before doing weeding, but after verifying the ZKPs. This makes it difficult to implement certain policies, like the last valid vote counts; however, it fits nicely with the policy that a random selection from the valid votes count. Unfortunately, this does not prevent the attack. The coercer could mark his ballot by casting it a certain number of times which is likely to be unique. He then checks if he sees this number of duplicates or one more. Even if mix between each duplicate removal, which would be horrible for an efficiency perspective, we do not get a leak-free tally. The distribution of time until a PET reveals a duplicate will depend on whether the PIN was correct or not. Especially the coercer could cast a lot of votes with the same trial PIN which would make detecting this more visible. There are other methods to

[3] In general this is not good for coercion-resistance since a coercer might detect a voter not following instructions across elections, see [8].

[4] Note that the coercer does not have to let the voter know that he follows this strategy. The voter only knows that the coercer has access to the card for some short time. Based on this, she could also decide not to cast her true vote at all, but then the protocol could not really be called coercion-resistant since the coercer has a very efficient strategy to force abstention.

limit the the information leak in the tally which we will present below. Further, we will present a protocol that does not leak information about the number of duplicates per voter, and does have linear tally complexity (compared to the quadratic in JCJ), but which has an obfuscated form of participation privacy.

Fake election identifier: This is an attack on verifiability. As mentioned in the original JCJ paper, the zero-knowledge proofs need to include a unique election identifier. This identifier is announced by the election administrator and prevents that ballots are copied from one election to another, i.e. the proofs would not verify when the wrong identifier is used. However, the smart card needs to be updated with this identifier before vote casting. However, we cannot trust this is done correctly, i.e. an adversary e.g. controlling the voter's computer could try to provide a wrong credential.
Repair: The voter could enter the election identifier herself, but this is error prone. The simplest solution is that the voter checks that the submitted ballot has a zero-knowledge proof that verifies according to the real election identifier. This could be done when the hash of the full ballot is checked, but will mean that the voter has to wait a bit longer before being able to do this check.

Smart card removal: An obvious forced abstention attack is that the coercer simply demand to hold the smart card during the election period.
Repair: This problem seems quite inherent to the smart card approach. We could let the voter hold several smart cards. However, holding several cards would be physical evidence which a voter with a local coercer probably would not want to risk. Further, the number of cards allowed per voter could necessarily not be bounded. If each voter were allowed to hold e.g. 5 cards, the coercer would simply ask for five cards. If this is troublesome it seems better to leave the smartcard only approach and allow the voter to also hold the credential as a piece of data as in standard JCJ. This can more easily be hidden (steganography could be an option here) even though theoretically this also has problems [19]. Our protocols below can be implemented with or without smart cards.

2.3 Security Problems

In this section we discuss some problems with the protocol, that do not fall under the category of attacks.

The main usability and verifiability problem with the protocol is that PIN entry is error prone, as was already stressed in the papers by Neumann et al. An obvious solution is to have a PIN check, e.g. a checksum check. However, this would mean that only certain PINs are valid PINs, and in order for a voter to present a fake PIN to a coercer, she would first have to prepare a valid fake PIN, which is less usable.

An option with higher usability is to have a policy of allowed PIN errors and accept full credentials that corresponds to the PIN being entered with allowed

errors. This is the approach we will essentially follow in this paper, however our solutions will also work for checksum checks.

If JCJ had a method of verifying the cast votes, we would also be able to at least detect such PIN errors. Such a verification mechanism was suggested in [8] using the Selene approach. However, this check can only be made after vote casting has ended, thus too late to update a PIN typo.

Another problem is the assumption that the smart card is trustworthy. This does not seem like a valid assumption, at least for important election. The smart card could simply use a wrong credential in a ballot, which would invalidate the vote. Further, this cannot be detected since the smart card is the only holder of the credential. At least the encryption of the PIN could be Benaloh tested, but not the credential. Futher, the smart card reader is also trusted. However, this might not be enough in practice. As an example, if the middleware on the reader allows the voter's computer or the network to display messages on the screen, e.g. to say it is waiting for a connection, then it could e.g. try to display fake hash values. A corrupted smart card could also easily break privacy by using the encryption choice as a subliminal channel for the vote choice. In light of this the smartcard can also be seen as *a single point of failure*. We will thus focus on hardware-independent protocols.

3 Protocol Description

In this section we will present two protocols which tolerate PIN errors and prevents leak of information in the deduplication phase.

In our voting scenario the voter has two keys: a long key which is stored on her device (smart card or another device) and a short PIN, which is memorized. To efficiently evaluate whether a PIN is allowed we will use polynomial evaluation. To this end, given a user's PIN a, we generate an $\mathsf{ErrorList}_a = \{a_1 = a, a_2, \ldots, a_k\}$ of allowed PINs. Note the number of PINs here is constant for every voter and might contain duplicates. From this, we generate a polynomial, $\mathsf{poly}_{\mathrm{PIN}}(x) = \prod_{i=1}^{k}(x - a_i) = \sum_{i=0}^{k} p_i x^i$ which has all $\mathsf{ErrorList}_a$ members as its root. In order to check the validity of the PIN, typed by the voter, it is then sufficient check whether the polynomial value on this PIN is equal to zero or not.[5] It is obvious that this polynomial should kept secret otherwise an adversary can recover the PIN by factorizing the polynomial. Therefore we have to work with encrypted polynomials and a main challenge is the polynomial evaluation under this encryption. Assume we have $\mathsf{Enc}(\mathsf{poly}_{\mathrm{PIN}}(x)) = \sum_{i=0}^{k} \mathsf{cp}_i x^i$ and $\mathsf{CT}_{\mathrm{PIN}} = \mathsf{Enc}(\hat{a})$, we need to find a way to efficiently compute $\mathsf{Enc}(\mathsf{poly}_{\mathrm{PIN}}(\hat{a}))$.

The next challenge is to find a way to prove publicly that the individual voter's polynomial are correctly evaluated without endangering the coercion-resistance. This would e.g. rule out voters evaluating the polynomials on voter side only.

[5] Note there is a small problem here since we are in composite order groups and the polynomials might have more roots than the allowed PINs. However, the probability in general is negligible.

Further, while solving this problem, we will also focus on efficient protocols to obtain a practical JCJ scheme with (almost) linear tally time in the number of voters.To obtain this we need to sacrifice perfect privacy. In the first scheme we only have participation privacy by obfuscation inspired by [6,11]. Here ballots are submitted with an ID and homomorphic Paillier encryption can then be used to evaluate the polynomial. Everybody, e.g. also a separate authority, can cast votes labelled with ID which will later be discarded as invalid. Thus the actual participation of the voter is obfuscated and the voter can deny having participated in the election. Optionally, we could also follow the JCJ alternative method in [6] to achieve perfect privacy, however the cost will be that the voters twice have to defy the coercer and interact with the voting system. In the second scheme using BGN encryption, the information leak from duplicate removal will not be negligible, but bounded, and this scheme does not satisfy linear tally efficiency.

Due to space limitations, we will just explain the basic building blocks and their algorithm and suppress some details about ballot integrity and non-malleability from the zero-knowledge proofs, e.g. the inclusion of election identifiers and the correct form of the Fiat-Shamir transformations. Also, for simplicity, we describe the protocol with a single trusted party, but it is possible to distributively run this protocol. We will also not specify all parts of the distributed registration phase and the Benaloh challenges, this can be implemented as in the NV12 scheme with some obvious modifications and with the repairs mentioned above.

3.1 Paillier Instantiation

The first instantiation relies on the Paillier public-key cryptosystem which is a partially homomorphic and its security is based on the hardness of the decisional composite residuosity assumption. A ciphertext on message $m \in \mathbb{Z}_n$ has the form $\mathsf{CT} = (g^m \cdot r^n \mod n^2)$ which $n = pq$ and p, q are two same-length prime numbers, and g is a proper member of group $\mathbb{Z}^*_{n^2}$. Its homomorphic property allows us to evaluate the polynomial without decrypting the coefficients of the polynomials. Further it allows an efficient multi-party computation protocol to compare and (and hence sort) ciphertexts by plaintext values without decryption [13]. This algorithm is linear in the bit length, i.e. logarithmic in the security parameter, and can be made public verifiable [12]. Using this technique allows us to do the weeding process secure and efficient, but at the cost of all ballots being submitted with a voter identifier. To achieve participation privacy, obfuscating votes needs to be cast too.

eVoting Protocol with Paillier instantiation: In Set-Up phase, CA generates the pair of keys, for Paillier cryptosystem: $\mathsf{pk} = (n = pq, \mathbb{G}, g)$, $\mathsf{sk} = (p, q)$

1. **Registration Phase:** For voter $\mathsf{V_{id}}$ the registrar, does the following steps:
   - Long credential: Pick $\mathsf{crd} \leftarrow \mathbb{Z}_n$, store crd on voter's device.
   - Short credential: Pick random PIN $a \in$ PIN-Set and send it to voter $\mathsf{V_{id}}$.
   - Compute the error list for a based on the election policy: $\mathsf{ErrorList}_a = \{a_1 = a, a_2, \ldots, a_k\}$and set $\mathsf{poly_{id}} = \prod_{i=1}^{k}(x - \mathsf{crd} - a_i) = \sum_{i=0}^{k} p_i x^i$

- Encrypt polynomial coefficients: For $i = 0, \dots k : \mathsf{cp}_i = \mathsf{Enc}(p_i)$
- Provide a designated proof of validity for the ciphertexts, cp_i, $i = 0, \dots k$.
- Publish $\mathsf{V}_{\mathtt{id}} : \big(\mathsf{CP} = (\mathsf{cp}_0, \dots, \mathsf{cp}_k), \mathsf{Enc}(\mathsf{crd})\big)$ on bulletin board.

2. **Casting ballot:** Voter chooses her candidate m, and enter her choice of PIN, $\hat{a}$. The voting algorithm runs the following steps:
   - Encrypt m and long credential, $\mathsf{CT}_{\mathsf{vote}} = \mathsf{Enc}(m), \mathsf{CT}_{\mathsf{crd}} = \mathsf{Enc}(\mathsf{crd})$
   - For $i = 1, \dots, k$ compute $\mathsf{cp}_i^* = \mathsf{cp}_i^{(\hat{a}+\mathsf{crd})^i} \cdot \mathsf{r}_i^{*n}$ and $\mathsf{CT}_i = \mathsf{Enc}((\hat{a} + \mathsf{crd})^i)$ for random number $\mathsf{r}_i, {\mathsf{r}_i}^*$. Provide a proof, π_{ballot}, (also proof of knowledge) for the following relation:

$$
\begin{aligned}
\mathrm{R}_{\mathsf{ballot}} = \Big\{ & (x, w), x = \big(\mathsf{CT}_{\mathsf{vote}}, \mathsf{CT}_{\mathsf{crd}}, \mathsf{CT}_i, \mathsf{CP} = (\mathsf{cp}_i)_{i\in[k]}, \mathsf{CP}^* = (\mathsf{cp}_i^*)_{i\in[k]}\big) \\
& w = \big(\mathsf{vote}, \mathsf{r}_{\mathsf{vote}}, \hat{a}, \mathsf{crd}, \mathsf{r}_{\mathsf{crd}}, \{\mathsf{r}_i, \mathsf{r}_i^*\}_{i\in[k]}\big) : \\
& \mathsf{CT}_{\mathsf{vote}} = g^{\mathsf{vote}} \cdot h^{\mathsf{r}_{\mathsf{vote}}}, , \mathsf{vote} \in \mathtt{List\ of\ candidats}, \mathsf{CT} = g^{\mathsf{crd}} \cdot h^{\mathsf{r}_{\mathsf{crd}}}, \\
& i = 1, \dots, k : \mathsf{CT}_i = g^{(\mathsf{crd}+\hat{a})^i} \cdot h^{\mathsf{r}_i}, \mathsf{cp}_i^* = \mathsf{cp}^{(\mathsf{crd}+\hat{a})^i} \cdot h^{\mathsf{r}_i^*} \Big\}
\end{aligned}
$$

     This proof can be implemented efficiently using Sigma protocols and will rely on the DDH assumption, and will be given in a long version of the paper. They can be made non-interactive using the strong Fiat-Shamir heuristic. Note that the hash should contain all parts of the ballot.
   - Cast $\mathsf{ballot}_{\mathsf{V}} = (\mathsf{CT}_{\mathsf{crd}}, \mathsf{CT}_{\mathsf{vote}}, \{\mathsf{cp}_1^*, \dots, \mathsf{cp}_k^*\}, \pi_{\mathsf{ballot}})$ with her ID.
   - Obfuscate: Everybody can cast (invalid) votes with any voter ID. This will obfuscate whether voter ID participated in the election as in [6,11]
3. **Tally Phase:** Using the Paillier encryption scheme, allows us to efficiently sort ciphertexts based on plaintext values without decrypting them, see [13]. This techniques can be done in a multi-party computation which provide privacy for the e-voting protocol. $\mathsf{MPC}_{\min}$ the algorithm that takes as input the ciphertexts $\mathsf{ct}_1 = \mathsf{Enc}(m_1), \mathsf{ct}_2 = \mathsf{Enc}(m_2), \dots, \mathsf{ct}_t = \mathsf{Enc}(m_t)$ and outputs the index i^* such that $\mathsf{ct}_{i^*} = \mathsf{Enc}(m_{i^*}) : m_{i^*} = \min\{m_1, \dots, m_t\}$. We use this algorithm in the Tally phase:
   - **Ballot Validity check:** In the first step, we remove exact ballot copies and all ballots with invalid proof π_{ballot}. In the next step we need to remove extra ballots for each voter, making sure a valid ballot is kept, if existing.
   - **Weeding:** Since each voter will be associated with possibly more than one ballot, we need to weed them. We make sure a valid ballot is chosen - if existing. Assume there are q ballots with the same ID, $\mathsf{ballot}_1, \dots, \mathsf{ballot}_q$, We now homomorphically combine the public ciphertext cp_0 with the submitted encryptions to obtain an encrypted polynomial evaluation for each ballot: $\mathsf{Enc}(\mathsf{poly}_{\mathtt{id}}(\mathsf{crd}_i + \hat{a}_i)) = \mathsf{cp}_0 \cdot \prod_{j=1}^{k} \mathsf{cp}_j^*, i = 1, \dots q$. Denote by $t_i = \mathsf{poly}_{\mathtt{id}}(\mathsf{crd}_i + \hat{a}_i)$ and note this is zero if the ballot has a valid credential and pin. We now verifiably mix the pairs $\mathsf{Enc}(t_i), \mathsf{Enc}(\mathsf{vote}_i)$ and run the $\mathsf{MPC}_{\min}$ algorithm on the first ciphertexts to determine the one with the minimal t_i. We only keep this ciphertext and the corresponding

encrypted vote and discard the rest. Note that this will select valid ballots having $t_i = 0$ if they exist.[6]

- **Ballot anonymization:** We delete the ID, run all the remaing pairs $\mathsf{Enc}(t), \mathsf{Enc}(\mathsf{vote})$ through a verifiable parallel mixnet for re-encryption and permutation.
- **Final PIN and Credential validity check:** Finally, for each ballot, we decrypt the polynomial evaluation. All ballots with non-zero polynomial evaluation will be discarded. We need to do this step without revealing any information about t_i for non-zero evaluation. Thus the tally tellers first jointly and verifiably multiply some random number onto t_i and then decrypt. We accept ballots with output zero and discard the rest.
- **Vote decryption:** Decrypt the remaining vote ciphertexts and compute the voting result.

Error tolerance property of the scheme: Note the following computation:

$$\mathsf{cp}_i = g^{p_i} \cdot \mathsf{r}_i^n \ , \ \mathsf{cp}_i^* = \mathsf{cp}_i^{(\hat{a}+\mathsf{crd})^i} \cdot \mathsf{r}_i^{*n} \Rightarrow \mathsf{cp}_i^* = g^{(\hat{a}+\mathsf{crd})^i p_i} \cdot \mathsf{r}_i'^n$$

$$\Rightarrow \mathsf{cp}_0 \cdot \prod_{i=1}^{k} \mathsf{cp}_i^* = g^{\sum_{i=0}^{n}(\hat{a}+\mathsf{crd})^i p_i} \cdot \mathsf{r}^n = g^{\mathsf{Poly}_{\mathtt{id}}(\mathsf{crd}+\hat{a})} \cdot \mathsf{r}^n$$

Decrypting this gives us the polynomial evaluation. Note that this evaluation will only check if $\hat{a} + \mathsf{crd}$ is valid. This should be sufficient for security. However, to check that both the credential is corrected and the PIN is in the allowed space, we can use a distributed plaintext equivalence test [21] between the submitted credential and the registrered credential and add the outcome under encryption to the polynomial evaluation.

Security analysis: The main advantage of this instantiation is sorting the ciphertexts without decrypting them. Note that $\mathsf{poly}_{\mathtt{id},\mathrm{PIN}}$ has the range in nonnegative integers. Therefore if there is any ballot with valid credential and PIN, the output of $\mathsf{MPC}_{\min}$ will be a valid ballot. On the other hand, it does not reveal whether any ballot has a valid pin or not, thus sidestepping the attack on the standard duplicate removal.

3.2 BGN Instantiation

The second instantiation is based on composite order groups introduced by [3] and the Groth-Sahai NIWI-proof system [7] with security are based on the Subgroup decision assumption.

The main point of using those in this instantiation are, BGN is a homomorphic encryption scheme which can be efficiently implemented in a bilinear group. Having bilinear map allows us to do the polynomial evaluation in an efficient and secure way and also having the efficient NIWI-proof system.

[6] This will give a random correct vote. The policy "Last valid vote counts" can be implemented by adding the received order to t_i.

Definition 1. ***BGN Cryptosystem*** *works as follows. Its Key-Generation algorithm,* KGen *outputs a pair of keys:* $\Big(\mathsf{pk} = (n, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g, h = g'^q), \mathsf{sk} = (p, q)\Big)$ *which* $\mathbb{G} = \langle g \rangle$ *and* $\mathbb{G}_T$ *are two groups of order* n *and the secret key consists of two primes* p, q *such that* $n = pq$. $\mathbf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ *is bilinear* $(\forall a, b \in \mathbb{Z}, g \in \mathbb{G} : \mathbf{e}(g^a, g^b) = \mathbf{e}(g, g)^{ab})$*, non-degenerate* $(\mathbb{G} = \langle g \rangle \Rightarrow \mathbf{e}(g, g) \neq 1_{\mathbb{G}_T})$ *and commutable map. A ciphertext on message* $m \in [T]$*, for* $T < q$ *has the form* $\mathsf{CT} = g^m h^{\mathsf{r}} \in \mathbb{G}$ *for some random number* r*. Decryption: raise the ciphertext to power* p *and compute the discrete log.*

BGN E-voting Protocol:

1. **SetUp Phase:** The central authority runs the BGN key-generation algorithm to generate $(\mathsf{sk}_{\mathsf{BGN}} = p, q, \mathsf{pk}_{\mathsf{BGN}} = (n, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g, h)$. Then chooses four random group elements $f_1, f_2, f_3, f_4 \in \mathbb{G}$. Note that $\mathbb{G} = \langle g \rangle$ is a cyclic group so there exists a unique integers $z_i, i \in [4]$ such that $f_i = g^{z_i}$. Set the secret key of election as $\mathsf{SK}_{\mathsf{election}} = (p, f_1, f_2, f_3, f_4)$ and public key of election as $\mathsf{PK}_{\mathsf{election}} = (n, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g, h)$. Publish $\mathsf{PK}_{\mathsf{election}}$ on the bulletin board.
2. **Registration Phase:** Registrar, $\mathcal{R}$, for voter V does the following steps:
   - Generate credential and pin: crd, a as in the Paillier instantiation.
   - Generate the list of errors, $\mathsf{ErrorList}_a = \{a_1 = a, a_2, \ldots, a_k\}$. Then compute $\mathsf{poly}_a = \prod_{i=1}^{k}(x - a_i) = \sum_{i=0}^{k} p_i x^i$ and the following ciphertexts: $i \in [k] : \mathsf{cp}_i = \mathsf{Enc}(p_i) = g^{p_i} h^{\mathsf{r}_i}, \mathsf{cp}_0 = g^{p_0} \cdot f_1^{\mathsf{crd}} h^{\mathsf{r}} = \mathsf{Enc}(p_0 + \mathsf{crd} \times z_1)$. Note that, technically cp_0 is the encryption of $p_0 + \mathsf{crd} \times z_1$. Although z_1 is not a known value to any parties, the registrar can compute cp_0 without knowing its value.
   - Generates a designated proof of validity of the polynomial poly_a and all cp_i, for $i = 0, \ldots k$.
   - Store $\mathsf{CP} = (\mathsf{cp}_0, \mathsf{cp}_1, \ldots, \mathsf{cp}_k), \mathsf{CRD} = g^{\mathsf{crd}}$ in the user device and publish $\mathsf{Enc}(\mathsf{crd}) = g^{\mathsf{crd}} \cdot h^{\mathsf{r}}, \mathsf{CP}$ on bulletin board.
3. **Casting ballot:** Voter V chooses her candidate vote, and enter her choice of PIN, $\hat{a}$. The voting algorithm runs the following steps:
   - Compute, $\mathsf{CT}_{\mathsf{vote}} = \mathsf{Enc}(\mathsf{vote})$ and $\mathsf{CT}_{\mathsf{crd}} = \mathsf{Enc}(\mathsf{crd}) = \mathsf{CRD} \cdot h^{\mathsf{r}}$.
   - PIN encryption: For $i = 1, \ldots, k$ compute $\mathsf{CA}_i = \mathsf{Enc}(\hat{a}^i)$.
   - Re-randomize cp_i for $i = 0, \ldots, k$ by multiplying in a random $h^{\mathsf{r}_i^*}$ to generate cp_i^*.
   - Set $\mathsf{CA} = (\mathsf{CA}_1, \ldots, \mathsf{CA}_k), \mathsf{CP}^* = (\mathsf{cp}_0^*, \ldots, \mathsf{cp}_k^*)$ and provide a proof (Proof of knowledge), π_{ballot} for the following relation, including a joint proof of plaintext-knowledge for all the other ciphertexts in the ballot and include the rest of the ballot in the hash for non-malleability. This proof can be generated using the Groth-Sahai technique.

$$\mathrm{R}_{\mathsf{ballot}} = \Big\{(x, w), x = \big(\mathsf{CT}_{\mathsf{vote}}, \mathsf{CT}_{\mathsf{crd}}, \mathsf{CA}\big), w = \big(\mathsf{vote}, \mathsf{r}_{\mathsf{vote}}, \mathsf{CRD}, \mathsf{r}_{\mathsf{crd}}, \hat{a}, \{\mathsf{r}_i\}_{i \in [k]}\big) :$$
$$\mathsf{CT}_{\mathsf{vote}} = g^{\mathsf{vote}} \cdot h^{\mathsf{r}_{\mathsf{vote}}}, \mathsf{vote} \in \texttt{List of candidates},$$
$$\mathsf{CT}_{\mathsf{crd}} = \mathsf{CRD} \cdot h^{\mathsf{r}_{\mathsf{crd}}}, \{\mathsf{CA}_i = g^{(\hat{a})^i} \cdot h^{\mathsf{r}_i}\}_{i=1,\ldots,k}\Big\}$$

   - Cast $\mathsf{ballot} = (\mathsf{CT}_{\mathsf{vote}}, \mathsf{CT}_{\mathsf{CRD}}, \mathsf{CA}, \mathsf{CP}^*, \pi_{\mathsf{ballot}})$

Polynomial evaluation: The following computation shows how to evaluate the polynomial on the input value $\hat{a}$, the PIN that was used by the voter:

$$\mathbf{e}(\mathsf{CT}_{\mathsf{crd}}, f_1)^{-1} \cdot \mathbf{e}(\mathsf{cp}_0^*, g) \cdot \mathbf{e}(\mathsf{cp}_1^*, CA_1) \cdots \mathbf{e}(\mathsf{cp}_k^*, CA_k) =$$

$$\mathbf{e}(\mathsf{CRD} \cdot h^{\mathsf{r}}, f_1)^{-1} \cdot \mathbf{e}(g^{p_0}(f_1)^{\mathsf{crd}} h^{\mathsf{r}_0}, g) \cdot \mathbf{e}(g^{p_1} h^{r_1}, g^{\alpha_i} h^{\gamma_i}) \cdot \ldots \mathbf{e}(g^{p_k} h^{\mathsf{r}_k}, g^{\alpha_k} h^{\gamma_k}) =$$

$$\mathbf{e}(\mathsf{CRD}, f_1)^{-1} \cdot \mathbf{e}(h, f_1)^{-\mathsf{r}} \mathbf{e}(g^{p_0} f_1^{\mathsf{crd}} h^{\mathsf{r}_0}, g) \cdot \mathbf{e}(g^{p_1} h^{\mathsf{r}_1}, g^{a^i} h^{\gamma_i}) \cdot \ldots \mathbf{e}(g^{p_k} h^{\mathsf{r}_k}, g^{a^k} h^{\gamma_k}) =$$

$$\mathbf{e}(\mathsf{CRD}, f_1)^{-1} \mathbf{e}(f_1, h^{\mathsf{r}}) \mathbf{e}(f_1, \mathsf{CRD}) \mathbf{e}(g^{p_0} h^{\mathsf{r}_0}, g) \cdot \mathbf{e}(g^{p_1} h^{\mathsf{r}_1}, g^{a^i} h^{\gamma_i}) \cdot \ldots \mathbf{e}(g^{p_k} h^{\mathsf{r}_k}, g^{a^k} h^{\gamma_k}) =$$

$$\cdot\, e(h^{\mathsf{r}}, f_1)(\prod_{i=0}^{k} \mathbf{e}(g^{p_i}, g^{\alpha_i})) \cdot (\prod_{i=0}^{k} \mathbf{e}(g^{p_i}, h^{\gamma_i})) \cdot (\prod_{i=0}^{k} \mathbf{e}(g^{\alpha_i}, h^{\mathsf{r}_i}))(\prod_{i=0}^{k} \mathbf{e}(h^{\gamma_i}, h^{\mathsf{r}_i}))$$

$$\mathbf{e}(g, g^{\sum_{i=0}^{k} p_i \alpha_i})) \cdot \mathbf{e}(g, h^{\mathsf{r}}) = \mathbf{e}(g, g^{\mathsf{poly}_a(\hat{a})}) \cdot \mathbf{e}(g, h^{\mathsf{r}})$$

Hence, if we raise above term to power p, if $\mathsf{poly}_a(\hat{a}) = 0$ the result is equal to 1 and otherwise not. Due to the secret f_1 and zero-knowledge proofs, malicious voters cannot construct a zero-evaluation dishonestly.

• **Tally Phase:** First, we check the validity of the proofs, π_{ballot}. In case any of any failure, the ballot will be discarded.

- Step 1: Compute the encrypted polynomial evaluation as above and provide a proof of its validity (efficient using the Groth-Sahai technique). Call this $\mathsf{Enc}_T(t)$ with t being the polynomial evaluation which can be seen as an encryption in the target space. Note that this is computed from the ballot alone. Now verifiably mix the tuples $(\mathsf{CT}_{\mathsf{crd}}, \mathsf{CT}_{\mathsf{vote}}, \mathsf{Enc}_T(t))$. For each ballot we now create $\mathsf{Enc}_T(\mathsf{crd} + t)$ and remove duplicates ballot having the same $\mathsf{crd} + t$ which basically means same credential and same error-equivalent PIN for honest ballots. We will do this via PETs. If we have a small number of voters, we can mix between each duplicate removal. For a larger number we suggest to split the board in two, remove duplicates separately, then mix and do duplicate removal again. This will decrease the information from the distribution of confirmed duplicates to a coercer carrying out the "leaky duplicate removal attack" mentioned in Sec. 2.
- Step 2: We now want to select eligible valid votes. We mix the above list and the list of registered encrypted credential. Then we perform PETs between each registered credential and the submitted credential and homomorphically add the polynomial value to this before decrypting the result. This will be one if the credential is correct and the polynomial evaluation is correct. When we get a positive test result we do a further PET against the credentials. This will reveal malicous authorities creating valid polynomial evaluations on their own. If this is positive too, we decrypt the vote and continue to the next registered credential.

4 PIN Space Coverings

Our voting protocol ensures that the voter's credential is validated even if they make certain typos in their PIN. This could e.g. be a transposition error or a single wrong digit.

The interesting question from a security viewpoint is now how much this reduces the entropy of the PINs. To have a precise research question, we investigate how many PINs an attacker needs to try to cover the whole PIN space. This is related to the brute force attack of an attacker holding the real credential e.g. in the smart card. We will not solve this exactly in generality, but give some upper and lower bounds. Note also, that users generally are not good at choosing random PINs as revealed in PIN frequency analyses. We thus recommend that the PIN should be generated uniformly at random and not chosen by the voter.

We first focus on the case where we allow PIN swaps and an error in one digit. Let us denote the PIN by $p_1p_2\cdots p_k$. We first compute the number of PINs covered by a PIN try. Let us start with the case $k = 2$. By $[p_1p_2]$, we mean the set of numbers covered by this PIN. Clearly $[p_1p_2] = \{p_1p_2, p_2p_1, p_1*, *p_2\}$, where $* \in \{0, 1, 2, \ldots, 9\}$. After removing the repeated cases we will have $|[p_1p_2]| = 20$ for the case $p_1 \neq p_2$ and it will be 19 for the case $p_1 = p_2$. Actually, for $2r$ distinct digits $p_1, \ldots, p_{2r}$, one can verify that the r 2-digits numbers $p_1p_2, p_3p_4, \ldots, p_{2r-1}p_{2r}$ will cover a total of $20r - 2\binom{r}{2}$ PINs. The formula can also be used to give an upper bound of PINs cover by r PIN trys, and thus it shows that the attacker needs at least 8 PINs to cover the entire PIN space of all 2-digits numbers.

Since the attacker is trying to cover the PIN space with the minimum number of attempts, a good strategy seems to be to add PINs with distinct digits as much as possible to the basis. In the case there is no possible new PIN with distinct digits, we will then add a PIN which increase the size of current basis the most, and so forth until the PIN space is covered. We have implemented an algorithm in Python following this idea, but using random sampling to find the next optimal element for efficiency. For the case of 2-digits PIN, a basis of size 9 was found which is close to the theoretical lower bound.

Let us now consider the case of 3-digit PINs. For any PIN $p_1p_2p_3$ the maximum size of all covered PIN, $|[p_1p_2p_3]|$ is 30. Therefore 34 will be an lower bound for the size of basis of PIN space in this case.

Assume that only swapping errors are tolerated. For 2-digit PINs, finding a basis is equivalent to finding a basis for upper triangular matrices. There the basis size is 55 which the Python code also finds. For $k \geq 3$, an upper estimate of the cover of a single PIN is k (including itself) thus $10^k/k$ is a lower bound.

We collect the lower theoretical bounds and the upper bounds resulting from our Python code for PIN lengths between 2 and 5 in Table 1. We ran the code 1000 times in the case of 2,3 and 4 and just one time for the case 5.

PIN Length	2	3	4	5
S+W Lower Bound	8	34	250	2000
S+W Upper Bound	9	78	713	6490
S Upper Bound	55	465	4131	

Table 1. S+W means the system accepts swapping errors and wrong digit errors, where S means a system that just tolerate swapping errors.

5 Conclusions and Outlook

In this paper we have presented attacks and repairs on the NV12 scheme, especially, we have also presented protocols which are resilient to human errors in the form of PIN typos. It is interesting to notice that the digitally stored key could be combined or replaced with a key derived from biometric data. An important future direction is to make the error correction here so efficient that we can allow using noisy biometric data without fuzzy extraction.

For the Paillier-based system that we have presented it would be natural to add the tally system from Ordinos [12] since this is also based on Paillier encryption. Ordinos will only reveal the winner or the ranking of the candidates in the election, and will thus help for coercion-resistance in the case where there are candidates which expected to only get few or no votes. Another method that could used in both protocols is the risk-limiting tally method described in [9] which gives plausible deniability for the voter.

The PIN space analysis might be of general interest, and more precise results should be found. Interestingly, the one-digit error in k-digit PINs is related to Rook-polynomials, [1], in a k-dimensional chessboard.

Finally, some socio-tehcnical research questions are: 1) Which type of PIN errors do voters do when the are in a vote setting and do not get any feedback on the correctness of the PIN. 2) Related to this, what it the optimal PIN policy that corrects as many PIN typos while still keeping the entropy of the PIN space sufficiently high. 3) If we do not use a smart card, or use both a smart card and key storage: how well can voters be trained to handle, fake and hide secret keys.

Of course a main missing part is to provide proofs of security for our protocols.

Acknowledgments. This work was supported by the Luxembourg National Research Fund (FNR) and the Research Council of Norway for the joint project SURCVS and by the FNR CORE project FESS.

References

1. R.B.J.T. Allenby and A. Slomson. *How to Count: An Introduction to Combinatorics, Second Edition*. Discrete Mathematics and Its Applications. Taylor & Francis, 2011.
2. R. Araújo, A. Barki, S. Brunet, and J.s Traoré. Remote electronic voting can be efficient, verifiable and coercion-resistant. In *International Conference on Financial Cryptography and Data Security*, pages 224–232. Springer, 2016.
3. D. Boneh, E. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *In TCC*, pages 325–341, 2005.
4. M. R. Clarkson, S. Chong, and A. C. Myers. Civitas: Toward a secure voting system. In *2008 IEEE Symposium on Security and Privacy, 18-21 May 2008, Oakland, California, USA*, pages 354–368. IEEE Computer Society, 2008.
5. C. Feier, S. Neumann, and M. Volkamer. Coercion-resistant internet voting in practice. In E. Plödereder, L. Grunske, E. Schneider, and D. Ull, editors, *44. Jahrestagung der Gesellschaft für Informatik, Informatik 2014, Big Data - Komplexität meistern, 2014*, volume P-232 of *LNI*, pages 1401–1414. GI, 2014.

6. P. Grontas, A. Pagourtzis, A. Zacharakis, and B. Zhang. Towards everlasting privacy and efficient coercion resistance in remote electronic voting. In *International Conference on Financial Cryptography and Data Security*, pages 210–231. Springer, 2018.
7. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. *Electronic Colloquium on Computational Complexity (ECCC)*, 14, 01 2007.
8. V. Iovino, A. Rial, P.B Rønne, and P. YA Ryan. Using Selene to verify your vote in JCJ. In *International Conference on Financial Cryptography and Data Security*, pages 385–403. Springer, 2017.
9. Wojciech Jamroga, Peter B Roenne, Peter YA Ryan, and Philip B Stark. Risk-limiting tallies. In *International Joint Conference on Electronic Voting*, pages 183–199. Springer, 2019.
10. Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *Towards Trustworthy Elections*, pages 37–63. Springer, 2010.
11. O. Kulyk, V. Teague, and M. Volkamer. Extending helios towards private eligibility verifiability. In R. Haenni, R. E. Koenig, and D. Wikström, editors, *E-Voting and Identity*, pages 57–73, Cham, 2015. Springer International Publishing.
12. R. Küsters, J. Liedtke, J. Mueller, D. Rausch, and A. Vogt. Ordinos: A verifiable tally-hiding e-voting system. *IACR Cryptol. ePrint Arch.*, 2020:405, 2020.
13. H. Lipmaa and T. Toft. Secure equality and greater-than tests with sublinear online complexity. In F. Fomin, R. Freivalds, M. Z. Kwiatkowska, and D. Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, 2013, Proceedings*, volume 7966 of *Lecture Notes in Computer Science*, pages 645–656. Springer, 2013.
14. A. Silva Neto, M.Leite, R.Araújo, M. Pereira Mota, N. Sampaio Neto, and J. Traoré. Usability considerations for coercion-resistant election systems. In M. Mota, B. Serique Meiguins, R. Prates, and H.Candello, editors, *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems, IHC 2018, Brazil, 2018*, pages 40:1–40:10. ACM, 2018.
15. S. Neumann, C. Feier, M. Volkamer, and R. Koenig. Towards a practical jcj/civitas implementation. *INFORMATIK 2013–Informatik angepasst an Mensch, Organisation und Umwelt*, 2013.
16. S. Neumann and M. Volkamer. Civitas and the real world: Problems and solutions from a practical point of view. In *Seventh International Conference on Availability, Reliability and Security, Prague, ARES 2012, Czech Republic, August 20-24, 2012*, pages 180–185. IEEE Computer Society, 2012.
17. Peter B. Roenne. JCJ with improved verifiability guarantees. In *The International Conference on Electronic Voting E-Vote-ID 2016*, 2016.
18. P. B Rønne, A. Atashpendar, K. Gjøsteen, and P. YA Ryan. Coercion-resistant voting in linear time via fully homomorphic encryption: Towards a quantum-safe scheme. *arXiv preprint arXiv:1901.02560*, 2019.
19. Adi Shamir and Nicko Van Someren. Playing 'hide and seek'with stored keys. In *International conference on financial cryptography*, pages 118–124. Springer, 1999.
20. O. Spycher, R. Koenig, R.and Haenni, and M. Schläpfer. A new approach towards coercion-resistant remote e-voting in linear time. In *International Conference on Financial Cryptography and Data Security*, pages 182–189. Springer, 2011.
21. Pei-Yih Ting and Xiao-Wei Huang. Distributed paillier plaintext equivalence test. *I. J. Network Security*, 6(3):258–264, 2008.
22. S. Wiseman, P. Cairns, and A. Cox. A taxonomy of number entry error. In *Proceedings of the 25th BCS Conference on Human-Computer Interaction*, pages 187–196. British Computer Society, 2011.

Developments in Technology in Elections from European Examples

The Election Information System in Finland in 2035 – A Lifecycle Study

Juha Mäenalusta[1][0000-0003-1954-6399] and Heini Huotarinen[2][0000-0002-0235-9952]

[1] Legal Register Centre, Hämeenlinna, Finland
juha.maenalusta@om.fi
[2] Ministry of Justice, Helsinki, Finland

Abstract. This paper summarizes a lifecycle study of the Finnish Election Information System. The system is the basis for future digital development of Finnish elections. The paper describes the vision for an election information system, summarizes areas studied as well as current and possible future voting methods. The way forward is presented as well, covering many areas that need to be considered.

Keywords: elections, cybersecurity, lifecycle.

1 Background

Electoral integrity is a key priority to election officials everywhere, also in Finland. For successful introduction of any technical system or tool in elections, it is of utmost importance that citizens trust the voting system. As IT systems are a major tool for running elections, the systems need to be trustworthy. Finland has had a comprehensive and centralized Election Information System already since the 1990s. The Election Information System is a vital tool in organizing national and municipal elections.

The Election Information System includes data on constituencies, polling stations, voting register, data on candidates and election results and a centralized calculation system so it affects the conduct of elections during the whole electoral cycle. There are about 5 000 users of the System in every election and the whole constituency, approximately 4,4 million eligible voters, is listed.

As IT systems age, the decisions made during design of the system might no longer be valid due to changes in requirements or components. For instance, the maintenance of some software libraries might have ended. Introducing major information systems takes time, and in case of elections, this process must be coordinated with the electoral cycle. At some point maintaining older systems can be even more expensive than creating a new system. Therefore, the Ministry of Justice set up a project for the term from 7 June 2019 to 30 May 2020 to examine the technical life cycle of the Election Information System and the costs and challenges that should be considered in the development of the aging system in the coming years. This study was published in in June 2020 [1].

The Election Information System is used in all general elections: Parliamentary Elections (every 4th year), Presidential Elections (every 6th year), Municipal Elections (every

4th year) and European Parliamentary Elections (every 5th year). In Parliamentary Elections the country is divided into 13 electoral districts, in Municipal Elections it is a municipality that constitutes an electoral district and in Presidential and European Parliamentary Elections the country is not divided into electoral districts.

There is no separate Electoral Management Body (EMB) in Finland, but the tasks often handled by the EMBs are divided among the Ministry of Justice, the Legal Register Centre, the Digital and Population Data Services Agency and the electoral district committees. The Ministry of Justice, which has overall responsibility over elections in Finland, owns the Election Information System. The Legal Register Centre takes care of development and technical use of the system together with private IT-companies.

2 Technical lifecycle of the Election Information System

The current Election Information System was built in 2006-2012. The election authorities are familiar with it and consider it reliable and dependable. However, there were two main topics to study at this point: technologies used and security. The technological solutions are over a decade old, so it was considered appropriate to see what their lifecycle is likely to be, considering supplier support and developer availability. These considerations are not specific to election systems, but general for all information systems. However, the study was also a reaction of the Ministry of Justice to changes in the perception and importance of cybersecurity in elections. The public debate surrounding e-voting was also highlighting importance of security and trust. The study included ideas for developing the features and usage of the system. [1]

Multiple themes were studied: technologies, security, features, usability, accessibility, contracts, documentation and organization. For this, election experts from the Ministry of Justice and election information system experts from Legal Register Centre were organized into a project group. External consultants were used only to facilitate technology workshops and provide input on technology choices. That way it was possible to utilize all silent knowledge from within the organizations and to build competencies. Many relevant organizations were consulted, including local election authorities, cybersecurity authorities and election experts from neighboring countries. [1]

The main output of the study was to form a vision of how an election information system should be, by the target year 2035. This vision serves as the basis for development of the Election Information System. The vision is presented in Finnish, but this is an unofficial translation by authors. [1]

1. The election information system is reliable, credibly secure and enables elections with lighter processes than before.
2. Authorities have critical expertise over the system and decision-making power is clearly divided.
3. The system guides users and does not require external user guides. The system adapts to different user needs and processes.
4. The system is transparent and the public information it generates is easily available to all.
5. The system withstands time and changes in the operating environment.

The study gives various results from various aspects of elections. The good news is that the study shows that most of the technologies currently in use in the Election Data System are also feasible in the future, and expertise is available for now. However, some technologies such as front-end libraries are not sustainable, and should be changed. Accessibility legislation is one example of changing legislation that requires substantial changes. More importantly, the system was not designed to provide credible transparent cybersecurity, that was included in the vision. Credible transparent cybersecurity is often a goal of e-voting system design so this one aspect that would enable introducing i-voting in the future.

The study concluded that election authorities should have the necessary expertise to understand the system in order to be able explain the security features. As the timeframe of the vision is 15 years into the future, it is not possible to give detailed recommendations about all technology choices. However, a valuable part of the study was determining what could be the possible threats or necessary changes in the future. These challenges, such as post-quantum cryptography and authentication methods available in the Finnish market, require that the situation is closely monitored. Updating to current software development good practices was also recommended. Recently the Election Data System was improved to be able to handle two elections at the same time (municipal and upcoming regional elections). The study suggests that such changes would be easier by a more modern system and more modern development practices. [1]

The conclusion of the study was that either a new system should be built or the old one modernized. Continuing with the old system was not seen as an option due to expense and risk analysis. The next step is a feasibility study of the two options.

One conclusion was that the importance of election security is increasing. It is seen both in Finland [2] [3] and abroad that the transparency about security in i-voting has influenced the expectations about securing other election-related systems. Development of the election security discussion internationally was also seen as something that should be closely monitored. [1]

3 Voting methods now and in the future

The voting methods in use are several types of advance voting and election day voting. In the last parliamentary election, 50,7% of voters voted in advance [4]. Advance voting is equally popular in all general elections. Convenience and confidence explain the popularity of advance voting in Finland. Advance voting is convenient as there are several advance voting stations in shopping centres, libraries and other popular public places. Voter may vote at any advance voting station of their choosing, in Finland or abroad. Advance voting is especially popular among voters from rural areas, voters who are attached to a certain political party and among the elderly. [5]

It is possible to vote from any advance voting station, and this is usually made very simple by using the Election Information System and online voter rolls on advance polling stations. The project suggests [1] expanding electronic voting rolls to advance voting at homes and institutions such as hospitals and prisons, where the advance voting

is now handled manually. Introducing electronic voting rolls to all advance voting stations could also be used to enable voting at any polling station on election day, but it was not recommended, as it would slow down result calculation. The study also recommended changing as much as possible of the advance voting process, except actual ballots, to an online system. That would help to overcome delays in mailing the ballots as the level of postal services is degrading. Digitalization of the overall electoral process would help in introducing i-voting in the future.

Finland does not have i-voting in place, but the possibility has been studied twice in the previous years, which shows the interest in the topic. Especially the interesting example from the southern neighbor Estonia has been a lively topic in public debate for several years. The first recent study on i-voting [2] recommended implementing i-voting to municipal referendums and the second study concerning use of i-voting in general elections [3] concluded that the risks outweigh the benefits. This conclusion was accepted by all political parties that were represented in the parliament at the time. As Finland has a central Election Information System, i-voting would have been done by integrating an i-voting system into the Election Information System to get voter rolls and combine results with other voting methods. This means that the Election Information System would be crucial for any i-voting in Finland. [3]

Trust in i-voting or any other development of electronic systems relies on the trust in the Election Information System, as it is the system controlling voter rolls and combining the different votes into one result. The current system is designed so that extra voting methods do not require big changes in the logic of the system, just a simple interface. The lifecycle study suggests that this would be the best way to combine these systems in the future.

Postal voting from abroad is the only voting method where the actual voting is not supervised by authorities. It has only been in use since 2019 and it seen as an anomaly. There is only a very short history of unsupervised remote voting in Finland and it is less than 10 000 voters that have ever used this option. The COVID-19 pandemic might affect the perception of the benefits of remote voting but since the 2017 report [3] there has been very little public debate around the subject. The turnout in Finland has remained rather high (2019 parliamentary election 72,1%) so it seems that voters are content with the existing voting methods.

4 Lessons learned

The project reinforced the understanding that it is necessary to link the reform of the contractual structure of the Election Information System to its technical reform. This is because the contracts are as old as the system and reflect the time of their writing. In addition, the need for legislative and procedural developments became evident during the project. For example, some fixed processes and due dates in the legislation make reforms too complicated.

A well-functioning, reliable and user-friendly election information system is a solid basis for any steps towards digitalization of electoral processes. Therefore, even if e.g. any form of e-voting is not currently being considered [3], it is important to bear in

mind that the Election Information System must adapt to any changes in the electoral cycle before and after year 2035. These changes could be for example new election types, changes in electoral districts of digitalization of processes such as voting card delivery.

Many benefits of e-voting can be achieved by many less exciting improvements in voting methods than e-voting. The project highlighted the need for more digitalized advance voting that would utilize the already existing wide use of electronic voter lists at advance voting stations. Also, many developments can be introduced, such as wider use of electronic voter rolls and introducing electronic candidate registration. Even though Finland has very digitalized elections compared to many other countries, international examples of best practices should be closely monitored.

Citizens' trust in election information systems requires that the system is understandable and transparent enough. The system should be transparent so that for example NGOs would have access to it and could examine it. Trust should be built throughout the process by consulting various stakeholders, peer review and mutual learning. Voters and temporary staff of polling stations should be involved by adapting UX design principles.

Finland is not immune to international phenomena such as election interference or disinformation even though so far, the situation remains calm and no major attempts to interfere with the Finnish elections were not detected in previous elections [6]. It seems that high levels of societal trust and media literacy make Finland a less attractive target for election interference. The Election Information System is also an important building block in countering election interference, when it is dependable and secure. Election system has recently been considered as part of critical infrastructure in Finland.

The project proposes that work for the development of the Election Information System shall be continued. Several questions concerning the costs, technical and functional details must be further examined and solved. Therefore, the next step towards year 2035 in technical development of Finnish elections will be a feasibility study that will begin shortly by the Legal Register Centre.

References

1. Huotarinen, H, Nurminen, L, Tjurin, A, Vornanen, A, Koskinen, A, Kujanen, M, Mäkelä, M, Kolehmainen, M, Mäenalusta, J, Kariniemi, U.: Vaalitietojärjestelmän elinkaariselvitys. Oikeusministeriön julkaisuja, Toiminta ja hallinto 2020:6 (2020).
2. Nettiäänestystyöryhmä: Nettiäänestystyöryhmän loppuraportti. Oikeusministeriön julkaisuja, Mietintöjä ja lausuntoja 28/2015 (2015)
3. Nettiäänestystyöryhmä 2017: Nettiäänestys Suomessa. Oikeusministeriön julkaisuja, Mietintöjä ja lausuntoja 59/2017 (2017)
4. Official Statistics of Finland (OSF): Parliamentary elections. http://www.stat.fi/til/evaa/index_en.html , last accessed: 2020/07/02.
5. Wass, H, Borg, S: Muutosvaalit 2011 Oikeusministeriön julkaisuja, Selvityksiä ja ohjeita 16/2012 (2012)
6. Jääskeläinen, A, Sillanpää, A, Valtonen, V, Huotarinen, H, Toivanen, J: Vaalivaikuttamisen koulutushankkeen loppuraportti. Valtioneuvoston julkaisuja 2019:22. (2019).

Pushing water uphill;
Renewal of the Dutch electoral process [1]

Peter Castenmiller[1] and Arjan Dikmans[2]

[1] Dutch Electoral Council, PB 20011, 2500 EA, The Hague, Netherlands
[2] Former managing director for the transition of the Dutch Electoral Council
p.castenmiller@planet.nl

Abstract. This contribution addresses current developments in the reorganization of the electoral process in the Netherlands, a process that centers on a renewal of digital support tools. It will be clear that these developments have been stormy since 2017 and have not yet produced a satisfactory result.

Keywords: Elections, Electoral Process, Digital support of Elections

1 Introduction

In many democracies around the world, the organization and execution of elections are clearly and increasingly becoming points of dispute in polarizing political relations. Institutions are being questioned. If the result for a specific party or person is disappointing, one is quick to point out deficiencies in the execution of the elections. As a result, it not only becomes increasingly important, but also increasingly difficult to ensure and maintain a neutral organization and execution of elections. One has also to be aware that the digitization of elections has all kinds of risks, ranging from failing IT facilities (IOWA caucus 2020) [2] to the possibility of people systematically influencing elections (USA presidential election 2016)[3 Since the US presidential election at the end of 2016 there has been more emphasis on hacking. All in all, the process of ensuring safe, transparent and credible elections is under pressure.

Apart from attention to interference by external parties, there is discussion about the responsible use of all kinds of digital technology to support the logistically challenging electoral process. For each individual country, the final choice for the use of digital technology in the electoral process is the result of a balance between the specific requirements associated with elections.[4] These are, of course, general quality requirements that can be set on elections, whether or not laid down in laws or in internationally approved standards. Whether district elections or general elections are held, however, also makes a fundamental difference in the way the process is organized. Furthermore, the organization of the electoral process is partly determined by political culture and past experience.

This contribution addresses current developments in the reorganization of the electoral process in the Netherlands, a process that centers on a renewal of digital support tools. It will be clear that these developments have been stormy since 2017[5] but haven't yet materialized in a satisfactory result.

2 Outline of the baseline situation

In the 20th century, a start was made in the Netherlands with automated and digital support of the electoral process, such as the use of voting machines. This really took off at the turn of the century. However, after an extensive reflection on the reliability and transparency of the use of voting machines, these were banned from 2007 onwards. The main argument is the need for physical evidence (a ballot paper), so recounting is possible. After the abolition of voting machines, the Dutch Electoral Council started developing software in 2008 to support the process of determining the results. Within years, a program was developed which in the last decade became almost indispensable in order for the results to be determined within the statutory time limits. Shortly before the parliamentary elections in 2017, a public debate was held about the reliability and security of the application of this software. Under pressure from this debate, the Minister of the Interior decided to prohibit the use and application of components of this software – such as the exchange of digital files from the approximately 10,000 polling stations. It was only with great difficulty that the Electoral Council managed to present the results in time.[6]

About the organization of elections in the Netherlands

Every election in the Netherlands involves a system of proportional representation. Although political parties present their candidates as a single list, the law only allows a vote to be cast on a person. This means that all candidates must be given a place on the ballot paper.

In the Netherlands, only the national government can establish laws. For the implementation of these laws, the national government is heavily dependent on local authorities, which have a high degree of autonomy in this implementation. This also applies to the organization of the elections. Laws and regulations are laid down by the national government, but most of these laws and regulations are implemented by municipalities. In order to guarantee independence in the electoral process, an Electoral Council was set up, which acts as a central electoral committee in the elections to the two national Houses of the Dutch parliament, as well as in European elections. In provincial council or municipal council elections, the Electoral Council has no direct responsibility, although the Electoral Council does provide support in the form of advice and the provision of supporting software.

Almost 10,000 polling stations are set up on election day. All Dutch citizens who are entitled to vote can go on foot or otherwise by the most popular national means of transport, the bicycle, to a polling station from 7.30 am to 9.00 pm and cast their vote. In this respect, the voting process poses very few logistical challenges to voters. It does pose challenges from the moment the voters have cast their votes. Staffing nearly 10,000 polling stations is not an easy job. It is estimated that between 60,000 to 80,000 people are involved as volunteers in the exe-

cution of elections. It involves the staffing of the polling stations and the determination of the results. In practice, it is becoming increasingly difficult for some municipalities to find enough volunteers, even without the COVID-19 virus.
Immediately after the polling stations are closed, the votes cast on paper are counted by hand, an activity which may take long into the next morning. In the case of elections to the national parliament, the votes cast on all candidates must first be tallied at a municipal level, then at electoral district level, and must eventually be combined to form a single national result, and all this within a few days.

Following the execution of the 2017 elections, which were characterized by many incidents [7], all those involved were convinced that this should be ‘once, but never again’. A structural solution for a better electoral process was sought in an innovation of the supporting software that can be used to determine the results. It soon became clear that merely an innovation of the software would be of little significance if it would not be part of an overall review and reorganization of the electoral process. For in addition to the software, it would also require a different management of the process, with a different division of responsibilities between the actors concerned.

3 Considerations for modernizing the electoral process

The basis of the electoral process was laid with the introduction of universal suffrage in 1919. At that time, there were only a few political parties and all candidates of these parties could be clearly placed on a single ballot paper. In the course of the 20th century, the number of parties involved in the elections increased, a development which continued strongly in the 21st century. In 2017, there were 28 parties that nominated candidates for the national parliament. This results in an ever larger ballot paper (70 x 50 cm), which becomes increasingly difficult to read. This poses a major challenge when counting all these ballot papers. It takes many hours to determine the result at the level of the polling station, an activity which, pursuant to the law, must take place within the polling station immediately after closure, at least for the time being.[8]

When voting machines were used on an ever-increasing scale at the beginning of the 21st century, the legislator saw cause to extend the closing time of the polling stations to nine o'clock in the evening. Although the voting can only be done on paper since 2007, this closing time has been maintained. This means that the counting can only start after 9.00 pm and in many polling rooms it takes until after midnight. This result must then be brought to the town hall that same evening, where an initial result at municipal level is determined that same night.[9] This could take until early in the morning, especially in large municipalities. The fact that these intensive and precise activities take place in the middle of the night, along with the fact that many of those involved have also been active during the day, increases the chance of inaccuracies

and errors. Municipalities are therefore increasingly critical of this procedure, and the call for the reintroduction of voting machines is growing stronger. In this respect, municipalities have public opinion on their side, and jokes are often made about the supposed archaic process; Statements such as "*it's hopelessly old-fashioned that you still have to vote with a pencil*" or "*If you can bank from home, why can't you vote from home?"* are often heard. However, the legislator is unyielding in its position, supported by good arguments, that the use of voting machines does not meet the requirements that can be set on elections.

The use of supporting software for determining the results only begins at the level of the municipalities, not in the individual polling stations. The total of all polling stations is entered at municipal level and recorded in a software system, which essentially only supports the counting process. Until 2017, the available digital results could then first be transferred to the electoral district and then to the Electoral Council by means of a password-protected USB stick. However, it was exactly the vulnerability of this digital recording and transfer of the results which was the topic of debate in 2017. Could the software be trusted? Were the software or the USB sticks not prone to error or easy to hack? Did the digital transfer not offer external powers, and perhaps also malicious foreign powers, the opportunity to hack and adjust the results? In the end, digital transfer via USB stick was banned at the 2017 elections, which led to an unexpected, almost uncontrollable, extra burden on the process.

The challenges are therefore of a different nature. Because of the social importance of elections, there is a great need for rapid publication of the results, causing permanent time pressure. But there is also a need for a safe and credible process when the results are determined. This means that there must be safeguards against fraud or inaccuracies, as well as for preventing external interference. This need for safety and reliability may rather require that the process must not be rushed.

There are several causes for the resulting pressure on the electoral process. It is therefore also very doubtful whether there is a single solution to achieving a much-desired secure, credible and transparent process. Trust is a fundamental part of democracy. It is the key for the credibility of an election. Everyone must be assured that one man has just one vote and that your vote will be counted. That is also why the institutions involved in the election process must be above doubt and totally independent.[10]

4 Elections in the Dutch polder and consensus democracy

Each country has different bodies that are involved in the electoral process. Particular examples are an Electoral Management Board, the legislator and executive bodies such as municipalities in the Netherlands. It is therefore relevant to note that all these bodies also have specific interests in the renewal of the electoral process. The Dutch

situation very much shows that these interests can sometimes be at odds with each other. For instance, relations between the Electoral Council and the Ministry of the Interior had gradually come under pressure since the abolition of voting machines. The Ministry was of the opinion that the Electoral Council was too hesitant and not sufficiently ambitious to take the lead in the renewal of the electoral process, in particular the development of a new, safer version of the supporting software. After all, it was also the Electoral Council that acted as the contracting authority for 'the first generation' of this software.

Around 2017, the Electoral Council held the view that, given the stricter requirements to be set on the new software, it would be better for the Ministry to take the initiative and bear responsibility. The software could then also be based on legal requirements yet to be developed. Moreover, only the legislator could make it compulsory for the software to be used in the calculation of the results. Furthermore, the Electoral Council believed that the increased complexity of designing and securing software would be beyond the knowledge available within its relatively small staff. It was said that the Ministry had more knowledge and experience available in areas such as software security or current cyber threats.

These difficult relations led to a deadlock in consultations, to growing irritation and frustration of the municipalities. After all, the logistical problems in the Netherlands mostly lie with the municipalities, which more and more believed that no one wanted to make an effort to relieve their problems. In the end, an independent mediator, a so called 'scout', had to be called in to break the deadlock and to outline a plan of action to work on these solutions. He delivered his opinion in February 2019. The opinion showed that there was consensus between the Ministry of the Interior, the Electoral Council and municipalities on the problem analysis and the tasks at hand. These were the following:

- Making a design and preparing for a tendering procedure for a new digital tool;
- Redesigning governance and reorganizing tasks and powers between the parties;
- Drawing up and implementing the necessary legislative amendments.

At the end of April 2019, these ambitions were laid down in an assignment that was given to an independent managing director, who began his work in mid-June 2019. As said, the central focus of his work was on modernizing the supporting software used to determine the results. The assignment was also to ensure a transition from Electoral Council to 'Electoral Authority'. This would not only make the Electoral Council responsible for the development, maintenance and management of digital tools in the electoral process, but would also create a hierarchy in the chain, the Electoral Council being granted far-reaching powers to monitor the desired quality. This hierarchy meant a breakthrough in the relations between the Ministry, municipalities and the Electoral Council existing in the Netherlands up to then. The strengthen-

ing of the role and powers of the Electoral Council in all elections meant an additional safeguard of the independence of elections. An additional advantage was that this would combine knowledge and experience.

Shortly after the managing director had taken office, it became clear that there was not enough time left to assume with certainty that 'reliable and tested new software would be available at the forthcoming parliamentary elections in March 2021. After all, a complete schedule of requirements had to be developed, followed by a public procurement procedure. Once this problem was identified, it was decided to develop a short-term strategy as well. The strategy consisted of a thorough update of the existing software, which tackles the current vulnerabilities. With a view to the elections to the House of Representatives in March 2021, a mandate was given at the beginning of 2020 to update the current Supporting Software for Elections (*Ondersteunende Software Verkiezingen*, OSV) to ensure that its use at the forthcoming parliamentary elections is still extremely credible and therefore justified. As the improvements are not expected to be sufficient in the long term (because of evolving hacking tools), work has also been done to develop and describe a new security concept and requirements for a completely new digital tool.

5 The role of cyber security

The importance of cyber security needs no explanation and that's why cyber security specialist were consulted. The involvement of the national security services has increased since the start of the programme to modernize the electoral process. For instance, they prepare a threat analysis and reviewed the security concept as well as the requirements to be set on the new software. This security concept is undergoing a number of major changes compared to the current software. So far, the Netherlands has been using software that is distributed to all municipalities. The municipalities install the software on their own computers, which may not have (or have had) an internet connection. This is not supervised or controlled in any manner whatsoever. In the new security concept, the software is managed centrally and a secure connection allows the various (also municipal) users to use the software. The use and network traffic are monitored centrally by a Security Operation Centre. So the concept changes from decentralized to centralized and from 'own responsibility' to 'central control'. It is proposed that the security concept be extended further with the creation of a second independent verification trail. Results of all polling stations are made available in digital form at one central location to allow citizens to check the allocation of seats themselves.

The proposed verification audit trail is off the upmost importance. The transparency in every step (from polling station via municipalities to the national level) secures the

trust of the public in the results (you can do your own check), but it also makes manipulation even more difficult. You have to infiltrate two systems instead of one! And because of this second trail there is no single point of failure. But – as often – it also has a downside. You could create confusion. Suppose the second trail leads to other results than the mean trail than one could argue that something is wrong and the results are manipulated. That is why also the verification trail must have a high level of cyber security. Moreover, in the Netherlands there is also a third (or better to say a first) audit trail: paper. The voting bills are secured and manual checks are always possible and will be taken.

6 From Electoral Council to Electoral Authority

In addition to the new IT, the role and responsibility of a new Electoral Authority is described. We want it to be an even more independent organization. For example: in the Netherlands the Electoral Council is financed by the Ministry and the Minister has to approve the results. The Authority will take over and strengthen the position and responsibilities of the Electoral Council. It is recorded how the Electoral Authority can determine that the election results are and will be credible. It will give an independent assessment of the credibility. Furthermore, there will come one commander. In the existing situation all executive parties (municipalities, districts and the election council) have an equal position. In the coming years there will be a hierarchy were the Authority can force executive parties in action. This demands that the roles of the various parties involved in the electoral process are described in detail. This has been done and these agreements form the basis for the legislation to be drawn up.

A new instrument e for the new Electoral Authority will be a report describing the course of the elections at the level of polling stations and municipalities. These facts and opinions are drawn up by an ‘election coordinator’ to be designated for each municipality, who can be compared with ‘election officers’ or ‘returning officers’ in some other countries. This report allows the Electoral Authority to decide to revise a result or require a recount. In turn, the Ministry will no longer draw up any regulations on the execution of elections. This role will also be assumed by the Electoral Authority. So this authority will be fully responsible for organizing and ensuring a reliable and credible course of these elections. The Authority will be given legal powers and means to do so. The secretariat of the Electoral Council will be expanded so that it can fulfil its new, larger role.

7 Failed at the last hurdle

After the rough start from the 2017 elections onwards, the implementation of the programme was speeded up in 2019. At the end of 2019, there was light at the end of the tunnel. Agreements had been made with the current supplier on a thorough update of the existing software. In addition, preparations for a tendering procedure for the development of completely new digital support were at an advanced stage. The plans for the other division of responsibilities in the electoral chain were also detailed. As in all other countries around the world, the Corona pandemic completely changed the situation in the early spring of 2020. Despite the drastically changed economic conditions and the severe strain this has put on public finances, the Ministry has made budget available for new electoral software and new governance. However, these funds are not sufficient for the full implementation of the proposed plans. As a result, it was necessary to decide to extend the useful life of the existing software, which, as already said, would be thoroughly adjusted. This would also entail additional costs, meaning that the financial viability of all long-term ambitions became uncertain. Given the partial cancellation of the promised funds and the impossibility to achieve all objectives of the programme within the scope of the original agreements, the managing director decided to resign as from June 2020. As a result, at the time of writing, the future of the programme is uncertain, or the programme is in any case delayed. However, it can be ascertained that all parties concerned have expressed and confirmed the will to successfully complete the programme. The usefulness and necessity of changes remain unabated.

8 Lessons for the future

The complexity of organizing and executing credible, secure and transparent elections has increased dramatically in a short time. Both from society and from various IT companies, there is pressure to rapidly digitize elections, whereas other experts, who are often also well-informed about the possibilities and limitations of IT, emphasize the risks. Furthermore with a 24/7 economy and all social media, the pressure to have fast, credible results increases and the citizens want it to be as easy as possible for them to vote.

All these tensions and risks are experienced differently in each country, which also means that they are addressed differently. In the Netherlands, the division of powers and responsibilities in elections between municipalities, the national government and the Electoral Council creates an additional complication. The necessary and inevitable renewal of the supporting software in the Netherlands has also been seized as an opportunity to improve cooperation in the chain, to strengthen the role and responsibility of the Electoral Council and to make the system of elections more resistant to un-

wanted interference. With so many different goals it is not strange that there were also many risks of failure. In this light, we have actually come a long way in the Netherlands. In the end, it was the Corona pandemic, which was unforeseeable until the beginning of 2020, which prevented the desired success.

Apart from the specific Dutch circumstances, there are the following, more general lessons for the future:

- developing a schedule of requirements for new election software is complex. It requires the use of specific expertise and therefore requires a lot of time and money;
- in this connection, the organization of a public tendering procedure for the actual development of that software is also complex;
- due to the increasing importance of securing the entire electoral process, an active role of national security services is becoming increasingly important, not to say inevitable; due to the involvement of various parties in the electoral chain, the overview and control of the entire chain, with all the requirements arising from legislation and regulations, the guarantee of a proper application of hardware and software and the protection against unwanted interference are becoming increasingly complex;
- an audit trail like our verification process is recommended. Trust and transparency are key for a democracy;

Based on our experience, we anticipate that maintaining and promoting a safe and credible organization and execution of elections, along with a well-considered and responsible use of digital tools, threatens to go beyond the knowledge and expertise of individual countries. We are aware that each country has its own system of elections, so the software has to be partly tailor-made for each country. But security concepts are the same. In a digital world cyber security is excellent or the software is not secured. And cyber security is more than bits and bytes. Organization, audit, transparency, back-up systems are part of a security concept. In view of the requirements to be imposed on parties interested in the development of the required software, increasingly fewer companies within Europe are able to do so. All in all, there is growing dependence on an increasingly limited number of commercial companies that will be able and willing to meet all requirements; That is why we want to call for more international cooperation, especially at a European level.[11] In pursuit of this, we envisage a European Centre of Expertise for elections, which could be tasked with the following activities:

- exchanging, pooling and enriching experience with elections;
- offering a platform to staff members of security services from different countries in order to exchange information about elections and threats to elections and to share knowledge about the security thereof;

- exchanging, pooling and enriching knowledge of and experience with digital support of elections;
- exchanging, pooling and sharing knowledge of the development and procurement of hardware and software for election purposes;
- and in time it can perhaps do the tendering and organise a purchasing power for the highest level of cyber security, an audit process and credible counting software.

Let's not do it on our one, let's work together. Cybercrime has no borders and the enemies of democracy in France are the same as in the Netherlands. Of course we know that there are existing platforms (i.e. the European Conference of Electoral Management Bodies organized by the Venice Commission), but these platforms are in our opinion without obligations. What we are suggesting is a permanent and more compelling organization.

It is conceivable to extend the tasks of such a centre of expertise to include, for example:

- developing basic components for election hardware and software that can be used in more than one country;
- supporting tendering procedures
- developing schedules of requirements for hardware and software to be used for elections.

It is time!

References

1. The views and opinions expressed in this article are those of the authors and do not necessarily reflects the opinion of the Electoral Council or the Ministry.
2. See: Q. Scanlann ea. Iowa caucus: what we know and what went wrong. In: ABC news, 2020
3. See: A. Bovet and H. Makse. Influence of fake news in Twitter during the 2016 US presidential election. In: Nature, 2020
4. See: P. Castenmiller and M. Bom. Digitizing the vote. In: Yearbook of the Association of Registrars, 2018; See also: A. Driza Maurer, Digital technologies in Elections, Questions, lessons learned, perspectives. Council of Europe, 2020.
5. This article can in some respects be seen as a follow-up to two articles that one of the authors published earlier on the use of digital technology in Dutch elections. These are: P. Castenmiller K. Uijl., The use of supporting software in elections, the peculiar case of the Netherlands 2017. In: R. Krimmer et al (eds.), Proceedings E-Vote-ID 2017, pages 315-325; P. Castenmiller and P. Young, Paper and digital: in search of 'the best of both worlds' in establishing the outcome of elections. In: Robert Krimmer et al (eds.), Proceedings E-Vote-ID 2018, pages 170 – 178
6. See: Kiesraad, Evaluatie-advies Tweede Kamerverkiezing 15 maart 2017, 2017

7. This course of affairs is described in: P. Castenmiller and K. Uijl, The use of 'supporting software' in elections, The peculiar case of the Netherlands 2017, see note 5.
8. For some years now, the legislator has been giving municipalities the opportunity, by way of experiment, to transfer the ballot papers from the polling station to a central location, where the votes are counted the next day.
9. As said, there are municipalities that participate in the experiment of central counting of votes on the day after the elections.
10. This is also laid down in the Recommendation of the Council of Europe on standards for E-voting (Recommendation CM/rec(2017)5).
11. We do not take a position on whether this should be done at the level of the European Community or another organization like the OSCE or Council of Europe.

Blockchain-Enabled Electronic Voting: Experiments in Ukraine

Dmytro Khutkyy[1][0000-0003-0786-2749]

[1] European University Institute, Florence, Italy
khutkyy@gmail.com

Abstract. Proponents of blockchain technology state that it facilitates transparency, verifiability, and auditability. Thereby, sometimes it is promoted and tested as an experimental e-voting design. In this relation, the contribution overviews the three cases of e-voting in Ukraine based on blockchain. Considering the available data, the cases are compared in the aspects of integrity mechanisms, social consensus arrangements, and political functions. It was found that despite differences in technology, all viewed e-voting systems allow remote internet voting, rely on an e-voting organizer, and enable advanced voting transparency. Finally, each consecutive blockchain-based e-voting initiative is more large-scale in the number of voters and the role in policy making and politics.

Keywords: Electronic Voting, Internet Voting, Blockchain.

1 E-Voting on Blockchain

Being technically feasible, electronic voting (e-voting) is applied for non-binding surveys, politically binding referendums, elections of public officials and other representatives in many countries [1], as well as for voting for policies by policy makers themselves. Yet, e-voting might be contested because of technical malfunction, malicious hacking, coercion, corrupt ballot counting, or other concerns [2]. To address some of these challenges on technological level, distributed ledger technologies, such as blockchain, strive to develop a more decentralized, transparent, verifiable, and auditable e-voting. Since blockchain is a shared record of information stored in a way that is resistant to manipulation, it has features that make it attractive as an electoral tool: it might make elections more transparent and verifiable, given that voters could theoretically go back and check that their vote was properly registered on the blockchain and counted in the election, all without broadcasting their identity to the rest of the world [3]. The application of blockchain in online voting eliminates the need for recounting the votes, enables an immediate calculation of the final result and is supposed to increase the embattled trust in electronic voting [4]. Reportedly, e-voting systems based on blockchain are being piloted in at least seven countries for party, government, and other e-voting procedures, although on a small scale [5]. This paper summarizes the three cases of blockchain-based electronic voting systems in Ukraine, albeit without intending to evaluate their implementation, usage, success or failures.

2 Evidence from Ukraine

Considering possible risks associated with the struggle for power in real-life voting, besides technical solutions such as encryption mechanisms, e-voting system integrity highly depends on the scrutiny of its algorithms and accordance to the agreed voting rules. Democratic and egalitarian potential of blockchains depends on the underlying consensus protocols: considering potential problems selfishness and imbalances of power, especially of core developers and users, it should possess sufficient safeguards to protect the integrity of recorded transactions or votes [6]. Moreover, according to the Council of Europe's standards for e-voting, an e-voting system shall be auditable [7]. Therefore, wherever relevant information is available, technical consensus protocols, social consensus arrangements, and integrity mechanisms are analyzed.

2.1 Planned Electronic Voting at a Local Council

The first e-voting system on blockchain in Ukraine, E-VOX:NaRada, was announced in 2016. This voting system was envisaged as a tool for local councils to automatize the process of voting, collecting cumulative data and publishing it on the Internet [8]. Considering the challenges associated with e-voting abuse in Ukraine's authorities, including instances of legally forbidden but practiced 'delegated' non-personal voting in the Parliament, probably, advanced technical solution was sought to ensure and demonstrate transparency, verifiability, auditability, and facilitate accountability.

E-VOX:NaRada is offered as a free open source license allowing council members (first of all at Ukraine's town council of Balta) to vote on plenary sessions and a web-portal to publish voting results in full compliance with the Ukrainian legislation; this electronic voting system based on blockchain technology is designed to ensure a complete transparency of the voting process and eliminate falsifications [9]. The e-voting system architect describes the system as follows: NaRada is a decentralized application (dApp) developed as a smart contract on Ethereum with the client part as an application on Android; the user's level of this dApp consists of two groups of users: (1) organizers – the Head of the City Council and the secretary; and (2) voters – Members of the City Council; the organizer creates and validates voters accounts, creates agenda, starts and stops voting sessions, while voters cast their votes [10]. The e-voting system open code has been published in 2016 and 2017 and is available online [11]. Its design does introduce a transparent and verifiable technological solution for e-voting. It still relies on a kind of social consensus regarding roles in e-voting process and organization procedures. Therefore, it is also important that the voting procedures comply with the legal regulations of the local council in practice.

Besides, the system architect has outlined an alternative e-voting system concept in a separate article. According to him, the algorithm is the following: (1) each voter creates a blockchain address and informs it to the voting organizer; (2) the organizer generates a multitransaction with a list of voters' addresses and sends each person a coin to vote and a coin to pay the transaction fee; (3) the organizer determines the voting addresses (e.g. one address is assigned 'For' and the other 'Against', or there are multiple candidates' addresses) and informs the voters about them; (4) voters, in turn, send the received coin to the chosen address thereby expressing their will; (5) the address

which has scored more 'voting' coins wins, thereby the decision is transparent and automatic [12]. Such design also technically enables the possibility of a secret blockchain voting. It is similar to the previous procedure, but with several preceding stages: (i) having fulfilled the required authorization conditions (e.g. by sending passport data or via an authorized account) the voter creates a blockchain address, encrypts it and sends it to the organizer; (ii) the organizer imposes a cryptographic Chaum's blind signature [13] to the encrypted address and returns it to the voter; (iii) the voter removes the encryption from the received data and sends to the organizer the address with the organizer's signature, but anonymously; (iv) having recognized the signature the organizer adds the received address to the voting coins' distribution list, but without identifying the owner of the address [14]. Such algorithm does enable voting secrecy and automatic vote count. Similarly to the previous e-voting protocol, the responsibility of issuing ballots and identifying voters rests with the e-voting organizer. Therefore, for the system integrity, the organizer should be held accountable.

In any case, neither of these blockchain-enabled e-voting designs have been introduced in Ukrainian local councils yet.

2.2 Completed Internet Voting for the Elections to the Supervisory Board

An e-voting system using distributed ledger and applying blockchain principles has been utilized for politically binding internet elections (i-elections) to the Supervisory Board of the Ukrainian Cultural Foundation (the Supervisory Board) in 2017. Ukrainian Cultural Foundation (the Foundation) is a public institution on cultural policy guided and coordinated by the Ministry of Culture of Ukraine (the Ministry). The Supervisory Board is an advisory body to the Foundation comprised of nine persons: one is the Foundation Head; two are appointed by the President; two are appointed by the Ministry; two are elected via an internet voting by cultural institutions, and two are elected via an internet voting by civil society organizations in cultural sphere [15]. The representatives elected by cultural institutions and by civil society organizations will be further named 'the Representatives.' Elected by stakeholders and comprising almost half of the Supervisory Board they bring a democratic voice to national policy making in arts and culture. The format of internet voting is supposed to increase transparency similarly to other cases of e-voting in Ukraine [16]. For example, since 2015 internet elections were held to public councils at the National Anti-Corruption Bureau of Ukraine [17], although using different, non-blockchain technologies.

There were two internet elections to the Supervisory Board: one for the representatives of cultural institutions and another – for the representatives of civil society organizations. As both i-voting webpages read: (a) each vote is inscribed in a voting ballot and is saved as an electronic document signed with an electronic digital signature of a person entitled to vote; (b) it is impossible to modify a signed ballot without ruining its integrity and its link with the electronic digital signature; (c) the use of an electronic digital signature allows to verify the integrity of a ballot and to identify its signer; (d) legally electronic digital signature is equal to a handwritten signature; (e) the electronic protocol is formed automatically [18, 19]. Indeed, the webpage of the first e-voting lists the full electronic protocol, which starts with a digitally signed control token, continues

with 102 digitally signed entries, and ends with a digitally signed control token. The same is true for the webpage of the second e-voting, the only difference is the number of digitally signed entries – 76. All these transactions are dated consecutively with the precision to minutes and have been performed during the two weeks of 13-27 October 2017. Each transaction is accompanied by a ballot and an encrypted digital signature. Each protocol entry has a number, the name of the chosen candidate, the name of the candidate's organization, the voting date and time, as well as the link to the ballot and an encrypted digital signature.

Given its i-voting design, it is an open voting. Each ballot discloses the name of the voting organization. The e-voting source code is not explicitly published online. Probably, this is due to complex requirements to licensing digital software used by public agencies that is coordinated by the State Special Communications Service of Ukraine. Nevertheless, both e-voting webpages, both lists of candidates, and both tickets starting e-voting were programmed and are available online as open data in structured XML format. According to the system architect, during the voting the list of voters was available in XML format too [20]. This displays high transparency.

This e-voting structure is based on blockchain principles, although there is the voter organizer (the Foundation) with special functions unavailable to voters. The Foundation serves as the organizing node that signs the list of candidates (the voter registry), starts and finishes the e-voting. Thereby the organizer node prevents the emergence of alternative e-voting chains and 'throw-in' of votes before or after the voting period protecting the integrity of the i-elections. In turn, each voter (a Representative) acts as a node continuing the chain with a signature. Thus each voter node protects the integrity of previous votes and its own vote. In addition, according to the system architect, each voter received a signed copy of bulletin and all previous votes via email [21]. Due to this any voter was able to prove how its vote was cast. Thereby, transparency, verifiability, and auditability of this e-voting system were indeed ensured.

2.3 Experimented Electronic Voting for National Elections

Another blockchain e-voting system was announced in 2018. It was intended to be applied on the scale of nation-wide elections. Reportedly, the Head of the State Register at the Central Electoral Commission of Ukraine (the Commission) has run a blockchain voting pilot: he created a test vote using 28 blockchain nodes, invited the public to vote, and concluded that the Commission continues a series of experiments applying the secure blockchain technology within electoral voting [22]. The social media post of the Head of the State Register at the Central Electoral Commission demonstrates an electronic voting form, although the displayed form link is an offline file [23]. As neither public hyperlink to the voting form nor the source code were provided by the Commission in the post, this hindered an independent examination of this e-voting system. Therefore, this case can be labelled as a reported non-binding e-voting experiment for national elections allegedly utilizing blockchain technology. Since e-voting has been inscribed in the governmental Concept Paper and the Action Plan for the Development of Electronic Democracy in Ukraine in 2017 [24], e-voting is on government's agenda, yet its specific technology and prospects are still unclear.

3 Conclusions on Blockchain-Based E-Voting in Ukraine

The presented cases of Ukrainian e-voting systems based on blockchain demonstrate several marked differences and similarities. Evidently, these ventures differ in technical solutions. Also, the degree of an e-voting system technical documentation availability in open access is in reverse proportion to its e-voting scale. The nation-wide e-voting system lacks published comprehensive technical documentation, the e-voting system for the supervisory board is available in a structured open data format, and the e-voting system for a local council has its source code published online. Considering this, for public e-voting and e-elections an enhanced transparency will be required. Still, these e-voting initiatives reveal common patterns. All the three e-voting systems used allow remote internet voting. This shows their potential to scale up. All of them represent versions of a distributed ledger with an e-voting organizer. Given the necessity to administer voter registers, initiate and finish voting process this is anticipated. Overall, this dynamic experimentation performed independently by multiple actors employing diverse technologies reflects a vibrant Ukrainian civic-tech and gov-tech ecosystem, where IT community, civil society and authorities cooperate for the public good. Finally, each blockchain-based e-voting undertaking is more ambitious in terms of scope. While the first one was designed for the internal use of local council members, the second one was conducted publicly among institutions and civil society organizations online, and the third one aspires to be used for binding national elections. This reveals the evolution of e-voting ambition in Ukraine and demonstrates the potential of wider use of blockchain-based systems in non-binding and binding e-voting for policies as well as in e-elections for advisory councils and public offices.

Acknowledgements

This work has been prepared within the framework of the Policy Leader Fellowship at the School of Transnational Governance, European University Institute.

References

1. IDEA: ICT in Elections Database, https://www.idea.int/data-tools/data/icts-elections, last accessed 2020/06/12.
2. Khutkyy, D.: Internet Voting: Challenges and Solutions. Policy Paper (2020), https://europeandigital.org/files/19/Internet_Voting_Challenges_and_Solutions_ENG.pdf, last accessed 2020/08/27.
3. Magnusson, W.: Blockchain Democracy. Technology, Law, and the Rule of the Crowd. Cambridge University Press, Cambridge, UK (2020).
4. Racsko, P.: Blockchain and Democracy. Society and Economy 41(3), 353–369 (2019), DOI: 10.1556/204.2019.007.
5. Jun, M.S.: Blockchain Government - A Next Form of Infrastructure for the Twenty-First Century. Journal of Open Innovation: Technology, Market, and Complexity 4(7), 1–12 (2018), DOI 10.1186/s40852-018-0086-3.

6. Hermstrüwer, H.: Democratic Blockchain Design. Journal of Institutional and Theoretical Economics 175, 163–177 (2018), DOI: 10.1628/jite-2019-0023.
7. Council of Europe: Guidelines on the implementation of the provisions of Recommendation CM/Rec(2017)5 on standards for e-voting (2017).
8. E-Vox democracy platform, https://en-gb.facebook.com/pg/e.vox.democracy/posts/, last accessed 2020/06/12.
9. EGAP: Winners of the EGAP CHALLENGE, the first Ukrainian contest of IT-projects on electronic democracy, are announced, https://egap.in.ua/en/novyny/winners-egap-challenge-first-ukrainian-contest-projects-electronic-democracy-announced/, last accessed 2020/06/12.
10. Konashevych, O., online conversation with the author, August 23, 2020.
11. E-Vox Repositories, https://bitbucket.org/evoxvoting/, last accessed 2020/08/23.
12. Konashevych, O.: Open and Secret Online Voting on Blockchain. Medium, 27 May (2018), https://medium.com/@Konashevych/open-and-secret-online-voting-on-blockchain-c3ea97ccd694, last accessed 2020/06/12.
13. Chaum, D.: Blind Signatures for Untraceable Payments. Springer (1998), http://www.hit.bme.hu/~buttyan/courses/BMEVIHIM219/2009/Chaum.BlindSigForPayment.1982.PDF, last accessed 2020/08/23.
14. Konashevych, O.: Open and Secret Online Voting on Blockchain. Medium, 27 May (2018), https://medium.com/@Konashevych/open-and-secret-online-voting-on-blockchain-c3ea97ccd694, last accessed 2020/06/12.
15. Electronic Democracy: Current Voting, http://mincult-voting.ed.org.ua/, last accessed 2020/06/12.
16. Khutkyy, D.: E-voting in Ukraine: Advancements, Challenges and Perspectives. Brussels Ukraïna Review April, 11–13 (2020), https://www.researchgate.net/publication/340862515_E-voting_in_Ukraine_Advancements_Challenges_and_Perspectives_Brussels_Ukraina_Review, last accessed 2020/06/12
17. National Anti-Corruption Bureau of Ukraine: Civil Oversight Council, https://nabu.gov.ua/tags/rada-gromadskogo-kontrolyu, last accessed 2020/06/12.
18. Electronic Democracy: The Protocol of Voting for the Representatives of Cultural Institutions, http://mincult-voting.ed.org.ua/vote1.html, last accessed 2020/06/12.
19. Electronic Democracy: The Protocol of Voting for the Representatives of Public Associations, http://mincult-voting.ed.org.ua/vote2.html, last accessed 2020/06/12.
20. Flonts, V., online conversation with the author, August 27, 2020.
21. Flonts, V., online conversation with the author, August 27, 2020.
22. Suberg, W.: Ukraine Electoral Commission Uses NEM Blockchain for Voting Trial. Cointelegraph, 8 August (2018), https://cointelegraph.com/news/ukraine-electoral-commission-uses-nem-blockchain-for-voting-trial, last accessed 2020/06/12.
23. Stelmakh, O.: Post. Facebook, 10 July (2018), https://www.facebook.com/permalink.php?story_fbid=1780800001987197&id=100001716433858, last accessed 2020/06/12.
24. The Cabinet of Ministers of Ukraine: The Concept Paper and the Action Plan for the Development of Electronic Democracy in Ukraine (2017), http://www.kmu.gov.ua/document/250417885/R0797.doc, las accessed 2017/11/08.

Schemes and Attacks

Privacy-preserving Dispute Resolution in the Improved Bingo Voting

Rosario Giustolisi and Alessandro Bruni

IT University of Copenhagen, Denmark
{rosg,brun}@itu.dk

Abstract. Dispute resolution mechanisms are important components of voting schemes, deterring a voting authority to change the election outcome as any alteration can be *proved* by such mechanisms. However, these mechanisms are useless if not triggered by voters, who should not have to choose to either raise a dispute or keep their vote private. Hence, voting schemes should include privacy-preserving dispute resolution.
In this work, we advance the formal analysis in the symbolic model of an improved version of the Bingo Voting scheme, whose enhancements include privacy-preserving dispute resolution mechanisms. Most of our analysis of several verification, dispute resolution, and privacy properties is done automatically using ProVerif, which we complement with manual induction proofs as necessary. We find that the scheme meets some properties only if one makes additional trust assumptions to those stated in [6]. For example, we find that dispute resolution is met assuming an honest voting authority. Moreover, our work provides an understanding of privacy-preserving dispute resolution in general, which can be beneficial to similar analyses of other voting schemes.

1 Introduction

Consensus on the election outcome and vote privacy are two main pillars of voting schemes. On the one hand, voting schemes that fail in achieving consensus are worthless, hence a voting scheme should provide high confidence in the result of the election despite voters do not necessarily trust the voting authority. On the other hand, failing to provide vote privacy opens to effective manipulation of voters and to control the outcome of the election. Intuitively, consensus on the election outcome and vote privacy seem to be two contrasting properties: more evidence would increase confidence in the election outcome at the risk of fewer privacy guarantees. Recent work [12] has shown that vote privacy implies individual verifiability. However, individual verifiability only enables a voter to *check* that her ballot has been counted, but not to publicly *prove* it. This means that a dishonest voting authority may still change the election outcome and there is no public evidence that could prove so.

One can deter a voting authority from changing the election outcome by introducing dispute resolution mechanisms that enable a voter to prove to any observer that her vote was not included in the tally. This should be possible

for the voter without giving up vote privacy, hence dispute resolution should be privacy-preserving.

In this paper, we provide a formal analysis of an improved version of the Bingo Voting scheme [6,18], which aims at ensuring privacy-preserving dispute resolution mechanisms among other features. We check automatically several verification, dispute resolution, and privacy properties in ProVerif, and identify the additional trust assumptions required by the scheme respect to the ones stated in [18]. To the best of our knowledge, this work represents the first formal treatment of the improved version of Bingo Voting. We provide the precise algorithm that enables an observer to dispute the outcome of an election and details the aftermath of a privacy-preserving dispute resolution at the voting phase, considering different mitigation scenarios. The outcome of our analysis pinpoints the difficulties in designing privacy-preserving dispute resolution mechanisms and can be useful for other voting schemes.

Outline. This paper is organised as follows. Section 2 details the improved Bingo Voting scheme as well as its properties and trust assumptions. Section 3 presents the formal analysis of verification, dispute resolution, and privacy properties in the improved Bingo Voting. Then, it discusses the outcome of the analysis. Section 4 presents some related work. Finally, Section 5 concludes the paper.

2 Background

Bingo Voting was originally proposed by Bohli, Müller-Quade and Röhrich in 2007 [7]. The underlying idea of Bingo Voting is that each voter receipt assigns to each candidate either a *dummy* random number or a *fresh* random number. The voting authority generates the dummy random numbers before the voting phase starts. A trusted random generator (TRNG) creates the fresh random numbers during the voting phase. The voting machine then assigns the fresh random number to the candidate chosen by the voter and a different dummy random number to each of the remaining candidates.

In Bohli et al. [6] and later in Henrich [18], several improvements are proposed to the original Bingo Voting system, including extensions to use Bingo Voting for more complex elections and ways to address usability limitations. In this paper, we consider two key improvements, hence we will refer to the resulting system as the *improved Bingo Voting*. The first improvement that we consider consists of two privacy-preserving dispute resolution procedures, one at the voting and the other at tallying. The other improvement regards the optimisation of the proof of correct distribution of dummy votes, which in the improved version is done after the voting phase. Figure 1 presents a message sequence chart of the scheme. The details of the scheme are outlined below.

Before the voting phase, the voting authority generates and publishes a set of *dummy votes*. A dummy vote consists of a pair of Pedersen commitments that hide both the dummy random number and the assigned candidate. Each candidate receives the same number of dummy votes, that is, the number of

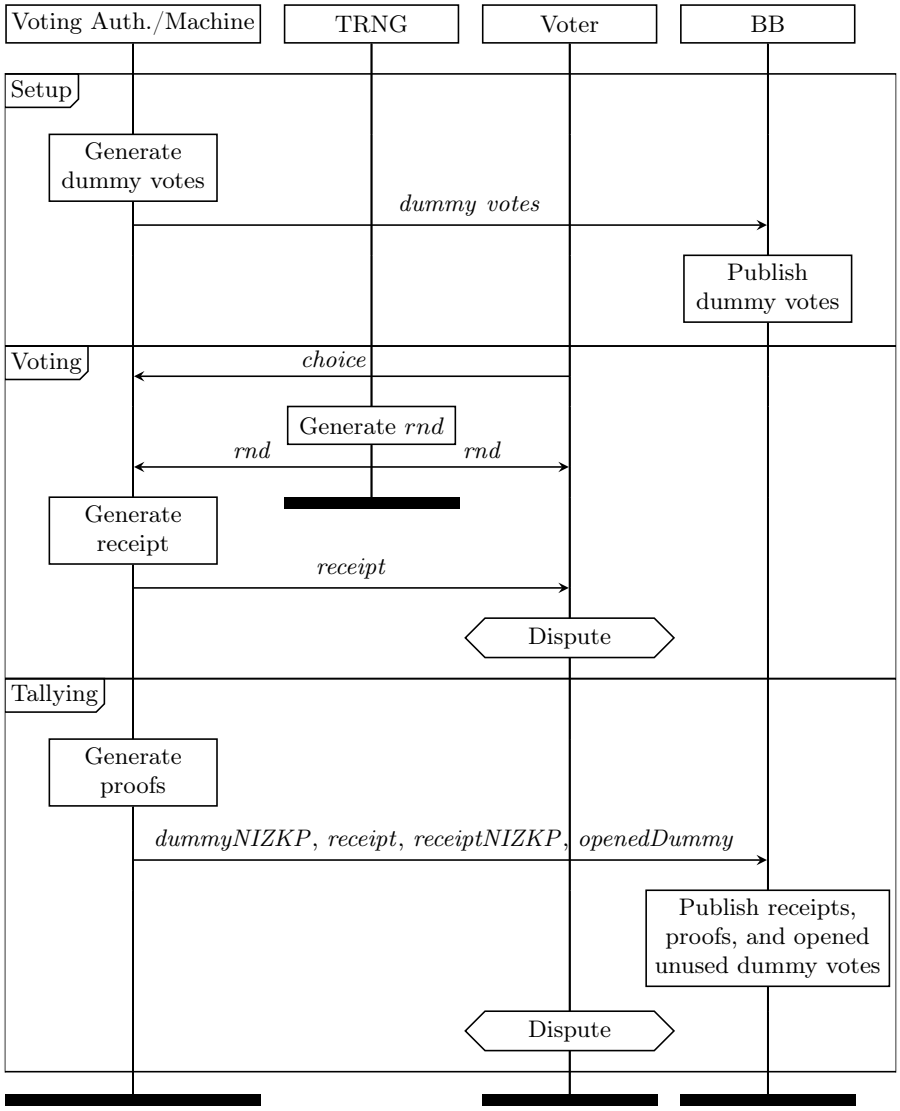
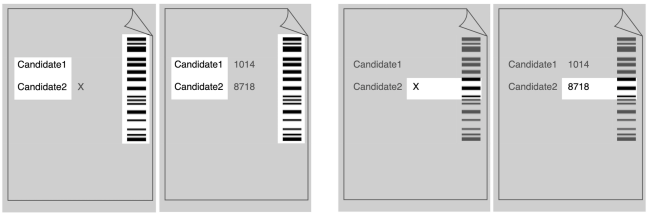


Fig. 1. Message sequence chart of the improved Bingo Voting

registered voters. Thus, the total number of generated dummy votes is equal to the product of the number of voters and the number of candidates.

Inside the voting booth, a display shows the fresh random number generated by the TRNG. The voter records her choice on a paper ballot and feeds it into the voting machine, which is equipped with a scanner-based interface. The



(a) Alignment sleeve (b) Encoding sleeve

Fig. 2. The privacy sleeves for the privacy-preserving dispute resolution at voting

voting machine scans the paper ballot and generates a receipt such that the fresh random number is printed next to the name of the candidate chosen by the voter. Unused dummy random numbers, which the voting authority generated before the voting phase, are instead printed next to any other candidates. The voting machine also prints an identical barcode onto both paper ballot and receipt, and keeps the paper ballot inside a special compartment unless the voter decides to raise a dispute should she receive an incorrect receipt.

In the case of a dispute, the voter can use two different pairs of privacy sleeves to prove that the printed receipt is incorrect, without revealing the way she voted. Each pair of privacy sleeves is to be used with both paper ballot and receipt. The first type of privacy sleeve leaves uncovered candidate names and the barcodes (see Figure 2a) and enables a third party to check whether the candidates are not placed identically in respect to the barcode on the paper ballot and the receipt. The second type of privacy sleeve leaves uncovered the marking area for one candidate on the paper ballot and one row of random numbers on the receipt (see Figure 2b). This enables a third party to check whether there is a discrepancy between the voter choice and the receipt as the printed random number differs from the one displayed on the TRNG.

At tallying, the voting authority publishes the final result of the election along with the following sets of data on an append-only bulletin board

- A non-interactive zero-knowledge proof of correct distribution of dummy votes showing that each candidate gets the same number of dummy votes.
- A non-interactive zero-knowledge proof for each receipt showing that it contains the correct amount of dummy random numbers and that each dummy random number is assigned to the right candidate.
- The list of all printed receipts.
- The list of opened unused dummy votes, which determines how many votes each candidate has received.

Since all the receipts are published, every voter can verify whether their vote is correctly counted. If not, they can raise a privacy-preserving dispute resolution at tallying proving that their receipt has not been published. Morever, any observer can check the correctness of the election outcome by verifying that the tally is indeed the sum of all votes cast.

Properties. The improved Bingo Voting aims at the following properties:

- *Individual verifiability*: a voter can check that the receipt encodes her vote.
- *Privacy-preserving dispute resolution at voting*[1]: a voter can prove that the receipt incorrectly encodes her vote, without revealing her vote.
- *Privacy-preserving dispute resolution at tallying*[2]: a voter can prove that her receipt is not in the bulletin board, without revealing her vote.
- *Global verification*: anyone can prove that the tally is incorrectly computed.
- *Vote privacy*: No one knows how the voter votes.
- *Receipt freeness*: The voter has no evidence proving how she voted.
- *Coercion resistance*: A voter deviating from the intended voting process receives no evidence that may be used to prove how she voted.

The improved Bingo Voting requires a number of trust assumptions to meet the security properties outlined above. The most important are that only eligible voters get access to a voting machine and that each voter casts a single ballot. Also, it is assumed that voters are unobserved as they cast their ballot, which is known as the *voting booth assumption*. Bulletin board (BB) and TRNG are always considered uncorrupted. For vote privacy, it can be assumed that both voting authority and the voting machine can be dishonest as soon as they do not communicate. For receipt freeness and coercion resistance, the voting authority should be uncorrupted, and the voting machine should not be able to communicate with an attacker. In the next section, we analyse the improved Bingo Voting in ProVerif to determine any necessary additional assumptions.

3 Formal Analysis

ProVerif [5] allows one to analyse reachability and equivalence-based properties in the symbolic attacker model. We chose ProVerif mainly because its input language fits well with our approach in modelling the verification and dispute resolution mechanisms. It is also one of the few tools that enable the automated analysis of privacy properties using observational equivalence. The input language of ProVerif is the applied π-calculus [1], which the tool automatically translates to Horn clauses. Cryptographic primitives can be modelled by means of equational theories. An equational theory E describes the equations that hold on terms built from the signature. Terms are related by an equivalence relation $=$ induced by E. For instance, the equation $dec(enc(m, pk(k)), k) = m$ models an asymmetric encryption scheme. The term m is the message, the term k is the secret key, the function $pk(k)$ models the public key, the term enc models the encryption function, and the term dec models the decryption function.

[1] In our formal analysis we separate this property into *dispute resolution at voting*, which checks the correctness of the test as a reachability property, and *vote privacy after a dispute*, which checks vote privacy in terms of observational equivalence.

[2] Since all the receipts are eventually published, vote privacy implies that dispute resolution at tallying is privacy-preserving.

Table 1. Equational theory modelling the improved Bingo Voting

Primitive	Equation
Digital signature	$checksign(sign(m, ssk), spk(ssk)) = m$
Commitment & Dummy vote	$openCommit(com(val, r)) = (val, r)$ $openDummyVote(dvp(com0, com1)) = (com0, com1)$
NIZKP dummy vote	$checkzkp1(cA, cB, dvp(com0A, com(cA, cr0)),$ $dvp(com0B, com(cB, cr1)),$ $zkp1(cA, cB, cr0, cr1, com(cA, cr0), com(cB, cr1) = OK$
NIZKP receipt (candidate A)	$checkzkp2(cA, cB, Rtrg, rX, dvp(com(Rtrg, tr), com(cA, cr0)),$ $dvp(com(rY, r0), com(cA, cr0)), dvp(com(rX, r1), com(cB, cr1)),$ $zkp2(cA, cB, rX, dvp(com(Rtrg, tr), com(cA, cr0)),$ $dvp(com(rX, r1), com(cB, cr1)),$ $cr1, cr0, r1, tr)) = OK$
NIZKP receipt (candidate B)	$checkzkp2(cA, cB, rY, Rtrg, dvp(com(Rtrg, tr), com(cB, cr1)),$ $dvp(com(rY, r1), com(cA, cr0)), dvp(com(rX, r1), com(cB, cr1)),$ $zkp2(cA, cB, rX, dvp(com(Rtrg, tr), com(cB, cr1)),$ $dvp(com(rY, r1), com(cA, cr0)),$ $cr0, cr1, r1, tr)) = OK$

The equational theory for the improved Bingo Voting is described in Table 1. It includes the equations for digital signature (in our case *checksign* returns the signed message only if one uses the correct verification key, and it fails otherwise), Pedersen commitment, dummy vote, and non-interactive zero-knowledge proofs (NIZKP) that prove the correctness of dummy votes and published receipts. To prove the correctness of the dummy votes, the voting authority uses the function *zkp1* showing that the content of the second commitment of each dummy vote is equal to the list of the two candidates cA and cB. The function *zkp2* allows the voting authority to prove that the content of a receipt (cA, cB, $Rtrg$, rX) is identical to the content of the used dummy vote pair $dvp(com(rX, r1), com(cB, cr1))$ and to random number displayed on the TRNG ($Rtrg$), which is hidden into the fresh dummy vote pair $dvp(com(Rtrg, tr), com(cA, cr0))$. An auditor can check both proofs against the dummy vote pairs and the receipts published on the BB.

We specify the processes modelling voting authority, voter, TRNG, and bulletin board into a ProVerif library and reuse it to check each property. This guarantees that all the properties are checked against the same model of the improved Bingo Voting.

3.1 Verification and Dispute Resolution

All the verification and dispute resolution properties of the improved Bingo Voting can be modelled as reachability properties. In line with the verification

approach defined in [24] and [8], we identify the *tests* that decide whether a goal of the improved Bingo Voting fails. We then check that each of the tests meets soundness, completeness, and sufficiency conditions, as outlined in Table 2.

Table 2. $\mathcal{A}(\cdot)$: external attacker; $\mathcal{A}(VA)$: attacker controlling the voting authority ; V: voter instances; V_{test}: voter instance running the test; τ: a trace representing a run of the improved Bingo Voting; $\mathcal{T}$: the set of all traces. BB and TRNG are always honest according to the improved Bingo Voting assumptions.

	Strategy			Condition
	Individual Verification	Dispute Resolution	Global Verification	
(soundness)	$\mathcal{A}(\cdot)$	$\mathcal{A}(V)$	$\mathcal{A}(V)$	$\forall \tau \in \mathcal{T} \mid \mathit{goal}$ holds in $\tau \implies \mathit{test}(\tau) : \texttt{true}$
(completeness)	$\mathcal{A}(VA, V \setminus V_{test})$	$\mathcal{A}(VA, V \setminus V_{test})$	$\mathcal{A}(VA, V)$	$\forall \tau \in \mathcal{T} \mid \mathit{test}(\tau) : \texttt{true} \implies \mathit{goal}$ holds in τ
(sufficiency)	$\mathcal{A}(VA, V \setminus V_{test})$	$\mathcal{A}(VA, V \setminus V_{test})$	$\mathcal{A}(VA, V)$	$\exists \tau \in \mathcal{T} \mid \mathit{test}(\tau) : \texttt{false}$

Soundness guarantees that if the goal holds, then the test always succeeds. For dispute resolution and global verification, it means that an honest voting authority should never be blamed by any test. Note that individual verification requires a different verification strategy than dispute resolution, as the former considers no inside attacker since the verification is based on (the honest) voter's knowledge of the way she voted. In fact, individual verification does not give the voter a way to prove that the voting authority misbehaved. Conversely, in case of dispute resolution or global verification, in which tests are decided upon public information, we consider no honest voters, who may try to feed the tests with incorrect information. We prove that an honest voting authority cannot be unfairly blamed.

Completeness guarantees that whenever a test does not blame the voting authority, then the goal holds. Note that this is logically equivalent to saying that whenever a goal does not hold, then the test blames the voting authority. Thus, we check that a dishonest voting authority cannot feed the tests with incorrect information so that the test succeeds but the goal fails. The verification strategy for completeness regarding global verification is different from the one regarding individual verification and dispute resolution: in principle, global verification should hold even if all voters are dishonest as any election observer can run the test. However, as we shall see later, global verification can provide only guarantees up to dishonest voters.

While soundness and completeness are conforming to [8], we introduce a third condition, *sufficiency*, which formalises that the misbehaviour of selected parties alone is sufficient to make the test fail. Without this condition, a protocol that does not permit any violation might still fulfil criteria to blame a party [23].

The conditions described in Table 2 show that the main difference between individual verification and dispute resolution boils down to be the verification strategy for checking soundness. Thus, a protocol that is dispute free for a specific

goal is also individually verifiable for that goal. This is the case for individual verification and dispute resolution at voting for the improved Bingo Voting.

Due to space limitations, we only discuss the details of the dispute arising due to the global verification test in the improved Bingo Voting. The ProVerif code for all properties is available in [16]. Global verification enables any observers, including those who have not participated in the election at all, to verify the correctness of the election outcome. Global verification ensures that all candidates have received the same number of dummy votes and that for each receipt all but one candidate lose one dummy vote. This is the most complex test in improved Bingo Voting and requires the voting authority to release some information. The original paper presenting the improved Bingo Voting does not detail a specific algorithm for the test, thus we propose the test as defined in Algorithm 1. Our test considers two candidates, cA and cB. The input data of the test is published by the voting authority on the bulletin board.

We can define the goal for global verification $goal_{gv}$ as follows. Let us consider the set of all voters V of type $\mathcal{V}$, the set of voters' choices C of type $\mathcal{C}$, the set of candidates K of type $\mathcal{K}$, the set of honest voters $V_h \subseteq V$, and the set of choices of honest voters $C_h \subseteq C$. Let us now consider the relation `Choice` as the votes accepted by the bulletin board according to the published receipts, linking voters to their choices such that $\texttt{Choice} \subseteq V \times C$. Similarly, consider the relation `Choice_h` that links honest voters to their choices such that $\texttt{Choice}_\texttt{h} \subseteq V_h \times C_h$. Let `Count`: $(\mathcal{V} \times \mathcal{C}) \rightarrow (\mathcal{K} \times \mathbb{N})$ be an ideal counting function that returns the number of votes for each candidate. We can say that the $goal_{gv}$ *holds in* τ if $\texttt{Choice}_\texttt{h} \subseteq \texttt{Choice}$ and the election result is equal to `Count`(`Choice`).

All our proofs consider an unbounded number of voters. While ProVerif can automatically prove sufficiency for global verification, it is not possible to prove soundness and completeness since, according to Algorithm 1, we need to iterate over all receipts, but ProVerif does not support loops. We thus prove the base case in ProVerif, in which we consider only one published receipt. Then, we provide a manual induction proof that generalises the ProVerif results to the general case with an arbitrary number of published receipts.

ProVerif proves soundness and completeness when only one published receipt is considered. To prove the general case that considers an unbounded number of published receipt, it is necessary to show that

$$test(\tau) : \texttt{true} \Leftrightarrow \texttt{Choice}_\texttt{h} \subseteq \texttt{Choice} \wedge \text{ the election results is equal to } \texttt{Count}(\texttt{Choice})$$

It can be assumed that the number of published receipts is equal to the number of the published dummy votes and of the opened dummies. Any observer can check that these numbers coincide by looking at the bulleting board.

Theorem 1. *Let* $test_k(\cdot)$ *be the test applied to an execution that considers* k *receipts; let* $test_k(\cdot) \rightarrow^*$ `true` *denote the test that outputs* `true` *after some steps; let* τ *be a trace that has* n *receipts; let* τ_j *be a version of* τ *that only considers the* j^{th} *receipt that is associated with a honest voter* i_j *and corresponding choice* c_j*. For soundness, we prove that*

$$\forall 1 \leqslant i \leqslant n : test_1(\tau_j) \rightarrow^* \texttt{true} \Rightarrow (i_j, c_j) \in \texttt{Choice} \wedge \textit{ the election results is equal to } \texttt{Count}(\texttt{Choice})$$

For completeness, we prove that

$$\forall 1 \leqslant i \leqslant n : (i_j, c_j) \in \texttt{Choice} \wedge \textit{ the election results is equal to } \texttt{Count}(\texttt{Choice}) \Rightarrow test_1(\tau_j) \rightarrow^* \texttt{true}$$

Proof. $test_n(\tau)$ checks all the receipts, dummy votes, and proofs published in the bulletin board as defined in Algorithm 1. Similarly, the test $\forall 1 \leqslant j \leqslant n : test_1(\tau_j)$ does the same check for the j^{th} entry in the bulletin board. It follows that

$$\begin{array}{c} test_n(\tau) \rightarrow^* \texttt{true} \\ \Downarrow \\ \forall 1 \leqslant j \leqslant n : test_1(\tau_j) \rightarrow^* \texttt{true} \\ \Downarrow_{(by\ ProVerif)} \\ \forall 1 \leqslant j \leqslant n : (i_j, c_j) \in \texttt{Choice} \wedge \text{ the election results is equal to } \texttt{Count}(\texttt{Choice}) \\ \Downarrow \\ \texttt{Choice}_\texttt{h} \subseteq \texttt{Choice} \wedge \text{ the election results is equal to } \texttt{Count}(\texttt{Choice}) \end{array}$$

which proves soundness also for the general case.

$$\begin{array}{c} \texttt{Choice}_\texttt{h} \subseteq \texttt{Choice} \wedge \text{ the election results is equal to } \texttt{Count}(\texttt{Choice}) \\ \Downarrow \\ \forall 1 \leqslant j \leqslant n : (i_j, c_j) \in \texttt{Choice} \wedge \text{ the election results is equal to } \texttt{Count}(\texttt{Choice}) \\ \Downarrow_{(by\ ProVerif)} \\ \forall 1 \leqslant j \leqslant n : test_1(\tau_j) \rightarrow^* \texttt{true} \\ \Downarrow \\ test_n(\tau) \rightarrow^* \texttt{true} \end{array}$$

which proves completeness also for the general case.

3.2 Privacy

Like in the verification of the verifiability and dispute resolution properties, we prove privacy by encoding the protocol into one ProVerif library – with a few modifications compared to the previous one – and then check privacy of different setups. The main practical change required for proving privacy is to remove the channel that voter, voting authority, and bulletin board use to feed the test with the evidence, and let the attacker read all public data and impersonate misbehaving parties, including an unbounded number of dishonest voters. As the improved Bingo Voting requires that voters are unobserved as they cast their vote, all communications between honest voters, the voting machine, and the TRNG are done over private channels.

In the privacy setting, we observe two voters in particular, hence the bulletin board needs to shuffle the votes specifically to avoid trivial attacks to privacy. We check vote privacy, receipt freeness, and coercion resistance considering an honest voting authority. We also check vote privacy, and vote privacy of disputed

Algorithm 1: Global Verification

Data: $cA, cB, receipt : (cx, cy, rx, ry, barcode), zkp1, dummy\_vote, zkp2,$ $new\_dummy, opened\_dummy : (ca, ra)$

foreach *receipt* in BB **do**
  if $\texttt{checkzkp1}(cA, cB, dummy\_vote, zkp1) = \texttt{OK}$ **then**
    if $cx = cA \wedge cy = cB \wedge$
     $\texttt{checkzkp2}(cA, cB, rx, ry, new\_dummy, dummy\_vote, zkp2) = \texttt{OK} \wedge$
     $rx \neq ry \wedge rx \neq cx \wedge rx \neq cy \wedge ry \neq cx \wedge ry \neq cy \wedge$ **then**
      if $\texttt{dummy}(ca, ra) \in dummy\_vote \wedge ra \neq rx \wedge ra \neq ry$ **then**
        return `true`
      else
        return `false`
    else
      return `false`
  else
    return `false`

receipt at the voting phase consider a dishonest voting authority. First, we check whether vote privacy holds in the improved Bingo Voting. Specifically, we check that if two honest voters swap their votes in two different runs of the protocol then the attacker cannot distinguish the two resulting systems as in [22]:

$$S[V_A\{^a/_v\} \mid V_B\{^b/_v\}] \approx_l S[V_A\{^b/_v\} \mid V_B\{^a/_v\}]$$

Similarly, we check whether vote privacy holds after a dispute at the voting phase. We let the honest voters reveal the fresh random number obtained by the trusted random number generator and the dummy random number on the receipt that is revealed by the privacy sleeve.

To check receipt freeness, we additionally let the voters publish their receipts on the public channel, and verify that privacy still holds:

$$S[V_A\{^a/_v\} \mid V_B\{^b/_v\}] \approx_l S[V' \mid V_B\{^a/_v\}]$$

where V' is a process such that $V'^{\backslash \mathsf{out}(chc,\cdot)} \approx_l V_A\{^b/_v\}$, i.e. V' is the process that acts like V_A voting for candidate B, but pretends to cooperate with the attacker.

Finally, to check whether the scheme is coercion resistant, we set up the protocol so that one of the voters receives the instruction on how to vote from the attacker and then provides the receipt to the attacker. We check that

$$S[C[V_A\{^?/_v\}^{c_1,c_2}] \mid V_B\{^a/_v\}] \approx_l S[C[V'] \mid V_B\{^c/_v\}]$$

where $V_A\{^?/_v\}^{ch,a}$ is the coerced voter process that votes for candidate B, no matter their original intention, reveals all its private information to the attacker via channels c_1, c_2, while V_B is the other voter process intended to balance the resulting votes, that is, if V_A votes for candidate A, then V_B votes for candidate B and vice versa. Note that with the setup described here there is a trivial attack, which only appears in the model, as the bulletin board should not reveal whether the votes were swapped or not. In practice, this is done by shuffling.

Thus, we let the bulletin board swap the order of published ballots if and only if the voters actually swap their choice following the attacker's instruction.

3.3 Findings

ProVerif proves individual verification and both dispute resolution at voting and at tallying automatically. It also proves global verification for one receipt, then we provide a manual inductive proof for the unbounded case. The outcome of our analysis shows that the improved Bingo Voting meets some properties only if one makes additional assumptions to the ones already defined in [6,18]. The additional assumptions are reported in Table 3. For dispute resolution at voting, we need to assume that the test does not blame the voting authority if the barcode printed on the paper ballot does not match with the one printed on the receipt. This avoids an attack due to a dishonest voter handing her receipt to another voter [8]. Without this assumption, the latter, isolated in the voting booth, may swap the receipt printed by the voting machine with the ones handed by the dishonest voter, leading to a successful blaming of the voting authority.

We also need to make additional assumptions for proving global verification. As already noted by in [24], it is only possible to have global verification up to the votes of dishonest voters since a dishonest voting authority can alter votes cast by such voters without being detected. Moreover, we found that honest voters should check that their receipts are well-formed at voting and at tallying, and raise disputes otherwise.

As regards privacy properties, we found that vote privacy, receipt freeness, and coercion resistance hold if the voting authority is honest and the voting machine cannot decide which dummy vote should be assigned to which receipt. This can be achieved by prearranging dummy votes in *clusters* [18], which limits the voting machine's choice on selecting the dummy votes. Considering two candidates, each cluster contains two dummy votes, one per candidate. The voting authority publishes the clusters in the same order in which the voting machine uses them for the receipts. The voting authority can prove in zero-knowledge that each receipt used the dummy votes from the expected cluster. However, the verification process of the correct order of clusters requires that the bulletin board publishes the receipts as they are issued. Revealing the order in which the receipts are issued may not be acceptable for many elections. In fact, ProVerif finds that if the bulletin board does not randomly shuffle the receipts before publishing them, the voting authority can easily break vote privacy by just looking at the order of voters, which is normally available in the voter registration record at the polling place. Thus, for vote privacy, it is not enough assuming that a dishonest voting authority does not communicate with a dishonest voting machine as suggested in [18]. We need to assume that at least either the voting authority or the voting machine is honest.

ProVerif can prove that vote privacy holds after a dispute if the disputed receipt is not published on the bulletin board and the dummy vote corresponding to the dummy random numbers revealed by the privacy sleeve is not opened. In fact, if the receipt is published, vote privacy does not hold any more because the

Table 3. The additional assumptions required in the improved Bingo Voting respect to the ones stated in [6,18], according to the outcome of our formal analysis

Property	**Assumptions in [6,18]**	**Additional assumptions**
Individual Verification	Honest *TRNG* and *BB*	–
Dispute Resolution at voting	Honest *TRNG* and *BB*	Do not blame the *VA* if barcodes are different
Dispute Resolution at tallying	Honest *TRNG* and *BB*	–
Global Verification	Honest *TRNG* and *BB*	Up to dishonest voters. Voters check and dispute incorrect receipts at voting and at tallying
Vote Privacy if dispute at voting	Honest *TRNG* and *BB*. *VA* has no access to the voting machine	Honest *VA*
Vote Privacy	Honest *TRNG* and *BB*. *VA* has no access to the voting machine	Honest *VA* or voting machine
Receipt Freeness	Honest *TRNG*, *BB*, *VA*, and voting machine	–
Coercion Resistance	Honest *TRNG*, *BB*, *VA*, and voting machine	–

random number generated by the TRNG is revealed during the dispute. If the dummy vote is opened, vote privacy does not hold as well because this would reveal one of the candidates not chosen by the voter. However, we found that not revealing the receipt and not opening the dummy vote after a dispute might break vote privacy.

Privacy attack due to dispute resolution. Let us consider the scenario with two candidates in which a voter mistakenly disputes a valid receipt at voting. This vote should not be counted because the receipt is not published. Also, we require that a pair of dummy votes that are not in any receipts should not be opened

- The *disputed* dummy vote containing the disputed dummy random number associated with the candidate not chosen by the voter printed on the receipt.
- A dummy vote associated with the candidate chosen by the voter so that the disputed receipt is not counted at tallying.

Then, the voting authority should prove in zero-knowledge that the pair of dummy votes contain the list of the candidates. However, we observe that *any* pair of dummy votes containing the list of the candidates can serve for such proof since the corresponding receipt will not be published. Thus, a dishonest voting

machine can signal a different dummy random number to the voting authority and print the disputed dummy random number again into another receipt, which will be published on the bulletin board. This would reveal how the disputing voter voted, breaking vote privacy. If one considers a dishonest voter, this attack is even more harmful. A dishonest voter can dispute a vote on purpose to learn how another voter voted since the dishonest voter knows the disputed dummy random number.

Note that the voting machine does not need to communicate with the dishonest voter to break vote privacy of another voter, and that this attack works even considering an honest voting authority. Of course, the attack is not possible if one considers an honest voting machine but there would not be need of dispute resolution at all in the first place if one makes such an assumption.

None of the papers presenting the improved Bingo Voting describes what happens after a dispute. Prearranging dummy votes may mitigate the attack at the cost of assuming an honest voting authority. Another possible mitigation to such an attack might be to allow voters who dispute their votes to revote. Revoting requires to generate additional dummy votes. The total amount of needed dummy votes should be the double of the original amount in order to avoid denial of voting attacks. However, this is a partial solution as it would not mitigate attacks due to dishonest voters.

4 Related Work

Several voting schemes have considered notions of dispute resolution or related properties. The FOO protocol [14] is one of the first voting schemes that enables voters to prove certain frauds due to a dishonest voting authority. Pret â Vòter [27] and vVote [13] provide some dispute resolution and accountability guarantees as a voter can use invalid proof and a ballot confirmation check as evidence. Remotegrity [31], Scantegrity II [9], and Scantegrity III [29] detail dispute resolution processes that allow voters to file disputes in case of incorrect designated ballots or *confirmation codes*, which are invisible random codes preprinted on the ballots. sElect [25] features a fully automated verification procedure that performs cryptographic checks without requiring any voter interaction. The procedure is capable to single out a specific misbehaving party and producing the necessary evidence of the misbehaviour. Schoenmarkers [28] and Kiayias and Yung [20] design dispute-free voting schemes, whose aim is to neutralise faults rather than providing mechanisms to address them. Some of the above protocols have been formally checked for accountability and/or privacy properties. However, no formal analysis has been done to check whether disputes leak any information regarding how the voter voted.

Prior works on the formalisation of dispute resolution and related properties, such as accountability, include the seminal work by Küsters et al. [24], who advance accountability notions in the symbolic and computational models. Moreover, they provide an analysis of accountability and coercion resistance [26] of the original Bingo Voting scheme. Bruni et al. [8] propose formal definitions

of accountability that are amenable to automated verification. One of their case studies is the improved Bingo Voting, which they analyse up to the voting phase, finding that it does not meet dispute resolution at voting. In contrast, we find that, if the dispute resolution test does not blame the voting authority when the barcodes are different between the paper ballot and the receipt, then the improved Bingo Voting achieves that property. Künneman et al. [23] give verification conditions that imply accountability based on *counterfactual relations*, capturing what actually happened to what could have happened. Basin et al. [2] proposed a definition of dispute resolution for voting requiring that voters get evidence that their ballot is incorrectly recorded before the end of of the election.

The notions of individual verifiability and universal verifiability have been extensively studied in voting [11,4,19,10,3]. Kremer et al. [21] formalised both individual and universal verifiability in the applied pi-calculus, including the requirement of *eligibility verifiability*, which expresses that auditors can verify that each vote in the election result was cast by a registered voter, and there is at most one vote per voter. Smyth et al. [30] used ProVerif to check verifiability in three voting protocols expressing the requirements as reachability properties. Gallegos-Garcia et al. [15] studies how to achieve verifiability without any trust assumptions. Giustolisi et al. [17] observe that privacy-preserving verifiability can be achieved using non-interactive zero-knowledge proofs and functional encryption techniques. More recently, Cortier and Lallemand [12] have shown that a voting scheme that does not meet individual verifiability fails to achieve vote privacy, when one considers the same trust assumptions. This line of work opens up to interesting questions on how stronger properties such as dispute resolution and coercion resistance relate.

5 Conclusion

Dispute resolution mechanisms are essential components of a voting scheme, enabling the correctness of an election outcome. They can provably expose a misbehaving voting authority, hence deterring it by doing so. However, dispute resolution is useless if it is not triggered when it should be, and voters should not have to choose to either raise a dispute or keep their vote private. In this work, we have looked at the privacy-preserving dispute resolution mechanisms described in the improved Bingo Voting.

The formal analysis of the improved Bingo Voting allows us to identify precisely the necessary assumptions that enable the scheme to meet all the stated properties. It is found that global verification, which enables any observer to dispute the correctness on an election, cannot be achieved without dispute resolution both at voting and at tallying. To the best of our knowledge, it is an open question whether this is just for the improved Bingo Voting or it is a requirement for any voting scheme.

It is also found that assuming that the voting authority has not illegitimate access to the voting machine is not enough to guarantee vote privacy: either the voting authority or the voting machine must be honest at least. However,

it is found that dispute resolution at voting can be achieved only assuming an honest voting authority as prearranging dummy votes would enable the voting authority to link votes to voters.

The results of this work also show that designing privacy-preserving dispute resolution mechanisms with minimal trust assumptions is not a trivial task in voting. The voting booth assumption should ideally be the sole assumption made in a voting scheme. Also, the details of the aftermath of a dispute resolution procedure in voting need to be described and thought with the same precision and care as are the *standard* voting procedures. For the improved Bingo Voting, we observe that, while cancelling an election due to a dispute is not an option, allowing voters who wrongly contest a receipt to revote mitigates an attack due to a dishonest voting machine. However, it does not help against a voting machine colluding with a dishonest voter.

Other voting schemes might achieve privacy-preserving dispute resolution with fewer assumptions than the improved Bingo Voting. With this work, we stress the importance of detailing the aftermath of disputes and aim at stimulating the voting community to make similar analyses to other voting schemes.

Acknowledgments. We are grateful to Rasmus Dilling Møller and Sean Wachs for helping out with the privacy analysis of an early model of Bingo Voting.

References

1. Abadi, M., Fournet, C.: Mobile values, new names, and secure communication. In: POPL. pp. 104–115. ACM, New York (2001)
2. Basin, D.A., Radomirovic, S., Schmid, L.: Dispute resolution in voting. CoRR **abs/2005.03749** (2020), `https://arxiv.org/abs/2005.03749`
3. Benaloh, J.: Verifiable Secret-Ballot Elections. Ph.D. thesis, Yale University (December 1996)
4. Benaloh, J., Tuinstra, D.: Receipt-free secret-ballot elections (extended abstract). In: STOC. pp. 544–553. ACM (1994)
5. Blanchet, B.: An efficient cryptographic protocol verifier based on prolog rules. In: CSFW. pp. 82–96. IEEE Computer Society (2001)
6. Bohli, J.M., Henrich, C., Kempka, C., Muller-Quade, J., Rohrich, S.: Enhancing electronic voting machines on the example of bingo voting. IEEE Transactions on Information Forensics and Security pp. 745–750 (2009)
7. Bohli, J.M., Müller-Quade, J., Röhrich, S.: Bingo voting: Secure and coercion-free voting using a trusted random number generator. In: VOTE-ID. pp. 111–124. Springer (2007)
8. Bruni, A., Giustolisi, R., Schuermann, C.: Automated analysis of accountability. In: Information Security Conference. pp. 417–434. Springer (2017)
9. Carback, R., Chaum, D., Clark, J., Conway, J., Essex, A., Herrnson, P.S., Mayberry, T., Popoveniuc, S., Rivest, R.L., Shen, E., Sherman, A.T., Vora, P.L.: Scantegrity II municipal election at takoma park: The first e2e binding governmental election with ballot privacy. In: USENIX Conference on Security. USENIX (2010)

10. Cohen, J., Fischer, M.: A robust and verifiable cryptographically secure election scheme (extended abstract). In: FOCS. pp. 372–382. IEEE (1985)
11. Cortier, V., Galindo, D., Küsters, R., Müller, J., Truderung, T.: SoK: Verifiability notions for e-voting protocols. In: IEEE Symposium on Security and Privacy. pp. 779–798 (2016)
12. Cortier, V., Lallemand, J.: Voting: You can't have privacy without individual verifiability. In: CCS. pp. 53–66. ACM (2018)
13. Culnane, C., Ryan, P.Y.A., Schneider, S., Teague, V.: vvote: a verifiable voting system (DRAFT). CoRR **abs/1404.6822** (2014)
14. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: AUSCRYPT, pp. 244–251. Springer (1992)
15. Gallegos-García, G., Iovino, V., Rial, A., Rønne, P.B., Ryan, P.Y.A.: (universal) unconditional verifiability in e-voting without trusted parties. CoRR **abs/1610.06343** (2016), `http://arxiv.org/abs/1610.06343`
16. Giustolisi, R., Bruni, A.: The ProVerif code used to verify the Improved Bingo Voting. `https://itu.dk/people/rosg/code/evoteid20code.tar.gz`
17. Giustolisi, R., Iovino, V., Lenzini, G.: Privacy-preserving verifiability - A case for an electronic exam protocol. In: SECRYPT. pp. 139–150. SciTePress (2017)
18. Henrich, C.: Improving and Analysing Bingo Voting. Ph.D. thesis (2012). https://doi.org/10.5445/IR/1000030270
19. Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. In: EUROCRYPT. pp. 539–556. Springer (2000)
20. Kiayias, A., Yung, M.: Self-tallying elections and perfect ballot secrecy. In: Public Key Cryptography. pp. 141–158. Springer (2002)
21. Kremer, S., Ryan, M., Smyth, B.: Election verifiability in electronic voting protocols. In: ESORICS. pp. 389–404. Springer (2010)
22. Kremer, S., Ryan, M.: Analysis of an electronic voting protocol in the applied pi calculus. In: Programming Languages and Systems. pp. 186–200. Springer (2005)
23. Künnemann, R., Esiyok, I., Backes, M.: Automated verification of accountability in security protocols. In: CSF. pp. 397–413. IEEE (2019)
24. Küsters, R., Truderung, T., Vogt, A.: Accountability: definition and relationship to verifiability. In: CCS. pp. 526–535. ACM (2010)
25. Küsters, R., Müller, J., Scapin, E., Truderung, T.: select: A lightweight verifiable remote voting system. In: CSF. pp. 341–354. IEEE (2016)
26. Küsters, R., Truderung, T., Vogt, A.: A game-based definition of coercion-resistance and its applications. In: CSF. pp. 122–136. IEEE (2010)
27. Ryan, P.Y.A., Bismark, D., Heather, J., Schneider, S., Xia, Z.: PrÊt À voter: a voter-verifiable voting system. IEEE Transactions on Information Forensics and Security pp. 662–673 (2009)
28. Schoenmakers, B.: A simple publicly verifiable secret sharing scheme and its application to electronic. In: CRYPTO. pp. 148–164. Springer (1999)
29. Sherman, A.T., Fink, R.A., Carback, R., Chaum, D.: Scantegrity III: automatic trustworthy receipts, highlighting over/under votes, and full voter verifiability. In: Shacham, H., Teague, V. (eds.) EVT/WOTE. USENIX (2011)
30. Smyth, B., Ryan, M., Kremer, S., Mounira, K.: Towards automatic analysis of election verifiability properties. In: ARSPA-WITS. pp. 146–163. Springer (2010)
31. Zagórski, F., Carback, R.T., Chaum, D., Clark, J., Essex, A., Vora, P.L.: Remotegrity: Design and use of an end-to-end verifiable remote voting system. In: Applied Cryptography and Network Security. pp. 441–457. Springer (2013)

Ballot Logistics: Tracking Paper-based Ballots Using Cryptography

Kristian Gjøsteen[1], Clémentine Gritti[2], and Kelsey N. Moran[1]

[1]NTNU – Norwegian University of Science and Technology, Trondheim, Norway
{kristian.gjosteen,kelsey.n.moran}@ntnu.no
[2]UC – University of Canterbury, Christchurch, New Zealand
clementine.gritti@canterbury.ac.nz

Abstract. A paper-based voting system can be seen as a non-trivial, security-critical logistics system for transporting ballots from voting precincts to where they will be counted. The security of this system is usually wholly physical in nature, and the modern practice of election observation is quite good at producing evidence for fraud in the logistics part of the voting system.
We initiate the study of how to use cryptography to increase the robustness of the logistics part of a paper-based voting system. We define security notions and provide concrete constructions, one of which is based on an existing system for monitoring supply chains. We also discuss how to realise our schemes in practice.

Keywords: Paper-based voting · Ballot tracking · Voting system design

1 Introduction

Abstractly, a paper-based voting system is a very simple affair. The ballots are placed in a ballot box and then counted. In reality, a paper-based voting system is a non-trivial, security-critical logistics system, transporting ballots from the voting booths to centralised counting centres, usually through multiple intermediates. In some cases, these intermediates are also responsible for doing partial counts.

Since such a system is security-critical, studying ways to detect and prevent fraud in this system is important, and election observers have become quite good at this. One example is that statistics reported by the various players in this logistics system offer opportunities to detect election errors and fraud. In mature democracies, such a monitoring system often discovers logistics mistakes such as losing a ballot box.

However, even with good election observation and strong physical security, there is still room for various attacks on election frameworks[1], in particular by

[1] See for example https://mz.usembassy.gov/press-101819/ for a recent example of potential weaknesses.

well-resourced and well-connected adversaries. Studying ways to apply cryptographic technology to make this logistics system more resistant to manipulation even by insiders therefore makes sense.

In this work, we follow up on the idea of treating the voting system as a logistics system. We apply techniques from supply chain management to the voting system, and arrive at several useful techniques for detecting some attacks on voting systems.

We first define a model for the logistics part of the paper-based voting system (shown in Figure 1), based on a collection of voting precincts where ballots enter the system and a collection of intermediates transporting the ballots to a central manager that will eventually tally the ballots. In addition, we have a central party managing and distributing key material (which would typically require the use of more sophisticated cryptographic techniques such as multi-party computation or similar, but this is not the focus of this work).

The parties record certain information on the ballots as they pass through the system. When the central manager receives a ballot, it uses this information to decide if the recorded information matches the ballot's supposed path through the system.

The adversary is able to corrupt any subset of the players (except the issuer manager). If a voting precinct is corrupt, the adversary may insert any number of ballots, and better logistics will not prevent that. But logistics should prevent corrupt intermediates from doing anything to the flow of ballots from precincts to manager, except lose ballots.

We formalise our security notion by requiring that the flow of ballots (i.e. the path the ballots have followed) inferred by the manager should match the partial path, or "sub-flow", of the ballots that are observed by honest intermediates.

Note that this security notion still allows ballots to be lost, which does not guarantee election integrity. However, given that an honest manager accepts the ballots, integrity follows from a simple counting argument: If the number of ballots that the manager receives equals the number of ballots inserted into the system, the logistics system has behaved correctly. Our goal is not to prevent fraud or manipulation but to detect it.

We study two different schemes. A straightforward scheme based on digital signatures, and a better, but more involved scheme based on Tracker [2], a system for monitoring supply chains.

1.1 Physical Realisation

This work is about applying cryptographic techniques to a physical system. We describe and study cryptographic systems, but in order to show that this work is plausible, we must discuss their physical implementation.

The simplest option is to record information directly on the ballot. This is simple, but it also introduces additional logistics and security problems, since a box of ballots must be opened (thereby possibly invalidating physical security measures) in order to record information.

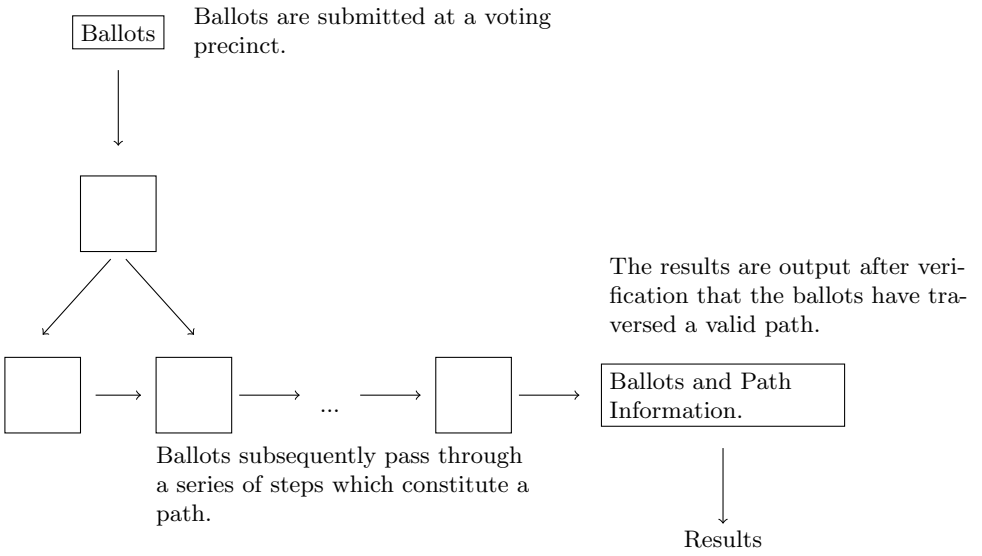


Fig. 1. The conceptual model for a ballot logistics system.

A more realistic option is to use cheap RFID tags affixed to the physical paper ballots. This simplifies the logistics, since the required information can be recorded on all the ballots in a locked box, without opening the box.

In this case, it is vitally important that a RFID tag cannot be correlated with an identifiable voter. In many voting systems, ballots are stamped before being put in a ballot box. RFID tags can be implemented as stickers, so affixing a tag to the physical paper ballot could be done as part of the ballot submission process, perhaps at the same time as stamping the ballot, before the voter places the ballot in a ballot box. Note that not only must it be impossible to correlate the RFID tag to the voter, but it must be obvious to the voter that the tag cannot be correlated. There are many ways this correlation could jeapordize privacy, necessitating care in the design of how the physical ballots will be handled once cast. This suggests that it is probably better to physically separate the authentication and the RFID affixing processes (for instance affixing the RFID tags only after the ballots are taken out of a ballot box).

Note that cheap RFID tags have very limited storage, so if two solutions are otherwise equal, the one that minimizes the required storage space should be preferred.

To illustrate this we give a short description of one of our proposed protocols. The protocol we describe utilizes a system known as Tracker [2] which is built around the use of RFID tags to track goods through a supply chain. Each ballot is initially affixed with an RFID tag and each election worker who will handle the ballots at a location is given a key. A manager, an election official with authority to verify the integrity of a ballot's path, is given these keys as well. With these keys and knowledge of which path the ballots should take the manager is able to

compute what the resulting signature should be. The ballots then move through the locations until reaching the manager at a checkpoint where the manager can then compare the signature on each ballot to the expected signature based on their own calculations. Given a match, the manager can then conclude that the ballots have traversed the correct path.

1.2 Related Work

In this paper, we focus on voting systems that use paper-based ballots. In general, such design requires voters to go to polling stations, rather than voting online, from their home computer for instance.

The natural approach to securing paper-based voting systems is to go for a full-blown cryptographic voting systems, such as punched card systems, optical scan systems, scratch-card voting systems and digital-pen systems [5–7,9]. Their use has been limited due to security issues, deployment difficulties and, most of all, usability issues [1]. This suggests that such voting systems are not yet ready for deployment, and that creating such systems is currently beyond the state of the art.

We do not intend to propose a new paper-based ballot voting design, but rather to develop a system enabling to track paper-based ballots that are already available on the market. Our goal is therefore not to create an end-to-end secure voting system, but rather to make existing systems more robust. Such a limited goal will provide a usable increase in security. We are not aware of such systems in the literature.

1.3 Road Map

In Section 2, we describe the model for tracking ballots in a paper-based voting system. In Section 3, we recall the definition of digital signature schemes and describe Tracker [2]; both are useful to describe our voting solutions. In Section 4, we present our two voting systems that enable tracking of paper-based ballots. In Section 5, we discuss pros and cons of our two solutions and possible future work.

2 Tracking Model

There are three types of entities involved in our tracking model:

- **Election Issuer I:** This central party prepares the ballots to be deployed into the voting chain.
- **Election Officials O_k:** An election official is an intermediate at a given location, represented by vertex v_k, which interacts with the ballots in some capacity at this location.
- **Election Manager M:** The central manager receives the ballots at the checkpoint, represented by the vertex v_l, in the chain and checks the signature(s) on each ballot to ensure their validity. In doing so, it is verified that each ballot has passed through a valid path in the chain so far.

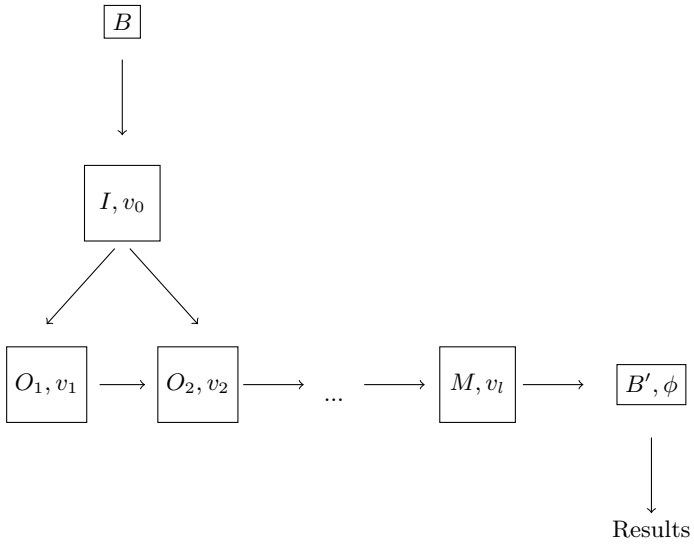


Fig. 2. Tracking model

We model the ballots in our system using two parts: The actual ballot and some associated data storage space. The entities are organised using a directed graph, $\mathbb{G} = (V, E)$, comprised of vertices, V, and edges, E. Each vertex, $v_i \in V$ represents a step in the voting chain, a physical location the ballots will pass through. At each step, v_i, there is an associated election official, O_i, who will attest that the ballots have passed through their step by updating the data stored on the ballot. The aggregate of these updates is what we refer to as a pathmark, ϕ, which is an encoded form of the ballot path. Each directed edge $e \in E, e := \overrightarrow{v_i v_j}$, from vertex v_i to v_j, indicates that it is possible for ballots to move from v_i to v_j. If movement from v_i to v_j is invalid, then $\overrightarrow{v_i v_j} \notin E$. The election official O_j interacts with ballots when they are moved from v_i to v_j.

The issuer I is associated with the vertex v_0, the only vertex without incoming edges. Note that this placement of the issuer is a technical trick. It does not imply that the ballots pass physically through the issuer, but that the issuer is involved in preparing the ballots, typically before the ballots are cast. A path P is a finite sequence of steps $P = \{v_0, v_1, ..., v_l\}$, the length of the path is l and v_l, the last step in the path, is a checkpoint. After passing through v_l the validity of the path is checked by the central manager M. The set of all valid paths is represented with P_{valid}. Valid paths are those that represent a series of allowed steps through the voting chain, the full chain being composed of a sequence of valid paths, $C = \{p_{valid_1}, p_{valid_2}, ..., p_{valid_z}\}$.

Our tracking model is depicted in Figure 2. B is the set of cast ballots, and B' is the set ballots received by the manager M.

2.1 Tracking Protocol

A tracking protocol consists of three entities and three algorithms. The entities are the election issuer I, a set of the election officials O_k, and the election manager M.

Key Generation Algorithm. The first algorithm takes no input and outputs a secret key and a verifying key pair (sk, vk).

Signature Algorithm. The second algorithm takes as input a message m, a secret key sk, and the current pathmark ϕ_{i-1}, and outputs an updated pathmark ϕ_i.

Verification Algorithm. The third algorithm takes as input a message m, the verifying key vk, a path P, and a pathmark ϕ, and outputs valid or invalid.

2.2 Tracking Game

In this game the experiment, $\mathcal{E}$, simulates the honest players while the adversary, $\mathcal{A}$, controls the corrupt players. The game then simulates the interaction between the honest players and corrupt players as ballots move through a voting chain. The issuer and manager can not be corrupt in this game. We assume there is a secure communication channel between players, meaning the adversary will only be able to observe messages between the corrupt players it controls. We allow the adversary to have control over which ballots and paths will be used.

We outline the game between an adversary, $\mathcal{A}$, and an experiment, $\mathcal{E}$, in Figure 3. As a summary, the steps in the tracking game are the following:

1. $\mathcal{A}$ generates the path and the ballots.
2. $\mathcal{E}$ generates the keys which will be used by each official to form their signature.
3. $\mathcal{A}$ chooses which players it will corrupt.
4. $\mathcal{E}$ distributes the keys. $\mathcal{A}$ receives the keys of the corrupt players.
5. $\mathcal{A}$ inserts the ballots into the voting chain.
6. The ballots move between locations in the chain and the respective honest or corrupt officials, being signed at each location.
7. The ballots reach the end of the chain and are received by the manager M. (Note that is possible some ballots were lost, modified or inserted by corrupt players as they moved through the chain.)
8. The manager M reconstructs the path taken using the pathmark on the ballot.
9. $\mathcal{E}$ checks if the reconstructed path matches the path that was observed by honest players during ballot movement.
10. $\mathcal{E}$ outputs valid or invalid.

As mentioned above, we make the assumption that we have secure communication channels and that the adversary $\mathcal{A}$ only observes messages that are sent

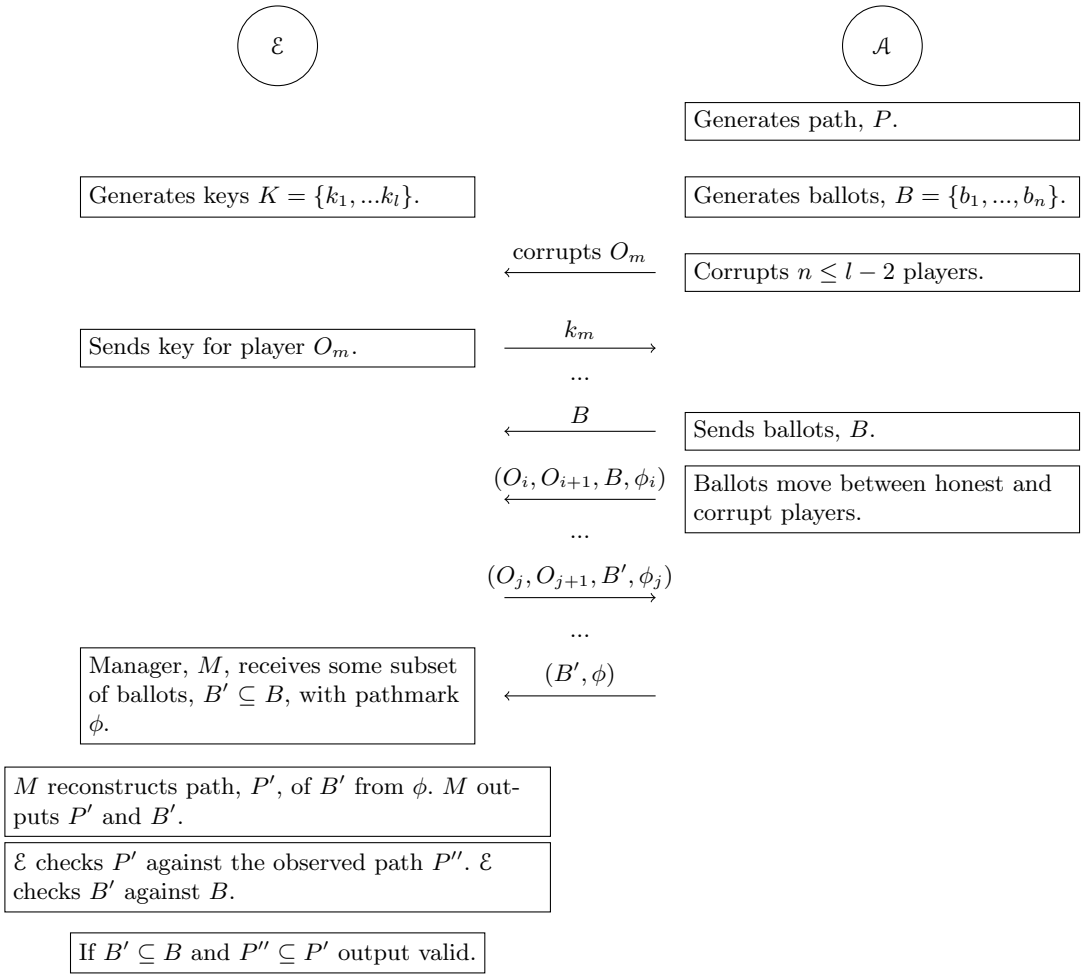


Fig. 3. Tracking Game. Note: m, i, j are all arbitrary indices.

from and to corrupt players. In this game, $\mathcal{A}$ chooses the path the ballots will traverse as well as choosing the set of ballots, B.

The experiment $\mathcal{E}$ generates the keys to be used by players to sign ballots as they pass through the point controlled by those players. As $\mathcal{A}$ corrupts players, it receives the players' corresponding keys.

Once all players, honest and corrupt, have received their keys, the ballots are put into the path and move between the honest and corrupt players. As the ballots move and the players' signatures accumulate, a pathmark, ϕ, is generated. This pathmark ϕ contains information about the path the ballots have traversed. At the end of the ballot movement, the manager, M, receives some set, B', of ballots as well as the pathmark, ϕ, of the ballots. From ϕ, M reconstructs the path that was taken by B'. The reconstructed path is represented by P'. M then outputs its findings: B' and P'.

The experiment $\mathcal{E}$ will compare the outputs from M with the observed path and ballot subset. If the path determined by M is a sub-flow of the path observed by $\mathcal{E}$ and the set of ballots received by M is a subset of the original ballots set, B, then $\mathcal{E}$ outputs valid, otherwise $\mathcal{E}$ outputs invalid.

Remark 1. While the actual path the ballot has travelled may be stored in the ballot's data storage, it is more likely to be inferred from auxillary information. For instance, if a box of ballots travels through the system, the path may be recorded on the box. In this case, the manager can infer the path taken by the ballots by looking at the log written on the box.

We say that a protocol is complete if every ballot arrives at the manager and is accepted when the adversary is passive. We say that the protocol is secure if the set of ballots with associated paths accepted by the manager is a "sub-flow" of the real flow of ballots through the system.

Definition 1 (Completeness). *We say that the protocol is* complete *if the tracking game from Figure 3 is run with $\mathcal{A}$, a passive adversary, that has corrupted $n = 0$ players, then the subset of ballots received by the manager M is the same set of ballots originally generated, $B' = B$, the observed flow is the same as the flow reconstructed by M from the received pathmark, $P'' = P'$, and $\mathcal{E}$ outputs valid.*

Definition 2 (Security). *Let $\mathcal{A}$ be an adversary in the tracking game from Figure 3 which successfully corrupts n players. M receives a set of ballots, B', during the final step of ballot movement. From the ballot's attached pathmark, ϕ, M generates the path that the ballots traversed, P'. The experiment $\mathcal{E}$ then compares both the set of ballots, B', and the path generated by M, P', to what was observed during the game by $\mathcal{E}$ (B and P''). If M's outputs are consistent with what $\mathcal{E}$ observed, then $\mathcal{E}$ will output valid. In the event that M's output is not consistent with what was observed, meaning $\mathcal{A}$ was able to falsify a pathmark or inject ballots, then $\mathcal{E}$ will output invalid.*

The success rate *of $\mathcal{A}$ is the probability that $\mathcal{E}$ outputs invalid:*

$$\text{Success}(\mathcal{A}) = \Pr(Output_{\mathcal{E}} = invalid).$$

3 Background

In this section, we present the definition of a digital signature scheme and the description of an existing signature-related scheme, named Tracker [2]. Our first solution will use any existing digital signature scheme, for instance the Gap Diffie-Hellman Signature Scheme [4]. Our second system involves Tracker that was designed for supply chain environments.

3.1 Digital Signature Scheme

A digital signature scheme enables a user, upon reception of a message and a claimed signature, to assess that the message was created by a known signer, and that the message was not altered in transit. A digital signature scheme, $\mathcal{D} = (\mathcal{K}, \mathcal{S}, \mathcal{V})$, consists of three algorithms.

Key Generation Algorithm: $\mathcal{K} = (sk, vk)$**.** $\mathcal{K}$ is a randomized key generation algorithm which takes no input and outputs a secret key sk and a verifying key vk. There are two message sets M_{sk} and M_{vk} associated with the secret key and the verifying key respectively.

Signature Algorithm: $\mathcal{S}(m, sk) = (m, \sigma)$**.** $\mathcal{S}$ is a possibly randomized signature generation algorithm which takes a message $m \in M_{sk}$, and the secret key sk, and outputs the message m and a signature σ.

Verification Algorithm: $\mathcal{V}(vk, m, \sigma) = 0/1$**.** $\mathcal{V}$ is a deterministic verification algorithm that takes as input the message m, the verifiying key vk, and the signature σ, and outputs 1 if the signature is valid and 0 if not.

It is required for any key pair (vk, sk) output by $\mathcal{K}$ and any message $m \in \mathcal{M}_{vk}$ that:

$$\mathcal{V}(vk, m, \mathcal{S}(sk, m)) = 1.$$

A digital signature scheme is considered secure if it is unforgeable under chosen message attacks meaning that an adversary is unable to forge a valid signature even with access to a signature oracle.

3.2 Tracker

Tracker [2] is a system which was created in order to trace physical goods through supply chains and detect if the path taken by those goods was either valid or invalid. Tracker uses RFID tags to do this. Intermediates at each step in the chain update the RFID tag, eventually creating a polynomial which can be evaluated by the central manager once the goods have reached a checkpoint in the chain. Depending on the evaluation of the polynomial and the resulting value [8], the manager can determine if the path taken was valid. Additionally, Tracker supplies a method for detecting duplicates and extra security features

in regards to fraudulent objects being inserted into the chain by virtue of its design.

We now give an overview of how the Tracker system works, followed by a more formal description. Tracker uses directed graphs to model the supply chain that a set of goods are intended to move through. Each good in the supply chain is affixed with an RFID tag that will store a state which can be read and updated as the goods move. Each vertex in the graph is a step in the supply chain. The primary entities are issuers, readers, and managers. Issuers are responsible for initialization, including generation of keys. Issuers issue the keys to each reader, as well as issuing all keys to the manager, and sets the initial state on each RFID tag before releasing the goods into the supply chain. The initial tag state stores a polynomial evaluated at a value chosen by the issuer. This polynominal value is called the pathmark. When readers receive a ballot they use their keys and arithmetic operations to update the polynomial, eventually creating an effectively unique value which describes where the goods have traveled. Managers use their knowledge of the keys to generate the value for each acceptable path. When the good reaches a checkpoint in the supply chain, where a manager is located, it is then simple for the manager to verify if the received value is correct given the expected path of the goods.

In order to soften the reading, we directly use the voting terminology to describe Tracker. We consider an issuer I and a manager M, as well as m different election officials O_k. Let $\mathbb{G} = \{V, E\}$ be a directed graph representing the voting chain, with V the set of vertices and E the set of edges such that $E = |m|$. Let $v_i \in V$ be a step in the voting chain and $e_i \in E$ be a directed edge. Let $\mathcal{T}$ be a set of n different tags t. Let P_{poss} denote the set of possible paths and S denote the set of possible tag states s_t^j (of tag t at step j). There are m state transition functions $f_i : S \to S$, grouped in the set F. Let P_{valid} be the set of valid paths p_{valid_i} and S_{valid} be a set of valid states s_{valid_i}. There is a database DB_{clone} stored at the manager M to protect against cloned tags.

Three functions are defined as follows:

- a function $READ : T \to S$ that reads out a tag t and returns t's current state.
- a function $WRITE : T \times S \to S$ that writes a new state s_t^{j+1} into tag t.
- a function $CHECK : S \to \begin{cases} p_{valid_i} & \text{if tag } t \text{ went through } p_{valid_i} \\ 0 & \text{if } \nexists\ p_{valid_i} \text{ that } t \text{ went through} \end{cases}$ that based on t's current state s_t^j decides about which valid path in the voting chain tag t has taken.

The state of a tag t in Tracker represents the path that the tag, and attached ballot, has traversed. At the end of each valid path the state, s_t^l, will match the evaluation of a unique polynomial $Q_{p_{valid}}(x)$ for a fixed value x_0. A path in the voting chain is represented by $Q_{p_{valid}}(x_0) \in \mathbb{F}_q$.

Key Generation Algorithm. Issuer I writes the initial state, s_t^0 into a new tag t.

Signature Algorithm. Election officials at each step sequentially compute the evaluation of a polynomial. In order to achieve the evaluation of the complete polynomial, $Q_{p_{valid}}(x_0)$, when the tag reaches the end of the valid path, each election official computes tag t's new state, s_t^i by applying arithmetic operations represented by the function f_i on t's current state, s_t^{i-1}.

Verification Algorithm. Manager M checks the tag's state, s_t^l. M has a set of evaluations of valid polynomials $Q_{p_{valid}}(x_0)$. M compares the computed value to the set of valid values it has. If the computed value is in the set, M knows that the tag traversed a valid path.

Each tag t stores three elements which compose the state, s_t^i, of a tag: A unique ID, a keyed HMAC of the unique ID, and $Q_{p_{valid}}(x_0)$ multiplied by the keyed HMAC of the unique ID. We denote $\phi(p) = Q_p(x_0)$, and define t's pathmark as $\phi_{ID}(p) := HMAC_{sk}(ID) \cdot \phi(p)$. The use of HMAC proves that tags are issued by a legitimate authority and prevents injection of fake tags by an adversary. Storing the pathmark, $HMAC_{sk}(ID) \cdot Q_{p_{valid}}(x_0)$, prevents a fake tag from having a valid path copied into it and being injected into the voting chain. The tag stores a probabilistic elliptic-curve Elgamal encryption of the state.

We next address the construction of the polynomial $Q_{p_{valid}}(x_0)$. For each step v_i, $1 \leq i \leq m$, in the voting chain, v_i is associated with a unique random number $a_i \in \mathbb{F}_q$, where q is a large prime. $a_0 \in \mathbb{F}_q$ is the random number associated with v_0 and the issuer I. The polynomial in $\mathbb{F}_q$ which corresponds to $p = \overrightarrow{v_0 v_1 ... v_l}$ is defined as follows:

$$Q_p(x) := a_0 x^l + \sum_{i=1}^{l} a_i x^{l-i}$$

All operations are in $\mathbb{F}_q$.

When an election official receives a tag, it reads the tag's current pathmark, updates it, and then writes the updated pathmark into the tag. Updating the pathmark as an official is fairly straightforward. Consider a tag t that is traveling along the path $p = \overrightarrow{v_0 v_1 ... v_{i-1} v_i v_{i+1} ... v_l}$. When the tag arrives at an official O_i, or step v_i in the voting chain, t has traveled through the path $p_{i-1} = \overrightarrow{v_0 v_1 ... v_{i-1}}$ and has ID, $HMAC_{sk}(ID)$, and pathmark $\phi_{ID}(p_{i-1})$. To update the tag's pathmark to $\phi_{ID}(p_i)$, O_i computes f_{O_i}.

$$f_{O_i}(x) := x_0 x + HMAC_{sk}(ID) \cdot a_i$$

So, $\phi_{ID}(p_i) := f_{O_i}(\phi_{ID}(p_{i-1})) = x_0 \phi_{ID}(p_{i-1}) + HMAC_{sk}(ID) \cdot a_i$.This constructs $\phi_{ID}(p_i) = HMAC_{sk}(ID) \cdot (a_0 x_0^l + \sum_{j=1}^{i} a_j x_0^{i-j}) = HMAC_{sk}(ID) \cdot \phi(p_i)$, the updated pathmark for the official O_i.

To check a tag, the manager M uses the state, s_t^l, of the tag which consists of a 3-tuple: $s_t^l = (ID, HMAC_{sk}(ID), \phi_{ID}(p_{valid}))$ (assuming a valid tag which has traveled a valid path). M has been provided with the secret key sk by I, and

reads ID, the first element of the 3-tuple, from the tag. M uses these in order to initially compute $HMAC_{sk}(ID)$. If $HMAC_{sk}(ID)$ is valid them M multiples $\phi_{ID}(p_{valid})$ by $HMAC_{sk}(ID)^{-1}$ to obtain $\phi(p_{valid})$. M then can easily compare $\phi(p_{valid})$ to its list of values to verify the validity.

Tracker has been shown to be secure under the security of HMAC and the Computational Diffie-Hellman (CDH) assumption[2], and to provide tag unlinkability and step unlinkability under the Decisional Diffie-Hellman (DDH) assumption[3] [2].

4 Paper-Based Voting Systems

In this section, we describe how the two aforementioned primitives are applied to the tracking model. First, we depict how we carry out lots of signatures in our model, and then how we embed Tracker [2] in it.

4.1 Using Digital Signatures

In this scheme we will use digital signatures, these signatures can be based on any secure digital signature scheme, in order to create the pathmark ϕ on each ballot.

Key Generation Algorithm. Issuer I generates both secret and verifying keys for each step, v_i, in the voting chain based on the chosen digital signature scheme and distributes them via a secure channel to each corresponding election official, O_i.

More precisely, the issuer, I, generates the keys according to the key generation algorithm for the chosen digital signature scheme for each of the m election officials, O_i. Each election official O_i, will receive their secret key sk_i, verifying key vk_i, and the verifying keys which correspond to the other election officials. Each potential role/location in the voting chain has a key associated with it, and so if an official holds multiple roles or will be at multiple locations then they also hold the appropriate keys which correspond to each of these. We assume there exists a secure channel to transmit this information over. The manager, M, will also receive the set of all public verifying keys for the election officials. Both the elections officials and the manager are sent the path, P.

Signature Algorithm. Each election official signs the ballot according to the chosen digital signature scheme as follows.

[2] The CDH problem is as follows: given three random elements (g, u, v) of G, compute $h = g^{log_g u \cdot log_g v}$.

[3] The DDH problem is as follows: given four elements (g, u, v, h), which with equal probability can be either all random elements of G or have the property that $log_g u = log_v h$, decide to output 0 in the case that the four elements are random, and 1 in the case that the four elements possess the mentioned property.

Once the keys have been distributed to both the officials and the manager, then the issuer I initiates the movement of the ballots from their point of origin into the voting chain path. As the ballots move to each election official O_i, the latter uses its secret key sk_i in order to sign the ballot. Each official also indicates, as part of its signed ballot, which official in the path should next receive the ballot and where the ballot came from (by including previous signatures). When an official receives a ballot, it first will use the verifying key of the previous official to verify that it was the intended recipient. We consider the set of signatures that are affixed to the ballots at the end of the path, in the pathmark, ϕ, of the ballots.

Verification Algorithm. The manager M verifies each individual signature based on the chosen digital signature scheme as follows.

When the manager M receives the set of ballots it then use the public verifying keys of each official in order to verify their signature and to reconstruct the path, P'.

Theorem 1. *The above scheme is complete.*

Sketch of proof. The correctness of the underlying digital signature scheme implies the completeness of our paper-based voting solution.

Theorem 2. *Suppose $\mathcal{A}$ is an adversary playing the game described in Definition 2. Then there exists an adversary $\mathcal{B}$ against the digital signature scheme, such that the success rate of $\mathcal{A}$ is upperbounded by the success rate of the forger $\mathcal{B}$.*

Sketch of proof. We sketch the argument. First, we assume that the adversary $\mathcal{A}$ does not create any signature forgeries, otherwise we get an adversary against the underlying digital signature scheme. Note that every ballot's claimed path can then be verified by verifying the signatures. Since every signature by an honest player (i.e. honest election official) was really created by an honest player, the only way the manager M can accept a given path for a ballot is if the ballot actually passed through every honest player on its way. The honest players also use the signatures to verify the sender (i.e. the official election from who they received the signature) and indicate the next recipient (i.e. the official election that should receive the currently generated signature). It follows that the paths deduced by the manager M must correspond to actual paths taken by ballots. It follows that if the manager M accepts, the experiment from Definition 2 outputs valid.

4.2 Using Tracker

In this scheme, we will use Tracker [2].

Key Generation Algorithm. Issuer I generates keys, a_i, for each step, v_i,

in the voting chain and sends them to the official, O_i, at that step over a secure channel. The keys for each official are sent to the manager M.

More precisely, the issuer I will set up Tracker as is specified in Section 3.2, including writing the initial state to each tag. I generates a value a_i for each election official O_i that will be used as their secret key sk_i in order to sign the ballots as they move through the path. I distributes these values via an assumed secure channel to both the corresponding election officials as well as the manager M. M, from these values, is able to predetermine valid pathmarks and store them in a database to later be referenced when checking a ballot's pathmark.

Signature Algorithm. Election officials, O_i, at each step, v_i, compute the evaluation of a polynomial using their secret key, a_i. This evaluation of $Q_{p_{valid}}(x_0)$ using their key produces a value which is the updated pathmark, ϕ, for the ballot.

In order for an official O_i to sign the ballot, it uses the $READ$ and $WRITE$ functions. When O_i receives a ballot, it reads the state using $READ$, updates the pathmark using the state change function f_i, and updates the state of the tag using $WRITE$. We consider the updated pathmark to be the signature generated in this scheme.

Verification Algorithm. Manager M has precomputed valid values for the polynomial $Q_{p_{valid}}(x_0)$. M compares the computed value of the pathmark to the set of valid values it has. If the computed value is in the set, M knows that the tag traversed a valid path.

Indeed, once the ballot has traversed the full path, M receives the ballot at a checkpoint. At this point, M is able to read and decrypt that tag state of the ballot. M first checks that the ballot ID is unique as well as verifying the keyed hash of the ID, then recovers the pathmark of the ballot and compares it to the predetermined list that was generated during protocol set up. If the pathmark is valid, M outputs valid, otherwise M outputs invalid.

Theorem 3. *The above scheme is complete.*

Sketch of proof. Given the case with a passive adversary $\mathcal{A}$ who has corrupt $n = 0$ players (i.e. election officials), the set of ballots that the manager M receives at the checkpoint, B' will be the same as the ballots which are originally submitted, B. And given that all players are honest, the pathmark and corresponding path, P', recovered by M will be the same as the flow observed by $\mathcal{E}$, $P'' = P'$. Then $\mathcal{E}$ outputs valid and the protocol has completeness.

Theorem 4. *Suppose $\mathcal{A}$ is an adversary playing the game described in Definition 2. Then there exists an adversary $\mathcal{B}$ against Tracker such that the success rate of $\mathcal{A}$ is upperbounded by the success rate of the Tracker forger $\mathcal{B}$.*

Sketch of proof. The claim follows directly from the security properties of Tracker. The polynomial involved in Tracker encodes and authenticates the ballot path,

which means that if the manager accepts any ballot that has not passed through the corresponding path, we get a Tracker forgery. Otherwise, the "sub-flow" inferred by the manager matches the actual flow, and the experiment does not output invalid.

5 Concluding Remarks

In this paper, we proposed two logistics systems for paper-based ballot elections. Our first proposal is the simplest one, using digital signatures. While its design and implementation are simple, performance is unfortunately limited. Our second proposition is more complex and uses a system originally designed for tracking goods in supply chains. The resulting solution is more efficient, more practical scheme.

5.1 Comparing our Paper-based Voting Solutions

Digital Signatures: Our system with digital signatures is the simplest design that one could imagine. Such a system is thus easily implementable. Signatures prevent the manager M to validate, cast and count fraudulent, forged ballots. However, there are as many signatures on a ballot as there are election officials who checked it. Therefore, ballot storage size linearly increases with steps in the path. Also, the manager M is required to verify each signature, making the computational verification cost linear in the number of steps in the path.

Tracker: Our Tracker-based system is the most sophisticated one. Thus, its implementation may require more work, from defining various functions (e.g. $READ$, $WRITE$ and $CHECK$) to specifying cryptographic tools (e.g. HMAC).

However, the gain in efficiency is considerable. There is one global signature to be verified by the manager M, not depending on the number of steps on the path that the ballot followed, thanks to the polynomial evaluation mechanism.

5.2 Future Work

Future work will consider the application of aggregate signatures in our tracking model. An aggregate signature scheme [3] gives us one single signature on behalf of multiple signers, in a given order. At first sight, such primitive seems to be an attractive option for tracking paper-based ballots. Nevertheless, the performance and security may not be as good as our Tracker-based solution. It is also interesting to consider how this could be perhaps be applied to other systems which use paper ballots such as mail-in voting. Additionally, Tracker does not currently have a good mechanism for tracking ballots to enable individual accountability but could potentially be adapted in order to increase individual accountability. In the future we may also consider ways to mitigate attacks we did not consider here, which were exclusively insider attacks.

Careful thought and analysis is also required for any physical implementation of our system. RFID tags vary widely in capability, reliability and speed. We cannot use the simplest, write-once tags, but the tags do not need very sophisticated capabilites such as tamper-resistance or cryptographic processing ability. The tags need to be sufficiently reliable so that no significant manipulation may hide in "statistical noise" of expected failures. Processing the ballots must also be sufficiently fast, so that ballot tracking does not become a bottle-neck.

Careful thought is also needed for the physical process of affixing RFID tags to ballots, such that the correlation between voter and ballot is broken.

References

1. Claudia Z. Acemyan, Philip Kortum, Michael D. Byrne, and Dan S. Wallach. Usability of voter verifiable, end-to-end voting systems: Baseline data for Helios, Prêt à Voter, and Scantegrity II. In *2014 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 14)*, San Diego, CA, August 2014. USENIX Association.
2. Erik-Oliver Blass, Kaoutar Elkhiyaoui, Refik Molva, and Eurecom Sophia Antipolis. Tracker: Security and privacy for RFID-based supply chains. In *In NDSS'11, 18th Annual Network and Distributed System Security Symposium, 6-9 February 2011*. Citeseer, 2011.
3. Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 416–432. Springer, 2003.
4. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 514–532. Springer, 2001.
5. David Chaum, Richard Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L. Rivest, Peter Y. A. Ryan, Emily Shen, Alan T. Sherman, and Poorvi L. Vora. Scantegrity II: end-to-end verifiability by voters of optical scan elections through confirmation codes. *IEEE Trans. Information Forensics and Security*, 4(4):611–627, 2009.
6. David Chaum, Aleksander Essex, Richard Carback, Jeremy Clark, Stefan Popoveniuc, Alan T. Sherman, and Poorvi L. Vora. Scantegrity: End-to-end voter-verifiable optical-scan voting. *IEEE Secur. Priv.*, 6(3):40–46, 2008.
7. David Chaum, Peter Y. A. Ryan, and Steve A. Schneider. A practical voter-verifiable election scheme. In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *Computer Security - ESORICS 2005, 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005, Proceedings*, volume 3679 of *Lecture Notes in Computer Science*, pages 118–139. Springer, 2005.
8. G. Noubir, K. Vijayananda, and H.J. Nussbaumer. Signature-based method for run-time fault detection in communication protocols. *Computer Communications*, 21(5):405 – 421, 1998.
9. Ron Rivest. The ThreeBallot voting system. MIT Report – `https://people.csail.mit.edu/rivest/Rivest-TheThreeBallotVotingSystem.pdf`, 2006.

How to fake zero-knowledge proofs, again

Véronique Cortier[1], Pierrick Gaudry[1], and Quentin Yang[1,2]

[1] Université de Lorraine, CNRS, Inria, Nancy, France
[2] École Polytechnique, Palaiseau, France

Abstract. In 2012, Bernhard et al. showed that the Fiat-Shamir heuristic must be used with great care in zero-knowledge proofs. We explain how, in the Belenios voting system, while not using the weak version of Fiat-Shamir, there is still a gap that allows to fake a zero-knowledge proof in certain circumstances. Therefore an attacker who corrupts the voting server and the decryption trustees could break verifiability.

1 Presentation of the result

Context. Zero-knowledge proofs are heavily used in e-voting protocols. A voter typically proves that her vote belongs to a set of valid choices and authorities show that they correctly decrypted the result. A standard way to move from an interactive proof to a non-interactive one is the Fiat-Shamir heuristic [5] where the randomness of the verifier is simulated by hashing the inputs of the prover. Bernhard et al. [2] have highlighted that great care must be taken in the part of the inputs that are hashed. In particular, for proofs on messages encrypted using ElGamal, they showed that it is not sufficient to hash the commitment (weak Fiat-Shamir). Instead, the full context must be hashed (strong Fiat-Shamir), otherwise decrypting authorities could, for example, change the election result yet producing a valid proof of correct decryption, in the context of the Helios [1] protocol. A similar idea [7] has been used in 2019 to attack the voting protocol of Scytl that were to be deployed in Switzerland.

Our contribution. Belenios [4] is a variant of Helios that prevents ballot stuffing and that has been used in more than 500 elections since 2018. In Belenios, a version of the Fiat-Shamir heuristic is used that does not fully follow the strong Fiat-Shamir as described in [2], since, apart from the commitment, only the ciphertext is hashed, and not the other parts of the context. In particular, like in Helios, the generator g of the group is, in principle, a parameter that can be chosen to be different for each election. This gives one degree of freedom for the authority in charge of setting up the election. Another degree of freedom can come from the public key when the decryption authorities collude with the authority. Combining both, we show that an attacker can produce a ciphertext and a valid proof of set membership, while the ciphertext will actually decrypt to an arbitrary vote controlled by the attacker. Compared to [2], the attack is slightly more involved since in Belenios, the whole ciphertext is hashed in the zero-knowledge proof, instead of just the commitment in Helios. In particular, our attack requires to adversarially select the group generator.

To illustrate this weakness, we consider the context of an election organized with Belenios, where voters select at most one candidate over a list of k choices. An attacker who can choose the group generator and the decryption key will be able to produce a full ballot that would be considered as valid but contains an arbitrary value. Interestingly, the "proof" that the ballot is valid involves several zero-knowledge proofs, but even if the attacker can fake only a single proof of set membership, it can be used several times to construct an entirely valid fake ballot.

How does it work? The first step is to create an ElGamal ciphertext with a fake proof of set membership, following the specification of Belenios for this zero-knowledge proof [6]. Without going into the details (see next section), we start from a pair of elements (x, y) for which the discrete logarithms are known w.r.t. a generator γ, and then re-interpret them as an ElGamal ciphertext using another generator $g = \gamma^a$. The value a can be chosen by the attacker and gives a degree of freedom that allows to construct all the data needed for a fake proof that the encrypted value is either 0 or 1. In this construction, the secret key that serves to decrypt the ballots must be known from the attacker. If it can be chosen by the attacker, this gives an additional degree of freedom to force a certain decryption value instead of having a random one when decrypting (x, y).

The second step is to use this ciphertext (x, y) to build a fake ballot. The structure of a ballot is a list of k encrypted bits, each of them telling whether or not the corresponding candidate is chosen. There are individual proofs that these are indeed encrypted bits, and an overall proof that the homomorphic sum of these ballots is also in $\{0, 1\}$. The construction uses (x, y) as a fake bit-encryption for the first candidate, and produces genuine encrypted bits for the others, but takes advantage of the randomness to enforce the homomorphic sum of all of them to be (x, y) as well. Therefore the fake proof of set membership for (x, y) can be used twice and the ballot looks valid.

Impact. In a setting where the attacker controls the server and the decryption trustees, our attack breaks verifiability since the result of the election can be arbitrarily modified by the attacker. This attack has been missed in the security proof of [3] because the group generator was assumed to be a fixed, known, parameter.

The fix is pretty simple: the strong Fiat-Shamir heuristic should be used, that is the group generator and the public key of the election should be included in the hash, as advised in [2] for Helios. Interestingly, the current implementation of Helios still uses the weak Fiat-Shamir heuristic and is hence still subject to the attacks mentioned in [2] as well as our attack where the attacker plays with the group generator instead of the ciphertext.

2 Technical details

For the sake of completeness, we provide here the full details on how to build a fake ballot with a valid proof by choosing the group generator appropriately,

in the context of a Belenios election where voters choose at most one candidate among k choices.

We start by recalling how to build a zero-knowledge proof of set membership, using the Fiat-Shamir heuristic. This relies on a hash function that maps to $\mathbb{Z}_q$; we denote it by hash. We consider a group G of prime order q and we write $w \in_r \mathbb{Z}_q$ to say that w is drawn uniformly at random in $\mathbb{Z}_q$, independently of all the other random choices. We follow the Belenios specification which uses an intermediate between weak and strong Fiat-Shamir.

Set membership proof. Let g be a group generator in G, and let $h = g^s$ be a public key corresponding to a secret key $s \in \mathbb{Z}_q$. Given an ElGamal encryption $(x, y) = (g^\alpha, g^m h^\alpha)$ of a cleartext m, and given α, the prover wishes to prove that m belongs to some fixed set $\{m_1, \cdots, m_n\}$.

The prover proceeds as follows:

1. Let i in $[1, n]$ be such that $m = m_i$. Let $w \in_r \mathbb{Z}_q$ and compute $A_j = g^w$ and $B_j = h^w$.
2. For any $j \neq i$, $1 \leq j \leq n$, let $\sigma_j, \rho_j \in_r \mathbb{Z}_q$ and compute $A_j = g^{\rho_j} x^{-\sigma_j}$ and $B_j = h^{\rho_j} \left(y/g^{m_j}\right)^{-\sigma_j}$.
3. Let $c = \mathsf{hash}(x||y||A_1||B_1|| \cdots ||A_n||B_n)$ and compute $\sigma_i = c - \sum_{j \neq i} \sigma_j$ and $\rho_i = w + \alpha\sigma_i$.

The set membership proof is then

$$\pi = (\sigma_1, \rho_1), \ldots, (\sigma_n, \rho_n).$$

To check the validity of this proof, the verifier proceeds as follows:

1. Compute $A_i = g^{\rho_i} x^{-\sigma_i}$ and $B_i = h^{\rho_i} \left(y/g^{m_i}\right)^{-\sigma_i}$ for all $1 \leq i \leq n$.
2. Check that $\mathsf{hash}(x||y||A_1||B_1|| \cdots ||A_n||B_n) = \sum_{i \in [1,n]} \sigma_i$.

The security properties of this Σ protocol, *i.e.* completeness, zero-knowledge and special-soundness are very classical and can be found for instance in [6].

Forging a set membership proof in {0, 1}. We show here how an attacker can select a group generator g and a secret key s such that she can forge a ciphertext (x, y) and a proof π that passes the validity check while (x, y) is an encryption of a message V chosen by the attacker.

The construction proceeds as follows:

1. Let γ be a generator of G.
2. Let $\alpha, \beta, r_1, r_2, r_3, r_4 \in_r \mathbb{Z}_q$.
3. Let $x = \gamma^\alpha$, $y = \gamma^\beta$.
4. Compute $c = \mathsf{hash}(x||y||\gamma^{r_1}||\gamma^{r_2}||\gamma^{r_3}||\gamma^{r_4})$.

Recall that one has to find $g, h, \sigma_1, \rho_1, \sigma_2, \rho_2$ such that $\sigma_1 + \sigma_2 = \mathsf{hash}(x||y||$ $g^{\rho_1} x^{-\sigma_1}||h^{\rho_1} y^{-\sigma_1}||g^{\rho_2} x^{-\sigma_2}||h^{\rho_2} \left(y/g\right)^{-\sigma_2})$. In order to do so, one can choose $a, s \in \mathbb{Z}_q$ such that $g = \gamma^a$ and $h = g^s$, along with the corresponding σ_1, ρ_1, σ_2, ρ_2 such

that $\mathsf{hash}(x||y||g^{\rho_1}x^{-\sigma_1}||h^{\rho_1}y^{-\sigma_1}||g^{\rho_2}x^{-\sigma_2}||h^{\rho_2}\,(y/g)^{-\sigma_2}) = c$. This leads to the following equations

$$\begin{cases} g^{\rho_1}x^{-\sigma_1} = \gamma^{r_1}, \\ h^{\rho_1}y^{-\sigma_1} = y^{r_2}, \\ g^{\rho_2}x^{-\sigma_2} = \gamma^{r_3}, \\ h^{\rho_2}\,(y/g)^{-\sigma_2} = \gamma^{r_4}, \\ \sigma_1 + \sigma_2 = c. \end{cases}$$

Using the logarithm in base γ, we obtain the following system of equations to be verified modulo q:

$$\begin{cases} a\rho_1 - \alpha\sigma_1 = r_1 \\ as\rho_1 - \beta\sigma_1 = r_2 \\ a\rho_2 - \alpha\sigma_2 = r_3 \\ as\rho_2 - \sigma_2(\beta - a) = r_4 \\ \sigma_1 + \sigma_2 = c \end{cases}, \text{ hence } \begin{cases} \rho_1 = \frac{r_1+\alpha\sigma_1}{a} \\ \sigma_1 = \frac{r_2-sr_1}{\alpha s-\beta} \\ \rho_2 = \frac{r_3+\alpha\sigma_2}{a} \\ \sigma_2 = \frac{r_4-sr_3}{\alpha s-\beta+a} \\ \frac{r_2-sr_1}{\alpha s-\beta} + \frac{r_4-sr_3}{\alpha s-\beta+a} = c. \end{cases}$$

The first four equations give explicit formulas which allows one to derive $\sigma_1, \rho_1, \sigma_2$ and ρ_2 from a and s while the last one gives a sufficient condition for the proof to be accepted. In addition, (x, y) decrypts into g^V if and only if $a = \frac{\beta-\alpha s}{V}$. Therefore, one can choose s as the solution of the following equation that becomes linear in s after clearing denominators:

$$\frac{r_2 - sr_1}{\alpha s - \beta} + \frac{r_4 - sr_3}{(\alpha s - \beta)(1 - \frac{1}{V})} = c.$$

Consequently, the remaining of the procedure consists of the following steps:

5. Let $s = \left(\beta c + r_2 + \frac{Vr_4}{V-1}\right)\left(\alpha c + r_1 + \frac{Vr_3}{V-1}\right)^{-1}$ and $a = \frac{\beta-\alpha s}{V}$.
6. Let $\sigma_1 = \frac{r_2-sr_1}{\alpha s-\beta}$, $\rho_1 = \frac{r_1+\alpha\sigma_1}{a}$, $\sigma_2 = \frac{r_4-sr_3}{\alpha s-\beta+a}$ and $\rho_2 = \frac{r_3+\alpha\sigma_2}{a}$.
7. Return the elements $g = \gamma^a$ as a generator, $h = g^s$ as a public key, s a secret key, (x, y) as a ciphertext, and $\pi = (\sigma_1, \rho_1), (\sigma_2, \rho_2)$ as a fake set membership proof for (x, y).

The computations of s, σ_1, ρ_1, σ_2, ρ_2, and a involve divisions by quantities that could in principle be zero. However, these are random elements in $\mathbb{Z}_q$, so that the probability of these events to occur is negligible (and one could still start again with other randoms).

As explained above, the verifier will accept the proof π, yet (x, y) does not encrypt 0 nor 1 but V, where V is chosen by the attacker. Thus, she can use this approach to build a ciphertext that encrypts a particular value of her choice or that could be used to add or subtract some amount of votes.

Forging a fake Belenios ballot. A full Belenios ballot for an election with k candidates is actually of the form $(x_1, y_1), \ldots, (x_k, y_k), \pi_0, \pi_1, \ldots, \pi_k$ where:

- π_0 is a proof that $\prod_{i=1}^{k}(x_i, y_i)$ is the encryption of 0 or 1 (the voter should select at most one candidate);

– for $i \geq 1$, π_i is a proof that (x_i, y_i) is the encryption of 0 or 1 (the voter can choose the ith candidate, or not).

In Belenios, a ballot is also signed by a credential but this part is irrelevant here (the adaptation is straightforward).

To forge a ballot that looks valid but contains an arbitrary value V, we proceed as follows:

– let $(x_1, y_1) = (x, y)$ and $\pi_1 = \pi$ where (x, y) is the forged ciphertext for which we have a valid proof π that (x, y) is the encryption of 0 or 1, while it is the encryption of V;
– pick random $\alpha_2, \ldots, \alpha_{k-1} \in_r \mathbb{Z}_q$ and let $(x_i, y_i) = (g^{\alpha_i}, h^{\alpha_i})$ be an encryption of 0, and compute π_i a (honest) proof that (x_i, y_i) encrypts 0 or 1;
– let $\alpha_k = -\sum_{i=2}^{k-1} \alpha_i$ and $x_k = g^{\alpha_k}$ et $y_k = h^{\alpha_k}$, so that (x_k, y_k) is an encryption of 0, and compute π_k a (honest) proof that (x_k, y_k) encrypts 0 or 1;
– Finally, $\prod_{i=1}^{k}(x_i, y_i) = (x, y)$ hence we can simply take $\pi_0 = \pi$ for the last proof.

The forged ballot will pass all the tests hence will be accepted. However, (x_1, y_1) is an encryption of V. Since only the homomorphic sums of all the votes for each candidate are decrypted, the tally will show V additional votes for the first candidate, where V is arbitrarily chosen by the attacker.

The attacker could also choose to set V to a huge value, so that decryption, which is based on a small discrete-logarithm computation will run forever.

References

1. Adida, B.: Helios: Web-based open-audit voting. USENIX Security 2008. pp. 335–348. USENIX Association, 2008
2. Bernhard, D., Pereira, O., Warinschi, B.: How not to prove yourself: Pitfalls of the Fiat-Shamir heuristic and applications to Helios. ASIACRYPT 2012. LNCS, vol. 7658, pp. 626–643. Springer, 2012.
3. Cortier, V., Dragan, C.C., Dupressoir, F., Schmidt, B., Strub, P.Y., Warinschi, B.: Machine-checked proofs of privacy for electronic voting protocols. 2017 IEEE Symposium on Security and Privacy. pp. 993–1008. IEEE, 2017.
4. Cortier, V., Gaudry, P., Glondu, S.: Belenios: A simple private and verifiable electronic voting system. Foundations of Security, Protocols, and Equational Reasoning: Essays Dedicated to Catherine A. Meadows. LNCS, vol. 11565, pp. 214–238. Springer, 2019
5. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, 1987.
6. Gaudry, P.: Some ZK security proofs for Belenios. `https://hal.inria.fr/hal-01576379`, 2017, informal note
7. Lewis, S.J., Pereira, O., Teague, V.: Trapdoor commitments in the SwissPost e-voting shuffle proof. `https://people.eng.unimelb.edu.au/vjteague/SwissVote`, 2019, blog entry

Proposed Effective Responses to Technology in Elections Problems

Verify My Vote: Voter Experience

Mohammed Alsadi[0000−0003−1635−5132] and
Steve Schneider[0000−0001−8365−6993]

Surrey Centre for Cyber Security, University of Surrey, UK
{m.alsadi,s.schneider}@surrey.ac.uk
http://www.surrey.ac.uk/sccs

Abstract. Online voting typically requires an election provider to run the election process. Relying on election providers as trusted entities to deliver the election does not typically provide transparency. Verifiability has been proposed as a way for voters to independently verify whether their votes have been recorded as they were cast, and included in the tally. This paper reports on an application of the Selene approach to verifiability, which provides a way for voters to confirm at the end of the election that their vote has been included correctly. In conjunction with Civica Election Services, a commercial elections organisation, we have deployed a trial system by integrating Selene-based verifiability with their existing balloting system, allowing us to investigate its use 'in the wild'. We used this system to explore the experience and opinions of voters around verifiability through analyzing their responses and feedback to a survey following a live ballot. Our results show that (1) survey respondents were happy to verify their vote, (2) survey respondents found it easy to use the system, and considered it kept their vote private, and (3) checking the vote increased their confidence in election results which led to positive perception of the system. We also explore the relationship between different factors that affect survey respondents' perception such as privacy, confidence and design issues.

Keywords: E-voting · Selene · Voter experience · Verifiability · Trust.

1 Introduction

Electronic voting (e-voting) is an approach to balloting that uses electronic technology in some element of the vote processing, from vote capture and transfer through to vote tallying. The introduction of computer technologies into the area of voting has led to other advantages such as improved accessibility and remote voting, and also the possibility of efficiency in tallying. However, concerns about cyber security for e-voting systems act as a key challenge to their adoption for high-stakes elections.

One key approach to addressing e-voting systems' challenges is cryptographic election verification [7], which produces a verifiable cryptographic proof allowing any particular voter to verify whether her vote has been included in the tally phase exactly as it was cast and also any observer to confirm the correctness

of the tally results. This allows verification that the election has delivered the correct result.

Enhancement of security in e-voting systems should be a factor in their trustworthiness, however, security will not in itself increase trust among the public since the system must not only be secure but must be seen to be secure. Verifiability provides the mechanism by which the voters and the public can observe that the system is behaving properly, and so plays a role in increasing the trust in the election [10]. Verifiability in e-voting systems is classified into:

- Individual Verifiability : any particular voter should be able to check whether her vote has been recorded, tallied and counted correctly
- Universal Verifiability : any third party should be able to verify the final result of the election.

In order to achieve verifiability, voters are typically provided with some kind of receipt which they can use—together with additional data published on a bulletin board, such as encrypted ballots and zero-knowledge proofs—to check that their votes were counted and that voting machines/authorities followed the prescribed procedure.

In this paper, we focus on voters' experience with verifiable e-voting through the deployment of a system incorporating verifiability based on the Selene voting protocol [11]. This was used in two commercial elections run by Civica Election Services (CES) for the Royal College of Nursing and for the College of Podiatrists to select their representatives. Voters who verified their vote were invited to complete a questionnaire about their attitudes and perception of the system.

This paper is organized as follows; Section 2 gives an overview of Selene and its architecture. Section 3 provides details about VMV from voter point of view. Experiment analysis is explained in details in Section 4. Relative works regarding usability in verifiable voting is summarized in Section 5. Finally, our work is concluded in Section 6.

2 Selene Overview

Selene [11] is an end-to-end verifiable election cryptographic protocol which is designed to allow voters to verify their votes directly in plaintext. This is seen as more intuitive for the voter compared to the majority of end-to-end verifiable schemes in which the encrypted ballots are verified, preventing voters from directly seeing the vote that they have cast. The design intention of Selene is to provide verifiability in a direct and easy to understand manner on one hand and improve the overall voting experience on the other hand.

Verifiability in Selene is achieved by allocating each voter with a unique tracker number which is only revealed to the voter after the election has taken place. Using tracker numbers helps to ensure voters' privacy as election providers are unable to link a tracker number to its corresponding voter. Further, it helps to enable coercion-resistance, as voters only know their tracker numbers after the election tally. Furthermore, Selene allows voters to generate a fake tracker

number to show a coercer if necessary, as a further defence against having to reveal her real vote to a coercer.

The stages of the Selene protocol are: (1) election set-up, including the generation of the election keys and encrypted trackers, (2) generation of tracker commitments and distribution to voters of the first part of their commitment, (3) voting, where encrypted and signed votes are recorded for each voter against their encrypted tracker, (4) mixing and decryption, where all of the encrypted tracker and vote pairs are securely shuffled and then decrypted, and (5) notification of trackers where each voter receives the information required for them to recover their tracker and hence verify their vote. Throughout these stages, Selene relies upon the use of a mix-net: a distributed system that securely and verifiably shuffles its input data according to some unknown permutation. Shuffling within the mix-net takes advantage of an encryption scheme which can re-encrypt data without first decrypting it.

Selene also relies upon an append-only web bulletin board (WBB), to broadcast data in such a way as to guarantee that data—once broadcast—cannot be amended or deleted. In Selene, the WBB is used to provide a record of the election data for verification and audit. This includes publishing, for example, the election public key, encrypted trackers, encrypted votes and shuffled plaintext votes.

3 The VMV System: Voter Perspective

Verify My Vote (VMV) is an end-to-end verifiable e-voting approach designed for use with existing internet voting systems [2]. The implementation used in this study is considered a first step towards a full system, and does not at this stage incorporate all of the elements that would be required for a full implementation of Selene. For example, untappable channels between the system and the voters are not present, the voters do not manage their own keys, and the full coercion-resistance mechanisms in Selene are not included. However the implementation does provide the verifiability functionality for the users, enabling investigation of its usability and voters' attitude to it. This step towards full Selene enables integration with existing systems without the need to have major changes either in the structure of the existing systems or of the voters' overall experience. Selene makes it easier for voters to verify their vote as they can see their vote in plaintext when they come to confirm it. VMV makes use of Distributed Ledger Technology for the WBB to store verifiability parameters in advance, during and at the end of the election. The intention is to enhance trust in the voting system since the stored parameters are immutable once they are on the WBB and voters are able to check them at any time once they are published. Further, this approach enables us to distribute control over several election trustees rather than having only a single election provider controlling the whole election process. This way, election trustees have to reach a consensus on the election information before being stored on the WBB which results in increased integrity and trust in the election.

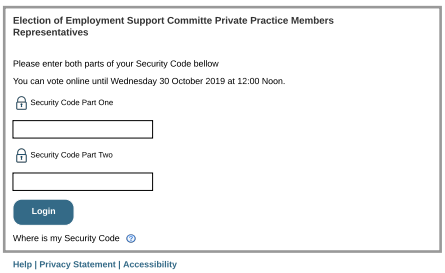


Fig. 1. Login Screen

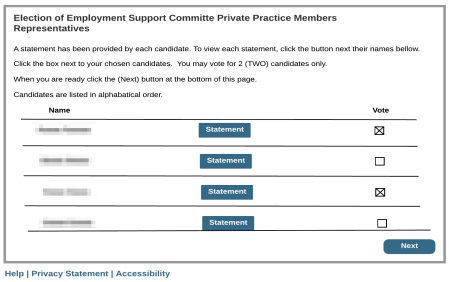


Fig. 2. Completed ballot form.

From the voter's perspective, VMV is designed to be simple since the underlying cryptographic complexities and calculations are kept away from voters. Moreover, at each step voters are provided with a simple, direct, short and clear instructions which help them to understand what they have to do and what is the next action to be taken.

Voters log in to the system as shown in Figure 1 via HTTPS using the security credentials supplied to them by email prior to election start time (following the election provider's standard approach). Once verified, the voter is presented with the ballot where she is able to select an appropriate number of candidates according to the election, as illustrated in Figure 2. The voter makes her selection(s) and clicks next, and is then redirected to the submission screen where a summary of her vote is displayed. At this point, the voter has the chance to either modify her choice(s) or submit the ballot. If the voter chooses to proceed with the current choice(s), her confirmed vote will be recorded within the vote database and a confirmation message will be displayed.

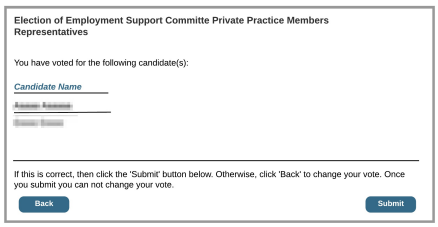


Fig. 3. Submission screen

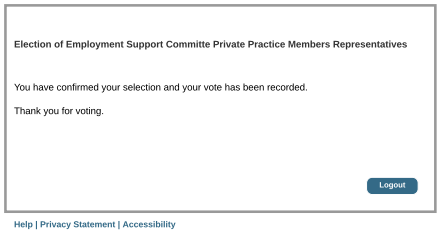
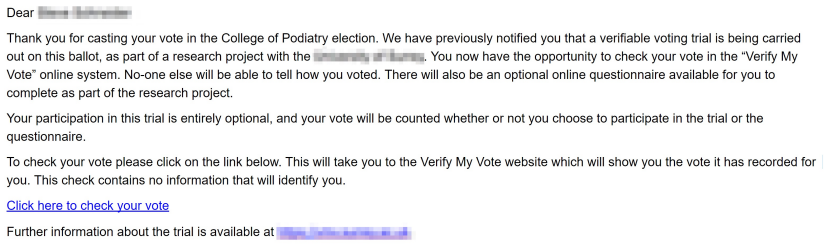


Fig. 4. Confirmation screen

Once the election period has closed and the results have been published, voters are offered the opportunity to access the web service to verify their vote. This is achieved using their encrypted tracker numbers' commitment values supplied to them via email. As shown in Figure 5, voters can verify their vote using the URL supplied, where the cryptographic information is hidden in the URL. When

a valid URL containing the corresponding election name and correct values is used, then the verification result will be displayed as shown in Figure 6. If the voter sees that the verification result does not correspond to their cast vote, the voter can easily report this issue by using the link "Report if this is not your vote" provided on the verification page.



Dear [redacted]

Thank you for casting your vote in the College of Podiatry election. We have previously notified you that a verifiable voting trial is being carried out on this ballot, as part of a research project with the [redacted]. You now have the opportunity to check your vote in the "Verify My Vote" online system. No-one else will be able to tell how you voted. There will also be an optional online questionnaire available for you to complete as part of the research project.

Your participation in this trial is entirely optional, and your vote will be counted whether or not you choose to participate in the trial or the questionnaire.

To check your vote please click on the link below. This will take you to the Verify My Vote website which will show you the vote it has recorded for you. This check contains no information that will identify you.

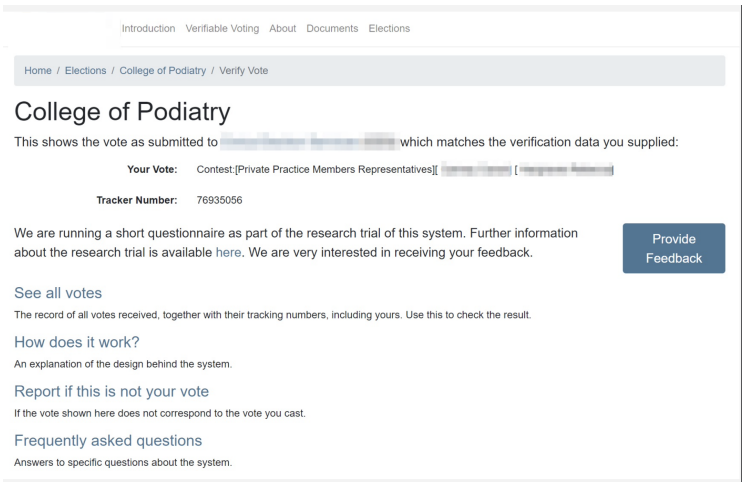
Click here to check your vote

Further information about the trial is available at [redacted]

Fig. 5. Verification Email sent to Voters.

Selecting the "See all votes" link presents the full list of plaintext votes along with their corresponding tracker numbers. This list is extracted from the WBB to ensure that voting information has not been manipulated.

The website also provided pages explaining the system including the verifiability mechanism, and a page about the VMV project. These received respectively 106 and 294 distinct visits during the period of the trials (though for anonymity reasons we did not check if any were repeat visits).



Introduction Verifiable Voting About Documents Elections

Home / Elections / College of Podiatry / Verify Vote

College of Podiatry

This shows the vote as submitted to [redacted] which matches the verification data you supplied:

Your Vote: Contest:[Private Practice Members Representatives][[redacted]

Tracker Number: 76935056

We are running a short questionnaire as part of the research trial of this system. Further information about the research trial is available here. We are very interested in receiving your feedback.

Provide Feedback

See all votes

The record of all votes received, together with their tracking numbers, including yours. Use this to check the result.

How does it work?

An explanation of the design behind the system.

Report if this is not your vote

If the vote shown here does not correspond to the vote you cast.

Frequently asked questions

Answers to specific questions about the system.

Fig. 6. Verification of Vote

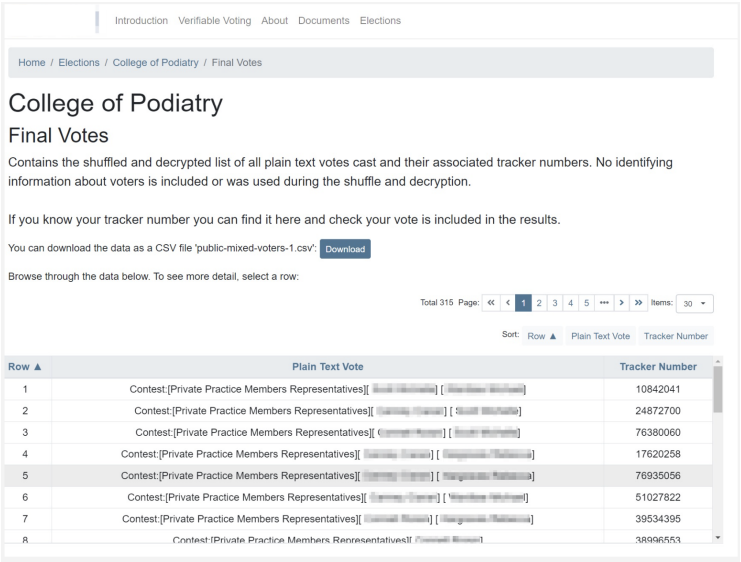


Fig. 7. List of all votes and tracker numbers

4 Voter Trials

The system was first used in an in-house trial which was followed by another election in June 2019 to select student representatives of the Department of Computer Science at Surrey. Based on survey feedback the system was tuned to make the election setup easier and to simplify the VMV website.

The trials reported here arise from two commercial ballots run by CES; a ballot to elect a West Midlands representative for the Royal College of Nursing and a ballot to elect two representatives for the College of Podiatrists, in August and October 2019 respectively. In both of these elections, voters were invited to verify their vote after the close of the ballot, and to complete a questionnaire as described below. The results reported in this paper are concerned with the responses to the questionnaire in these two elections.

Voters participating in the Royal College of Nursing ballot were asked to select one from two candidates. Voters were members of the Royal College of Nursing within the West Midlands region. The total number of votes cast were 942, and there were 503 verification checks. The College of Podiatrists ballot was conducted to select two representatives from six candidates. By the end of the election there were 315 ballots cast, and 148 verification checks were carried out.

The questionnaire consists of 14 statements (shown in Figure 9) about the VMV system they had just used, using a 6-point Likert scale, with levels of agreement ranging from Strongly Agree to Strongly Disagree; one open-ended question asking voters about their feedback or any further comments; and two categorical questions representing Gender and Age range. The questionnaire

was only available for voters who had followed the link from the confirmation email and verified their vote. We obtained 162 responses (30% of those who verified their votes) for the Royal College of Nursing ballot, and 32 responses (21.4% of those who verified their votes) for the College of Podiatrists ballot. The questionnaire and the anonymous responses are available at [3].

The questionnaire aimed to check to what extent respondents were happy with the VMV system and their experience of verifiability: how they perceived, used and evaluated it. In particular, we were interested in their perceptions of VMV and factors that play an important role in determining their attitude toward it.

The data obtained after the Royal College of Nursing and the College of Podiatrists elections shows that 82.6% of respondents were female and 16.3% were male, broadly matching the gender distributions in those professions. Age and Gender distribution is shown in Figure 8.

	<25	25-34	35-44	45-54	55-64	>64	prefer not to say
Female	3	9	18	56	54	10	2
Male	0	1	5	10	10	3	1

Fig. 8. Voters Age and Gender Distribution.

Our main findings from the questionnaire were that respondents were generally pleased with the opportunity to verify their vote, and with the ease of doing so, and did not feel that the system was complicated. 89.2% agreed with the statement "I am pleased to check my vote", 95.8% agreed with the statement "It was easy to check my vote", and 94.0% disagreed with the statement "It was difficult to check my vote". Furthermore, 84.9% of respondents expressed their agreement that the system gave them confidence that the election result is correct, and 72.1% of voters with the statement that everyone should check their vote if the facility is available. The majority of voters indicate that such a system is an alternative to paper-based voting as 83.4% disagree with the statement "I would prefer to vote on paper rather than over the Internet". The full breakdown of answers for agreement questions is shown in Figure 9.

In exploring impressions related to verifiable voting in general and VMV in particular, as well as understanding the actual experience voters had with verifying their vote, we asked questions in several ways, framed both positively and negatively. For example, we asked about how easy the system is and separately asked them if the system is difficult. Also, we asked about their perception of the privacy of votes, through their levels of agreement with the sentence "With this system other people cannot tell which vote is mine" and with "With this system I can tell how a particular person has voted". The list of questions is as follows:

– $v1$:"I am pleased to check my vote".

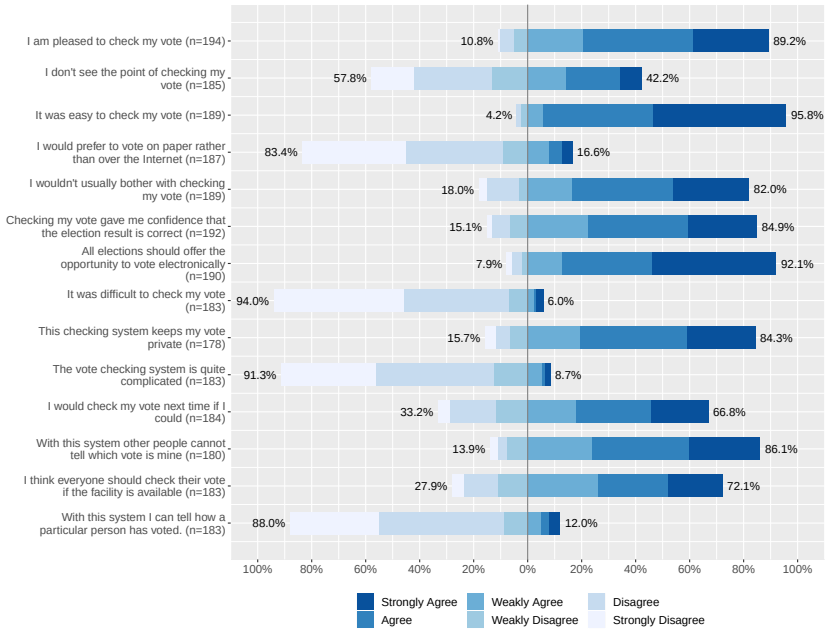


Fig. 9. Voters' Agreement Level Responses.

- $v2$:"I don't see the point of checking my vote".
- $v3$:"It was easy to check my vote".
- $v4$:"I would prefer to vote on paper rather than over the Internet".
- $v5$:"Checking my vote gave me confidence that the election result is correct".
- $v6$:"All elections should offer the opportunity to vote electronically".
- $v7$:"It was difficult to check my vote".
- $v8$:"This checking system keeps my vote private ".
- $v9$:"I would check my vote next time if I could".
- $v10$:"With this system other people cannot tell which vote is mine".
- $v11$:"I think everyone should check their vote if the facility is available ".
- $v12$:"With this system I can tell how a particular person has voted".

We evaluated correlations between the responses to these questions using Spearman's Rank Correlation test. Spearman's correlation is a nonparametric test which measures the strength and direction of association that exists between two variables measured on at least an ordinal scale. The matrix of correlations between pairs of statements is shown in Figure 10. This figure is used to highlight the most correlated variables in the questionnaire. Correlation coefficients range from [-1 to +1] where -1 indicates a negatively very strong correlation and +1 indicates a positively very strong correlation. Correlation coefficients are colored according to their value. Insignificant correlations are crossed.

The results show a number of both positive and negative strong correlations among the predefined factors, where a value above 0.6 is considered a strong

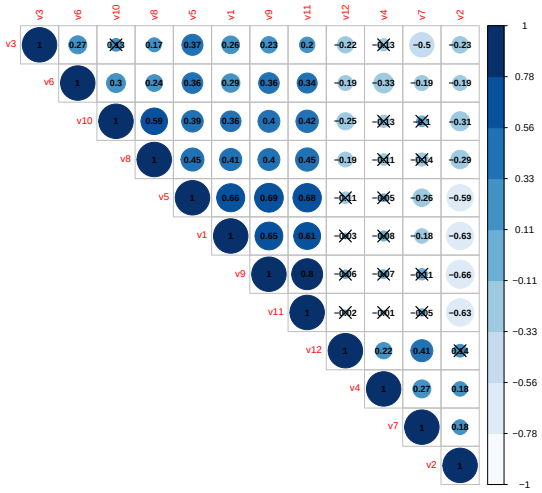


Fig. 10. Correlation Matrix.

correlation. The strongest positive correlations were between $v9$ and $v11$ (ρ = 0.8), $v5$ and $v9$ (ρ = 0.69), and $v5$ and $v11$ (ρ = 0.68). These findings indicate the importance of the main feature provided by VMV: verifiability. The ability to verify their vote increased voters' confidence in election results which leads them to recommend the system to other people. Moreover, the findings show other positive correlations such as the correlation between $v1$ and $v9$ (ρ = 0.65), $v1$ and $v11$ (ρ = 0.61). These correlations highlight the impact of satisfaction level respondents had while verifying their vote on the possibility to verify their vote next time in the future if they could and also on recommending the system to others. Respondents who indicated that they were pleased to check their vote are more likely to check their vote again in the future and recommend the system to other people. Thus we see consistency in the appreciation of being able to check the vote.

The strongest negative correlations among the predefined factors were between $v2$ and $v9$ (ρ = -0.66), $v2$ and $v11$ (ρ = -0.63), and $v2$ and $v1$ (ρ = -0.63). These findings lead us to the conclusion that voters agreed with $v2$ "I don't see the point of checking my vote" had negative impact on their attitude toward the possibility to verify their vote next time in the future if they could, recommend the system to other voters, as well as toward their general experience with the system. The other negative correlation was between $v3$ and $v7$ which is expected as we tend to ask the same question in opposite way.

We evaluated these negative correlations in more detail by using the Mann-Whitney test. This test evaluates whether the respondents in two groups (those that do not see the point of checking their vote, and those who do) gave similar responses to the questions, or whether they were different. The Mann Whitney test indicated that the mean value of responses of the two groups were differ-

ent and that the null hypothesis (that the two groups are similar) was rejected. Figure 11 shows the box plots for the two groups against Confidence and Verifiability. They show that voters' negative feedback about the system almost always arose from those respondents who indicated that they were unable to see the point of checking their vote.

Results show that respondents' negative feedback about the system almost always arose from those who indicated that they were unable to see the point of checking their vote.

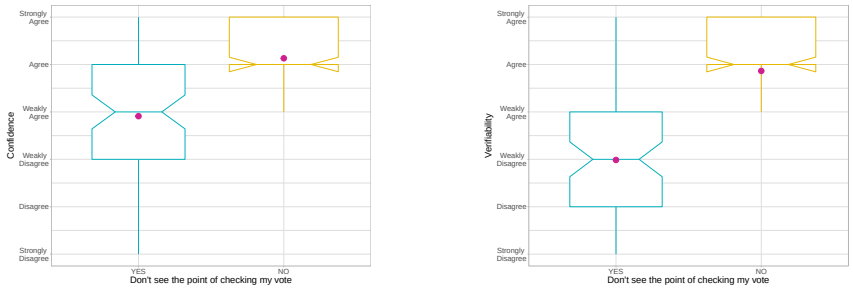


Fig. 11. Comparisons against Confidence and against Verifiability

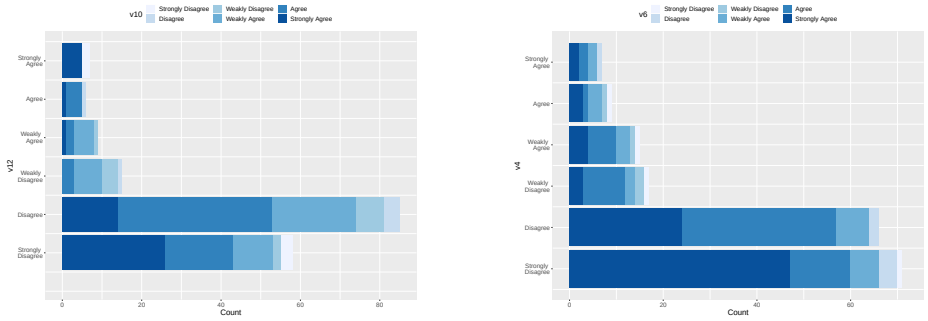


Fig. 12. Response Distribution among Reverse Questions.

One interesting finding in passing is the weak strength of correlations between the reverse questions. While we were expecting to get very strong negative correlations between $v3$ and $v7$, $v10$ and $v12$, and $v6$ and $v4$, as these were asking about the same aspect in different ways, results have shown moderate and weak correlations with ρ = -0.5, ρ = -0.25, and ρ = -0.33 respectively.

As illustrated in Figure 12 and Figure 13, we noticed that the same response (i.e. agreement level) had been given to reverse questions. For example, some voters who strongly agreed with $v3$ (It was easy to check my vote) strongly

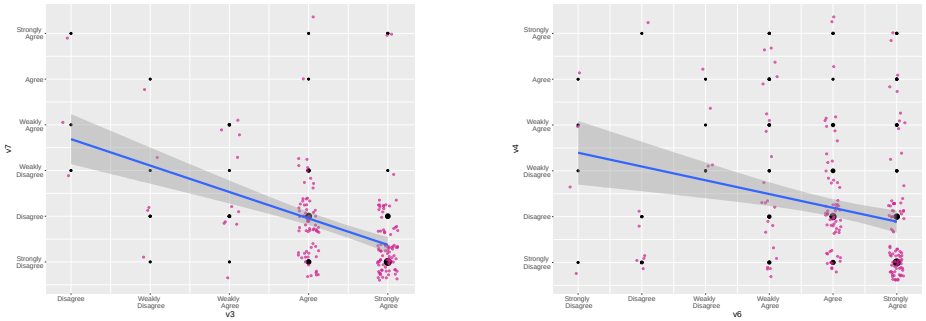


Fig. 13. Response Distribution among Reverse Questions.

agreed with $v7$ (It was difficult to check my vote). This issue leads to outlier values which had weakened the strength of the correlations.

4.1 Sentiment Analysis

The questionnaire contained one open-ended question which provided an opportunity for respondents to provide free form comments on any aspect of the system. This has provided qualitative feedback giving the broad range of responses to the system. The number of responses obtained by the end of the elections was 67. In order to analyze these comments "Sentimentr" R package was used.

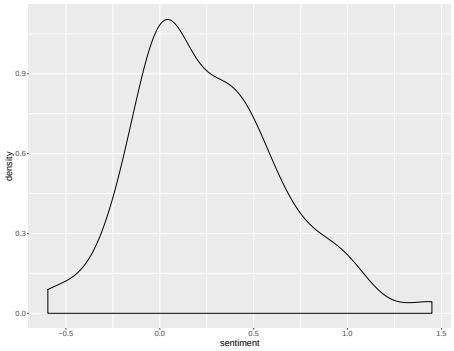


Fig. 14. Sentiment Density.

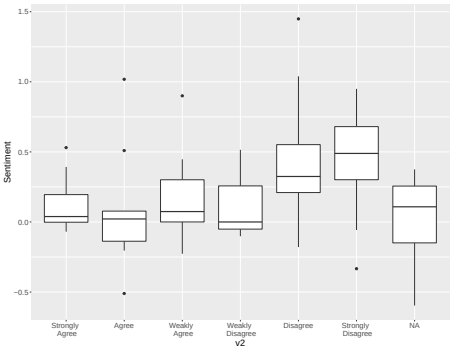


Fig. 15. Sentiment Scores against $v2$ Responses.

Generally, the sentiment analysis results show that 69% of responses were positive while 31% were negative. There were many positive comments around ease of use and confirming the vote, for example: "*Was easy and nice to know it*

was correct", "*Very easy to use and reassuring that vote registered as I voted*", and "*I like and appreciate the opportunity to check my vote. In something as seemingly trustworthy as an RCN election it is a bonus. If I were voting in a General Election I would regard it as a necessity.*"

Some respondents felt that the election authority should be trusted to ensure the system is secure and therefore there should be no need for verifiability, or could not see the point of verifiability: "*It isn't something I would use. I would trust that an organisation such as yours would have a secure system*", "*Very easy but I don't understand the need to check. Do you feel that we don't trust the system ?*", "*I don't see the point! Sorry! Once my vote is cast I assume it has been correctly processed or there are issues with the system. It tells me nothing about the robustness of the system and is a step that I would find not worthy of my time.*"

Some respondents considered the question around privacy more deeply and identified some issues that would need to be answered for voters: "*Since there seems to be a permanent record of how someone votes there is risk of breach of anonymity. I would prefer no record to be held.*", "*As this is the first time I have used the vote checking system, the privacy of my vote is purely on a trust basis relating to data protection.*", and "*I have no idea if this system keeps my vote private and/or that others can see how I or other people have voted.*"

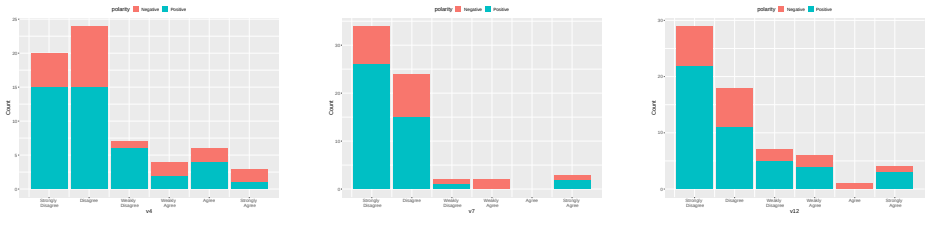


Fig. 16. Comments Polarity against v4, v7 and v12.

In order to get further details about the causes of negative comments, we compare the sentiment scores we obtained against some factors like $v2$, $v4$, $v7$ and $v12$. While Figure 16 illustrates sentiment polarity distribution with regard to $v4$, $v7$ and $v12$, Figure 15 shows sentiment scores of comments based on voters' agreement level with $v2$: negative and neutral score comments were made mostly by those who agreed that they did not see the point of checking their vote.

Additionally, we tried to detect the rate of emotions at the comment level. The "sentimentr" package uses a simple dictionary lookup to find emotion words and then compute the rate per comment. The emotion score ranges between 0 (no emotion used) and 1 (all words used were emotional). It provides a list which consists of seven emotions; anticipation, anger, joy, fear, sadness, surprise, trust and disgust. The results show that the highest emotions extracted from voters'

comments are trust and anticipation. The weight associated with trust emotion is 32% and 15% with anticipation.

5 Related Work

Voter experience with verifiable voting systems as well as usability of these systems has been gaining much attention from industrial and academic communities. The common feature among almost all of end-to-end verifiable voting systems is their heavy reliance on cryptography to solve various security problems surrounding voting. In such systems, voters have to verify votes that are encrypted (so the vote itself is not directly visible). This approach results in adding more complexity that might negatively impact their usability. Poor usability could lead to higher error rates and inefficient performance.

Usability studies of Selene have been carried out in both [6] and [14]. The former studied the impact of displaying security mechanisms on the user experience, while the latter expanded the previous research by studying the mental models revealed during voter interviews and comparing them with theoretical security notions. In each studies, the system was tested with 38 participants, and data was collected via questionnaires and semi-structured interviews. These interviews took place in a lab setting which has some limitations such as increased participants' feeling of security and partially biasing the evaluation of UX on specific aspects which are harder to assess in a controlled environment. The results indicated that hiding the cryptographic interaction has consequences on trust assumptions for the voting protocol. Further, they highlighted that the understanding of the verification phase has to be facilitated. Lack of understanding could lead to trust issues as participants question the integrity of the election and the purpose of the verification phase.

Assessing e-voting systems is a complex task due to the various numbers of factors that should be taken into consideration such as usability, user acceptance, privacy, security, trust and others [5]. The System Usability Scale (SUS) [4] and Unified Theory of Acceptance and Use of Technology (UTAUT) [12] are examples of well-known approaches for evaluating systems usability and user acceptance respectively.

Authors in [1] presented a lab-based experimental usability assessment of Helios, Pret a Voter, and Scantegrity II verifiable voting systems. In their approach, 37 participants recruited through an online advertisement were involved in three elections run using the aforementioned voting systems. After voting on a system, participants were asked immediately to complete the SUS. Then, they were instructed to evaluate the voting system they had just used. Next, they verified their vote using the same system and completed another SUS. At the end of the experiment, participants completed a final survey composed of 49 questions. Evaluations depended on three main factors: (1) effectiveness, which is measured in term errors occurred during voting/verification, (2) efficiency which is measured based on voting and verification completion times, and (3) satisfaction which is measured based on the SUS scores. Their results found that the

tested systems were difficult to use, and satisfaction was generally low. Further, verification completion rates were even lower than those for vote casting.

In [9], authors evaluated the usability and UX of the Swiss Internet Voting Neuchâtel scheme. Their assessment consisted of 3 stages: (1) a group of 12 HCI experts were recruited to evaluate the scheme through interaction with the interface and a semi-structured interview. (2) Based on experts' recommendations, a new design interface was proposed to address the weakness in stage 1. The new design interface was then investigated through a study with 36 participants. Participants were involved in a prototype elections using both the original and the new design interface. (3) Based on the data collected in the previous stage, a new redesign was proposed and tested with 49 participants. The system was assessed using the rate of manipulation as a measure of effectiveness, the execution time of verification as a measure of efficiency, whereas satisfaction was measured with SUS. Additionally, the user experience was assessed using user experience questionnaire (UEQ).

Another usability study of Helios was presented in [13]. In this work, a user study was conducted by creating a mock student government election. 20 participants were selected to take part in this study. However the study did not cover the verification part or clearly specify the factors used for assessing the system. The result indicated that half of the participants were unable to complete the election. Karayumak et al. [8] analyzed the usability of the Helios e-voting system with the main focus on voters' interactions with the system and in particular on ballot casting combined with verifiability mechanisms using the cognitive walkthrough approach by security, e-voting and usability experts. From the voter point of view, each step in the ballot casting process was carefully analyzed to check whether functionality and instructions provided to the voter in each step support voters to decide which functionality to use and to understand the corresponding next step. The results showed a number of usability flaws in general design, and detected some complexities in verification which led to very few voters being able to make use of the verifiability feature.

There are several contrasts between these approaches and the approach reported in this paper. Our primary focus is not on usability but on voters' attitude and views of verifiability as they have experienced it. (1) Being in the wild rather than in the lab, participants involved in our study were real voters who care about the election and its result, and who were in the real-world situation of the election. Their attitudes to the system are in the context of having cast a real vote; (2) The number of participants involved in our study was approaching 200; (3) L;ength of questionnaire: we tend to cover the whole factors necessary to evaluate the e-voting model through a single short questionnaire. Participants involved in other approaches were asked to fill multiple long questionnaires, more appropriate to a lab experimental situation. (4) evaluation factors: we aimed to comprehensively cover factors that have a potential impact on e-voting usability and verifiability. This point was one of the main reason why we did not use SUS in our study, since it is a general usability satisfaction tool rather than tailored towards the particular aspects we sought to explore around verifiability.

6 Conclusion

Verifiability is a key feature in e-voting which aims to enhance trust and integrity in the whole election process. In this paper, we studied the usability of VMV model along with voters' experience in both voting and verification stages. The main focus of this work is to check to what extent voters were happy with their experience of verifiability, how they perceived, used and evaluated it, and what was their attitude towards it. By carrying out the experiment "in the wild" we had over 50% of voters follow the verification step and we received the responses to our questionnaire from these voters. While these respondents were self-selecting and therefore cannot be considered representative of the voting population overall they were still over 15% of the voters. However it is worth noting that only a subset of the voters need to carry out the verification to give a high level of confidence that the result could not have been switched through ballot tampering (the level of confidence depending on the winning margin). For each election in this study the level of confidence in the winners was greater than 99% given the number of verifications performed in each case.

Generally, the results show that respondents to the questionnaire were happy with the system. Our main results can be summarized as follows:

- Respondents found this approach to verifiability very usable, with a significant majority (94.8%) responding that the system was easy to use.
- Verifiability was a key feature appreciated by respondents that had a key impact on their attitude towards other aspects of the system, their confidence in the result, and their willingness to recommend it to others.
- Respondents who did not see the point of checking the vote were less positive about verifiability, even if they found it easy to perform. The implications of this finding are not clear and require further investigation.

Acknowledgements

Thanks to Peter Ryan, Peter Roenne, Marie-Laure Zollinger, Constantin Catalan Dragan, Francois Dupressoir, Matthew Casey and Muntadher Sallal for discussions around the trials and the questionnaire, and Chris Fife-Schaw for advice on the questionnaire. Thanks also to Phil Wright, Simon Hearn and Rufus George of Civica Election Services for their support of the trials.

This research was funded by EPSRC through the VOLT project EP/P031811/1, with support for development of VMV and integration with the CES system through the Surrey Impact Acceleration Account EP/R511791/1.

References

1. Acemyan, C.Z., Kortum, P., Byrne, M.D., Wallach, D.S.: Usability of voter verifiable, end-to-end voting systems: Baseline data for Helios, Prêt à Voter, and Scantegrity II. In: 2014 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 14) (2014)

2. Alsadi, M., Casey, M., Dragan, C., Dupressoir, F., Riley, L., M., S., Schneider, S., Treharne, H., Wadsworth, C., Wright, P.: Towards end-to-end verifiable online voting: adding verifiability to established voting systems, University of Surrey (2020)
3. Alsadi, M., Schneider, S.: Verify My Vote: Voter experience questionnaire results (2020). https://doi.org/10.5281/zenodo.4002106
4. Brooke, J., et al.: SUS-a quick and dirty usability scale. Usability evaluation in industry **189**(194), 4–7 (1996)
5. Carter, L., Bélanger, F.: The utilization of e-government services: citizen trust, innovation and acceptance factors. Information systems journal **15**(1), 5–25 (2005)
6. Distler, V., Zollinger, M.L., Lallemand, C., Roenne, P.B., Ryan, P.Y., Koenig, V.: Security-visible, yet unseen? In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. pp. 1–13 (2019)
7. Dzieduszycka-Suinat, S., Murray, J., Kiniry, J., Zimmerman, D., Wagner, D., Robinson, P., Foltzer, A., Morina, S.: The future of voting: end-to-end verifiable internet voting-specification and feasibility study. US Vote Foundation pp. 30–38
8. Karayumak, F., Olembo, M.M., Kauer, M., Volkamer, M.: Usability analysis of helios-an open source verifiable remote electronic voting system. EVT/WOTE **11**(5) (2011)
9. Marky, K., Zimmermann, V., Funk, M., Daubert, J., Bleck, K., Mühlhäuser, M.: Improving the usability and ux of the swiss internet voting interface. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1–13 (2020)
10. Mursi, M.F., Assassa, G.M., Abdelhafez, A., Samra, K.M.A.: On the development of electronic voting: a survey. International Journal of Computer Applications **61**(16) (2013)
11. Ryan, P.Y., Rønne, P.B., Iovino, V.: Selene: Voting with transparent verifiability and coercion-mitigation. In: International Conference on Financial Cryptography and Data Security. pp. 176–192. Springer (2016)
12. Venkatesh, V., Thong, J.Y., Xu, X.: Unified theory of acceptance and use of technology: A synthesis and the road ahead. Journal of the Association for Information Systems **17**(5), 328–376 (2016)
13. Weber, J., Hengartner, U.: Usability study of the open audit voting system Helios, University of Waterloo (2009)
14. Zollinger, M.L., Distler, V., Roenne, P., Ryan, P., Lallemand, C., Koenig, V.: User experience design for e-voting: How mental models align with security mechanisms. Electronic Voting (2019)

You can do RLAs for IRV: The Process Pilot of Risk-Limiting Audits for the San Francisco District Attorney 2019 Instant Runoff Vote

Michelle Blom*, Andrew Conway**, Dan King***, Laurent Sandrolini†, Philip B. Stark‡, Peter J. Stuckey§ and Vanessa Teague¶

Abstract. The City and County of San Francisco, CA, has used Instant Runoff Voting (IRV) for some elections since 2004. This report describes the first ever process pilot of Risk Limiting Audits for IRV, for the San Francisco District Attorney's race in November, 2019. We found that the vote-by-mail outcome could be efficiently audited to well under the 0.05 risk limit given a sample of only 200 ballots. All the software we developed for the pilot is open source.

1 Introduction

Post-election audits test a reported election result by randomly sampling paper ballots.[1] A *Risk Limiting Audit (RLA)* of a trustworthy paper trail of votes either finds strong statistical evidence that the reported outcome is correct, or reverts to a full manual tabulation to set the record straight.[2] (The outcome is the political result—i.e., who won—not the exact vote counts.) The maximum chance that a RLA will fail to correct the reported outcome if the reported

* School of Computing and Information Systems, University of Melbourne. michelle.blom@unimelb.edu.au

** Silicon Econometrics Pty. Ltd., andrewelections@greatcactus.org

*** Viewpoint Technical, Chula Vista, California. dan.king@sfgov.org, dan.king@vptech.io

† work done while at Dept of Technology, City of San Francisco. laurent.sandrolini@gmail.com

‡ Department of Statistics, University of California, Berkeley. stark@stat.berkeley.edu

§ Department of Data Science & AI, Monash University. Peter.Stuckey@monash.edu

¶ Thinking Cybersecurity Pty. Ltd. vanessa@thinkingcybersecurity.com and the Australian National University vanessa.teague@anu.edu.au

[1] We use the terms "ballot," "ballot card," and "card" as synonyms, even though a ballot might comprise more than one physical card. Election audits generally sample cards rather than ballots: most voting systems cannot identify separate cards that comprise a single voter's multi-card ballot.

[2] A careful, accurate full hand count finds the correct winner(s) if the paper trail is trustworthy—which is not automatic.

Initially, all candidates remain standing (are not eliminated)
While there is *more than one* candidate standing
  For every candidate c standing
    Tally (count) the ballots in which c is the highest-ranked
      candidate of those standing
  Eliminate the candidate with the smallest tally
The winner is the one candidate not eliminated

Fig. 1. The IRV counting procedure.

outcome is wrong is the *risk limit*. RLAs are becoming the *de facto* standard for post-election audits that check the tabulation. They are required by statute in Colorado, Nevada, Rhode Island, and Virginia, and have been piloted in over a dozen US states and in Denmark. California AB2125 authorizes RLAs.

Instant Runoff Voting (IRV) allows voters to express their preference order (ranking) for some or all candidates. IRV elections are counted by iteratively eliminating the least-popular candidate, as described in Figure 1. When a candidate is eliminated, each of their votes is passed to the next-preferred candidate on each ballot. The winner is the last remaining candidate when all the others have been eliminated. IRV is the normal form of voting in Australia, and is used or will be used in numerous US counties including San Francisco, Aspen, Oakland, and New York.

RLAs have been conducted for a variety of social choice functions (plurality, majority, super-majority, multi-winner plurality) but never for IRV. These can be audited by statistical tests of simple assertions about the ballots such as candidate A getting more votes than candidate B. The complexity of IRV introduces challenges for RLAs because it may not be clear what assertions about the election need to be audited. In some IRV races, the only contest that really matters is the comparison between the last two uneliminated candidates; in others, a change in the early stages of the elimination sequence can cascade into a different election outcome. The San Francisco pilot relied on theory derived only recently by Blom et al. [2] for analyzing the cast votes records (CVRs) to determine a set of simple, auditable, assertions which, taken together, imply that the reported election outcome is correct.

1.1 Overview of the San Francisco DA pilot audit

The San Francisco RLA pilot audited the vote by mail ballots for the 2019 San Francisco District Attorney's race. Obviously, auditing only the votes cast by mail does not truly test the accuracy of the election outcome. In this case it happened that the outcome for the vote by mail ballots was different from the overall outcome—Susan Loftus won the vote by mail ballots quite comfortably, though Chesa Boudin won the election overall. Hence the audit itself does not actually prove anything about the overall winner. Instead, it tested whether Susan Loftus would have won if the vote by mail ballots had been the only ballots

cast. Nevertheless it makes an interesting case study with which to explain how the general IRV RLA process works.

We call this a "process pilot" because it tested the feasibility of the process, not the election result itself. It was not a true RLA in part because it considered only ballots cast by mail, since the voting system was able to match paper ballots to CVRs for those ballots but not for ballots cast in person. Nevertheless, it gives us a good estimate of the amount of work that would be required to administer a meaningful RLA of an election with similar parameters. The audit required a sample of only 200 ballots even though the margin was small, and terminated with an estimated risk of only 0.003, well under the 0.05 risk limit. With three pairs of people entering the ballot data, the elapsed time for the audit was less than one hour, not including the time required to retrieve the paper ballots.

This encourages optimism that RLAs for IRV is feasible, particularly when individual ballots can be compared with their CVRs. It dispels the previously common but mistaken belief that IRV audits should take longer than audits of simpler voting systems. They don't; they're just a little harder to understand. Any audit can require inspecting many ballots when the margin is close or the error rate is high, but there is no evidence that IRV audits are likely to require substantially more work than audits of other social choice functions.

Section 3 contains a discussion of how to extend the process pilot to a full election audit.

1.2 The Software

Two important new ideas were put into practice for the first time for this pilot. The first was the RAIRE IRV assertion generator, which turns a complete set of IRV CVRs into a set of simple assertions that can be tested by existing RLA methods. The second was the SHANGRLA auditing framework, which presents a very general and flexible interface for RLAs and can incorporate RAIRE's assertions as well as assertions for other voting methods such as Borda, Condorcet, STAR-Voting, multi-winner plurality, and super-majority. The audit also used a new "risk-measuring" function, the Kaplan Martingale (KMart).

The project produced five main pieces of software, all open source and easily available online:

A format converter and election counter reads the CVRs and counts the votes to check that the outcome implied by the CVRs matches the reported election outcome.
`https://github.com/pbstark/SHANGRLA/blob/master/ConvertCVRToRAIRE.html`

The RAIRE Assertion-generator inputs the reformatted CVRs and calculates a set of assertions which, if true, imply that the reported election outcome is right. RAIRE uses heuristics to choose assertions that can be audited efficiently. See Section 2.1 for an explanation and Blom et al. [2] for more detail.
`https://github.com/michelleblom/audit-irv-cp/tree/raire-branch`

The IRV assertion visualiser displays a visual representation of all possible IRV election outcomes, allowing auditors to check directly that the assertions generated by RAIRE are sufficient to prove the reported election outcome.
`https://github.com/pbstark/SHANGRLA/blob/master/Code/RAIREExampleDataParsing.ipynb`

The SHANGRLA RLA tool is a general tool for conducting RLAs involving complex elections and a variety of possible statistical tests. It inputs the assertions from RAIRE and constructs assertions for other social choice functions (e.g., plurality, multi-winner plurality, or super-majority) and administers the audit. See Section 2.2 for an explanation and [10] for more detail.
`https://github.com/pbstark/SHANGRLA`

The Manual Ballot Entry Tool inputs the list of randomly-selected ballot cards for audit, and allows the auditors to record what they see on the ballot. This information is then fed back into SHANGRLA to decide whether the audit can stop or must examine more ballot cards.
`https://github.com/dan-king/RLA-MVR`

2 How the software works

Here we show how to adapt existing RLAs to IRV. The key insight is that we don't have to verify all the complicated steps of an IRV count—we find a few simple assertions that imply that the election outcome is right, then conduct an audit to test whether those assertions are true. If the RLA doesn't find sufficiently strong evidence that those assertions are true, the audit eventually expands to a full manual tabulation.

Before any votes have been tallied, we can imagine all possible elimination sequences $l_1, l_2, \ldots, l_k, w$, meaning that l_1 is eliminated first, followed by l_2, etc., in sequence, until l_k and w are the last two candidates standing, and l_k has fewer votes (and is therefore eliminated). The last candidate in the list is the winner—the one who remains after everyone else has been eliminated. Without knowing anything about the votes, we know that if there are $k+1$ candidates there must be $(k+1)! = (k+1) \times k \times (k-1) \times \ldots \times 3 \times 2$ different possible elimination orders.

These $(k+1)!$ elimination orders can be arranged into $k+1$ trees, one for each winning candidate. The root of each tree is the winner, while each path from a leaf to the root represents a possible elimination order, with the first-eliminated candidate at the leaf, the next eliminated candidate as its parent node, and so on. In the San Francisco DA race, the apparent elimination order (for VBM ballots) was Dautch, Tung, Boudin, Loftus—this is shown in Figure 2, which is copied from `https://www.sfelections.org/results/20191105/data/20191125/da/20191125_da_short.pdf`. An example of the complete list of elimination trees is shown in Figure 3, with the reported election outcome marked in red. Other paths in the same tree also represent wins for Loftus, by different elimination sequences.

Candidate	Round 1			Round 2			Round 3		
	Votes	Percentage	Transfer (Elimination)	Votes	Percentage	Transfer (Elimination)	Votes	Percentage	Transfer
SUZY LOFTUS	60,002	31.06%	6,500	66,502	35.63%	17,363	83,865	49.17%	0
LEIF DAUTCH	27,027	13.99%	-27,027	0	0.00%	0	0	0.00%	0
NANCY TUNG	37,347	19.33%	9,274	46,621	24.98%	-46,621	0	0.00%	0
CHESA BOUDIN	68,792	35.61%	4,745	73,537	39.40%	13,159	86,696	50.83%	0
Continuing Ballots Total	**193,168**			**186,660**			**170,561**		
Blanks	12,392		0	12,392		0	12,392		0
Exhausted	0		6,439	6,439		15,976	22,415		0
Overvotes	525		69	594		123	717		0
Non Transferable Total	**12,917**			**19,425**			**35,524**		

* Tie resolved in accordance with election law.

Fig. 2. Official results, including elimination order, for the San Francisco DA race. Source: `sfelections.org`

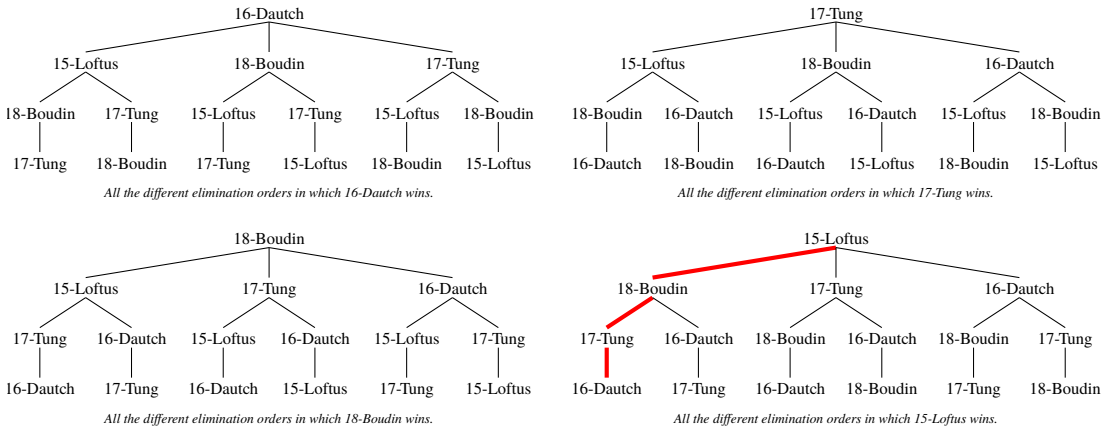


Fig. 3. Complete Elimination Trees for the San Francisco DA race

First observe that to test whether Loftus truly won there is no need to check the exact elimination sequence. If the reported winner truly won, but by a different elimination sequence, the reported election outcome is still correct. Therefore there is no need to audit anything about the tree of possible ways in which Loftus won. Instead, we concentrate on checking that no elimination sequence with a different winner is possible—visually, this corresponds to pruning every other tree so that every path from a leaf to the root is broken somewhere by an assertion that can be excluded by auditing.

The next section explains how RAIRE constructs assertions that perform this pruning.

2.1 Overview of RAIRE

Sometimes the only way to audit an IRV election is to to check that, at every step of the process, the right candidate was eliminated. In general, however, that strategy is inefficient because it may take a lot of work to verify comparisons that don't matter—for instance if a change to the early elimination order makes no difference to the final result. The next sections describe quicker ways in which it is sometimes possible to be confident that the reported winner truly won, without checking the entire elimination sequence.

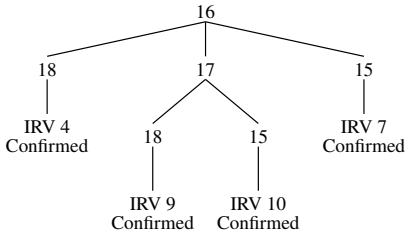
If every possible elimination sequence that produces a different winner (other than the reported winner w) can be contradicted by a true assertion, then every other possible winner has been excluded and w really won. RAIRE produces a set $\mathcal{F}$ of assertions such that *if* all the assertions in $\mathcal{F}$ are true, *then* w truly won. We conduct the overall RLA by checking each assertion in $\mathcal{F}$ as if it were the reported outcome of a 2-candidate plurality contest. The same sample can be used to check all the assertions.

RAIRE generates the assertions that can be used to prune the tree, but it is not necessary to trust RAIRE to do this correctly. The tree visualisation software allows any observer to check for themselves that every tree in which some candidate other than the reported winner wins has been completely pruned. Figures 4 and 6 show examples of tree visualisations for the San Francisco DA race—you can check for yourself that there is no remaining unpruned path from a leaf all the way to the root.

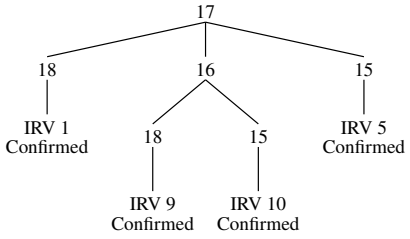
"IRV-elimination" assertions Suppose that one branch we wish to prune is an elimination sequence $l_1, \ldots, l_k$ with candidate w' the (alternative) winner. If w' is not the true winner, there must be at least one step along this sequence of eliminations that we can rule out. Consider the r-th step, in which l_r is eliminated. This elimination step is like a multi-winner plurality (first-past-the-post) election with one loser (l_r) and $k-r+1$ winners $l_{r+1}, \ldots, l_k, w$. We disregard all the candidates that have already been eliminated ($l_1, \ldots, l_{r-1}$) and attribute each ballot to whichever candidate in the set $l_r, l_{r+1}, \ldots, l_k, w$ it ranks highest. RAIRE can prune this branch by checking the assertion that that l_r must beat one of $l_{r+1}, \ldots, l_k, w$ at this step.

IRV$(l_r, c, \{l_{r+1}, \ldots, l_k, w\})$ is the assertion that l_r beats $c \in \{l_{r+1}, \ldots, l_k, w\}$ when each ballot card is counted as a vote for the candidate in $l_r, l_{r+1}, \ldots, l_k, w$ ranked highest on that card.

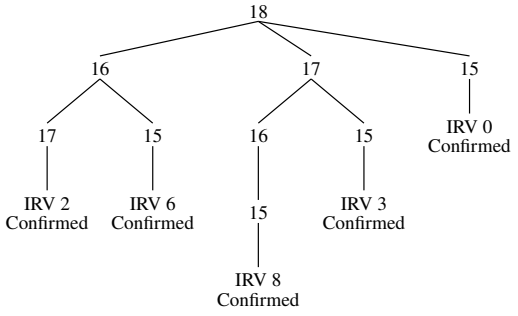
The visualisation of alternative trees for the San Francisco DA race is shown in Figure 4. Note that every branch of every tree (other than the tree in which Loftus wins) is pruned by an IRV assertion. The explanation of each assertion is shown in Figure 5.



Pruned tree in which 16-LEIF DAUTCH wins.



Pruned tree in which 17-NANCY TUNG wins.



Pruned tree in which 18-CHESA BOUDIN wins.

Fig. 4. Pruned Elimination Trees used for the San Francisco DA RLA. IRV n means that section of the tree is impossible if the assertion IRV n (listed in figure 5) is true. Confirmed means that the audit has confirmed that assertion.

IRV assertions:
Confirmed: IRV 0: Candidate 15 cannot be eliminated next when {16, 17} are eliminated.
Confirmed: IRV 1: Candidate 18 cannot be eliminated next when {16, 15} are eliminated.
Confirmed: IRV 2: Candidate 17 cannot be eliminated next when {15} are eliminated.
Confirmed: IRV 3: Candidate 15 cannot be eliminated next when {16} are eliminated.
Confirmed: IRV 4: Candidate 18 cannot be eliminated next when {15, 17} are eliminated.
Confirmed: IRV 5: Candidate 15 cannot be eliminated next when {16, 18} are eliminated.
Confirmed: IRV 6: Candidate 15 cannot be eliminated next when {17} are eliminated.
Confirmed: IRV 7: Candidate 15 cannot be eliminated next when {18, 17} are eliminated.
Confirmed: IRV 8: Candidate 15 cannot be eliminated next when {} are eliminated.
Confirmed: IRV 9: Candidate 18 cannot be eliminated next when {15} are eliminated.
Confirmed: IRV 10: Candidate 15 cannot be eliminated next when {18} are eliminated.

Fig. 5. Explanation of assertions for the Elimination Trees of Figure 4.

"Not-eliminated-before" assertions "Not-eliminated-before" auditing is a surprisingly powerful technique for proving that a certain candidate cannot win. It compares the highest possible tally of a reported loser to the lowest possible tally of the reported winner. The lowest tally that w can possibly have at any elimination stage is its total number of first preferences—IRV adds but never subtracts votes from not-yet-eliminated candidates as the algorithm progresses. The highest tally loser l can possibly have (assuming w is not eliminated) is the total number of mentions of l at any preference, when there is no higher preference for w on the same ballot card. If w's first preferences are greater than l's total mentions (excluding the ones listed below w), then l can never achieve a tally as large as w's. Therefore w cannot be eliminated before l in any elimination sequence.

We call this hypothesis *Not-Eliminated-Before*, NEB(l, w). (It is called Winner-only auditing in [3].)

> NEB(l, w) is the assertion that the number of cards that have w as the first preferences is greater than the total number of cards that mention l and do not prefer w to l.

If this assertion is true, w cannot be eliminated before l, so l cannot win. This assertion is most often useful when w is the reported winner of the election, but can sometimes be applied for other candidates too. Sometimes the assertion NEB(l_i, w) is true for every reported loser l_i, which is enough to prove that w won.[3]

[3] This argument can be extended to consider minimum and maximum tallies given that a certain set of candidates has already been eliminated—see [3] for details.

An example for an alternative method of auditing the San Francisco DA race incorporating an NEB assertion is shown in Figure 6, based on the 11th round of preliminary results. For those preliminary results, candidate 16 (Dautch) could be excluded immediately by an NEB assertion (i.e. at least one other candidate could not be eliminated before her). The assertions are all unconfirmed because this collection of assertions was never tested—the set shown in Figure 4 was. However, *if* these assertions had been checked, they would have provided an alternative valid way of confirming the election outcome. The assertions are explained in Figure 7.

Summary of What RAIRE does RAIRE takes the reported set of votes, computes the apparent winner w, and finds a collection $\mathcal{F}$ of assertions that imply w won. As described above, each assertion in $\mathcal{F}$ is either an IRV-elimination or NEB. These assertions should then be audited with an RLA. RAIRE uses heuristics to try to find the $\mathcal{F}$ most likely to terminate in a successful audit in the shortest time. This assumes, of course, that the reported outcome is correct—if it is not, then at least one of the assertions in $\mathcal{F}$ must be false, and this should be detected by the RLA with probability at least $1 - \alpha$. If the audit of any assertion $f \in \mathcal{F}$ does not support f, a full manual recount should be performed.

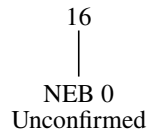
2.2 Overview of SHANGRLA

SHANGRLA is a very general method of auditing a variety of election types, by expressing an apparent election outcome as a series of assertions. Each assertion is of the form "the mean of a list of non-negative numbers is greater than 1/2." For example, consider an election with only two candidates, A and B, in which A is the reported winner. We test the assertion that "of those ballots that contain one candidate selection, more than half chose A." This can be audited in SHANGRLA by counting a vote for A as 1, a vote for B as 0 and a blank ballot (or a ballot that selects both) as 1/2. Now A is the true winner of the election if and only if the mean of those numbers is greater than 1/2.

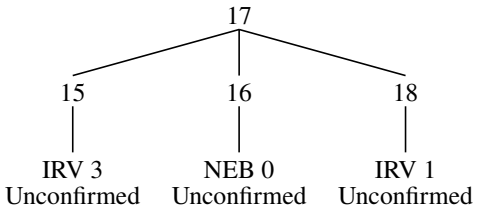
Each assertion is tested using a sequential test of the null hypothesis that its complement holds, i.e. the hypothesis that the mean is in fact less than or equal to 1/2. If all the null hypotheses are rejected, the election outcome is confirmed. If not, we proceed to a full manual recount. SHANGRLA incorporates several different statistical risk-measurement algorithms and extends naturally to plurality and super-majority contests with various election types including Range and Approval voting and Borda count.

SHANGRLA is specifically designed to support auditing Instant Runoff Voting (IRV) using the RAIRE assertion-generator. RAIRE produces a set of assertions sufficient to prove that the reported winner truly won, then SHANGRLA interprets these as assertions of the form "the mean of a list of non-negative numbers is greater than 1/2" and tests those assertions.

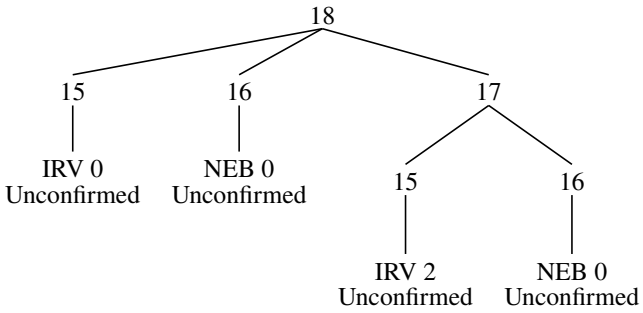
SHANGRLA also implements the "manifest phantoms to evil zombies" approach of [1] which allows the audit to sample only cards with CVRs that contain



Pruned tree in which 16-LEIF DAUTCH wins.



Pruned tree in which 17-NANCY TUNG wins.



Pruned tree in which 18-CHESA BOUDIN wins.

Fig. 6. A valid alternative set of assertions for testing the outcome of the San Francisco DA race

Not-Eliminated-Before assertions:
NEB 0: Candidate 15 cannot be eliminated before 16.
IRV assertions:
IRV 0: Candidate 15 cannot be eliminated next when {16, 17} are eliminated.
IRV 1: Candidate 18 cannot be eliminated next when {16, 15} are eliminated.
IRV 2: Candidate 15 cannot be eliminated next when {16} are eliminated.
IRV 3: Candidate 15 cannot be eliminated next when {16, 18} are eliminated.

Fig. 7. Explanation of assertions for the Elimination Trees of Figure 6.

particular contests, while ensuring that the risk limit is met even if the CVRs are wrong. An upper bound on the number of ballot cards that contain each contest under audit is required. The ability to target the sample makes it possible to audit contests that are not on every ballot card—such as partisan primaries and contests that are not jurisdiction-wide—much more efficiently. This is especially helpful for small contests with small margins, where it avoids "diluting" the sample be ensuring every selected card is informative and avoids "diluting" the contest margin by limiting the population of ballots to those that (putatively) contain the contest.

See [10] for additional detail.

SHANGRLA code is available at `https://github.com/pbstark/SHANGRLA`, with a detailed explanation by Stark [10].

Expressing IRV assertions in SHANGRLA Consider testing the assertion $IRV(l_r, c, \{l_{r+1}, \ldots, l_k, w\})$. In this assertion, l_r is treated as the winner and c as the loser, so a vote with l_r as the highest-ranked candidate in $\{l_r, l_{r+1}, \ldots, l_k, w\}$ is counted as 1. A vote with c as the highest-ranked candidate in $\{l_r, l_{r+1}, \ldots, l_k, w\}$ is counted as zero. Anything else is counted as 1/2. Thus l_r beats c at this point in the elimination sequence if and only if the mean of those numbers is greater than 1/2.

Expressing NEB assertions in SHANGRLA Consider testing the assertion $\text{NEB}(l, w)$. To express this using SHANGRLA, count a first preference for w as a 'vote' for w, i.e., a value of 1. Count any mention of l with no higher preference for w as a 'vote' for l, i.e., 0. Anything else is worth 1/2.

3 Completing the steps for a full audit

There are several generic steps necessary for a true audit that were omitted from the process pilot, such as a *compliance audit* to ensure that the paper trail was trustworthy, a public dice-rolling ceremony to generate the seed, and public retrieval of the paper ballots from storage. Since these are universal necessities for any RLA, we do not detail them here—see, e.g., [5], [11], [7] and [8] for instructions.

The main challenge in extending this pilot to a full, meaningful audit in San Franscisco is incorporating votes that were cast in precincts. An audit that considers only VBM ballots proves nothing about the overall election outcome—this was particularly obvious this year because the reported winner on VBM ballots was different from the reported overall winner of the DA race.

In San Francisco at present, ballots that are cast in the precinct are not amenable to a ballot-comparison audit, because the way they are stored electronically and physically does not allow an auditor to retrieve the paper ballot corresponding to a particular CVR. So there are three options.

1. Update the procedure for ballots cast in the precinct so that it is possible, without violating vote privacy, to link a particular CVR with its paper ballot.
2. It might also be possible to do batch-level comparison audits in a roundabout way: if the CVRs for physical batches are available (even if they can't be matched to specific ballots within the batch), one could use them to compute 'tallies' for the assertions, then check the tallies by hand if the batch is selected for audit.
3. Finally, it might be possible to combine RAIRE with the SUITE audit method [9], which allows auditing of ballots from two or more different strata, in this case ballot-comparison and ballot-polling.

The first option will result in examining the fewest ballots when the reported outcome is correct, though it requires some manual setup work.

The second option requires less setup but probably more auditing. A nice feature is that it wouldn't require stratification: batches can be drawn with probability proportional to an error bound as described in Section 3 of [6].

Option 3 requires more careful thought. RAIRE uses heuristics to generate a set of assertions that are likely to require the least auditing work, assuming there are no errors. These heuristics rely on an estimate of the expected sample size, which depends on the audit method being employed. SUITE does a complementary kind of optimization, choosing the most efficient ratio of sample probabilities in the different strata in order to minimize the expected audit cost. So SUITE can optimize for a given set of RAIRE assertions, and RAIRE can optimize given a particular choice of SUITE sampling ratios, but it is not obvious how to do the joint optimization to minimize overall expected sample size. Fortunately, this optimization affects efficiency but not soundness, and a suboptimal solution might still be quite efficient in practice. For example, we could instruct RAIRE to generate assertions as if it was doing a ballot polling audit, then use those assertions for both the ballot-polling and ballot-comparison strata, in the ratio determined by SUITE. However, RAIRE might in these cases over-estimate how much auditing is needed or even fail to produce any assertions because it seems much too hard.

4 Conclusion

You can do RLAs for IRV, using the open source software described in this report.

Our pilot took fewer than six person-hours of work, excluding the time to retrieve the paper ballots. The vote-by-mail outcome could be audited with a sample of only 200 ballots even though the margin was small, and terminated with an estimated risk of only 0.003, well under the 0.05 risk limit.

IRV audits can be as efficient as audits for simpler social choice functions, though of course a larger sample will be required if the margin is small, the error rate is large, or there is no way to match CVRs with their corresponding paper ballot.

All of the software developed for the San Francisco DA pilot audit is openly available online.

5 Acknowledgements

Many thanks to the San Francisco Department of Elections for their support of this project. Thanks also to Yuvi Panda for hosting Jupyterhub and Damjan Vukcevic for valuable discussions.

Bibliography

[1] J.H. Bañuelos and P.B. Stark. Limiting risk by turning manifest phantoms into evil zombies. Technical report, arXiv.org, 2012. Retrieved 17 July 2012.

[2] Michelle Blom, Peter J Stuckey, and Vanessa Teague. Risk-limiting audits for irv elections. *arXiv preprint arXiv:1903.08804*, 2019. `https://arxiv.org/abs/1903.08804`.

[3] Michelle Blom, Peter J Stuckey, and Vanessa Teague. Risk-limiting audits for irv elections. *arXiv preprint arXiv:1903.08804*, 2019. `https://arxiv.org/pdf/1903.08804.pdf`.

[4] Michelle Blom, Vanessa Teague, Peter J Stuckey, and Ron Tidhar. Efficient computation of exact irv margins. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, pages 480–488. IOS Press, 2016. ArXiv version: `https://arxiv.org/abs/1508.04885`.

[5] M. Lindeman and P.B. Stark. A gentle introduction to risk-limiting audits. *IEEE Security and Privacy*, 10:42–49, 2012. `https://www.stat.berkeley.edu/~stark/Preprints/gentle12.pdf`.

[6] Mark Lindeman, Neal McBurnett, Kellie Ottoboni, and Philip B Stark. Next steps for the colorado risk-limiting audit (corla) program. *arXiv preprint arXiv:1803.00698*, 2018.

[7] Jennifer Morrell. Knowing it's right, part one. a practical guide to risk-limiting audits. 2019. `https://www.democracyfund.org/media/uploaded/2019_DF_KnowingItsRight_Part1.pdf`.

[8] Jennifer Morrell. Knowing it's right, part two. risk-limiting audit implementation workbook. 2019. `https://www.democracyfund.org/media/uploaded/2019_DF_KnowingItsRight_Part2.pdf`.

[9] Kellie Ottoboni, Philip B Stark, Mark Lindeman, and Neal McBurnett. Risk-limiting audits by stratified union-intersection tests of elections (suite). In *International Joint Conference on Electronic Voting*, pages 174–188. Springer, 2018.

[10] Philip B. Stark. Sets of half-average nulls generate risk-limiting audits: Shangrla, 2019. `https://arxiv.org/abs/1911.10035`.

[11] Philip B. Stark and David A. Wagner. Evidence-based elections. *IEEE Security and Privacy*, 10:33–41, 2012.

A FAQ

1. What is the risk limit of the IRV audit?
   Answer: It inherits the risk limit from the RLAs conducted on each assertion in $\mathcal{F}$. If every assertion in $\mathcal{F}$ is audited with risk limit α, then the overall RAIRE audit detects a wrong election outcome with probability at least $1 - \alpha$.

2. How much auditing work will we need to do?
   Answer: It depends on factors such as the election margin and the number of discrepancies between the real and reported ballots. Even for ordinary first-past-the-post elections, RLAs can be very fast when the margin is large and there are no errors, or relatively time-consuming when the margins are close or there are significant differences between real and reported ballots. RAIRE also follows this pattern. It also depends on whether ballot-polling or ballot-level comparison audits are chosen.
3. Is it possible to estimate in advance how much auditing will be needed?
   Answer: Yes, but only on the assumption of a certain rate of error, which can't be predicted without inspecting the ballots. SHANGRLA provides estimated sample sizes given an estimated error rate.
4. Might RAIRE fall back to a full manual recount even when the reported outcome is correct? Is this more likely than for first-past-the-post audits?
   Answer: Yes, any RLA might fail to certify the result, and fall back to a full manual recount, even when the reported result is correct. RAIRE is more likely to do this than an otherwise-equivalent RLA on a first-past-the-post election of the same margin, because it conducts several simultaneous audits, any one of which might behave in this way.
5. Can we inspect the software?
   Answer: Yes, all the code is available at the links given in Section 1.2.
6. Do we need to trust the RAIRE software?
   No, you don't need to trust the software in order to be convinced by the audit—you can inspect the assertions $\mathcal{F}$ using the visualiser and check that they imply that the reported winner truly won.
   However, you do need a version of the RLA computations that you trust. There are many options—you can trust SHANGRLA or choose to reimplement your own.
7. Do we need to know the margin? Aren't margins hard to compute for IRV?
   Answer: The true margin in an IRV contest isn't obvious, though it can usually be computed in reasonable time [4]. It is often, but not always, half the difference between the last two candidates standing in the last round. RAIRE does not explicitly use the margin to construct the auditing assertions, but a lower bound on the margin is implied. Each assertion $f \in \mathcal{F}$ can be thought of as having its own margin, which is the number of votes that would need to be altered in order to make that assertion false. The overall IRV election margin cannot be smaller than the smallest margin of any assertion in $\mathcal{F}$.

Effective Cybersecurity Awareness Training for Election Officials

Carsten Schürmann, Lisa Hartmann Jensen, and Rósa María Sigbjörnsdóttir

IT University of Copenhagen, Copenhagen, Denmark
carsten@itu.dk, lisaha85@gmail.com, and rosa.sigbj@gmail.com

Abstract. Cybersecurity awareness training has a bad reputation for being ineffective and boring [21]. In this paper, we show the contrary, namely that it is possible to deliver effective cybersecurity awareness training using e-learning. We provide a general methodology on how to create cybersecurity awareness training and evaluate it based on Kirkpatrick's model of evaluation [22]. We have conducted a pilot study of the methodology in context of the European Parliament election 2019.

Keywords: Cybersecurity Awareness Training · E-learning · Human Factors · Attack Trees · Election Officials.

1 Introduction

Organizations rely on their staff for protection of their assets. No matter how many security polices are put in place, security always comes down to how the individual employee behaves. In March 2016, for example, the personal Google mail account of John Podesta, a former White House chief of staff and chair of Hillary Clinton's 2016 U.S. presidential campaign, was compromised in a data breach accomplished via a spear-phishing attack allegedly carried out by foreign Nation State. Allegedly, Podesta's assistant, following the advice of a security technician, complied and followed the instructions contained within the phishing mail [20].

Therefore, to protect an organization from security breaches, it is vital to protect the technical and organizational infrastructure including sensitive data *and* prepare users, employees, consultants, and guests to recognize and defend against cyberattacks. In this paper, we focus on the human factor. Social engineering attacks, where an adversary exploits human traits, such as modesty, altruism, empathy, and diligence of a victim to gain access to restricted resources, steal secrets, or causes other kinds of havoc. It seems natural that the only way to protect an organization against this kind of attack is by sharpening a user's common sense and the ability to recognize, react, and mitigate an imminent attack, and to install a designed behavior in connection with security [15]. Therefore, education is an important part of creating a security culture in organizations [6]. However, cybersecurity awareness training has the reputation of being ineffective [2].

Not wanting to accept this conclusion, we set out in this work to demonstrate that cybersecurity awareness training for short-term retention of knowledge, for example for election officials, *can be made* effective. The hypothesis of our work is that one of the reasons for the perceived ineffectiveness is that cybersecurity training is often unspecific, explaining concepts abstractly, such as confidentiality, integrity, and availability that are good to know, but often not directly relevant and difficult to translate into practice. Instead such training must be methodologically relevant, consistent, role-based and continuously adopted to an ever-evolving threat landscape [21].

As a corollary, effective cybersecurity awareness training can only take place, after a rigorous security analysis of the attack surface, the entire security context, and the security background of the target audience, i.e. users and course participants, has been conducted. These findings must inform the learning objectives of the cybersecurity awareness training, not more and not less. Concretely, in this paper, we develop a methodology consisting of a few easy to follow steps to prepare tailored security training for a particular target group to be deployed in a well-defined security context.

We evaluate this methodology empirically, in the context of the European Parliament election 2019, held in Denmark. In close cooperation with Copenhagen municipality, we conducted a security analysis of the voter identification system, deployed in each of the 53 polling stations in Copenhagen, and prepared an e-learning course for 53 election officials, the digital election secretaries, responsible for all technical equipment used in the polling station. The course was organized in modules, each tailored to the security needs of the election officials. All participants had to take an entry exam before the training and a final exam after the training. We could demonstratively measure a significant increase in cybersecurity preparedness for this limited target group election officials.

The cybersecurity awareness training was administered as part of the general training of election officials, who are are recruited within the municipality, some having served in this role already several times before. Election officials have to undergo training before each election, and the knowledge gained in the training is usually necessary only for the day of the election. In general, election officials were grateful to have the opportunity to learn about the attack surface. Long-term retention of knowledge was not measured. To our knowledge this is the first systematic study of e-learning with the short-term retention of cybersecurity knowledge.

The literature [21] defines three levels of security awareness: perception, comprehension, and projection. Perception is to be aware of that there are potential security risks. Comprehension is to understand and assess the dangers of security risks. Projection is to be able to anticipate future situations and how to act on potential security attacks. Based on our pilot training and the evaluative statistical analyses we conclude that cybersecurity awareness training for short-term retention delivered on all three levels of security awareness.

This paper is structured as follows. In Section 2, we discuss human factors in cyber security. In Section 3, we then design a methodology for designing cyber-

security awareness training to be delivered through e-learning. Next, we present a pilot study for the European Parliament election 2019 and an evaluation in Section 4 before we conclude and assess results in Section 5.

Acknowledgments We would like to thank the employees Copenhagen municipality's election office, the Ministry of Social and Internal Affairs, and KL, the association and interest organization of the 98 Danish municipalities.

2 The Human Factor

The attack surface of any system includes technical as well as human components. No system is stronger than its weakest component [5], and arguably, human performance is recognized as a critical part of securing critical infrastructure [5]. Depending on the adversary's objective, social engineering will always be considered as one way to achieve the goal: As opposed to technical cyberattacks that exploit vulnerabilities and always leave traces in log files or other media, social engineering is considered a viable alternative which allows adversaries to break a perimeter and operate somewhat undetected. In general, it is also more difficult to attribute a social engineering attack to an adversary. Therefore, measures to prevent or decrease the negative impacts of cybersecurity breaches must include all processes, policies and actors involved [7]. Technology alone cannot create a secure environment, since human factors are an integral part of any system, for example, during configuration, operation, or use. According to 2020 Verizon Data Breach Report social attacks are used in 22% of all cases recorded. These attacks are almost evenly split into phishing and pretexting attacks [4].

There are many factors that influence the security behavior of users i.e. the user's respective rank in an organization, their respective personal values, and their common sense regarding security [15]. Users are often not aware or do not consider the vulnerabilities in an organization, they make mistakes or are tricked into giving away sensitive information [1]. Therefore, common sense regarding security in an organization must be taught [15] and training in cybersecurity awareness is an important part of creating a security culture [1].

However, there seems to be a problem with existing cybersecurity awareness training as it does not change behavior as expected [2]. There are several reasons that this is the case. Firstly, cybersecurity awareness training is often designed as too general without a clear target group in mind, leading to users not finding it relevant. Secondly, incorrect assumptions about the targeted users and their skills and motivation tend to make cybersecurity awareness training too general.

3 Training Design Methodology

Next, we describe a methodology for how to create cybersecurity awareness training that avoids the above mentioned factors by tailoring training to a well defined target group and focusing the training content on what the target group need to know and nothing else. The methodology consists of five steps, which are summarized in Figure 1.

1. Target group
   - Define the target group, target setting, tasks, and responsibilities
2. Risk assessment
   - Define the adversarial environment
   - Define assets, including physical and logical, and processes
3. Threat modeling and risk analysis
   - Model the entire socio-technical system using CORAS/attack trees
   - Derive and prioritize potential attacks
   - Derive the attack surface and tailor it for the target group.
4. Training materials
   - Base training on knowledge gained from (1.-3.)
   - Create an e-learning platform
   - Consider using videos, audio, games as part of the training
5. Evaluate training

Fig. 1. Training Design Methodology

3.1 Target Group

The first step of creating good cybersecurity awareness training is to identify and characterize the target group, the target setting, and the target group's tasks and responsibilities in this setting. This can be achieved by ethnographic studies, long-time observation of work practices, and study of available procedures and documents. Usually, it is not sufficient to base this analysis only on printed materials, as common work practices often deviate from the described processes. A target group must be homogeneous, meaning all members should be assigned the same tasks and the same responsibilities. Heterogeneous target groups are not considered in this paper.

3.2 Risk Assessment

The next step is to identify assets and processes that are at risk, and define the security policies that should be enforced [3]. A good starting point for the risk assessment is to explore notions such as confidentiality, integrity, and availability, and refine them on demand. It is absolutely crucial that the target group identifies with this assessment. The cybersecurity training must be perceived as relevant by the target group for it to be effective.

A part of the risk assessment is the attack surface of the infrastructure, for which cybersecurity assessment training is to be offered. This presupposes a clear picture of the adversary's capacity and the adversary's objective. The attack surface includes all aspect of the infrastructure to be protected, including technology, networked computing equipment, air-gapped equipment, access control, cryptographic key distributions, physical access etc.

With the risk assessment in place, the next step is then to identify the weak points in the infrastructure that an adversary could exploit and to define the

role of the human to detect attacks and protect assets and processes. These insights and this knowledge form the basis of understanding of the infrastructure and feeds into the design process of the training materials, of which attacks participants should learn to spot, and which procedures they should learn follow to neutralize threats effectively.

3.3 Threat Modeling and Risk Analysis

In our experience, modern threat modeling tools, such as CORAS [16], attack trees [17] or even attack-defense trees [14] are useful tools to explore the threat model of any socio-technical system in a systematic and complete way. The CORAS method is a defensive risk analysis approach where the Unified Modeling Language (UML-diagrams) is used to model the target of the analysis. Unwanted behaviors are drawn as threat scenarios. The CORAS method comes with tool support, in particular, there exists a tool that supports drawing and analyzing diagrams. Alternative ways of conducting security analyses and modeling threats are described in this survey article [11]. In this paper, however, we focus on attack trees as a modeling tool.

An attack tree is a mathematical tree-like structure that organizes threats and attacks against a system. The root of the tree comprises the goal for the adversary, and the leaf nodes denote the different actions an adversary can execute to achieve this goal. Each node in a tree can be seen as a subgoal. The disjunctive "OR"-node represents alternatives, i.e. if *one* of the subtrees is successful then so is the subgoal. In contrast, the a subgoal rooted in a conjunctive "AND"-node is successful if an only if *all* subtrees are successful. There are also other variants of attack trees, that could in theory be considered, for example those supporting sequential conjunctions. The methodology presented here applies as well. The visual representations of "OR"-nodes, "AND"-nodes and leaf-nodes are depicted in Figure 2.

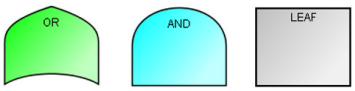


Fig. 2. Explanation of nodes in attack tree

Attack trees are known for their ability to express socio-technical systems and model human factors. We will be using them as well in our pilot study for securing polling stations during the European Parliament election 2019 that we describe in the Section 4.

3.4 Training Materials

Next, we identify the critical elements of the analysis and translate the attack tree into suitable training materials. We proceed in four steps, tagging, normalizing, prioritizing, and finalizing.

Tagging: When normalizing an attack tree, all information about the structure of the inner nodes, i.e. OR and AND nodes is lost. In practice, however, it is useful, to tag such inner nodes with keywords that help structure the content of the training materials, and collect them during the normalization procedure. Possible tags include, for example, social engineering attacks, man in the middle attacks, attacks against air-gapping, SQL-injection attacks, cross-site scripting attacks, buffer overflow attacks, and so on. An example of tagging can be seen in Figure 3.

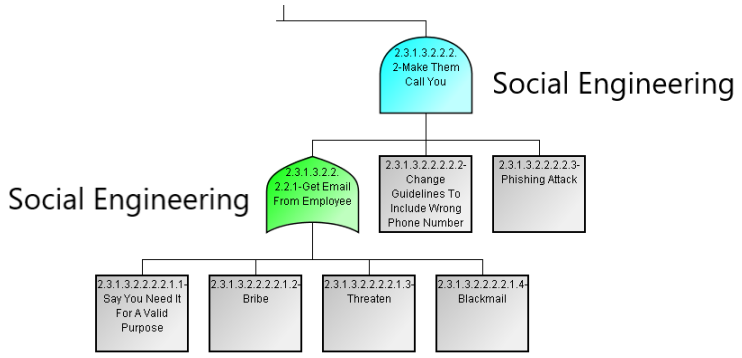


Fig. 3. Example of tagging sub-trees

Normalizing: Hereafter, the attack tree is normalized as to create a list of attack-chains in plain text. Attack-chains only include leaf nodes. Correspondingly the normalization procedure is augmented, to derive an additional tag-chain, of all of the tags that were encountered while constructing the attack chain. Below A and T are normalized attack-chain/tag-chain pair displaying the fragments derived from the attack tree depicted in Figure 3:

$A = \{\ldots$
Say You Need It For A Valid Purpose,
Change Guidelines To Include Wrong Phone Number,
Phishing Attack
$\ldots\}$

$T = \ldots$, Social Engineering, $\ldots$

The above step should result in a number of attack/tag-chains pairs. Duplicate attack chains should be removed while their tag-chains should be merged.

Prioritizing: Next, we identify precisely the topics that should be covered in the training materials. We therefore correlate the attack-chains with the tasks the target group is in charge of to determine what parts of the attack-chain, if not all, need to be included. It is critical for the training to be effective to educate the target group exactly in the topics they need to know - nothing more and nothing less. We use the tag chains as a guide to structure and organize the material.

Finalizing: In this last step, we create new or update existing training materials to create a consistent product. Recall that the success of effective training is to make sure the target group attains three levels of awareness of security risks, namely perception, comprehension and projection [21]. We propose to use e-learning as platform for the training, since interactive and adaptable material i.e. videos, also called hyper media-based material, can lead to effective cybersecurity training [21] and motivation for learning through such a platform tends to be high. Prior research has shown that video-based training is preferred over other methods and yields better results [1, 18]. The length of the video is important to get the participants engaged, and a study shows that videos that are 0-3 minutes have the highest engagement[9]. The training videos developed should train the target group to observe, identify, react, and defend against the individual steps laid out in the attack chains. Training material can be rearranged and reused for other target groups.

3.5 Evaluating E-learning

The final step of our methodology is that of evaluation. It is good practice to document the effects of security awareness training, to analyze the training objectively, and to measure the efficiency and effectiveness of it. Evaluation can help create a common understanding about the human factor defense capabilities, which areas of understanding among the target group are sufficient, and identify weaknesses that need to be strengthened [8].

Choosing an evaluation model to evaluate e-learning is dependent on the scale and the time frame of the e-learning. The state of the art is described in an article by Tripathi et al [23] where four different evaluation models are described in depth. We found that more models could be used in our case, and many of the models don't differ that much when measuring short-term effects, as we do. If we had to measure long term, we would have to go back and look at the evaluation models again. The two best models for our purpose are CIRO or Kirkpatrick's model of evaluation.

The CIRO model does not take the behavior of the learners into account and is, therefore, thought to be better suited for management focused training rather than for people working on lower levels of organizations [23, 24], therefore we chose to use Kirkpatrick's model of evaluation.

Kirkpatrick's model of evaluation was introduced in 1959. The model evaluates outcomes of training programs at four levels: reaction, learning, behavior and results. *Reaction* addresses how the participant felt and reacted to the training experience. *Learning* measures to which extent knowledge has increased and

how intellectual capability has changed from before the training. *Behavior* measures how the participant has changed behavior and applied the learning. *Results* addresses how the improved performance of the participant affect organizations [13].

Kirkpatrick's model is applied after training. The model is popular and still widely used among organizations. The main strength of the model is the focus on behavioral outcomes of the participants [13, 23].

Quizzes can be used to measure learning in Kirkpatrick's model. A quiz can be thought of as a survey, i.e. a quantitative method to collect data. The quiz, which must be taken both before and after training, consists of closed-ended questions. Participants can choose from a set of answers, where either one or more are correct. Participants can answer closed-ended questions fast and they can get instant feedback when they have taken the quiz. Another reason for using this type of question is that it is easy to analyze [19]. The quiz must be constructed in such a way that it measures the three levels of security awareness.

A survey can also be used to measure reaction in Kirkpatrick's model. The survey to measure this level consists of questions answered by a likert-scale and open questions. The likert-scale questions should give an indication of how relevant the participants find the e-learning. The open questions can help to discover unforeseen findings, and are essential to understand how the target group perceive the training [19].

4 Pilot Study: Digital Election Secretaries in the Election Context

In connection with the European Parliament election conducted in Denmark on Sunday 26$^{\text{th}}$ May, 2019, a group of election officials employed by Copenhagen municipality, called digital election secretaries, partook in cybersecurity awareness training. The staff at each polling station includes one *digital election secretary*, who is responsible for all computer equipment that is used in a polling station, that is, a digital voter identification system and a digital results transmission system. In Denmark, ballots are not interpreted and stored digitally, only the result of precinct-level tabulation is. The scope of our pilot was limited to cybersecurity awareness training with respect to the digital voter identification system. It was the first time that election officials had received any role-based cybersecurity training to recognize and act on attacks happening at the polling stations. The objective of our pilot study was to measure the improvement of their cybersecurity awareness.

4.1 Target Group

Copenhagen municipality has 53 digital election secretaries, one for each polling station. The main responsibilities of this group is to secure the equipment at the polling station and the electoral register including all the data in the above mentioned register. The digital election secretaries are recruited within the workers

of the municipality and differ in age and background. Some have served in the role of digital election secretary several times before. Despite the demographic differences, the group is highly homogeneous in the tasks they perform on election day. They will spend election day in similar environments, the different polling stations, and work with the same kind of election technologies, including the electoral register.

4.2 Risk Assessment

We conducted a detailed risk assessment of the processes connected with the digital election secretaries on election day, and identified a set of potential objectives of a hypothetical adversary. We consider confidentiality, integrity, and availability in turn.

Confidentiality: We consider an attacker who aims to get unauthorized access to information. If published by the attacker, it would weaken the trust in the security of the election and violate this security goal. It is the digital election secretaries' responsibility to protect voters' data at the polling stations and will, therefore, be considered in our cybersecurity awareness training.

Integrity: We consider an attacker who could try to violate election integrity by voting multiple times with the goal to change the election result in his or her favor. This is very difficult achieve given the organization of a Danish national election as several checks and balances were put in place for this not to happen. For example, every voter receives a voting card in the mail which they will have to bring to the polling station. All voting cards will be kept until the end of voting day and then counted to validate the number of votes cast. Once a voter is identified in the polling station, the physical poll book or in the electoral register will be updated, the former only if the electoral register fails. However, it is the digital election secretary's responsibility to ensure that no one voted more than once, and will hence be considered in our cybersecurity awareness training.

Availability: The attacker's objective could be to weaken public confidence in the voting process, by trying to make headlines in the press or on social media. To succeed, the attacker would have to break one or more security goals, for example, by rendering the electoral register at a polling station unavailable/unusable. To protect this asset, again, lies within the responsibilities of the digital election secretary and will hence be considered in our cybersecurity awareness training.

4.3 Threat Modeling

Based on the analysis in the previous section, we focus on all security goals, in particular an attacker's intent to weaken public confidence. We exclude insider attacks from our threat model. To succeed, the attacker would have to break one or more security goals, and it does not matter which one(s). With this objective in mind, we develop an attack tree of the election system from the vantage point of a digital election secretary.

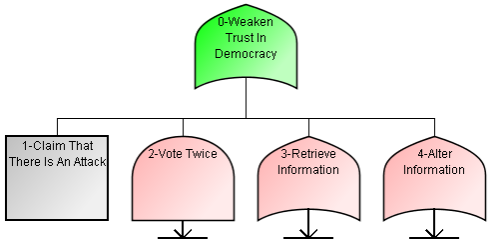


Fig. 4. Root and first level of the attack tree

Together with election experts from Copenhagen municipality, we identified 88 possible attack scenarios leaving us with an attack tree too large to include in this paper. The full attack tree can be found on the project's homepage[1]. Figure 4 depicts the top two levels of the attack tree. The leftmost singleton subtree (shaded in grey) states that a possible attack would be an attacker crying wolf and claiming that the election is under attack. Clearly, the digital election secretaries cannot stop people from lying, but still, such circumstances may arise, and the digital election secretary would need to know how to react. Hence, this must be a part of the cybersecurity awareness training.

The other three subtrees, describe ways on how an attacker could conceivable vote twice, gain access to privileged information, or alter the information stored in the electoral register. In the interest of space, we comment only the second subtree that is depicted in Figure 5. In our estimation, this attack is highly hypothetical and very difficult to execute. The nodes of the subtree are largely self-explanatory, except perhaps the unit that is called PCA, which refers to the laptop named "A" that contains the binding version of the digital electoral roll. In general, the polling place consists of several (through wired Ethernet) networked laptops. This network is not connected to other networks including the Internet during operation, but has been during configuration.

4.4 Training Materials

In our pilot study, we considered the entire attack tree[1], tagged the inner nodes, normalized to obtain attack/tag-chain pairs, prioritized them, and used this knowledge as input for the design of training materials. The training materials, which were created throughout a two months period, consist of an e-learning website with several modules and videos. The course page is online an can be accessed under `https://valgsikkerhed.dk`.[2] All 53 digital election secretaries

[1] See `https://www.demtech.dk/training/`

[2] The website is online, and anyone interested can make an account and access the teaching materials. Note, that the website is only in Danish.

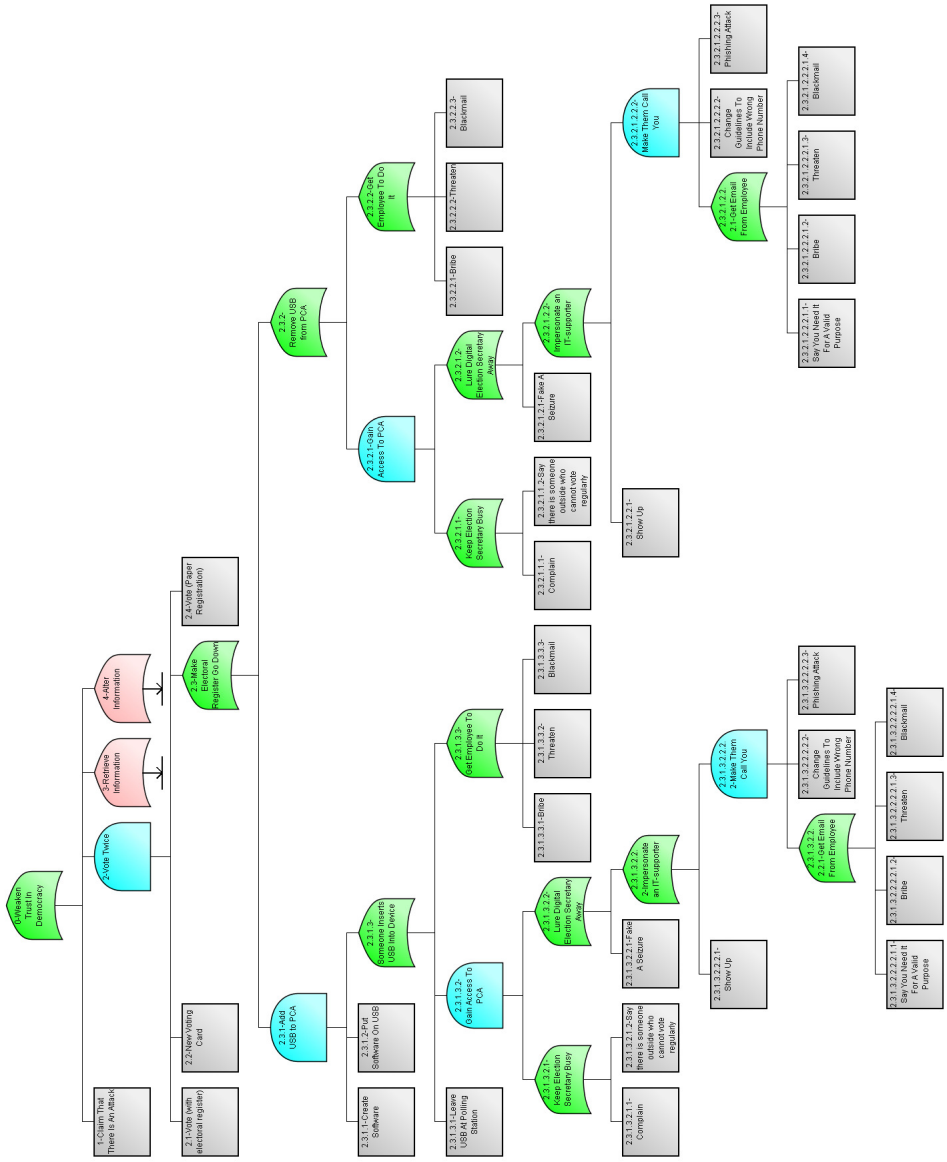


Fig. 5. Subtree 2: Vote twice

were invited to complete the e-learning course at their own pace and in their own time. Participating in the training was not mandatory.

All potential attacks are based on social engineering techniques aiming to coerce employees to retrieve desired confidential information or execute an attack on behalf of the adversary. Some potential attacks include also elements of man-in-the-middle attacks. Our training material therefore includes modules aimed to explain both, social engineering and man-in-the-middle attacks. The video on man-in-the-middle discusses devices that should not be present at polling stations, and how to react if they are spotted. The social-engineering videos focus on attacks that could be conducted before election day or at polling stations, i.e. exploiting common human traits resulting in that employees give access to confidential information to people with authority, follow instructions in phishing e-mails or gain access to any of the networked PCs in particular PCA, by creating a distraction. The training materials even include guidelines on how to calm worried voters in the case of an imminent cyberattack.

4.5 Evaluating E-learning

Learning outcome To evaluate if the digital election secretaries had gained cybersecurity awareness, they were tested both before and after the training with the same questionnaire.

The questionnaire was designed in such a way that each level of awareness was covered by more than one question. It is designed with reaction and learning levels from Kirkpatrick's model in mind. Since we are not measuring long term effects, there is no reason to evaluate the participants changed behavior nor how their changed behavior affect the organizations they work for.

77.4% of the target group signed up to the platform but only 71.7% completed the e-learning training. That means that 92% of those who started the e-learning finished it. The distribution of the grades can be seen in Figure 6.

A paired t-test can be used to check if the learning is effective by comparing before and after observations. This is done to show that there is statistical evidence that the difference of the means between the paired samples is significantly different from zero [12].

In order to do a paired t-test on this small data set, one need to make sure that the data is normally distributed. This was tested with a Q-Q Plot, that can be seen in Figure 7. It shows that the data is, indeed, normally distributed.

The t-test is run with the following hypotheses:

$$H_0 : \mu_{\mathrm{d}} = 0 \tag{1}$$

$$H_1 : \mu_{\mathrm{d}} \neq 0 \tag{2}$$

In other words, H_0 assumes that the security awareness training has no effect on the mean and the alternative hypothesis, H_1 assumes that there is a difference.

The grades before and after were used to run the paired t-test. Since the participants can also score less than before we do a two-tailed test.

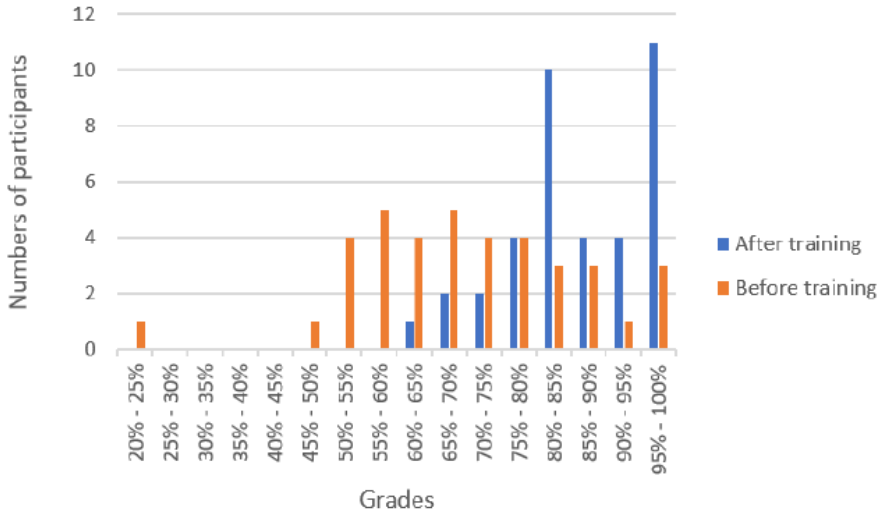


Fig. 6. Distribution of the grades before and after training.

SPSS is a widely used statics application created by IBM [10] and was used to run the paired t-test. The test was run with $\alpha = 0.05$. The result of the test is shown in Figure 8. As can be seen in the figure the digital election secretaries score, on average 1.6 points higher in the latter quiz. It also shows that the Sig.(2-tailed), also called the p-value, is much smaller than α. This means that we can reject the null-hypothesis.

To evaluate the awareness layers, as mentioned in [21], they are translated to this specific context. Perception is getting the digital election secretaries recognizing and understanding potential security risks in an election. Comprehension is to teach them to take in information from multiple sources, interpret them and be able to pass on information that can help others actors in the election. Projection is for them to be able to prevent future attacks.

The results in Table 1 shows that all the three levels of successful security awareness training was reached for the election officials that participated in the e-learning training.

Awareness Level	Before	After	t(37)	p
Perception	M=1.45, SD=0.57	M=1.64, SD=0.49	-2.113	0.041
Comprehension	M=2.24, SD=0.75	M=2.237, SD=0.41	-4.112	0.00209
Projection	M=3.3, SD=0.89	M=4.15, SD=0.59	-5.929	0.0000007835

Table 1. Table of scores for the three levels of security awareness

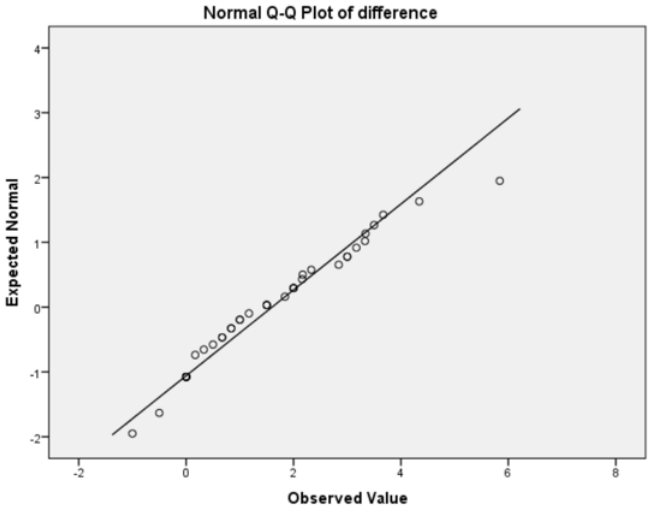


Fig. 7. Q-Q Plot of data.

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	before - after	-1.59789	1.50898	.24479	-2.09388	-1.10191	-6.528	37	.0000001217

Fig. 8. Paired t-test results

An analysis on the time spent on the quizzes, shows that the participants spend on average 4 minutes less on the latter quiz. However, we can not draw any conclusion by that in itself as we decided to give the participant the freedom to do the training at their own pace. Hence we have not measured the individual questions in the quizzes and, therefore, do not know how which questions they spend less time on in the latter quiz. We leave this to future work.

Participant evaluation 52.6% gave feedback on their experience of the e-learning. 85% said that they felt they had either gained new knowledge or refreshed knowledge they already had. 85% also said that they thought the content of the e-learning was good and relevant for their duties as digital election secretaries.

5 Conclusion

This paper provides a methodology for designing and delivering cybersecurity awareness training for short-term retention. The methodology was tested on 53 digital election secretaries who were deployed to 53 polling stations in Copenhagen municipality during the European Parliament election in 2019. We have evaluated the training using Kirkpatrick's model of evaluation found it to be effective. We are certain that our methodology carries over directly to the other 97 Danish municipalities, as their elections are organized in a manner similar to those in Copenhagen. We also believe that it is applicable beyond Denmark, as other European countries use digital voter identification and results transmission systems. The training material must be updated and adjusted to the respective target audiences and the specific technologies in use in a particular location.

Through understanding of the target group, the adversarial environment and the attack surface it was possible to create training materials tailored toward the job of the digital election secretaries. The training was delivered through a custom-made e-learning platform, containing short videos to deliver individual modules derived from potential attacks identified using attack trees. After training, we demonstrated that the target group reached all levels of successful security awareness: perception, comprehension and projection. In addition, a training evaluation showed that (1) the digital election secretaries perceived the training to be both good and relevant for their work on election day and (2) they also felt that they gained or at least refreshed their cyber security knowledge.

In future work, we would like to collect more evidence that this is a sustainable methodology to design and conduct cybersecurity awareness training. Firstly, we would like to compare a group that has been trained with a group that has not been trained to identify the difference, if any. Secondly, it would be interesting to analyze time spent on each task and correlate with retention of the concepts associated with each task. Hence do a more granular evaluation of the cyber security awareness training. Thirdly, we would like to conduct similar awareness training with the same group of digital election security at future elections to identify trends in the evaluation data. Fourthly, we would like to broaden the pilot to the whole of Denmark to examine if we can reproduce our results. Lastly, we believe that it would be interesting to apply the same methodology to elections in other countries and/or broaden cybersecurity awareness training beyond the elections to other sectors as well to study the robustness of the methodology.

References

1. Abawajy, J.: User preference of cyber security awareness delivery methods. Behaviour & Information Technology **33**(3), 237–248 (2014)
2. Bada, M., Sasse, A.M., Nurse, J.R.: Cyber security awareness campaigns: Why do they fail to change behaviour? arXiv preprint arXiv:1901.02672 (2019)
3. Basin, D., Schaller, P., Schläpfer, M.: Applied Information Security: A Hands-on Approach. Springer (2011)

4. Bassett, G., Hylender, C.D., Langlois, P., Pinto, A., Widup, S.: 2020 Verizon Data Breach Report (2020)
5. Boyce, M.W., Duma, K.M., Hettinger, L.J., Malone, T.B., Wilson, D.P., Lockett-Reynolds, J.: Human performance in cybersecurity: A research agenda. In: Proceedings of the Human Factors and Ergonomics Society annual meeting. vol. 55, pp. 1115–1119 (2011)
6. Dhillon, G.: What to do before and after a cybersecurity breach? Kogod Cybersecurity Governance Center, American University, Washington, DC (2015)
7. Dutton, W.H.: Fostering a cybersecurity mindset. Internet Policy Review **6**(1), 110–123 (2017)
8. Eminağaoğlu, M., Uçar, E., Eren, Ş.: The positive outcomes of information security awareness training in companies–a case study. Information Security Technical Report **14**(4), 223–229 (2009)
9. Guo, P.J., Kim, J., Rubin, R.: How video production affects student engagement: An empirical study of mooc videos. In: Proceedings of the first ACM conference on Learning@ scale conference. pp. 41–50. ACM (2014)
10. Hinton, P.R., McMurray, I., Brownlow, C.: SPSS explained. Routledge (2014)
11. Hussain, S., Kamal, A., Ahmad, S., Rasool, G., Iqbal, S.: Threat modelling methodologies: a survey. Sci. Int.(Lahore) **26**(4), 1607–1609 (2014)
12. Kent State University: SPSS tutorials: Paired samples t test (2019), `https://libguides.library.kent.edu/spss/pairedsamplesttest`
13. Kirkpatrick, D., Kirkpatrick, J.: Evaluating training programs: The four levels. Berrett-Koehler Publishers (2006)
14. Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Foundations of attack-defense trees. In: Proceedings of the 7th International Conference on Formal Aspects of Security and Trust. pp. 80–95. FAST'10, Springer-Verlag, Berlin, Heidelberg (2011)
15. Leach, J.: Improving user security behaviour. Computers & Security **22**(8), 685–692 (2003)
16. Lund, M.S., Solhaug, B., Stlen, K.: Model-Driven Risk Analysis: The CORAS Approach. Springer (2010)
17. Mauw, S., Oostdijk, M.: Foundations of attack trees. In: Proceedings of the 8th International Conference on Information Security and Cryptology. pp. 186–198. ICISC'05, Springer-Verlag (2006)
18. Merkt, M., Weigand, S., Heier, A., Schwan, S.: Learning with videos vs. learning with print: The role of interactive features. Learning and Instruction **21**(6), 687–704 (2011)
19. Neuman, W.L., Robson, K.: Basics of social research: Qualitative and quantitative approaches. Power **48**, 48 (2007)
20. Sciutto, J.: How one typo helped let russian hackers in. CNN, June **27** (2017)
21. Shaw, R.S., Chen, C.C., Harris, A.L., Huang, H.J.: The impact of information richness on information security awareness training effectiveness. Computers & Education **52**(1), 92–100 (2009)
22. Topno, H.: Evaluation of training and development: An analysis of various models. Journal of Business and Management **5**(2), 16–22 (2012)
23. Tripathi, J., Bansal, A.: A literature review on various models for evaluating training programs. IOSR Journal of Business and Management **19**(11), 1 (2017)
24. Warr, P., Bird, M., Rackham, N.: Evaluation of management training: A practical framework, with cases, for evaluating training needs and results. Gower Press (1970)

Confidence and Costs

The Oxymoron of the Internet Voting in Illiberal and Hybrid Political Contexts

Bogdan Romanov[1] and Yury Kabanov[2]

[1]University of Tartu
Tartu, Estonia
romanovbogdan4@gmail.com

[2]National Research University Higher School of Economics
St. Petersburg, Russia
ykabanov@hse.ru

Abstract. This paper explores the phenomenon of e-voting, in particular, new i-voting technologies, within the context of hybrid and authoritarian political regimes. While e-voting and i-voting are not particularly widespread, more and more illiberal countries are implementing these innovations, which has been overlooked in the academia so far. The paper attempts to fill in this gap. Firstly, we provide a general overview of the problem and identify the key features of non-democracies adopting e-voting and i-voting. Secondly, we explore the case of Russia, a hybrid regime, which may become a role model for other countries in the near future. The research exposes the potential of e-voting, and in particular, i-voting as a tool for the regime stability and provides some avenues of the future research.

Keywords: I-voting — E-voting — Autocracies — Hybrid regimes

1 Introduction

While the impact of the Internet on authoritarian politics is an emerging topic [1], little attention in this context has been given so far to e-voting technologies. This is partly due to the fact that unlike online repressions [2] and even e-participation [3], the use of e-voting is not so widespread among hybrid and authoritarian political regimes [4].

At the same time, this situation might change soon. The capacity of non-democracies to utilize the Internet is rapidly increasing [1; 5]. Moreover, the COVID-19 pandemic may give another powerful impetus for new practices of voting [6]. In this regard, it becomes vital to explore the possible causes and effects of such innovations, considering the social and political contexts in which they are implemented.

This paper attempts to preliminarily address this issue and answer the question as to whether e-voting technologies can be embedded into the resilience strategies of hybrid and authoritarian regimes. The study mostly deals with Internet voting, or i-voting, when a person casts a vote via the Internet or a mobile device [7, 8]. Yet,

some of the conclusions can be applicable to other e-voting practices and *new voting technologies* [8].

To answer the question, we first overview the general trends of e-voting and i-voting diffusion in non-democracies, to see if there are certain clues to understand when and why such countries introduce these innovations. Secondly, we explore the case of Russia, a hybrid regime, which is quite active in promoting Internet voting from 2019 onwards. We outline what happened during the 2019 elections for the Moscow State Duma and present some insights into the future development of i-voting in the country.

The ultimate goal of this paper is to provoke the discussion on the functions e-voting and i-voting may play in different social and political contexts. Furthermore, the paper overviews a preliminary framework to analyze future initiatives of that kind in authoritarian and hybrid political regimes.

2 E-Voting and I-Voting: Not So Democratic Anymore?

2.1 Patterns of Diffusion

The implications of ICTs within political regime dynamics heavily depend on the position we take within the debate between the Internet optimists and pessimists. The former claim that the Internet has a democratizing potential, as it creates new opportunities for free communication and political mobilization [9; 10; 11]. From this viewpoint, e-voting technologies increase the effectiveness of elections by reducing the human factor and making them more accessible for citizens [8; 12; 13].

Yet, there is more evidence to support the pessimistic view, claiming that the Internet has been successfully incorporated into authoritarian strategies of survival and governance [1; 2]. For example, autocrats use e-participation to get information and boost legitimacy [4; 14; 15], whilst the democratizing effects of the Internet are hindered by restrictions and prohibitions [2; 16].

E-voting and i-voting are rarely discussed in this context. However, scholars have recently started to pay more attention to the political and social context, in which new technologies are introduced [17; 18]. Cheeseman et al. note that "even the most advanced forms of technology depend on human input to no lesser extent than manual election management and are in certain cases actually more vulnerable to manipulation. Significantly, this risk is exacerbated by the difficulty of monitoring "black box" digital processes, especially in counties in which the ruling party is able to exert control over the electoral commission" [19, p. 1411]. This observation is in line with Oostveen and van den Besselaar, who suggest that "e-voting could possibly and relatively easily be used to reinforce control by the ruling party" [20, p. 19].

This viewpoint is further amplified by several case studies, tracing the experiments with e-voting and i-voting in hybrid and authoritarian regimes, including Russia [4], Pakistan [21], Oman [22], the UAE [23] and Kazakhstan [24]. Though these practices are still rare, such countries are no less interested in new electoral technologies.

Why do such countries implement e-voting (i-voting)? There are different answers to this question, and each state might have its own rationale. A possible factor that

would obviously facilitate e-voting (i-voting) is an institutional one. For instance, explaining the Internet elections of 2012 in Russia, Toepfl argued that since the elections had been practiced for a long time in the country, "[a]n additional flurry of semi-competitive Internet votes to fill advisory bodies to the government was thus, apparently, not perceived as a major threat" [4, p. 969]. In other words, new technologies are to be expected in the *electoral authoritarian regimes*, accustomed to voting procedures and familiar with the methods of electoral manipulations [25; 26].

Of course, other regime types may also benefit from Internet voting. For example, monarchies, as shown by Kneuer and Harnish [27], are counterintuitively active in promoting new participatory technologies. This can be explained by the fact that their political structure lacks institutionalized participation channels [27]. By analogy, i-voting seems a safe and easy to control alternative to collect citizens' preferences.

Another possible factor of e-voting (i-voting) adoption is the legitimacy quest. Cheeseman et al. argue that digital electoral technologies are often associated with the idea that they "boost the process's legitimacy – and hence that of the elected government" [19, p. 1398]. This idea falls into the emerging research area on legitimation as a source of authoritarian stability [28; 29; 30]. There are different legitimation strategies that autocrats may use. For instance, von Soest and Grauvogel [31] distinguish between six types of legitimacy claims: *foundational myth, ideology, personalism, procedures, performance, international engagement*, and measure the values of such claims for various types of non-democracies. Their findings suggest that certain legitimacy claims (e.g. *performance*) are of equal importance to most of the regime types. This methodology has been refined in the new *Varieties of Democracy* dataset [32; 58], where the classification encompasses four types of legitimation strategies: *performance, rational-legal, ideology* and *the personality of the leader*. In this regard, it can be speculated that i-voting will become more widespread within the countries that actively employ procedural legitimation strategies, "based on the carrying out of elections and other rule-based mechanisms for handing over power through 'orderly' process" [31, p. 291].

There is not enough evidence so far to prove these propositions, but the data we have demonstrate their plausibility. As shown in Table 1, different e-voting technologies can be found in both hybrid regimes and consolidated autocracies. According to the IDEA,[1] there are 10 multi-party and 2 monarchy regimes currently using e-voting, including 4 countries that use i-voting. This indeed suggests that e-voting can be easily adapted by the states already holding elections, but not exclusively, as other regimes may use ICTs to expand their regime resilience repertoires. It is notable that both monarchies use i-voting, in line with what has been said previously. The situation with legitimation is more complicated since many countries employ multiple strategies. Yet, again, most of the countries score 3 and higher on more "democratic" legitimation, like the *rational-legal* and *performance* legitimation [32].

[1] https://www.idea.int/data-tools/data/icts-elections

Table 1. E-Voting and I-Voting in Non-Democracies

Country[1]	Regime Type[2]	Legitimation Strategies[3]			
		Rational – Legal	Performance	Leader	Ideology
Armenia*	Multi-party	2.66	**3.24**	2.46	1.13
Bangladesh	Multi-party	0.76	**3.45**	3.82	3.33
Bhutan	Multi-party	**3.62**	**3.12**	2.24	2.61
Congo, DR	Military	1.53	1.5	2.8	1.34
Fiji	Multi-party	**3.3**	**3.52**	3.19	3.06
Honduras	Multi-party	2.83	**3.21**	0.63	2.37
Iran	Other	2.19	2.38	3.62	3.92
Iraq	Multi-party	2.13	1.21	3	3.29
Kyrgyzstan	Multi-party	**3.19**	**3.36**	0.85	0.93
Oman*	Monarchy	**3.19**	2.77	3.73	3.18
Pakistan*	Multi-party	**3.63**	**3.63**	1.72	1.81
Russia **	Multi-party	**3.29**	2.82	3.76	2.78
UAE*	Monarchy	**3.02**	**3.51**	3.28	3.27
Venezuela	Multi-party	1.15	3	2.96	3.65

Sources:

[1] Countries currently using e-voting. Source: ICTs in Elections Database. IDEA. URL: https://www.idea.int/data-tools/data/icts-elections.

[2] Typology of regimes by Wahman et. al [33], data for 2014.

[3] Legitimation strategies from the Varieties of Democracy Project, data for 2018 [32; 58].

* Countries, using i-voting, according to the IDEA Database

** Russia introduced the opportunity for i-voting in 2020

Here we may conclude that while the topic of e-voting (i-voting) in non-democracies is new, the emerging studies and new empirical cases suggest that this issue should be taken seriously. Like other IT-enabled and digital tools, voting technologies are being adapted and utilized by authoritarian and hybrid countries. Our preliminary analysis suggests that the regime institutional configurations and legitimation strategies are promising variables to explain this trend. However, it requires further testing using more rigorous techniques.

2.2 Effects

The question of what impact e-voting (i-voting) has on the regime dynamics is hard to answer empirically so far. Evidently, the effect will depend on which technology is used. For example, some e-voting applications, for instance, optical scan voting systems, were reported to prevent some falsifications [34]. Regardless, they are unlikely to provoke any democratization, since when the technology is fully controlled by the incumbent, it will rather *reinforce* existing power relations [35].

To know what effects new voting technologies may have on non-democracies, we need to understand why they organize them in the first place. Although there are various explanations, their basic functions are, first, to show the strength and legitimacy of the regime, and secondly, to obtain information about citizens' preferences [36, 37]. These goals may be contradictory, as they require different degrees of electoral

manipulation and fairness of the results [38]. And here the emerging i-voting technologies, in comparison to paper-based and even other e-voting options, appear to be more valuable in maintaining authoritarian practices, especially when it comes to *remote voting* [7].

First, Internet voting is assumed to increase turnout, since citizens may take part in elections from wherever they are [7, 39]. This assumption has found little support in reality, but it is still very popular with researchers and policymakers [40; 41, 42, 43]. In non-democracies, where the academic discourse towards e-voting is more positive [20], this claim can be even more profound.[2] This feature of i-voting may be of particular relevance to those regimes employing the high turnout strategy that "confers legitimacy, demonstrates the regime's invincibility, and allows the regime to gather information on societal grievances". [44, p. 1]

At the same time, in usual circumstances this strategy may be risky in various ways [44, p. 28], leading, for example, to the increase of votes for the opposition [45]. We argue that unlike traditional voting practices, i-voting may substantially reduce such risks, if the technology is fully controlled by the incumbent. Even in democracies, remote i-voting raises concerns about proper voters' authentication, absence of coercion and accurate votes' calculation in i-voting are raised in democracies [39]. As put by Goos et al., "there is no technical solution available which would guarantee transparency, accessibility, resistance to intimidation and vote selling and, last but not least, resistance to fraud or errors" [7, p. 136]. It is clear that authoritarian or hybrid regimes can demonstrate more instances of such malpractice.

If the general principles of i-voting *end-to-end verification* [46] are violated, the incumbents may substantially increase their capacity to control the elections at any stage. They, first, may benefit from the high turnout to legitimize their rule (naturally or by cheating). Secondly, as the votes do not need to be falsified when cast (it can be done in the later stages of votes' tallying or publication), the incumbents may get rather objective information about citizens' preferences.

3 Internet Voting in a Hybrid Regime: The Case of Russia

3.1 Framework: Electoral Authoritarianism goes Digital

There are different frameworks allowing the estimation of the integrity of i-voting, but most of them deal with technical questions of verifiability, privacy, secrecy etc. [46] Such problems occur rather often [47, 48] and they should not necessarily be regarded as a move away from democracy. What may potentially make them a repeated practice of non-democratic politics is the social and political context in which they happen. Thus, several additional theoretical frameworks will be of use.

Firstly, the development of Internet voting in a non-democratic country heavily depends on the level of control a government has over the online space. The Internet still poses a threat to authoritarian survival [49], and those risks should be mitigated

[2] Krivonosova I.: E-voting in Moscow: A Gratuitous Gimmick—RIDDLE. (n.d.). Retrieved June 2, 2020, from https://www.ridl.io/en/e-voting-in-moscow-a-gratuitous-gimmick/

before new technologies are introduced. As the literature suggests, many Internet-savvy dictators follow a *double strategy*: by developing online participatory tools they simultaneously strengthen their censorship, filtering, or other repressive capacities [3]. Such policy usually refers to as the *networked authoritarianism*, i.e. "[w]hen an authoritarian regime embraces and adjusts to the inevitable changes brought by digital communications" [16, p. 33]. Thus, to understand the perspectives of i-voting, we need to look at the general *capacity* of a country to control the Internet for regime resilience [5].

Secondly, we need to explore the whole electoral process – from defining the positions to be filled in with elections, to the validation of results. As shown by the research on electoral authoritarianism, at every stage of the process, dictators have a variety of tools to manipulate the choice, which are by no means limited to falsification of results. Here we use the framework developed by Schedler, who proposes a *menu of manipulation* across seven steps of elections: (1) *the object of choice*; (2) *the range of choice*; (3) *the formation of preferences*; (4) *the agents of choice*; (5) *the expression of preferences*; (6) *the aggregation of preferences*; (7) *the consequences of choice* [25, p. 39].

These frameworks help to describe what we may call the *digital electoral authoritarianism*, i.e. one that utilizes online repression and electoral manipulations to hold i-voting for regime resilience.

3.2 2019 Moscow City Duma Elections and Beyond

Russia seems to be a good case to analyze the transformation of i-voting in a hybrid regime. On the one hand, it is usually referred to as *competitive* or *electoral authoritarianism*, which "employs unfair electoral practices to an extent that deprives elections of their primary functions of political choice and elite circulation, and reduces them to a mere tool of legitimization and mobilization of support" [50, p. 623]. This set of practices is changing over time, shifting to more subtle manipulations, like changing electoral formulae [51] or denying oppositional candidates of registration [52].

On the other hand, it is usually emphasized that the government control over the Internet is increasing over time in the country, including various types of control and legal regulations [53; 54]. The country falls into a *double strategy* [3]: despite restrictive measures on the Internet, the government actively promotes e-government and e-participation to engage citizens into public policymaking [55]. One of the first initiatives in e-participation was the *Russian Public Initiative* e-petitions portal [56], followed by more successful regional portals, like the *Our Petersburg* portal in St. Petersburg [57] or the *Active Citizen* in Moscow, which have not only become important consultative instruments [4], but also have prepared the ground for further policy innovations.

Both factors – the developed stage of the *network* and *electoral authoritarianism* – make Russia an obvious candidate to introduce Internet voting. Although, e-voting in

the country had already been operational in other formats,[3] for many years it was not the case for Internet voting, as such practices were rare and related only to advisory bodies [4]. Yet, from 2019 this agenda became profound in relation to Moscow City Duma (MCD) elections. However local this case is, as will be shown further, it may be also considered either a rehearsal for a massive introduction of i-voting or a model of how such online elections can be held in the future.

Internet elections in Moscow were held on September 8, 2019, as an experiment, which was initially proposed by a liberal journalist Alexey Venediktov and then formulated as a federal bill by the State Duma deputies.[4] Eventually, only three electoral constituencies were included in the experiment.[5]

There were different opinions regarding the purpose of i-voting implementations. The explanation of the state officials was quite in line with the *procedural legitimation.* For example, Valentin Gorbunov, the chair of the Moscow Electoral Commission, emphasized that i-voting was a consequent step of the digital economy, which "creates additional circumstances for the realization of active suffrage".[6] Apparently, the increase of turnout became the most important goal, as many Muscovites were used to "solves all their issues via smartphone".[7] Alexey Shaposhnikov, the chair of the MCD, made it rather explicit: "Everyone supposes that the larger turnout is, the more legitimate the elections are. I think that the option of distant voting allows raising the turnout tremendously."[8]

Unlike the government, the so-called *non-system opposition* did not perceive that innovation as a positive step, claiming that it would become another instrument of electoral manipulation for the ruling party.[9] Many experts were also concerned about i-voting integrity. For instance, Dmitry Oreshkin, a political scientist, argued that "when the electronic voting is introduced, you do not have any observers. There will even be no primary protocols… The disappearance of voting results can be now justi-

---

[3] Krivonosova I.: E-voting in Moscow: A Gratuitous Gimmick—RIDDLE. (n.d.). Retrieved June 2, 2020, from https://www.ridl.io/en/e-voting-in-moscow-a-gratuitous-gimmick/

[4] V Gosdumu Vnesli Zakonoproekt o Testiro-vanii Elektronnogo Golosovaniya v Moskve – Vedomosti. (n.d.). Retrieved June 2, 2020, from https://www.vedomosti.ru/politics/news/2019/02/26/795201-elektronnogo-golosovaniya

[5] Moskvichi s 3 po 9 iyunya smogut vybrat' okruga dlya provedeniya elektronnogo goloso-vaniya—Moskva—TASS. (n.d.). Retrieved June 9, 2020, from https://tass.ru/moskva/6489105

[6] Cel' eksperimenta po vnedreniyu distancionnogo elektronnogo golosovaniya na vy-borah v Mosgordumu – sozdat' dlya moskvichej dopolnitel'nye vozmozhnosti realiza-cii aktivnogo izbiratel'nogo prava. (n.d.). Retrieved June 9, 2020, from https://duma.mos.ru/ru/34/news/novosti/tsel-eksperimenta-po-vnedreniyu-distantsionnogo-elektronnogo-golosovaniya-na-vyiborah-v-mosgordumu-sozdat-dlya-moskvichey-dopolnitelnyie-vozmojnosti-realizatsii-aktivnogo-izbiratelnogo-prava

[7] Ibid.

[8] A.Shaposhnikov: Elektronnoe golosovanie pozvolit kolossal'no podnyat' yavku na vyborah deputatov Mosgordumy—Agentstvo gorodskih novostej «Moskva»—Informacionnoe agentstvo. (n.d.). Retrieved June 7, 2020, from https://www.mskagency.ru/materials/2884983

[9] Aleksej Naval'nyj—Oficial'no: U nas est' to, chego byt' ne dolzhno. «Elektron-noe golosovanie» polnost'yu skomprometirovano. (n.d.). Retrieved June 9, 2020, from https://navalny.com/p/6234/

fied by even a short circuit".[10] Other experts were also skeptical about the capacity of the government to ensure the integrity, secrecy and privacy of the procedure.[11]

In terms of Schedler, concerns related to i-voting arose not only with the technologies *per se*, but also with the political context in which such technologies were introduced. For instance, there was an issue with the *range of choice*, as several oppositional candidates had been denied registration, which caused a series of public protests.[12] Though this is an "offline" issue, the rules of registration are applicable to all candidates, hence denial of access might also limit the choice for voters. The same "offline" problem was with the *formation of preferences*, as the candidates from the ruling party were competing as independent candidates without any party affiliation.[13] In terms of *expression* and *aggregation of preferences*, the problems with i-voting included the issues with the *end-to-end verification*,[14] as well as cases of the system malfunction, during which voters could not cast their votes or their votes were not counted.[15]

Regardless of the technical and organizational issues, the major goal set by the policymakers seems to have been achieved. First, the majority of votes in all three districts was given to the independent candidates, affiliated with the *United Russia* party.[16] Though the general turnout was rather average (20-25 per cent), the i-voting turnout was much higher: out of 11 228 registered voters, about 92% have cast their votes.[17] It is not clear whether the introduction of Internet voting has contributed to the victory of any candidates, but there were expert accounts that although the level of manipulation had been low in general, "experimental districts … have turned out to be much more pro-government, than in Moscow generally".[18]

---

[10] Dnevnoj fal'sifikat. Dmitrij Oreshkin ob"yasnyaet, kak vvedenie elektronnogo go-losovaniya pomozhet vlastyam vyigriyvat' vybory. (n.d.). Retrieved June 9, 2020, from https://novayagazeta.ru/articles/2020/05/14/85376-dnevnoy-falsifikat

[11] «Okej, golosujte bumazhno» Aleksej Venediktov otvetil na kritiku internet-vyborov v Mosgordumu. My poprosili ekspertov proverit' ego zayavleniya—Meduza. (n.d.). Retrieved June 7, 2020, from https://meduza.io/feature/2019/07/01/okey-golosuyte-bumazhno

[12] Eksperty prokommentirovali otkaz v registracii kandidatov na vyborah v MGD - RIA Novosti. 31.07.2019. (n.d.). Retrieved June 9, 2020, from https://ria.ru/20190731/1557026300.html

[13] «Edinaya Rossiya» ne vydvinula ni odnogo kandidata v Mosgordumu No frakciya edi-norosov v stolichnom parlamente vse ravno budet —Meduza. (n.d.). Retrieved June 8, 2020, from https://meduza.io/feature/2019/06/13/edinaya-rossiya-ne-vydvinula-ni-odnogo-kandidata-v-mosgordumu

[14] Meriya Obeshchala Prozrachnoe Internet-Golosovanie v Moskve. V Itoge Ona Mozhet Opublikovat' Lyubye Rezul'ta-ty, i Proverit' Ih Nikto Ne Smozhet — Meduza. (n.d.). Retrieved June 2, 2020, from https://meduza.io/feature/2019/09/06/meriya-obeschala-prozrachnoe-internet-golosovanie-v-moskve-v-itoge-ona-mozhet-opublikovat-lyubye-rezultaty-i-proverit-ih-nikto-ne-smozhet

[15] Onlajn-Golosovanie v Moskve Dvazhdy Priostanavlivali Iz-Za Sboya - Novosti – Politika – Kommersant. (n.d.). Retrieved June 2, 2020, from https://www.kommersant.ru/doc/4086901

[16] Ibid.

[17] V Moskve Podveli Itogi Eksperimenta s Elektron-nym Golosovaniem — Rossijskaya Gazeta. (n.d.). Retrieved June 1, 2020, from https://rg.ru/2019/09/09/reg-cfo/v-moskve-podveli-itogi-eksperimenta-s-elektronnym-golosovaniem.html

[18] Urnoterapiya. Poluchiv psihologicheskuyu travmu god nazad, vlast' stala otsekat' negativnye scenarii na dal'nih podstupah k uchastkam. CHto iz etogo vyshlo. (n.d.). Retrieved June 9, 2020, from https://novayagazeta.ru/articles/2019/09/10/81915-urnoterapiya?utm_source=tg&utm_medium=novaya&utm_campaign=matematik-sergey-shpilkin-proanalizirova

The fact that the experiment was considered rather successful by the elites can be also indirectly proved by the decision to continue and expand this practice. In 2020, during the COVID-19 pandemic, the government decided to hold the Internet voting on the constitutional amendments in two regions – Moscow and Nizhegorodskaya oblast. As of 15 June 2020, about 548 thousand people have registered to vote online.[19] What is even more important, in May 2020 a federal law was introduced, allowing the citizens to vote by mail or online in the elections and referendums of all levels.[20] It is possible that i-voting will be used in several regions in September 2020 for the additional elections for the State Duma, with the goal to increase turnout and raise their popularity.[21]

Of course, technical issues with elections may happen, especially during an experiment, and they *per se* do not prove the intention of the government to increase the control over electoral process. But there are enough facts to conclude that this experiment will continue and become a full-fledged practice. This, in turn, will require thorough analysis of its causes and effects.

In sum, several conclusions can be drawn. First, Russia, being a hybrid regime that is based on electoral institutions, and to a larger extent on a *procedural legitimation*, with a vast repertoire of the reactive and proactive Internet controls, can be considered a perfect example of the general trend that we have previously outlined. What is more, the Russian case can become a role model for other countries that either update their electoral rules or introduce elections from scratch. Secondly, the case of 2019 MCD elections reveals a set of issues with the use of the i-voting, which are not necessarily related to the technologies, but also to the broader social and political context.

4 Discussion and Conclusion

This paper is a preliminary attempt to explore the issue of e-voting and, especially, i-voting diffusion in the countries other than liberal democracies. Having reviewed some recent events occurring in the world and Russia, in particular, we argue that there might be a new trend of emerging online voting technologies in the context of authoritarian and hybrid political regimes, which, in turn, will question the status of e-voting and i-voting as a purely democratic innovation. Non-democracies that have suitable institutional configurations and relevant legitimation strategies, may become more interested in these innovations in the near future, thus opening the agenda of

[19] Bolee 548 tysyach chelovek podali zayavki na uchastie v onlajn-golosovanii—Parlamentskaya gazeta. (n.d.). Retrieved June 7, 2020, from https://www.pnp.ru/social/bolee-548-tysyach-chelovek-podali-zayavki-na-uchastie-v-onlayn-golosovanii.html

[20] Federal'nyj zakon ot 23 maya 2020 g. N 152-FZ "O provedenii eksperimenta po organizacii i osushchestvleniyu distancionnogo elektronnogo golosovaniya v gorode federal'nogo znacheniya Moskve"—Rossijskaya gazeta. (n.d.). Retrieved June 7, 2020, from https://rg.ru/2020/05/25/fz-o-golosovanii-v-moskve-dok.html

[21] Elektronnoe golosovanie na dovyborah v Gosdumu projdet v Kurskoj i YAroslavskoj oblastyah. (n.d.). Retrieved June 13, 2020, from http://actualcomment.ru/elektronnoe-golosovanie-na-dovyborakh-v-gosdumu-proydet-v-kurskoy-i-yaroslavskoy-oblastyakh-2007271358.html?fbclid=IwAR0UdmWrTsGKVDB8x4_ycnwRBOuz_KH4GNPQWYuIcHkC0wzuTCRgQZXnyxQ

authoritarian e-voting (i-voting), like it was a couple of years ago regarding e-participation [3]. Whereas China is considered one of the leaders in online consultations [1; 14], Russia has all the potential to become a role model in the case of Internet voting. The COVID-19 pandemic may also contribute to this trend [6], being a stimulus, or a "window of opportunities" to justify the changes in electoral rules.

We are of the opinion that the questions highlighted in this paper should become an important item of the *E-Voting Studies* research agenda in several respects. In the first place, more attention should be drawn to the importance of "offline" social and political context in which e-voting technologies are employed [20]. Modern Political Science and comparative authoritarianism offer a wide range of frameworks which may well complement the existing approaches that deal with the technical issues related to e-voting and i-voting *per se*. While our paper was working mostly with rationales and possible outcomes of e-voting, we have intentionally overlooked the technical aspect of online voting in non-democracies. However, this lacuna might be easily filled in by the theoretical framework, constituted by Heiberg et al., 2017 [46], in which the authors pinpoint core principles of verifiable elections in Estonian case — this potential projection would once again emphasize the contrast between e-voting in different contexts.

Secondly, while this paper proposes some clues to the factors that drive e-voting (i-voting) innovations in non-democracies (namely, institutional structures and legitimation strategies), more elaborate research techniques should be utilized to explore the determinants and incentives of e-voting adoption at the state and individual level. Here, both the large-N comparisons and deep case studies will be important.

Finally, as this issue is relatively new, it is rather early to estimate the possible effects of such innovations in terms of democratization and autocratization of adopting countries. Yet, this line of research will also be of great value, as the volume of empirical evidence continues to grow.

References

1. Keremoğlu, E., Weidmann, N. B.: How Dictators Control the Internet: A Review Essay. Comparative Political Studies. (2020). https://doi.org/10.1177/0010414020912278
2. Rød, E. G., Weidmann, N. B.: Empowering activists or autocrats? The Internet in authoritarian regimes. Journal of Peace Research. 52(3), 338-351 (2015)
3. Karlsson, M.: Carrots and sticks: internet governance in non–democratic regimes. International Journal of Electronic Governance. 6(3), 179-186 (2013)
4. Toepfl, F.: Innovating consultative authoritarianism: Internet votes as a novel digital tool to stabilize non-democratic rule in Russia. New media & society. 20(3), 956-972 (2018)
5. Christensen, B.: Cyber state capacity: A model of authoritarian durability, ICTs, and emerging media. Government Information Quarterly. 36(3), 460-468 (2019)
6. Krimmer, R., Duenas-Cid, D., Krivonosova, I.: Debate: safeguarding democracy during pandemics. Social distancing, postal, or internet voting—the good, the bad or the ugly? Public Money & Management. 1-3 (2020)
7. Goos, K., Beckert, B., Lindner, R.: Electronic, Internet-based voting. In: Electronic Democracy in Europe, pp. 135-184. Springer, Cham (2016)

8. Krimmer, R.: A Structure for New Voting Technologies: What They Are, How They Are Used and Why. In: K. Bergener, M. Räckers, A. Stein (eds.) The Art of Structuring, pp. 421–426. Springer International Publishing. (2019). https://doi.org/10.1007/978-3-030-06234-7_39
9. Berman, J., Weitzner, D. J.: Technology and Democracy. Social Research. 64(3), 1313–1319 (1997)
10. Calingaert, D.: Authoritarianism vs. The Internet. Policy Review. 14 (2010)
11. Diamond, L.: Liberation Technology. Journal of Democracy. 21(3), 69–83 (2010). https://doi.org/10.1353/jod.0.0190
12. Grider, M.: Securing the Vote: Electronic Voting in Theory and Practice. 9 (2018)
13. Vassil, K., Solvak, M., Vinkel, P., Trechsel, A. H., Alvarez, R. M.: The diffusion of internet voting. Usage patterns of internet voting in Estonia between 2005 and 2015. Government Information Quarterly. 33(3), 453–459 (2016). https://doi.org/10.1016/j.giq.2016.06.007
14. Kornreich, Y.: Authoritarian responsiveness: Online consultation with “issue publics” in China. Governance. 32(3), 547–564 (2019). https://doi.org/10.1111/gove.12393
15. Truex, R.: Consultative Authoritarianism and Its Limits. Comparative Political Studies. 50(3), 329–361 (2017). https://doi.org/10.1177/0010414014534196
16. MacKinnon, R.: Liberation Technology: China's" Networked Authoritarianism". Journal of Democracy. 22(2), 32-46 (2011)
17. Herrnson, P. S., Bederson, B. B., Lee, B., Francia, P. L., Sherman, R. M., Conrad, F. G., Traugott, M., Niemi, R. G.: Early Appraisals of Electronic Voting. Social Science Computer Review. 23(3), 274–292 (2005). https://doi.org/10.1177/0894439305275850
18. Kshetri, N., Voas, J.: Blockchain-Enabled E-Voting. IEEE Software. 35(4), 95–99 (2018). https://doi.org/10.1109/MS.2018.2801546
19. Cheeseman, N., Lynch, G., Willis, J.: Digital dilemmas: The unintended consequences of election technology. Democratization. 25(8), 1397-1418 (2018)
20. Oostveen, A. M., van den Besselaar, P.: The Academic Debate on Electronic Voting in a Socio-Political Context. E-Vote-ID 2019. 17 (2019)
21. Binte Haq, H., McDermott, R., Taha Ali, S.: Pakistan's Internet Voting Experiment. (2019). 1907.07765
22. Al Siyabi, M., Al Jabri, N., Al-Shihi, H., Al-Khod, A. K. A. K.: The Uptake of Voting Participations in Oman through E-voting. In: The International Information Systems Conference (iiSC) 2011 Sultan Qaboos University, Muscat, Sultanate of Oman. (2011)
23. Al-Khouri, A. M., Authority, E. I., Dhabi, A.: E-voting in UAE FNC elections: A case study. In: Information and Knowledge Management. Vol. 2, No. 6, pp. 25-84 (2012)
24. Kassen, M.: Politicization of e-voting rejection: reflections from Kazakhstan. Transforming Government: People, Process and Policy. (2020)
25. Schedler, A.: Elections without democracy: The menu of manipulation. Journal of democracy. 13(2), 36-50 (2002)
26. Schedler, A.: The politics of uncertainty: Sustaining and subverting electoral authoritarianism. OUP Oxford. (2013)
27. Kneuer, M., Harnisch, S.: Diffusion of e-government and e-participation in Democracies and Autocracies. Global Policy. 7(4), 548-556 (2016)
28. Dukalskis, A., Gerschewski, J.: What autocracies say (and what citizens hear): Proposing four mechanisms of autocratic legitimation. Contemporary Politics. 23(3), 251-268 (2017)
29. Gerschewski, J.: The three pillars of stability: legitimation, repression, and co-optation in autocratic regimes. Democratization. 20(1), 13-38 (2013)

30. Gerschewski, J.: Legitimacy in Autocracies: Oxymoron or Essential Feature? Perspectives on Politics. 16(3), 652-665 (2018)
31. von Soest, C., Grauvogel, J.: Identity, procedures and performance: how authoritarian regimes legitimize their rule. Contemporary Politics. 23(3), 287-305 (2017)
32. Tannenberg, M., Bernhard, M., Gerschewski, J., Lührmann, A., Von Soest, C.: Regime Legitimation Strategies (RLS) 1900 to 2018. V-Dem Working Paper. 86 (2019)
33. Wahman, M., Teorell, J., Hadenius, A.: Authoritarian regime types revisited: updated data in comparative perspective. Contemporary Politics. 19(1), 19-34 (2013)
34. Bader, M.: Do new voting technologies prevent fraud? Evidence from Russia. In: 2014 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 14). (2014)
35. Dutton, W. H., Danziger, J. N.: Computers and politics: High technology in American local governments. New York, Columbia University Press. (1982)
36. Brancati, D.: Democratic authoritarianism: Origins and effects. Annual Review of Political Science. 17, 313-326 (2014)
37. Gandhi, J., Lust-Okar, E.: Elections under authoritarianism. Annual review of political science. 12, 403-422 (2009)
38. Ananyev, M., Poyker, M.: Do Dictators Signal Strength with Elections? (2018). SSRN 2712064
39. Gibson, J. P., Krimmer, R., Teague, V., Pomares, J.: A review of e-voting: the past, present and future. Annals of Telecommunications. 71(7-8), 279-286 (2016)
40. Goodman, N., Stokes, L. C.: Reducing the Cost of Voting: An Evaluation of Internet Voting's Effect on Turnout. British Journal of Political Science. 1–13 (2018). https://doi.org/10.1017/S0007123417000849
41. Vassil, K., Weber, T.: A bottleneck model of e-voting: Why technology fails to boost turnout. New media & society. 13(8), 1336-1354 (2011)
42. Solvak, M., Vassil, K.: Could Internet Voting Halt Declining Electoral Turnout? New Evidence That E-Voting Is Habit Forming. Policy & Internet. 10(1), 4-21 (2018)
43. Germann, M., Serdült, U.: Internet voting and turnout: Evidence from Switzerland. Electoral Studies, 47, 1-12 (2017)
44. Reuter, O.J: Political Participation and the Survival of ElectoralAuthoritarian Regimes, http://ojreuter.com/wp-content/uploads/Turnout_Paper.pdf
45. Frantz, E.: Voter turnout and opposition performance in competitive authoritarian elections. Electoral Studies. 54, 218-225 (2018)
46. Heiberg, S., Martens, T., Vinkel, P., Willemson, J.: Improving the verifiability of the Estonian Internet Voting scheme. In: International Joint Conference on Electronic Voting, pp. 92-107. Springer, Cham (2017)
47. Springall, D., Finkenauer, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., Halderman, J. A.: Security analysis of the Estonian internet voting system. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 703-715. (2014)
48. Bull, C., Gjøsteen, K., Nore, H.: Faults in Norwegian internet voting. E-Vote-ID 2018 Proceedings. 166-169 (2018)
49. Ruijgrok, K.: From the web to the streets: internet and protests under authoritarian regimes. Democratization. 24(3), 498-520 (2017)
50. Golosov, G. V. The regional roots of electoral authoritarianism in Russia. Europe-Asia Studies. 63(4), 623-639 (2011)
51. Turchenko, M.: Electoral Engineering in the Russian Regions (2003–2017). Europe-Asia Studies. 72(1), 80-98 (2020)

52. Ross, C.: Regional elections in Russia: instruments of authoritarian legitimacy or instability? Palgrave Communications. 4(1), 1-9 (2018)
53. Maréchal, N.: Networked authoritarianism and the geopolitics of information: Understanding Russian Internet policy. Media and Communication. 5(1), 29-41 (2017)
54. Nocetti, J. Russia's' dictatorship-of-the-law' approach to internet policy. Internet Policy Review. (2015)
55. Maerz, S. F.: The electronic face of authoritarianism: E-government as a tool for gaining legitimacy in competitive and non-competitive regimes. Government Information Quarterly. 33(4), 727-735 (2016)
56. Chugunov, A. V., Kabanov, Y., Zenchenkova, K.: Russian e-petitions portal: exploring regional variance in use. In: International Conference on Electronic Participation, pp. 109-122. Springer, Cham. (2016)
57. Kabanov, Y., Chugunov, A. V.: Electronic "Pockets of Effectiveness": E-governance and Institutional Change in St. Petersburg, Russia. In: International Conference on Electronic Government, pp. 386-398. Springer, Cham. (2017)
58. Coppedge, M. et al. V-Dem [Country–Year/Country–Date] Dataset v10. Varieties of Democracy (V-Dem) Project (2020). https://doi.org/10.23696/vdemds20.

Does vote verification work: usage and impact of confidence building technology in Internet voting

Mihkel Solvak[0000−0003−0179−4036]

Johan Skytte Institute of Political Studies, University of Tartu, likooli 18, 51003 Estonia
mihkel.solvak@ut.ee

Abstract. The paper examines cast-as-intended verification usage in Estonia by looking at who verifies votes, how they do it and what is ithe effect on perceptions of election integrity. Using anonymized log and survey data a typical use case of verification is established - younger, Linux using male voting late at night - which suggest verification is used by more cyber risk aware users. Vote verifiers, when re-voting, are also more likely to change the voting environment compared to those re-voters who do not verify their vote, indicating verification is not simply used to check one's own mistakes in candidate selection. The effects of verifying on confidence in the vote being correctly taken into account are substantial - vote verifiers show a stronger belief in the election integrity. Overall, verification technology seems to be building confidence in the system and being used by the more risk aware voting population.

Keywords: Internet voting · cast-as-intended verification · voter confidence.

1 Introduction

Remote Internet voting gives voters a convenient location independent way of casting the vote. Though still not widely used in national elections, various theoretical and practical aspects of it have been examined over the years by computer science, behavioral and legal scholars, see for example [1] [2] [3] [4] [5]. One particularly interesting aspect of this is *individual vote verification*. Definitions of what is and what is not a verifiable voting system are presented below, but it suffices to say that the need for vote verification has seen a resurgence with Internet voting. There is more agreement on what constitutes individual verification. It should allow individual voters to verify that their vote was indeed *cast as intended* - the voting application correctly recorded the voters will - as well as *recorded as cast* - the vote was accepted into the virtual ballot box - or even *tabulated as recorded* - the vote was correctly tabulated. Though this is a disputable feature as it gives the voter final proof of their vote being counted, which compromises vote secrecy and might lead to electoral integrity problems.

Research on verification from the end user point of view has been so far more theoretical, as not many real-life applications - i.e. actual use cases in

elections - are available. This paper addresses that gap by using observational data on verification from Estonia to examine if the findings from the literature on verification being hard to comprehend by the average voter and being used for non-intended purposes do hold in an actual election settings as well.

Estonia introduced internet voting in 2005 and after a proof-of-concept vote manipulating malware demonstration in 2011 also introduced individual vote verification in 2013 which allows voters to check if the vote was cast as intended and recorded as cast. The ability of verify should in theory add to the voter's confidence that a technology intensive and hard to understand voting system is indeed performing as foreseen, is free from malicious manipulation and is overall trustworthy. All this of course presumes that the technology is actually used, used by those for whom the above mentioned features are important and that it truly functions as a trust building technology in an environment of increased cyber threats against the privacy and secrecy of the individual vote.

This research examines in detail the actual usage patterns of Internet vote verification and its effect on / or correlation with beliefs in the integrity of Internet voting itself. It will do so by examining two data sources, anonymized Internet voting logs from the 2019 Estonian parliamentary elections and survey data on Internet voting from the period between 2013-2019. The paper will proceed as follows, first it discusses potential explanations of usage and formulates hypotheses based on theses followed by an explanation of the data and design, and finally examines the empirical evidence from log and survey data on usage and attitudes to determine if the hypotheses hold and concluded by a discussion.

2 Theory

What defines if elections are end-to-end verifiable has not been fully agreed on and different authors list different features of such systems, one can however be certain that individual verification of cast as intended and recorded as cast is a subset of any such definition[6]. How exactly these features are implement might differ and cast-as-intended verification proposals are plentiful [7] [8] [9] [11] [10] [13]. The common feature is that the voter should be in a position to verify that the voting application has correctly recorded his or her will in the form of the candidate/party number or name for whom she wanted to vote for and that this vote has been also recorded as cast, meaning correctly placed into the ballot box. Given this it is surprising that verification usage or willingness to use it tends to be rather low in reality. This suggests risk awareness and the desire to protect one's vote against it can be presumed to be rather low. Kulyk et al propose to explain low verification usage with a "lack of awareness, lack of concern, lack of self-efficacy, lack of compulsion and lack of perseverance"[14] when faced with security and privacy risks of internet voting. Lets dub it the "five-lacks" explanation. They can be divided into three larger categories, one being risk perceptions, the second usability of the technology and third security practices. Lack of awareness and concern refer to inability to imagine potential risks or a perception that theses "won't affect me". Lack of

self-efficacy suggests a inability to use verification either due to being unable to grasp its function or overly complex design for the average voter. Finally, lack of compulsion and perseverance explain low usage through verification being an optional feature in voting as well as voters not adopting their other online risk mitigating practices when it comes to electronic voting.

The "five lacks" are in line with what has been observed in studies on voter perceptions of security in the voting context. Olembo et al [15] for example find that the most prevalent mental model used when faced with verification system is, first either extending the trust they usually hold towards paper voting also to the electronic voting system, which is mistaken as the risks differ considerably in their nature (deemed the *trusting* mental model) or second, simply not being able to imagine how the integrity of the voting process could be compromised in the first place and how verification could be used to counter that (deemed the *no knowledge* mental model). In sum, taking the reverse of explanations why verification usage tends to be low points to an expectation that verification user should be a clear non-random subgroup of voters with higher risk awareness. Therefore one can posit a hypothesis on likely users:

***H1**: Vote verifiers have a distinct profile with traits typical for more risk aware users*

Usability studies of cast-as-intended verification solutions point towards other expectations when turning from user profiles to actual verification practices. Design of the verification procedure could for example feed into the voter's perception that verification guards foremost against his/her own mistake in picking the correct candidate rather than actual malevolent actions by a third party[14] [16]. This leads to possible interesting expectation in the Estonian situation. To mitigate voter coercion threats Internet voters can re-vote, multiple times if needed, so as to leave the potential coercer unable to ensure the coerced vote will stand. This should particularly help against so called "over-the-shoulder coercion" [12] i.e. when the the immediate voting environment is somehow insecure or the vote privacy is under threat. The voter can simply change the environment and re-vote later on. However, if verification is simply used to check against mistakes made by the voter in picking a correct candidate as suggested above, and not to ensure against malicious vote manipulation, then upon noticing the mistake in candidate selection the voter would likely re-vote again immediately without any change in the environment. This leads to a hypothesis on verification practice:

***H2**: Vote verification correlates with re-voting in the same environment*

An alternative to this hypothesis is of course the intended use case of verification, i.e. verifying to make sure that the correct vote arrived at the authorities and security risks have not materialised. The corresponding hypothesis reads the following:

***H3**: Verification is more likely when voting environment is not private and/or deemed less secure*

This hypothesis rests on the conditions under which the vote is given, if it is more open to risks or presumed to be so by the voter, then verification should

give an additional guarantee that the risks did not come true. While we do not know the outcome of the verification, it could be additionally posited that voting under such conditions should lead more likely to re-voting as well.

The potential effects of verification on voter perceptions should be intuitive. Because of the particular logic of the *cast-as-intended* and *recorded-as-cast* verification logic we can further narrow down what part of the election process integrity should be solidified in the eyes of the voters - namely the belief that their vote was indeed correctly taken into account by the voting system. Even though this might already seem as part of the *tabulated as recorded* verification, which many verification features in fact do not allow to check, I doubt that the average voter makes this minute distinction. It is more likely that he presumes the cast-as-intended verification to show that the vote indeed is now safely cast and will be correctly taken into account. This leads to the fourth and final hypothesis:

H4*: Vote verifiers are more likely to believe that their vote was correctly taken into account*

It is clear that the belief in election integrity is influenced by a multitude of factors. Survey research has shown that voter perceptions are influenced by personal experiences on election day [17], voting for election losers[18], party affiliation[19] or even the propensity in general to believe in conspiracies [20]. The confidence level in electoral fairness also shows significant variation cross-nationally, which suggest cultural and institutional influences play a role [21]. Due to scope and space limitations I will however not attempt a comprehensive explanation of perceived electoral integrity but focus currently only in identifying if verification shows a tangible effect on belief in integrity while keeping other things constant. Next the data and design of the study are examined in more detail.

3 Data and design

The hypotheses are examined using two different dataset. First the verification patterns and the typical verifier profile are examined through the analysis of anonymized Internet voting logs from the 2019 parliamentary election. This dataset holds information on 247 232 Internet vote outer envelopes and contains the voter age, gender, identification type (ID smart card, Mobile ID or Digi ID), operating system of the computer used for voting, anonymized public IP and timestamps on when the vote was cast and if it was verified. Examining the verifier's profile will allow to evaluate hypothesis 1.

Hypothesis 2 and 3 will also be examined with log data. For hypothesis 2 the voting environment is defined through the anonymized IP, operating system and ID type of the voter. A change of any of these between two votes by the same voter would indicate some change in the immediate voting environment. Though these are mere proxies they do allow to see if something changed between casting the second vote. Changing locations will mean the IP has changed, a changed operating system will mean another computer was chosen to vote again and a

change in eID type could in theory reflect a reaction to a vote manipulating coercer who has access to or control of the voter's credentials. In the extreme this could mean a vote could be cast in the voters name [11]. In the Estonian case this would mean having access to the signing key stored inside the eID tool (smart card or Mobile-ID) [12]. A change in ID type would hence indicate that the voter has some issue with one type of ID and feels the need to vote again using another identification tool. If verification is however simply used to check one's own mistakes in candidate selection then none of the above described should be taking place and upon seeing the mistake the voter would simply re-vote after verifying the first vote without making any changes to the environment.

For hypothesis 3 I'll examine if verification is more frequent when the vote is cast using an IP that is shared meaning multiple votes are cast from a internet network that has the same public IP, indicating the same internet connection and most likely a shared office or home with another voter. In addition I'll examine if verification is more likely when the vote is given outside of Estonia as voting from abroad could lead to a additional need to reassure the voter that the vote indeed has successfully "travelled" across border or due to voting from a hotel or other internet connection they have not full confidence in compared to home networks.

Finally, hypothesis 4 is examined with survey data from the Estonian Internet voting survey 2005-2019, only data from the period 2013-2019 is included in this study as this covers the time when verification was possible. Therefore six post-election cross sectional surveys with questions on voting, usage of verification, attitudes and sociodemographics are utilized. The surveys included a question of election integrity, asked as: "How confident are you that your vote was counted as intended in the elections?". With a 4 category Likert scale from very confident to not at all confident. The same questions is asked about the confidence that votes by others were counted as intended. These questions will allow to examine if the vote verifiers show higher confidence levels in the elections as they have used the cast-as-intended and recorded-as-cast features of verification. The sample size in all surveys is roughly 1000 respondents and the full database consists of approximately 6 000 survey interviews.

4 Results

4.1 Who verifies the vote

Verification usage rates are displayed in Table 1. It is clear that verification is not used widely and there is no discernible increase in usage over the years. The share fluctuates between 3 to 5 percent out of all cast Internet votes. For the 2019 parliamentary election this translates into roughly 13 000 verified votes.

Figure 1 shows verification frequency according to age and gender, extracted from the logs. We see that a comparatively larger share of ivotes are verified by young males. Interestingly there seems almost a constant verification rate of 8-10% among males between 18-45 years of age, after which the share drops

Table. 1. Internet vote verification percent (2013-2019)

Election	Share of verified votes
Local 2013	3.4
EP 2014	4.0
Parliamentary 2015	4.3
Local 2017	4.0
Parliamentary 2019	5.3
EP 2019	4.1

Source: National Electoral Committee

significantly as age increases further. No such plateau among female voters is observable. The substantial gender difference in verification frequency disappears only among voters in their late 60's. On average men are about two times more likely to verify the vote compared to female voters. Women in Estonia nowadays make up a majority among ivoters with a share in the 2019 parliamentary election of 55%. The gender difference among ivote verifiers is however in the reverse direction, with males making up 61%.

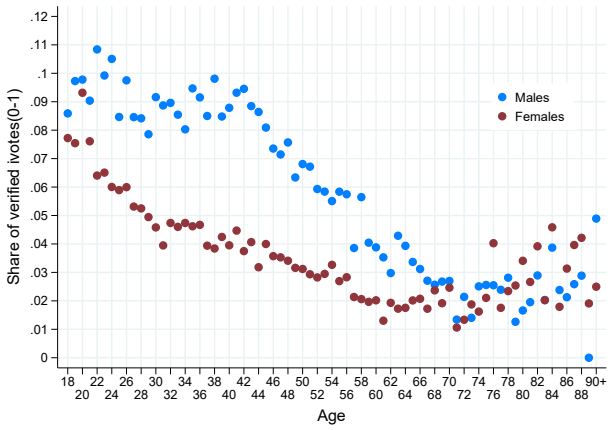


Fig. 1. Share of ivotes verified by age and gender

Examining the the typical verifier further we see that verification share is clearly higher during late voting hours. Figure 2 shows that verification share starts to increase when ivotes are given after 8pm and peaks at 3pm night-time. Time of course does not influence verification probability, it simply shows that the ivoters who vote at these hours are a somewhat distinct user group.

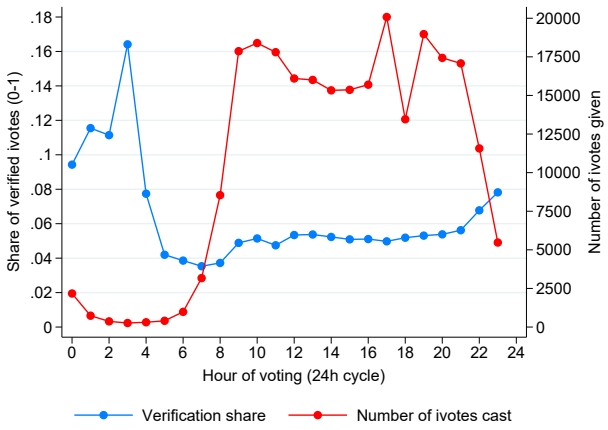


Fig. 2. Verification share and number of votes by time of day

Finally an examination of the other variables available in the logs (Table 2) shows verification to be more prevalent among Linux users with every 5th vote being verified compared to every 20th verified among Windows users. Also, Mac using voters are likely verifiers with a share that is twice that of Windows using voters. For ID types (Table 3) voters using Mobile-ID are much more likely to verify. This is explainable by the fact that they already have the smart device that can be used for verification at hand as they use it to authenticate themselves and digitally sign the vote.

Table. 2. Verification rate by operating system used for ivoting

Op. system	Verified the vote No	Yes	Total
Windows	215 612	10 923	226 535
	95.18%	4.82%	100.0%
Mac	22 941	2 127	25 068
	91.52%	8.48%	100.00%
Linux	1 552	417	1 969
	78.82%	21.18%	100.00%
Total	240 105	13 467	253 572
	94.69%	5.31%	100.00%

Table. 3. Verification rate by ID means used for ivoting

	Verified the vote		
eID	No	Yes	Total
ID-card	168 095	7 136	175 231
	95.93%	4.07%	100.00%
Mobile-ID	68 205	6 155	74 360
	91.72%	8.28%	100.00%
Digi-ID	3 805	176	3 981
	95.58%	4.42%	100.00%
Total	240 105	13 467	253 572
	94.69%	5.31%	100.00%

To sum up the profiles on can say that when the average ivoter is a 45 year old woman, but the average verifier a 40 year old male. Taking into account the very high verification share among males between 18 to 40, the disproportionate presence of Linux users as well as the voting hours it is safe to say that the ivote verifiers are clearly distinct from average ivoters, let alone voters. The verifier profile fits a more technology and computer literate user description. A logical conclusion from this is also that they are above average when it comes to awareness of the privacy and security risks connected to Internet voting.

4.2 How is verification used

Let us move from a typical user to more concrete usage patterns of verification. Figure 3 examines the paths voters took depending on whether they verified the vote or not and how did their voting environment changed in relation to that. A couple of interesting aspects appear. As a sidenote, the total figures differ from previous tables which included re-votes by the same voters, in Figure 3 only paths of unique voters are shown.

First, most of the 12 077 who verified left it at that and did not re-vote, the percentage is almost the same among verifiers (97.5%) and non-verifiers (97.9%) . Out of the ones who did vote a second time a majority did not change either the setting (change of IP), the device (change of operating system) or eID mode used for voting. But the same is apparent for those who did not verify. Changing the voting environment is however more likely among the verifiers compared to the non-verifiers, which runs counter to hypothesis 2. It is also apparent that once you verified the first vote you are also more likely to verify subsequent re-votes regardless of the change in environment as verification is the modal category for both (path 2 and 4) while not verifying the second vote is the modal category when you also did not verify your first vote (path 6 and 8). But lets examine time between the votes for the paths in Figure 3 as well before we conclude anything.

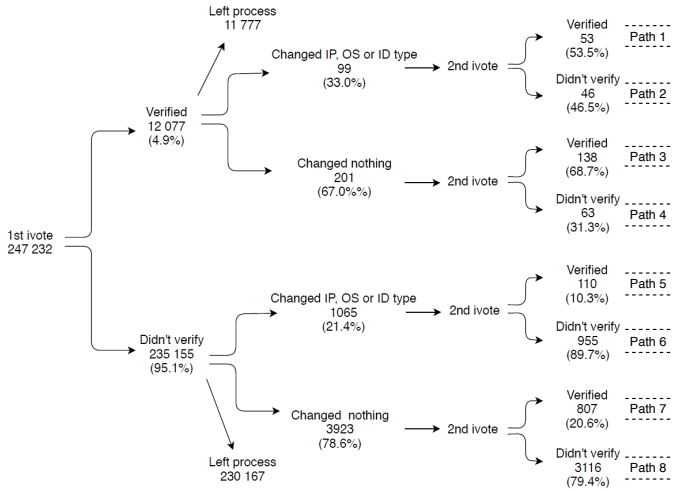


Fig. 3. Change patterns for those who voted second time (everyone included)

This is done in Figure 4. In both cases, for those who verified their first vote and for those who did not, the second vote is cast comparatively faster than among the group who actually changed either the IP, operating system or ID means. The IP change indicates a change of location and the operating system change means a change of the device used for voting. In other words, the second vote is given comparatively faster by those who leave everything unchanged. This is of course to be expected, if you vote from another network later on or purposefully move to another network to re-vote, there is bound to be more time between your two votes. The question is if this pattern supports the verification to "self-check and vote again" hypothesis (H2). A detailed look at the cumulative distributions shows that those who do not verify and simply vote again using the same setting do so substantially faster when compared to those who verify and then vote again using the same setting. Even zooming into the really fast re-voters, i.e. those who do so within 5 minutes after the first vote, we see that they make up 24% among the former group and only 11% among the latter. In fact, for every time period - be it 5, 10, 30 or even up to 720 minutes between two ivotes - the share is larger in the former group compared to the latter by a factor of two.

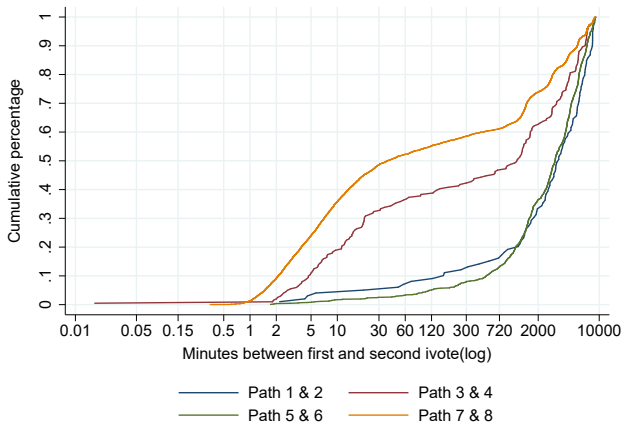
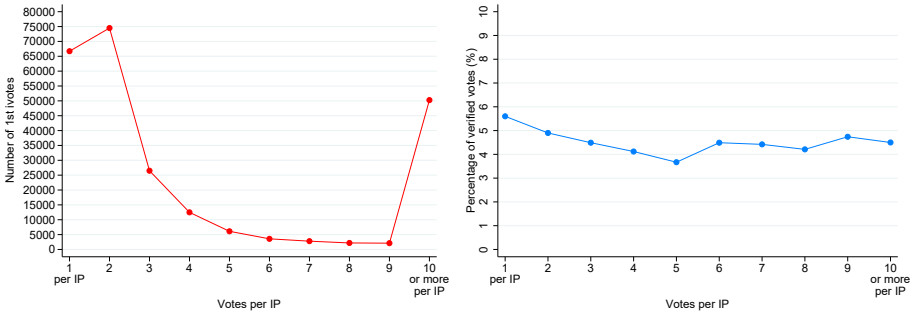


Fig. 4. Cumulative distributions of time between two ivotes for separate vote and verification paths in Figure 3

All in all the evidence does not seem to support H2. Verification compared to non-verification actually leads more likely to changing the environment and voting again, while it does not lead to a faster re-vote compared to non-verifying re-voters. And finally verifying your second vote is much more likely among voters who verified their first vote. All this does not seem to support the explanation that verification is used to simply checking if you did not make a mistake in candidate selection yourself.

Turning now to evidence for H3 leaves a final aspect to examined from the logs - is verification usage due to a possible vote secrecy and privacy concerns. Again, I have to employ a proxy for this in the form of a shared public IP, i.e. if two or more votes from separate voters share a public IP. This means they have voted from the same internet connection as some other voter. The logic being that voting in a setting that is perceived as not so private - shared network - leads more likely to verifying the vote. Figure 5 shows that this is not the case. Casting a vote from an IP from which more than one was cast does not lead to a higher share of votes being verified. Figure 5a shows that most ivotes are given from a public IP that is shared by two voters, but Figure 5b shows that verifying is most prevalent when voting from an IP that is not shared with any other voter. So voting from a network shared with others does not lead to more likely verification, the opposite is true. The caveat is of course that votes per IP does not tell us if it is a network indeed used only by one person, it merely shows no-one else used it for voting, it could still be some public work network. But this is the best proxy I have and it does not support H3.



(a) Number of votes given from shared IPs **(b)** Share of verified votes

Fig. 5. Number of 1st ivotes and share of verifying depending on how many votes were given from shared IPs

Another proxy for perceived security could be the location were the vote is cast from. If a vote is given abroad then the voter might need additional proof that the vote has made it safely "home" across the borders or that a unknown hotel/public/office network or computer used for voting can be trusted, something which verification can provide. This is done in Figure 6. We see that comparatively more ivotes are indeed verified if they are given abroad. This does suggest a need for further confirmation of the integrity given voting happens in a foreign country. In 2019 ivotes were cast from more than 140 countries across the globe and on average verification rate of votes from abroad was clearly higher when compared to the roughly 5% of verified votes in Estonia.

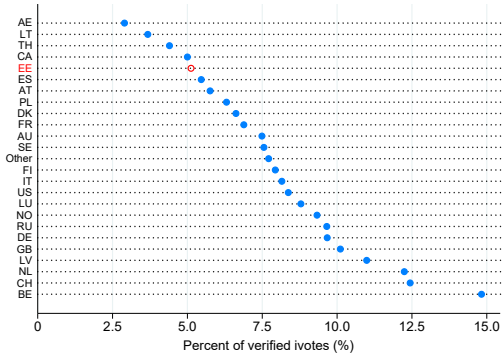


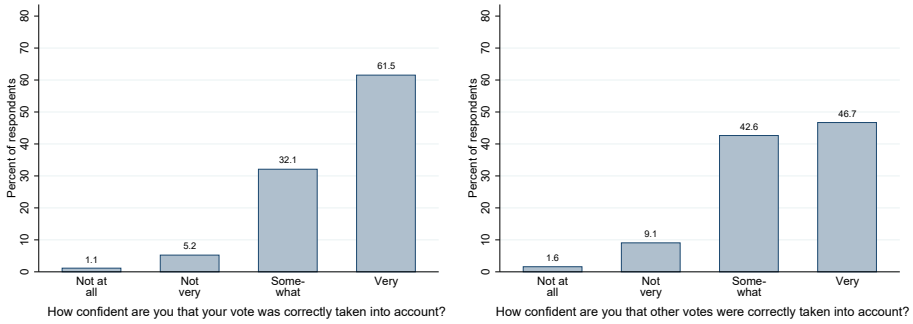
Fig. 6. Verification share by country (Estonia marked red)

All in all. The data shows that voting from an IP that is shared by some other voter, a proxy for a shared location and presumably less private voting

environment, does not translate into a higher probability to verify the vote. Voting from abroad however is indeed clearly connected to a higher verification usage frequency.

4.3 Effect of verification

Finally, lets examine also the connection between verification usage and belief in election integrity with the help of survey data. Confidence in the integrity of one's own vote is shown in Figure 7a and in other votes in 7b. Without bringing verification in yet Figure 7 shows that 87% are somewhat or fully confident that their vote has been correctly taken into account and 89% think so of the votes by other voters. But once this is broken down according to knowing about and using verification (Fig 8) we see that confidence in one's own vote is clearly higher for voters who are either aware of the verification option (Fig 8a) or who have also used it (Fig 8b).



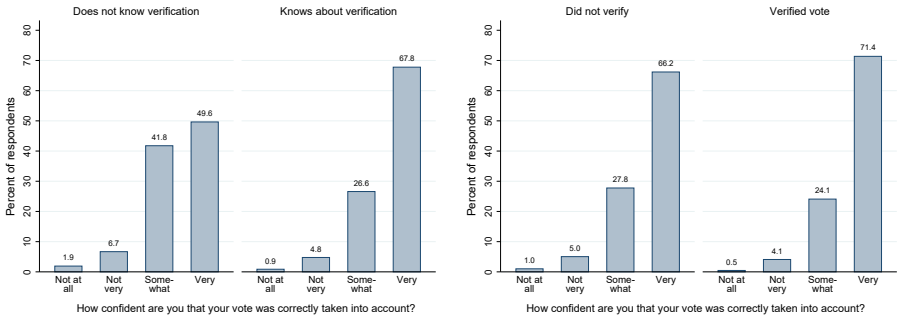
(a) Own vote was taken into account **(b)** Other votes were taken into account

Fig. 7. Confidence vote was correctly taken into account

This difference in confidence for those who know compared to those who don't know about the verification option persists after controlling for gender, age, income, education, computer literacy level and internet usage frequency in an ordered logistic regression model with standard errors clustered by election. Table 4 displays the average marginal effect of knowing about it as well as using verification extracted from the regression model. We see that even after controlling for other covariates that affect confidence, people who know about verification being available have a 17 percentage points higher probability in being very confident that their vote was correctly taken into account, while less likely to be somewhat or not very confident in this. As for actually using verification the effect on confidence is borderline non-significant.

Though this is not enough to claim that verification itself increases confidence as this data is cross sectional and post-election i.e. post-usage, one can clearly

state that awareness of the verification option being available and the actual user experience correlate with a clearly heightened belief in integrity, so the cast-as-intended and recorded-as-cast features of verification seem to be producing the expected results in voter confidence.



(a) Knowing verification and confidence (ivoters only)

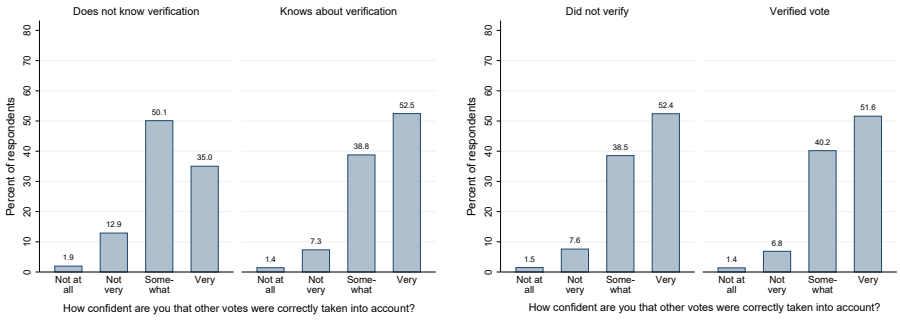
(b) Using verification and confidence (ivoters who know about verification only)

Fig. 8. Confidence own vote was correctly taken into account according to knowledge and usage of individual verification

Table. 4. Average marginal effects on confidence that votes were taken into account (standard errors in parentheses)

	Confident your vote taken into account			
	Not at all	Not very	Somewhat	Very
Knows of verification	-0.011	-0.039***	-0.117***	0.167***
(ref: does not know)	(0.006)	(0.010)	(0.027)	(0.040)
Verified vote	-0.004	-0.015	-0.056	0.076
(ref: did not verify)	(0.002)	(0.010)	(0.029)	(0.039)
	Confident other votes taken into account			
	Not at all	Not very	Somewhat	Very
Knows of verification	-0.012***	-0.059***	-0.095**	0.166***
(ref: does not know)	(0.001)	(0.014)	(0.021)	(0.034)
Verified vote	-0.002	-0.008	-0.021	0.030
(ref: did not verify)	(0.003)	(0.012)	(0.032)	(0.047)

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$



(a) Knowing verification and confidence (ivoters only)

(b) Using verification and confidence (ivoters who know about verification only)

Fig. 9. Confidence that other votes were correctly taken into account according to knowledge and usage of individual verification

Moving to the belief in the integrity of other votes we see somewhat surprisingly that people actually aren't overly confident that these were also correctly taken into account, as shown by Figure 9. The pattern in the figures is however the same as already seen above, verification awareness and usage correlates with a higher belief in the integrity of other votes. Again, the average marginal effects in table 4 confirm these associations to be robust to controls.

5 Discussion

In light of the evidence above the hypotheses can now be re-examined. The first hypothesised that verifiers have a specific profile that might show higher risk knowledge.The second posited that verification might be used to simply check ones own mistake and re-vote without changing much and not to guard against security risks or vote manipulation. The third hypothesised a contrary option that privacy concerns might lead to verification. The evidence is clearly supporting the first hypothesis, verifiers are younger male and Linux user with verification rate especially high in the 18 to 40 age group. Leaving out gender this is surely not a profile of a typical ivoter, let alone a voter, but does conform with a profile of a person who is more computer and technology literate and hence probably also more aware of potential privacy and security risks when it comes to Internet voting. As for the second hypothesis the data does not support it, verification leads actually more likely to changing the settings of the voting session for the re-vote and the data does not show that verifiers re-vote quickly after the initial self-check, the fastest re-voters are actually those who do not verify the first vote. I'm inclined to conclude that verification is indeed more likely used to check against cyber risks rather than voter made mistakes, at least the proxy measures suggest a pattern that is more in line with this. As for the third hypothesis the evidence is again somewhat mixed, while voting from

a network that is shared with other voters does not lead to more verification, voting from abroad clearly does so. The fourth hypothesis proposed that cast-as-intended verification leads more likely to higher confidence in the integrity of the form of being certain that ones vote was taken into account. This was clearly backed up by the data.

All in all cast-as-intended and recorded-as-cast verification does correlate with having higher confidence in the integrity of the vote. It does seem to be more likely used by the more risk aware subgroup of voters, who are more likely to make changes to their when voting environment before a re-vote compared to those who do not verify the vote, even though a large majority of verifiers and non-verifiers who re-vote do so without changing the setting. All this indicates that verification technology is fulfilling its intended purpose - being used to mitigate risks around Internet voting by the one's who are more risk aware. Its availability gives higher confidence in the vote integrity even without necessarily using the technology, knowledge of the possibility seems to suffice.

On a practical level these findings suggest a simple way how to increase the observed positive effects of verification. Raising awareness about this option would not only increase usage numbers, but also belief in the integrity of elections among the non-users. Those who have concerns or who are more risk-aware seem to be using it to mitigate the percieved risks. Those who are not so risk-aware as to be converted into using seem to be clearly more confident in election integrity if they know that the opportunity to verify is in principle available. Even though they choose to trust internet voting rather than verify, evidence at hand shows having the liberty to rely on trust with the option to exercise some verification clearly already boosts belief in election integrity.

The question raised in the title can be answered in the affirmative - verification works.

References

1. Krimmer, R., Volkamer, M., Cortier, V., Beckert, B., Ksters, R., Serdlt, U., Duenac-Cid, D. (eds.): Electronic Voting. LNCS, vol. 11759. Springer (2019)
2. Krimmer, R., Volkamer, M., Cortier, V., Duenac-Cid, D., Gore., Hapsara, M. Koenig, R., Martin, S., MCDermott, R. Roenne, P., Serdlt, U., Truderung, T. (eds.): E-Vote-ID proceedings. Austria, TUT Press (2018)
3. Krimmer, R., Volkamer, M., Binder, N., Kersting, N., Pereira, O., Schrman, C. (eds.): Electronic Voting. LNCS, vol. 10615. Springer (2017)
4. Krimmer, R., Volkamer, M., Barrat, J., Benaloh, J., Goodman, N. Ryan, P., Teague, V. (eds).: Electronic Voting. LNCS, vol. 10141. Springer (2016)
5. Haenni, R., Koenig, R., Wikstrm, D. (eds.): E-Voting and Identity. LNCS, vol. 9269. Springer (2015)
6. Popoveniuc, S., Kelsey, J., Regenscheid, A., Vora, P.: Performance Requirements for End-to-End Verifiable Elections. In: Proceedings of the 2010 International Conference on Electronic Voting Technology/Workshop on Trustworthy Elections, pp. 1–16. USENIX Association, Berkeley (2010)
7. Puiggali, J., Guasch Castell, S.: Internet Voting System with Cast as Intended Verifications. In: Kiayias, A., Lipmaa, H. (eds.) E-Voting and Identity. LNCS, vol. 7187, pp. 36–52. Springer, Heidelberg (2012)

8. Morales-Rocha, V., Soriano, M., Puiggali, J.: New voter verification scheme using pre-encrypted ballots. Computer Communications. 32, 1219–1227 (2009)
9. Helbach, J., Schwenk, J.: Secure Internet Voting with Code Sheets. In: Alkassar, A., Volkamer, M. (eds.) VOTE-ID 2007. LNCS, vol. 4896, pp. 166–177. Springer, Heidelberg (2007)
10. Clarkson, M., Chong, S., Myers, A.: Civitas: Toward a secure voting system. In: IEEE Symposium on Security and Privacy, pp. 354–368 (2008)
11. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: WPES, Alexandria, Virginia, USA, pp. 61–70 (2005)
12. Krips, K., Willemson, J.: On Practical Aspects of Coercion-Resistant Remote Voting Systems. In: Krimmer, R., Volkamer, M., Cortier, V., Beckert, B., Ksters, R., Serdlt, U., Duenac-Cid, D. (eds) Electronic Voting. LNCS, vol. 11759. Springer, Heidelberg (2019)
13. Kulyk, O., Teague, V., Volkamer, M.: Extending helios towards private eligibility verifiability. In: Haenni, R., Koenig, R.E., Wikstrm, D. (eds.) VoteID 2015. LNCS, vol. 9269, pp. 57–73. Springer, Cham (2015)
14. Kulyk, O., Volkamer, M.: Usability is not Enough: Lessons Learned from 'Human Factors in Security' Research for Verifability. In: Krimmer, R., Volkamer, M., Cortier, V., Duenac-Cid, D., Gore., Hapsara, M. Koenig, R., Martin, S., MCDermott, R. Roenne, P., Serdlt, U., Truderung, T. (eds.) E-vote-ID Proceedings, pp. 66–81. Austria, TUT Press (2018).
15. Olembo, M., Bartsch, S., Volkamer, M.: Mental Models of Verifiability in Voting. In: E-Voting and Identify. 4th International Conference, Vote-ID 2013. Proceedings, pp. 142–155, Guildford, UK (2013)
16. Schneider, S., Culnane, C.: Focus group views on Prt Voter 1.0n. 2011 International Workshop on Requirements Engineering for Electronic Voting Systems. REVOTE 2011, Trento, Italy (2011)
17. Kerr, N.: Election-Day Experiences and Evaluations of Electoral Integrity in Unconsolidated Democracies: Evidence From Nigeria. Political Studies. 66, 667–686 (2018)
18. Karp, J., Nai, A., Norris, P.: Dial F for fraud: Explaining citizens suspicions about elections. Electoral Studies 53, 11–19 (2018)
19. Bowler, S., Donovan, T.: A Partisan Model of Electoral Reform: Voter Identification Laws and Confidence in State Elections. State Politics & Policy Quarterly. 16, 340–361 (2016)
20. Norris, P., Garnett, H.A., Grmping, M.: The paranoid style of American elections: explaining perceptions of electoral integrity in an age of populism. Journal of Elections, Public Opinion and Parties. 30, 105–125 (2020)
21. Birch, S.: Electoral institutions and popular confidence in electoral processes: A cross-national analysis. Electoral Studies. 27, 305–320 (2008)

Reviewing the Costs of Multichannel Elections: Estonian Parliamentary Elections 2019

Iuliia Krivonosova[1] [0000-0001-7246-1373], David Duenas-Cid[1 2] [0000-0002-0451-4514] and Robert Krimmer[1] [0000-0002-0873-539X]

[1]Tallinn University of Technology, Akadeemia tee 3, 12618 Tallinn, Estonia
[2]Kozminski University, Jagiellonska 57/59, 03-301 Warsaw, Poland
{iuliia.krivonosova,david.duenas,robert.krimmer}@taltech.ee

Keywords: election administration, multichannel elections, convenience voting, Internet voting, Estonia

The 2019 Parliamentary elections in Estonia was a significant logistics exercise. For 887,420 of eligible voters, the election administration provided 10 voting channels (see Table 1), opened 451 polling stations (PSs) in-country and 40 PSs abroad, with the advance voting period starting at least 15 days before the Election Day.

In line with the findings of the 2017 case study [1], Internet voting stays the most cost-efficient voting channel in terms of cost per voter. The second most popular choice among voters and the second most cost-efficient voting channel is the most traditional way of voting - Election Day voting at the ordinary Voting District Committees (see Table 1). Those two voting channels accounted for above 80% of ballots cast, while the other eight voting channels - for 20%. The least cost-efficient voting channel, in the Estonian setting, is postal voting due to high both fixed and varied costs, and low usage rate of postal voting. The similar trend has been observed in other countries [2].

What drives the costs of multichannel elections? Following the electoral cost calculation framework developed by Krimmer, Duenas-Cid and Krivonosova [3], the cost pools for the paper-based voting channels include Labor, Transportation, Equipment, Stationery, and Printing costs, plus Postage costs for Postal voting; while costs of Internet voting include costs for software development and maintenance. In line with the findings from other countries [4], for most of the considered voting channels in Estonia the largest cost pool is staffing, with up to 80% of resources spent on it. Even for the Internet voting, which relies less on manual labor activities, the labor cost pool constitutes around 2/3 of all costs.

For ***Internet voting***, the most resource intensive activities were DDoS mitigation, updating servers and firewall, and the manual activity of consolidating Internet votes with advance votes cast on paper. The latter activity is inevitable for any elections where Internet voting is an additional voting channel; though, this activity is not necessary performed manually. In comparison to the 2017 elections, the cost per voter for Internet voting has increased, even though a higher number of voters utilized this voting channel in the 2019 elections. According to the stakeholder interviews, this cost change could be attributed to increased security risks, as more and more voters use Internet

voting. To deal with new risks, the Central Electoral Commission invites new stakeholders to cooperation like Information System Authority, outsources some electoral activities and adopts new measures, requiring more resources.

Table 1. Cost per voter in € with a range estimate of 20% in the 2019 Parliamentary elections.

Voting channel	Min Cost per Voter	Max Cost per Voter
Ordinary Voting District Committee		
Advance voting	16.88	18.16
Election Day voting	4.32	4.46
Home voting	14.40	15.51
County Centers Committee		
Early voting	9.25	9.56
Advance voting	8.58	8.99
Election Day voting	6.22	6.24
Home voting	16.01	16.82
Internet voting		
Internet voting	3.10	3.39
Voting from abroad		
Voting in diplomatic missions	28.32	30.39
Postal voting	110.09	126.39

Acknowledgments

This work received support from ETAG personal research grant 1361.

References

1. Krimmer, R., Duenas-Cid, D., & Krivonosova, I. (2020). New methodology for calculating cost-efficiency of different ways of voting: is internet voting cheaper?. Public Money & Management, 1-10.Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010).
2. Krimmer, R., Duenas-Cid, D., & Krivonosova, I. (2020). Debate: safeguarding democracy during pandemics. Social distancing, postal, or internet voting—the good, the bad or the ugly?. Public Money & Management, 1-3.
3. Krimmer, R., Duenas-Cid, D., Krivonosova, I., Vinkel, P., & Koitmae, A. (2018, October). How much does an e-Vote cost? Cost comparison per vote in multichannel elections in Estonia. In International Joint Conference on Electronic Voting (pp. 117-131). Springer, Cham.
4. Clark, Alistair. 2019. "The Cost of Democracy: The Determinants of Spending on the Public Administration of Elections." *International Political Science Review* 40(3): 354–69. https://doi.org/10.1177/0192512118824787 (March 22, 2020).

Audits and Verification

Bayesian Audits are Average but Risk-Limiting Audits are Above Average

Amanda K. Glazer, Jacob V. Spertus, and Philip B. Stark*

University of California, Berkeley, Department of Statistics
amandaglazer@berkeley.edu; jakespertus@berkeley.edu; pbstark@berkeley.edu

Abstract. Post-election audits can provide convincing evidence that election outcomes are correct—that the reported winner(s) really won—by manually inspecting ballots selected at random from a trustworthy paper trail of votes. Risk-limiting audits (RLAs) control the probability that, if the reported outcome is wrong, it is not corrected before the outcome becomes official. RLAs keep this probability below the specified "risk limit." Bayesian audits (BAs) control the probability that the reported outcome is wrong, the "upset probability." The upset probability does not exist unless one invents a prior probability distribution for cast votes. RLAs ensure that if *this* election's reported outcome is wrong, the procedure has a large chance of correcting it. BAs control a *weighted average probability* of correcting wrong outcomes over a hypothetical collection of elections; the weights come from the prior. In general, BAs do not ensure a large chance of correcting the outcome of an election when the reported outcome is wrong. "Nonpartisan" priors, i.e., priors that are invariant under relabeling the candidates, lead to upset probabilities that can be far smaller than the chance of correcting wrong reported outcomes. We demonstrate the difference using simulations based on several real contests.

Keywords: election integrity · risk-limiting audits · Bayesian audits

1 Introduction

The 2016 U.S. Presidential election was attacked by Russian hackers, and U.S. Intelligence agencies warn that several nation-states are already mounting attacks on the 2020 election [29, 30, 31, 22]. Almost every U.S. jurisdiction uses computers to count votes; many use computers to record votes. All computerized systems are vulnerable to bugs, misconfiguration, and hacking [26]. Voters, poll workers, and election officials are also bound to make mistakes [15]. Enough error from any source—innocent or malicious—could cause a losing candidate to appear to win.

The reported tallies will almost certainly be off by at least a little. Were the tallies accurate enough to ensure that the reported winner(s) really won—that the *reported outcome* is correct?

* Authors listed alphabetically.

An election is *evidence-based* [26] if it provides convincing public evidence that the reported winners really won. The only federally certified technology that can provide such evidence is trustworthy paper ballots kept demonstrably secure throughout the election and canvass, then audited manually [2]. However:

- 14% of registered voters live in in jurisdictions using Direct Recording Electronic (DRE) Systems for all voters. DREs do not retain a paper ballot [27].
- Some paper ballots are not trustworthy. For instance, touchscreen voting machines and ballot-marking devices are vulnerable to bugs, hacking, and misconfiguration that can cause them to print the wrong votes [3, 4].
- Rules for securing cast ballots and for ensuring the paper trail remains trusworthy are uneven and generally inadequate.

Nonetheless, to focus on statistical issues, we assume here that elections produce a trustworthy collection of paper ballots containing voters' expressed preferences [3, 2, 11, 26]. A trustworthy paper trail allows audits to check whether errors, bugs, or malfeasance altered the reported outcome. ("Outcome" means who won, not the exact vote tallies.) For instance, we could tabulate the votes on all the cast ballots by hand, as some recount laws require. But full manual recounts are expensive, contentious, and rare: according to Richie and Smith [19], only 27 statewide U.S. elections between 2000 and 2015 were manually recounted; three of the recounts overturned the original outcomes (11%).

Some states conduct tabulation audits that involve manually reading votes from some ballots. For instance, California law requires manually tabulating the votes on ballots in 1% of precincts selected at random.[1] Such audits typically do not ensure that outcome-changing errors will (probably) be detected, much less corrected. In contrast, risk-limiting audits (RLAs) [23, 11] have a known minimum chance of correcting the reported outcome if the reported outcome is wrong (but never alter correct outcomes). RLAs stop without a full hand count only if there is sufficiently strong evidence that a full hand count would find the same winners, i.e., if the P-value of the hypothesis that the reported outcome is wrong is sufficiently small.

RLAs have been endorsed by the National Academies of Science, Engineering, and Medicine [15], the American Statistical Association [1], and many other organizations concerned with election integrity. There have been roughly 60 pilot RLAs in 15 U.S. states and Denmark. Currently 10 U.S. states require or specifically allow RLAs. There have been statewide RLAs or pilot RLAs in five U.S. states: Alaska[2], Colorado [8], Kansas[3], Rhode Island [7], and Wyoming[3], and a pilot RLA in Michigan in which 80 of 83 counties participated [13].

[1] The law is a bit more complicated, including provisions to ensure that every contest gets some scrutiny and options for sampling vote-by-mail ballots (including not sampling them if they arrive after election day).

[2] Organized by J. Morrell; one of us (PBS) provided software and support.

[3] J. Morrell, personal communication, 2020

Bayesian audits (BAs, [20, 21]) have been proposed as an alternative to RLAs. BAs stop without a full hand count only if the "upset probability"—the posterior probability that the reported winner(s) actually lost, for a particular prior π, given the audit sample—is below a pre-specified threshold. They have been piloted in several states.

Bayesian and frequentist interpretations of probability are quite different. Frequentist probability is the long-run limiting relative frequency with which an event occurs in repeated trials. Bayesian probability quantifies the degree to which the subject believes an event will occur. A prior probability distribution quantifies beliefs before the data are collected; after the data are observed, Bayes' rule says how to update the prior using the data to obtain the posterior probability distribution.

Bayesian methods, including BAs, require stronger assumptions than frequentist methods, including RLAs. In particular, BAs require assuming that votes are random and follow a known "prior" probability distribution π.

Both RLAs and BAs rely on manually interpreting randomly selected ballots. In principle, both can use a wide range of sampling plans to accommodate differences in how jurisdictions handle and store ballots and variations in election laws and regulations. (To the best of our knowledge, BAs have been conducted only using "ballot polling" [9].) RLA methods have been developed to use individual ballots or groups of ballots as the sampling unit, to sample with or without replacement or to use Bernoulli sampling, to sample with and without stratification, and to sample uniformly or with unequal probabilities (see, e.g., Stark [23, 24, 25], Lindeman and Stark [11], Ottoboni et al. [18, 17]).

The manual interpretations can be used in two ways: *comparison audits* look at differences between the manual interpretation and the machine interpretation and tabulation, while *polling audits* just use the manual interpretation. (The two strategies can be combined in a single audit; see, e.g., Ottoboni et al. [18], Stark [25].) Comparison audits require more of the voting system and require more preparation than polling audits, but for a given size sampling unit, they generally require smaller samples. (The sample size scales like the reciprocal of the margin for comparison audits, and like the square of the reciprocal of the margin for polling audits.) Below, we focus on polling audits that use individual ballots as the sampling unit: *ballot-polling audits*. These are the simplest conceptually and require the least of the voting system: just the reported winner(s), but no other data export.

Both RLAs and BAs lead to a full hand count if sampling does not provide sufficiently strong evidence that the reported outcome is correct. If they lead to a full hand count, that hand count replaces the reported results. Thus, they might confirm a wrong outcome, but they never overturn a correct outcome. They make different assumptions, use different standards of evidence, and offer different assurances, as we shall explain.

```
while (!(full handcount) && !(strong evidence outcome is correct)) {
    audit more
}
if (strong evidence outcome is correct) {
    reported result is final
}
if (full handcount) {
    handcount result is final
}
```

Fig. 1. Pseudo code for sequential auditing procedures

2 Risk

The *risk* of an auditing procedure, given a trustworthy set of cast ballots and a reported outcome, is zero if the reported outcome is correct and is the chance that the procedure will not correct the reported outcome if the reported outcome is wrong. Formally, let θ denote a set of cast votes. For example, in a contest between (only) Alice and Bob in which n ballots were cast, all containing valid votes, θ is an element of $\{\text{Alice}, \text{Bob}\}^n$. (For sampling with replacement, we could also parametrize the cast votes as the fraction of votes for Alice; see Figure 2.)

RLAs treat θ as fixed but unknown. The only probability in RLAs is the probability involved in sampling ballots at random—a probability that exists by fiat and is known to the auditor, because the auditor designs the sampling protocol.

In contrast, BAs treat θ—the cast votes—as random rather than simply unknown. The probability in BAs comes not only from the sampling but also from the assumption that votes are random and follow a probability distribution π known to (or believed by) the auditor.

Let $f(\cdot)$ be the social choice function that maps a set of cast votes to the contest winner(s). Then

$$\text{risk}(\theta) \equiv \begin{cases} \Pr(\text{audit confirms reported outcome}), & \text{reported winner} \neq f(\theta) \\ 0, & \text{reported winner} = f(\theta). \end{cases}$$

RLAs ensure that the risk does not exceed a pre-specified limit (denoted α), no matter what votes were actually cast. Because θ is fixed, probabilities in RLAs come only from the random sampling of ballots.

BAs control a weighted average of the risk rather than the maximum risk (whence the title of this paper). The weights come from the prior probability distribution on θ. In symbols:

$$\text{risk}_{\text{RLA}} = \max_{\theta} \ \text{risk}(\theta)$$

$$\text{risk}_{\text{BA}} = \frac{1}{c} \sum_{\theta} \ \text{risk}(\theta)\pi(\theta)$$

where $\pi(\theta)$ is the prior on θ and $c = \sum_{\theta:\text{reported winner} \neq f(\theta)} \pi(\theta)$ makes the weights sum to 1.

BAs can have a large chance of correcting some wrong outcomes and a small chance of correcting others, depending on the prior π. If π assigns much probability to wrong outcomes where it is easy to tell there was a problem (e.g., a reported loser really won by a wide margin) the average risk (the upset probability) can be much lower than the risk for the actual set of ballots cast in the election.

An RLA with risk limit α automatically limits the upset probability to α for any prior, but the converse is not true in general. (The average of a function cannot exceed the maximum of that function, but the maximum exceeds the average unless the function is constant.) Below, we demonstrate that the upset probability can be much smaller than the true risk using simulations based on close historical elections.

3 Choosing the Prior for a BA

In a BA, the prior quantifies beliefs about the cast votes and the correctness of the reported outcome before the audit commences. Beliefs differ across the electorate. To address this, Rivest and Shen [20] considered a "bring your own prior" BA: the audit continues until everyone's upset probability is sufficiently small (see Figure 2A). Of course, if anyone's prior implies that a reported loser is virtually certain to have won, the audit won't stop without a full hand count.

Ultimately, Rivest and Shen [20] and Rivest [21] recommend using a single "nonpartisan" prior. A nonpartisan prior is one for which every candidate is equally likely to win, i.e., a prior that is invariant under permutations of the candidates' names (see Figure 2B). We doubt this captures anyone's beliefs about any particular election. Beliefs about whether the reported winner really won may depend on many things, including pre-election polls and exit polls, the reported margin, reports of polling-place problems, news reports of election interference, etc.

For instance, it seems less plausible that the reported winner actually lost if the reported margin is 60% than if the reported margin is 0.6%: producing an erroneous 60% margin would require much more error or manipulation than producing an erroneous 0.6% margin if the reported winner really lost. On the other hand, when the *true* margin is small, it is easier for error or manipulation to cause the wrong candidate to appear to win. Moreover, a tight contest might be a more attractive target for manipulation.

If every audit is to be conducted using the same prior, that prior arguably should put more weight on narrow margins. Taken to the extreme, the prior would concentrate the probability of wrong outcomes at the wrong outcome with the narrowest margin: a tie or one-vote win for a reported loser.

Indeed, Vora [28] and Morin et al. [14] show that in a two-candidate plurality contest with no invalid votes, a ballot-polling BA using a prior that assigns probability 1/2 to a tie (or one-vote win for the reported loser) and

probability 1/2 to correct outcomes is in fact a RLA (see Figure 2C): the upset probability equals the risk.

Constructing priors that make BAs risk-limiting for more complicated elections (e.g., elections with more than two candidates, elections in which ballots may contain invalid votes, social choice functions other than plurality, and audit sampling designs other than simple random samples of individual ballots or random samples of individual ballots with replacement) is an open problem.[4]

4 Empirical Comparison

How are risk and upset probability related? The upset probability is never larger than the risk, but the risk is often much larger than the upset probability for BAs with non-partisan priors, as we show using data from three recent close U.S. elections: the 2017 House of Delegates contest in Virginia's 94th district, the 2018 Congressional contest in Maine's 2nd district, and the 2018 Georgia Governor contest. The simulations, summarized in Table 1, treat the reported vote shares as correct, but re-label the reported winner as the reported loser. "Simulated Risk" is the estimated probability that a BA with 5% upset probability corrects the reported outcome. The simulations use the nonpartisan prior recommended by [21], with initial "pseudo-counts" of 0.5. Each audit begins with a sample of 25 ballots. Each step of each audit simulates 1,000 draws from the posterior distribution to estimate the upset probability. If the upset probability is above 5%, then the sample is increased by 20%, and the upset probability is estimated again. Each audit stops when the upset probability falls below 5%, or all ballots have been audited. We simulate 10,000 ballot-polling BAs for each scenario. Code for the simulations is available at `https://github.com/akglazer/BRLA-Comparison`.

A recount of the 2017 Virginia 94th district contest gave a 1-vote win for Simonds over Yancey. (A three-judge panel later determined that a vote counted as an overvote should be attributed to Yancey; the winner was determined by drawing a name from a bowl [12].) The 2018 Maine Congressional election used ranked-choice voting (RCV/IRV). While there are methods for conducting RLAs of IRV contests [6, 25], we treat the contest as if it were a plurality contest between the last two standing candidates, Golden and Poliquin, a "final-round margin" of 3,509 votes.[5]

In these experiments, the actual risk of the BA is 4 to 9 times larger than the upset probability, 5%. For example, in the Virginia 94th District contest, the BA failed to correct the outcome 43% of the time, 8.6 times the upset probability.

[4] This is related to the problem of constructing *least-favorable priors* in statistical decision problems. There is a deep duality between Bayesian and frequentist procedures: under mild regularity conditions the Bayes risk for a *least-favorable prior* is equal to the *minimax risk* [5]. (Here, risk is a term of art, a measure of the performance of the procedure.) That is to say, for a particular choice of prior, the Bayesian procedure is in fact the frequentist procedure that does best in the worst case. The least-favorable prior is generally not "flat" or "uninformative."

[5] The final-round margin of an IRV contest is an upper bound on the true margin.

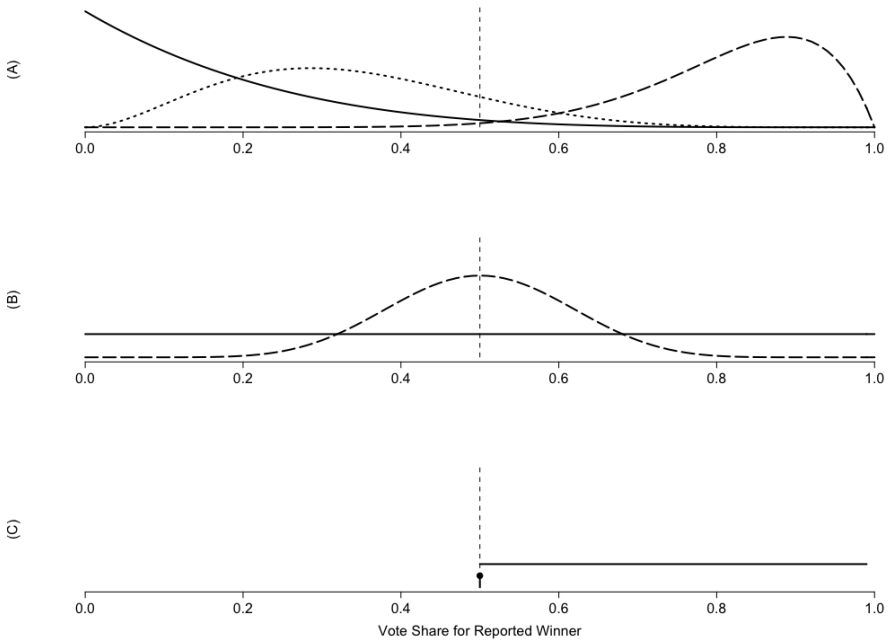


Fig. 2. *Exemplar priors for the true vote share for the reported winner in a two-candidate election. Values to the right of the vertical dotted line (at 1/2) correspond to correct reported outcomes: the winner got more than 50% of the valid votes. (A) plots three possible partisan priors. For BAs that allow observers to bring their own prior, a BA would stop only when all three posteriors give a sufficiently low probability to all outcomes where the reported winner actually lost: values less than or equal to 1/2. (B) plots two nonpartisan priors (the priors are symmetric around 1/2 and thus invariant under exchanging the candidates' names) including the flat prior recommended by Rivest and Shen [20]. The flat prior gives equal weight to all possible vote shares. (C) plots a least-favorable prior, a prior for which a BA is an RLA with risk limit equal to the upset probability. It assigns probability 1/2 to a tie, the wrong outcome that is most difficult to detect. The rest of the probability is spread (arbitrarily) across vote shares for which the reported outcome is correct. In this illustration, that probability is uniform. That choice affects the efficiency but not the risk.*

	Number of Votes Cast	Margin	BA Risk (simulated)
Virginia 94th	23,215 votes	1 vote (0.004%)	43%
Maine 2nd	281,371 votes	3509 votes (1.25%)	23%
Georgia Governor	3,902,093 votes	54,723 votes (1.4%)	22%

Table 1. Simulated risk of a Bayesian Audit using 5% upset probability with a "non-partisan" prior for the 2017 Virginia House of Delegates District 94 contest, the 2018 Maine 2nd Congressional District contest, and the 2018 Georgia gubernatorial contest. Column 2: the margin for each election in number of votes and percentage. Column 3: risk of the BA, i.e., the estimated probability that the BA audit will fail to correct the outcome.

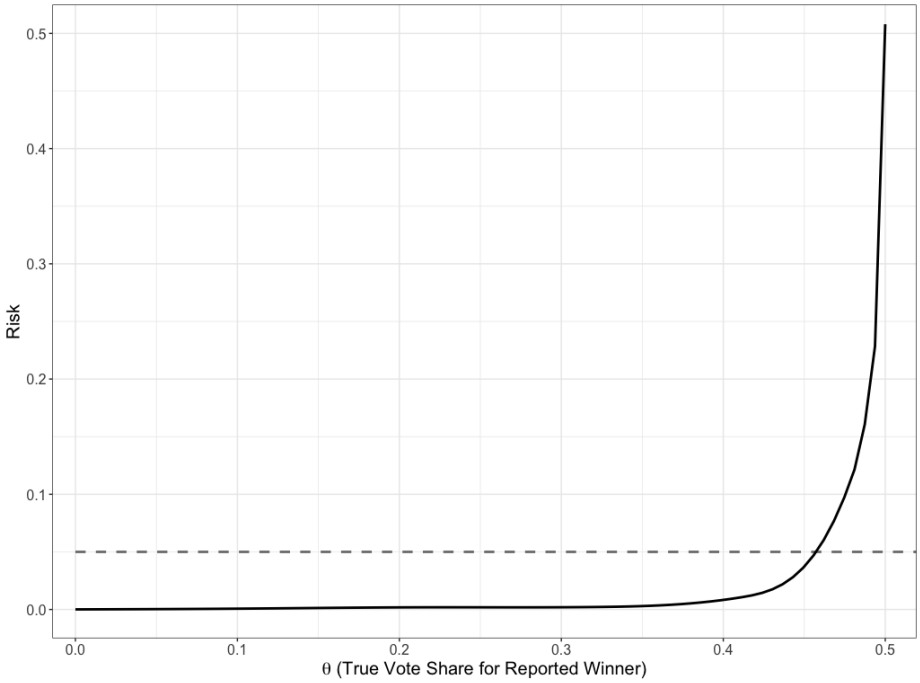


Fig. 3. Simulated risk (solid line) of a BA with nonpartisan prior for a two-candidate election with 1,000,000 total votes cast and no invalid votes. The x-axis is θ, the actual vote share for the reported winner. The reported winner really won if $\theta > 0.5$ and lost if $\theta < 0.5$. The y-axis is the actual risk, computed for $\theta < 0.5$ as the number of times the BA confirms the outcome over the total number of simulated audits. If $\theta > 0.5$ then the risk is 0. The dashed grey line at Risk = 0.05 is the upset probability threshold for the BA, and also the maximum risk for a RLA with risk limit 0.05.

This results from the fact that the upset probability averages the risk over all possible losing margins (with equal weight), while the actual losing margin was small. Figure 3 shows the simulated risk of a BA with a nonpartisan prior and

initial pseudo-counts of 0.5 for an election with 1,000,000 total votes cast. The risk is plotted as a function of the vote share for the winner. The empirical risk for a BA is very high for small margins, where auditing is especially important. As far as we know, there are situations where the risk can be an arbitrarily large multiple of the upset probability, depending on the actual cast votes, the social choice function, the prior, and details of the BA implementation (such as its rule for expanding the sample).

5 Conclusion

Elections are audited in part to rule out the possibility that voter errors, poll-worker errors, procedural errors, reporting errors, misconfiguration, miscalibration, malfunction, bugs, hacking, or other errors or malfeasance made losing candidates appear to win. We believe that controlling the probability that the reported outcome will not be corrected when it is wrong—the risk—should be the minimal goal of a post-election audit. RLAs control that risk; BAs control the upset probability, which can be much smaller than the risk.

Both RLAs and BAs require a trustworthy paper trail of voter intent. RLAs use the paper trail to protect against the worst case: they control the chance of certifying the reported outcome if it is wrong, no matter why it is wrong.

BAs protect against an *average* over hypothetical sets of cast votes (rather than the worst case); the weights in the average come from the *prior*.

The priors that have been proposed for BAs do not seem to correspond to beliefs about voter preferences, nor do they take into account the chance of error or manipulation. Moreover, BAs do not condition on a number of things that bear on whether the reported outcome is likely to be wrong, such as the reported margin and the political consequences. As Vora [28] shows, some BAs are RLAs if the prior is chosen suitably. Bayesian upset probabilities can never be larger than the maximum risk, but it seems that they can be arbitrarily smaller. Conversely, Huang et al. [10] discuss finding a threshold for the upset probability in a BA using a nonpartisan prior for a two-candidate, no invalid-vote contest so that using that threshold as a limit on the upset probability yields an RLA (with a larger risk limit).

Sequential RLAs stop as soon as there is strong evidence that the reported result is correct. When the outcome is correct by a wide margin, they generally inspect relatively few ballots. Thus, even though RLAs protect against the worst case, they are relatively efficient when outcomes are correct. (When outcomes are incorrect, they are intended to lead to a full hand tabulation.)

Partisanship, foreign interference, vendor misrepresentations [29], and suspicious results [16] all threaten public trust in elections, potentially destabilizing our democracy. Conducting elections primarily on hand-marked paper ballots (with accessible options for voters with disabilities), routine compliance audits, and RLAs can help ensure that elections deserve public trust.

Bibliography

[1] American Statistical Association. American Statistical Association statement on risk-limiting post-election audits. www.amstat.org/outreach/pdfs/Risk-Limiting_Endorsement.pdf, 2010.

[2] A. Appel and P. Stark. Evidence-based elections: Create a meaningful paper trail, then audit. *Georgetown Law Technology Review*, 4.2: 523–541, 2020. https://georgetownlawtechreview.org/wp-content/uploads/2020/07/4.2-p523-541-Appel-Stark.pdf.

[3] A. Appel, R. DeMillo, and P. Stark. Ballot-marking devices cannot assure the will of the voters. *Election Law Journal, Rules, Politics, and Policy*, 2020. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3375755.

[4] M. Bernhard, A. McDonald, H. Meng, J. Hwa, N. Bajaj, K. Chang, and J. Halderman. Can voters detect malicious manipulation of ballot marking devices? *41st IEEE Symposium on Security and Privacy*, 2020. https://jhalderm.com/pub/papers/bmd-verifiability-sp20.pdf.

[5] P. Bickel and K. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*. Pearson, 2006.

[6] M. Blom, P. Stuckey, and V. Teague. RAIRE: Risk-limiting audits for IRV elections. https://arxiv.org/abs/1903.08804, 2019.

[7] Brennan Center for Justice, Rhode Island RLA Working Group. Pilot implementation study of risk-limiting audit methods in the state of Rhode Island, 2019. URL https://www.brennancenter.org/our-work/research-reports/pilot-implementation-study-risk-limiting-audit-methods-state-rhode-island.

[8] Colorado Secretary of State. Audit Center, 2020. URL https://www.sos.state.co.us/pubs/elections/auditCenter.html.

[9] L. Howard, R. Rivest, and P. Stark. A review of robust post-election audits: Various methods of risk-limiting audits and Bayesian audits. Technical report, Brennan Center for Justice, 2019. https://www.brennancenter.org/sites/default/files/2019-11/2019_011_RLA_Analysis_FINAL_0.pdf.

[10] Z. Huang, R. Rivest, P. Stark, V. Teague, and D. Vukcevic. A unified evaluation of two-candidate ballot-polling election auditing methods. In *Proceedings of the 5th Annual Conference on Electronic Voting (E-Vote-ID '20)*, 2020.

[11] M. Lindeman and P. Stark. A gentle introduction to risk-limiting audits. *IEEE Security and Privacy*, 10:42–49, 2012.

[12] S. McCammon. Virginia Republican David Yancey wins tie-breaking drawing. https://www.npr.org/2018/01/04/573504079/virginia-republican-david-yancey-wins-tie-breaking-drawing, 2018.

[13] Michigan Secretary of State. Pilot audit of march presidential primary results showcases security, accuracy of Michigan elections systems, 2020. URL https://www.michigan.gov/sos/0,4670,7-127--531561--,00.html.

[14] S. Morin, G. McClearn, N. McBurnett, P. Vora, and F. Zagorski. A note on risk-limiting Bayesian polling audits for two-candidate elections. *Voting '20*, in press, 2020.

[15] National Academies of Sciences, Engineering, and Medicine. *Securing the Vote: Protecting American Democracy*. The National Academies Press, Washington, DC, 2018. ISBN 978-0-309-47647-8. https://doi.org/10.17226/25120. URL https://www.nap.edu/catalog/25120/securing-the-vote-protecting-american-democracy.

[16] K. Ottoboni and P. Stark. Election integrity and electronic voting machines in 2018 Georgia, USA. In *E-Vote-ID 2019 Proceedings*, 2019. Preprint: https://ssrn.com/abstract=3426250.

[17] K. Ottoboni, M. Bernhard, A. Halderman, R. Rivest, and P. Stark. Bernoulli ballot polling: A manifest improvement for risk-limiting audits. In *Proceedings of the 4th Annual Workshop on Advances in Secure Electronic Voting (Voting'19)*, 2018. Preprint: http://arxiv.org/abs/1812.06361.

[18] K. Ottoboni, P. Stark, M. Lindeman, and N. McBurnett. Risk-limiting audits by stratified union-intersection tests of elections (SUITE). In *Electronic Voting. E-Vote-ID 2018. Lecture Notes in Computer Science*. Springer, 2018. https://link.springer.com/chapter/10.1007/978-3-030-00419-4_12.

[19] R. Richie and H. Smith. A survey and analysis of statewide election recounts 2000–2015. https://fairvote.app.box.com/v/recounts, 2015.

[20] R. Rivest and E. Shen. A Bayesian method for auditing elections. In *Proceedings of the 2012 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE '12)*. USENIX, August 2012.

[21] R. L. Rivest. Bayesian tabulation audits: Explained and extended. https://arxiv.org/abs/1801.00528, January 1, 2018.

[22] Select Committee on Intelligence. Russian active measures campaigns and interference in the 2016 U.S. election. https://www.intelligence.senate.gov/sites/default/files/documents/Report_Volume1.pdf, 2019.

[23] P. Stark. Conservative statistical post-election audits. *Ann. Appl. Stat.*, 2: 550–581, 2008. URL http://arxiv.org/abs/0807.4005.

[24] P. Stark. Election audits by sampling with probability proportional to an error bound: dealing with discrepancies. https://www.stat.berkeley.edu/~stark/Preprints/ppebwrwd08.pdf, 2008.

[25] P. Stark. Sets of half-average nulls generate risk-limiting audits: SHANGRLA. *Voting '20*, in press, 2020. Preprint: http://arxiv.org/abs/1911.10035.

[26] P. B. Stark and D. A. Wagner. Evidence-based elections. *IEEE Security and Privacy*, 10:33–41, 2012. https://www.stat.berkeley.edu/~stark/Preprints/evidenceVote12.pdf.

[27] Verified Voting. The Verifier, 2020. URL https://verifiedvoting.org/verifier/#mode/navigate/map/ppEquip/mapType/normal/year/2020.

[28] P. Vora. Risk-limiting Bayesian polling audits for two-candidate elections. https://arxiv.org/abs/1902.00999, 2019.

[29] K. Zetter. The crisis of election security. *The New York Times*, 2018. https://www.nytimes.com/2018/09/26/magazine/election-security-crisis-midterms.html.

[30] K. Zetter. Critical U.S. election systems have been left exposed online despite official denials. *Vice*, 2019. https://www.vice.com/en_us/article/3kxzk9/exclusive-critical-us-election-systems-have-been-left-exposed-online-despite-official-denials.

[31] K. Zetter. How close did Russia really come to hacking the 2016 election? *Politico*, 2019. https://www.politico.com/news/magazine/2019/12/26/did-russia-really-hack-2016-election-088171.

A Unified Evaluation of Two-Candidate Ballot-Polling Election Auditing Methods

Zhuoqun Huang[1], Ronald L. Rivest[2][0000−0002−7105−3690], Philip B. Stark[3][0000−0002−3771−9604], Vanessa Teague[4,5][0000−0003−2648−2565], and Damjan Vukcevic[1,6][0000−0001−7780−9586]

[1] School of Mathematics and Statistics, University of Melbourne, Parkville, Australia
[2] Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA
[3] Department of Statistics, University of California, Berkeley, USA
[4] Thinking Cybersecurity Pty. Ltd.
[5] College of Engineering and Computer Science, Australian National University
[6] Melbourne Integrative Genomics, University of Melbourne, Parkville, Australia
damjan.vukcevic@unimelb.edu.au

Abstract. Counting votes is complex and error-prone. Several statistical methods have been developed to assess election accuracy by manually inspecting randomly selected physical ballots. Two 'principled' methods are risk-limiting audits (RLAs) and Bayesian audits (BAs). RLAs use frequentist statistical inference while BAs are based on Bayesian inference. Until recently, the two have been thought of as fundamentally different. We present results that unify and shed light upon 'ballot-polling' RLAs and BAs (which only require the ability to sample uniformly at random from all cast ballot cards) for two-candidate plurality contests, that are building blocks for auditing more complex social choice functions, including some preferential voting systems. We highlight the connections between the methods and explore their performance.
First, building on a previous demonstration of the mathematical equivalence of classical and Bayesian approaches, we show that BAs, suitably calibrated, are risk-limiting. Second, we compare the efficiency of the methods across a wide range of contest sizes and margins, focusing on the distribution of sample sizes required to attain a given risk limit. Third, we outline several ways to improve performance and show how the mathematical equivalence explains the improvements.

Keywords: Statistical audit · Risk-limiting · Bayesian

1 Introduction

Even if voters verify their ballots and the ballots are kept secure, the counting process is prone to errors from malfunction, human error, and malicious intervention. For this reason, the US National Academy of Sciences [4] and the American Statistical Association[7] have recommended the use of risk-limiting audits to check reported election outcomes.

[7] amstat.org/asa/files/pdfs/POL-ASARecommendsRisk-LimitingAudits.pdf

The simplest audit is a manual recount, which is usually expensive and time-consuming. An alternative is to examine a random sample of the ballots and test the result statistically. Unless the margin is narrow, a sample far smaller than the whole election may suffice. For more efficiency, sampling can be done adaptively: stop when there is strong evidence supporting the reported outcome [7].

Risk-limiting audits (RLAs) have become the audit method recommended for use in the USA. Pilot RLAs have been conducted for more than 50 elections in 14 US states and Denmark since 2008. Some early pilots are discussed in a report from the California Secretary of State to the US Election Assistance Commission.[8] In 2017, the state of Colorado became the first to complete a statewide RLA.[9] The defining feature of RLAs is that, if the reported outcome is incorrect, they have a large, pre-specified minimum probability of discovering this and correcting the outcome. Conversely, if the reported outcome is correct, then they will eventually certify the result. This might require only a small random sample, but the audit may lead to a complete manual tabulation of the votes if the result is very close or if tabulation error was an appreciable fraction of the margin.

RLAs exploit frequentist statistical hypothesis testing. There are by now more than half a dozen different approaches to conducting RLAs [8]. Election audits can also be based on Bayesian inference [6].

With so many methods, it may be hard to understand how they relate to each other, which perform better, which are risk-limiting, etc. Here, we review and compare the statistical properties of existing methods in the simplest case: a two-candidate, first-past-the-post contest with no invalid ballots. This allows us to survey a wide range of methods and more clearly describe the connections and differences between them. Most real elections have more than two candidates, of course. However, the methods designed for this simple context are often adapted for more complex elections by reducing them into pairwise contests (see below for further discussion of this point). Therefore, while we only explore a simple scenario, it sheds light on how the various approaches compare, which may inform future developments in more complex scenarios. There are many other aspects to auditing that matter greatly in practice, we do not attempt to cover all of these but we comment on some below.

For two-candidate, no-invalid-vote contests, we explain the connections and differences among many audit methods, including frequentist and Bayesian approaches. We evaluate their efficiency across a range of election sizes and margins. We also explore some natural extensions and variations of the methods. We ensure that the comparisons are 'fair' by numerically calibrating each method to attain a specified risk limit.

We focus on *ballot-polling audits*, which involve selecting ballots at random from the pool of cast ballots. Each sampled ballot is interpreted manually; those

[8] https://votingsystems.cdn.sos.ca.gov/oversight/risk-pilot/final-report-073014.pdf
[9] https://www.denverpost.com/2017/11/22/colorado-election-audit-complete/

interpretations comprise the audit data. (Ballot-polling audits do not rely on the voting system's interpretation of ballots, in contrast to *comparison audits*.)

Paper outline: Section 2 provides context and notation. Section 3 sketches the auditing methods we consider and points out the relationships among them and to other statistical methods. Section 4 explains how we evaluate these methods. Our benchmarking experiments are reported in Section 5. We finish with a discussion and suggestions for future work in Section 6.

2 Context and notation: two-candidate contests

We consider contests between two candidates, where each voter votes for exactly one candidate. The candidate who receives more votes wins. Ties are possible if the number of ballots is even.

Real elections may have invalid votes, for example, ballots marked in favour of both candidates or neither; for multipage ballots, not every ballot paper contains every contest. Here we assume every ballot has a valid vote for one of the two candidates. See Section 6.

Most elections have more than two candidates and can involve complex algorithms ('social choice functions') for determining who won. A common tactic for auditing these is to reduce them to a set of pairwise contests such that certifying all of the contests suffices to confirm the reported outcome [3,1,8]. These contests can be audited simultaneously using methods designed for two candidates that can accommodate invalid ballots, which most of the methods considered below do. Therefore, the methods we evaluate form the building blocks for many of the more complex methods, so our results are more widely relevant.

We do not consider *stratified audits*, which account for ballots cast across different locations or by different voting methods within the same election.

2.1 Ballot-polling audits for two-candidate contests

We use the terms 'ballot' and 'ballot card' interchangeably, even though typical ballots in the US consist of more than one card (and the distinction does matter for workload and for auditing methods). We consider unweighted *ballot-polling* audits, which require only the ability to sample uniformly at random from all ballot cards.

The sampling is typically sequential. We draw an initial sample and assess the evidence for or against the reported outcome. If there is sufficient evidence that the reported outcome is correct, we stop and 'certify' the winner. Otherwise, we inspect more ballots and try again, possibly continuing to a full manual tabulation. At any time, the auditor can chose to conduct a full hand count rather than continue to sample at random. That might occur if the work of continuing the audit is anticipated to be higher than that of a full hand count or if the audit data suggest that the reported outcome is wrong. One reasonable rule is to set a maximum sample size (number of draws, not necessarily the number of distinct ballots) for the audit; if the sample reaches that size but the

outcome has not been confirmed, there is a full manual tabulation. The outcome according to that manual tabulation becomes official.

There are many choices to be made, including:

How to assess evidence. Each stage involves calculating a statistic from the sample. What statistic do we use? This is one key difference amongst auditing methods, see Section 3.

Threshold for evidence. The decision of whether to certify or keep sampling is done by comparing the statistic to a reference value. Often the value is chosen such that it limits the probability of certifying the outcome if the outcome is wrong, i.e. limits the risk (see below).

Sampling with or without replacement. Sampling may be done with or without replacement. Sampling without replacement is more efficient; sampling with replacement often yields simpler mathematics. The difference in efficiency is small unless a substantial fraction (e.g. 20% or more) of the ballots are sampled.

Sampling increments. By how much do we increase the sample size if the current sample does not confirm the outcome? We could enlarge the sample one ballot at a time, but it is usually more efficient to have larger 'rounds'. The methods described here can accommodate rounds of any size.

We assume that the auditors read votes correctly, which generally requires retrieving the correct ballots and correctly applying legal rules for interpreting voters' marks.

2.2 Notation

Let $X_1, X_2, \cdots \in \{0, 1\}$ denote the sampled ballots, with $X_i = 1$ representing a vote in favour of the reported winner and $X_i = 0$ a vote for the reported loser.

Let n denote the number of (not necessarily distinct) ballots sampled at a given point in the audit, m the maximum sample size (i.e. number of draws) for the audit, and N the total number of cast ballots. We necessarily have $n \leqslant m$ and if sampling without replacement we also have $m \leqslant N$.

Each audit method summarizes the evidence in the sample using a statistic of the form $S_n(X_1, X_2, \ldots, X_n, n, m, N)$. For brevity, we suppress n, m and N in the notation.

Let $Y_n = \sum_{i=1}^{n} X_i$ be the number of sampled ballots that are in favour of the reported winner. Since the ballots are by assumption exchangeable, the statistics used by most methods can be written in terms of Y_n.

Let T be the *true* total number of votes for the winner and $p_T = T/N$ the true proportion of such votes. Let p_r be the *reported* proportion of votes for the winner. We do not know T nor p_T, and it is not guaranteed that $p_r \simeq p_T$.

For sampling with replacement, conditional on n, Y_n has a binomial distribution with parameters n and p_T. For sampling without replacement, conditional on n, Y_n has a hypergeometric distribution with parameters n, T and N.

2.3 Risk-limiting audits as hypothesis tests

Risk-limiting audits amount to statistical hypothesis tests. The null hypothesis H_0 is that the reported winner(s) did *not* really win. The alternative H_1 is that the reported winners really won. For a single-winner contest,

$$H_0 : p_T \leqslant \tfrac{1}{2}, \quad \text{(reported winner is false)}$$
$$H_1 : p_T > \tfrac{1}{2}. \quad \text{(reported winner is true)}$$

If we reject H_0, we certify the election without a full manual tally. The *certification rate* is the probability of rejecting H_0. Hypothesis tests are often characterized by their *significance level* (false positive rate) and *power*. Both have natural interpretations in the context of election audits by reference to the certification rate. The power is simply the certification rate when H_1 is true. Higher power reduces the chance of an unnecessary recount. A false positive is a *miscertification*: rejecting H_0 when in fact it is true. The probability of miscertification depends on p_T and the audit method, and is known as the *risk* of the method. In a two-candidate plurality contest, the maximum possible risk is typically attained when $p_T = \frac{1}{2}$.

For many auditing methods we can find an upper bound on the maximum possible risk, and can also set their evidence threshold such that the risk is limited to a given value. Such an upper bound is referred to as a *risk limit*, and methods for which this is possible are called *risk-limiting*. Some methods are explicitly designed to have a convenient mechanism to set such a bound, for example via a formula. We call such methods *automatically risk-limiting*.

Audits with a sample size limit m become full manual tabulations if they have not stopped after drawing the mth ballot. Such a tabulation is assumed to find the correct outcome, so the power of a risk-limiting audit is 1. We use the term 'power' informally to refer to the chance the audit stops after drawing m or fewer ballots.

3 Election auditing methods

We describe Bayesian audits in some detail because they provide a mathematical framework for many (but not all) of the other methods. We then describe the other methods, many of which can be viewed as Bayesian audits for a specific choice of the prior distribution. Some of these connections were previously described by [11]. These connections can shed light on the performance or interpretation of the other methods. However, our benchmarking experiments are frequentist, even for the Bayesian audits (for example, we calibrate the methods to limit the risk).

Table 1 lists the methods described here; the parameters of the methods are defined below.

Table 1. Summary of auditing methods. The methods in the first part of the table are benchmarked in this report.

Method	Quantities to set	Automatically risk-limiting
Bayesian	$f(p)$	—
Bayesian (risk-max.)	$f(p)$, for $p > 0.5$	✓
BRAVO	p_1	✓
MaxBRAVO	None	—
ClipAudit	None	—[†]
KMart	$g(\gamma)$[‡]	✓
Kaplan–Wald	γ	✓
Kaplan–Markov	γ	✓
Kaplan–Kolmogorov	γ	✓

[†] Provides a pre-computed table for approximate risk-limiting thresholds
[‡] Extension introduced here

3.1 Bayesian audits

Bayesian audits quantify evidence in the sample as a posterior distribution of the proportion of votes in favour of the reported winner. In turn, that distribution induces a (posterior) probability that the outcome is wrong, $\Pr(H_0 \mid Y_n)$, the *upset probability*.

The posterior probabilities require positing a *prior distribution*, f for the reported winner's vote share p. (For clarity, we denote the fraction of votes for the reported winner by p when we treat it as random for Bayesian inference and by p_T to refer to the actual true value.)

We represent the posterior using the posterior odds,

$$\frac{\Pr(H_1 \mid X_1, \ldots, X_n)}{\Pr(H_0 \mid X_1, \ldots, X_n)} = \frac{\Pr(X_1, \ldots, X_n \mid H_1)}{\Pr(X_1, \ldots, X_n \mid H_0)} \times \frac{\Pr(H_1)}{\Pr(H_0)}.$$

The first term on the right is the *Bayes factor* (BF) and the second is the prior odds. The prior odds do not depend on the data: the information from the data is in the BF. We shall use the BF as the statistic, S_n. It can be expressed as,

$$S_n = \frac{\Pr(X_1, \ldots, X_n \mid H_1)}{\Pr(X_1, \ldots, X_n \mid H_0)} = \frac{\int_{p>0.5} \Pr(Y_n \mid p)\, f(p)\, dp}{\int_{p \leqslant 0.5} \Pr(Y_n \mid p)\, f(p)\, dp}.$$

The term $\Pr(Y_n \mid p)$ is the *likelihood*. The BF is similar to a likelihood ratio, but the likelihoods are integrated over p rather than evaluated at specific values (in contrast to classical approaches, see Section 3.2).

Understanding priors. The prior f determines the relative contributions of possible values of p to the BF. It can be continuous, discrete, or neither. A *conjugate prior* is often used [6], which has the property that the posterior distribution is in the same family, which has mathematical and practical advantages.

For sampling with replacement the conjugate prior is beta (which is continuous), while for sampling without replacement it is a beta-binomial (which is discrete).

Vora [11] showed that a prior that places a probability mass of 0.5 on the value $p = 0.5$ and the remaining mass on $(1/2, 1]$ is *risk-maximizing*: for such a prior, limiting the upset probability to α also limits the risk to α.

We explore several priors below, emphasizing a uniform prior (an example of a 'non-partisan prior' [6]), which is a special case within the family of conjugate priors used here.

Bayesian audit procedure. A Bayesian audit proceeds as follows. At each stage of sampling, calculate S_n and then:

$$\begin{cases} \text{if } S_n > h, & \text{terminate and certify,} \\ \text{if } S_n \leqslant h, & \text{continue sampling.} \end{cases} \qquad (*)$$

If the audit does not terminate and certify for $n \leqslant m$, there is a full manual tabulation of the votes.

The threshold h is equivalent to a threshold on the upset probability: $\Pr(H_0 \mid Y_n) < \upsilon$ corresponds to $h = \frac{1-\upsilon}{\upsilon} \frac{\Pr(H_0)}{\Pr(H_1)}$. If the prior places equal probability on the two hypotheses (a common choice), this simplifies to $h = \frac{1-\upsilon}{\upsilon}$.

Interpretation. The upset probability, $\Pr(H_0 \mid Y_n)$, is **not** the risk, which we write informally as $\max_{H_0} \Pr(\text{certify} \mid H_0)$. The procedure outlined above limits the upset probability. This is not the same as limiting the risk. Nevertheless, in the election context considered here, Bayesian audits are risk-limiting, but with a risk limit that is in general larger than the upset probability threshold.[10]

For a given prior, sampling scheme, and risk limit α, we can calculate a value of h for which the risk of the Bayesian audit with threshold h is bounded by α. For risk-maximizing priors, taking $h = \frac{1-\alpha}{\alpha}$ yields an audit with risk limit α.

3.2 SPRT-based audits

The basic sequential probability ratio test (SPRT) [12], adapted slightly to suit the auditing context here,[11] tests the simple hypotheses

$$H_0 \colon p_T = p_0,$$
$$H_1 \colon p_T = p_1,$$

[10] This is a consequence of the fact that the risk is maximized when $p_T = 0.5$, a fact that we can use to bound the risk by choosing an appropriate value for the threshold. We include the mathematical details of this result in a technical appendix available at: https://arxiv.org/abs/2008.08536

[11] The SPRT allows rejection of either H_0 or H_1, but we only allow the former here. This aligns it with the broader framework for election audits described earlier. Also, we impose a maximum sample size, as we do for the other methods.

using the likelihood ratio:

$$\begin{cases} \text{if } S_n = \frac{\Pr(Y_n|p_1)}{\Pr(Y_n|p_0)} > \frac{1}{\alpha}, & \text{terminate and certify (reject } H_0), \\ \text{otherwise,} & \text{continue sampling.} \end{cases}$$

This is equivalent to (*) for a prior with point masses of 0.5 on the values p_0 and p_1 with $h = 1/\alpha$. This procedure has a risk limit of α.

The test statistic can be tailored to sampling with or without replacement by using the appropriate likelihood. The SPRT has the smallest expected sample size among all level α tests of these same hypotheses. This optimality holds only when no constraints are imposed on the sampling (such as a maximum sample size).

The SPRT statistic is a nonnegative martingale when H_0 holds; Kolmogorov's inequality implies that it is automatically risk-limiting. Other martingale-based tests are discussed in Section 3.4.

The statistic from a Bayesian audit can also be a martingale, if the prior is the true data generating process under H_0. This occurs, for example, for a risk-maximizing prior if $p_T = 0.5$.[12]

BRAVO. In a two-candidate contest, BRAVO [3] applies the SPRT with:

$$p_0 = 0.5,$$
$$p_1 = p_r - \epsilon,$$

where ϵ is a pre-specified small value for which $p_1 > 0.5$.[13] Because it is the SPRT, BRAVO has a risk limit no larger than α.

BRAVO requires picking p_1 (analogous to setting a prior for a Bayesian audit). The recommended value is based on the reported winner's share, but the SPRT can be used with any alternative. Our numerical experiments do not involve a reported vote share; we simply set p_1 to various values.

MaxBRAVO. As an alternative to specifying p_1, we experimented with replacing the likelihood, $\Pr(Y_n \mid p_1)$, with the maximized likelihood, $\max_{p_1} \Pr(Y_n \mid p_1)$, leaving other aspects of the test unchanged. This same idea has been used in other contexts, under the name MaxSPRT [2]. We refer to our version as *MaxBRAVO*. Because of the maximization, the method is not automatically risk-limiting, so we calibrate the stopping threshold h numerically to attain the desired risk limit, as we do for Bayesian audits.

3.3 ClipAudit

Rivest [5] introduces *ClipAudit*, a method that uses a statistic that is very easy to calculate, $S_n = (A_n - B_n)/\sqrt{A_n + B_n}$, where $A_n = Y_n$ and $B_n = n - Y_n$. Approximately risk-limiting thresholds for this statistic were given (found numerically),

[12] Such a prior places all its mass on $p = 0.5$ when $p \leqslant 0.5$.

[13] The SPRT can perform poorly when $p_T \in (p_0, p_1)$; taking $\epsilon > 0$ protects against the possibility that the reported winner really won, but not by as much as reported.

along with formulae that give approximate thresholds. We used ClipAudit with the 'best fit' formula [5, equation (6)].

As far as we can tell, ClipAudit is not related to any of the other methods we describe here, but S_n is the test statistic commonly used to test the hypothesis $H_0: p_T = 0.5$ against $H_1: p_T > 0.5$:

$$S_n = \frac{A_n - B_n}{\sqrt{A_n + B_n}} = \frac{Y_n - n + Y_n}{\sqrt{n}} = \frac{Y_n/n - 0.5}{\sqrt{0.5 \times (1 - 0.5)/n}} = \frac{\hat{p}_T - p_0}{\sqrt{p_0 \times (1 - p_0)/n}}.$$

3.4 Other methods

Several martingale-based methods have been developed for the general problem of testing hypotheses about the mean of a non-negative random variable. SHANGRLA exploits this generality to allow auditing of a wide class of elections [8]. While we did not benchmark these methods in our study (they are better suited for other scenarios, such as comparison audits, and will be less efficient in the simple case we consider here), we describe them here in order to point out some connections among the methods.

The essential difference between methods is in the definition of the statistic, S_n. Given the statistic, the procedure is the same: certify the election if $S_n > 1/\alpha$; otherwise, keep sampling. All of the procedures can be shown to have risk limit α.

All the procedures involve a parameter γ that prevents degenerate values of S_n. This parameter either needs to be set to a specific value or is integrated out.

The statistics below that are designed for sampling without replacement depend on the order in which ballots are sampled. None of the other statistics (in this section or earlier) have that property.

We use t to denote the value of $\mathbb{E}(X_i)$ under the null hypothesis. In the two-candidate context discussed in this paper, $t = p_0 = 0.5$.

We have presented the formulae for the statistics a little differently to highlight the connections among these methods. For simplicity of notation, we define $Y_0 = 0$.

KMart. This method was described online under the name *KMart*[14] and is implemented in SHANGRLA [8]. There are two versions of the test statistic, designed for sampling with or without replacement,[15] respectively:

$$S_n = \int_0^1 \prod_{i=1}^{n} \left(\gamma \left[\frac{X_i}{t} - 1 \right] + 1 \right) d\gamma, \text{ and } S_n = \int_0^1 \prod_{i=1}^{n} \left(\gamma \left[X_i \frac{\left(\frac{N-i+1}{N}\right)}{t - \frac{1}{N} Y_{i-1}} - 1 \right] + 1 \right) d\gamma.$$

This method is related to Bayesian audits for two-candidate contests: for sampling with replacement and no invalid votes, we have shown that KMart

[14] https://github.com/pbstark/MartInf/blob/master/kmart.ipynb

[15] When sampling without replacement, if we ever observe $Y_n > Nt$ then we ignore the statistic and terminate the audit since H_1 is guaranteed to be true.

is equivalent to a Bayesian audit with a risk-maximizing prior that is uniform over $p > 0.5$.[16] The same analysis shows how to extend KMart to be equivalent to using an arbitrary risk-maximizing prior, by inserting an appropriately constructed weighting function $g(\gamma)$ into the integrand.[16]

There is no direct relationship of this sort for the version of KMart that uses sampling without replacement, since this statistic depends on the order the ballots are sampled but the statistic for Bayesian audits does not.

Kaplan–Wald. This method is similar to KMart but involves picking a value for γ rather than integrating over γ [10]. The previous proof[16] shows that for sampling with replacement, Kaplan–Wald is equivalent to BRAVO with $p_1 = (\gamma + 1)/2$; while for sampling without replacement, there is no such relationship.

Kaplan–Markov. This method applies Markov's inequality to the martingale $\prod_{i \leqslant n} X_i / \mathbb{E}(X_i)$, where the expectation is calculated assuming sampling with replacement [9]. This gives the statistic $S_n = \prod_{i=1}^{n} (X_i + \gamma) / (t + \gamma)$.

Kaplan–Kolmogorov. This method is the same as Kaplan–Markov but with the expectation calculated assuming sampling without replacement [8]. This gives the statistic $S_n = \prod_{i=1}^{n} \left[(X_i + \gamma)\left(\frac{N-i+1}{N}\right)\right] / \left[t - \frac{1}{N}Y_{i-1} + \frac{N-i+1}{N}\gamma\right]$.[17]

4 Evaluating auditing methods

We evaluated the methods using simulations; see the first part of Table 1.

For each method, the termination threshold h was calibrated numerically to yield maximum risk as close as possible to 5%. This makes comparisons among the methods 'fair'. We calibrated even the automatically risk-limiting methods, resulting in a slight performance boost. We also ran some experiments without calibration, to quantify this difference.

We use three quantities to measure performance: maximum risk and 'power', defined in Section 2.3, and the mean sample size.

Choice of auditing methods. Most of the methods require choosing the form of statistics, tuning parameters, or a prior. Except where stated, our benchmarking experiments used sampling without replacement. Except where indicated, we used the version of each statistic designed for the method of sampling used. For example, we used a hypergeometric likelihood when sampling without replacement. For Bayesian audits we used a beta-binomial prior (conjugate to the hypergeometric likelihood) with shape parameters a and b. For BRAVO, we tried several values of p_1.

[16] We include the mathematical details of these results in a technical appendix available at: https://arxiv.org/abs/2008.08536

[17] As for KMart, if $Y_n > Nt$, the audit terminates: the null hypothesis is false.

The tests labelled 'BRAVO' are tests of a method related to but not identical to BRAVO, because there is no notion of a 'reported' vote share in our experiments. Instead, we set p_1 to several fixed values to explore how the underlying test statistic (from the SPRT) performs in different scenarios.

For MaxBRAVO and Bayesian audits with risk-maximizing prior, due to time constraints we only implemented statistics for the binomial likelihood (which assumes sampling with replacement). While these are not exact for sampling without replacement, we believe this choice has only a minor impact when $m \ll N$ (based on our results for the other methods when using different likelihoods).

For Bayesian audits with a risk-maximizing prior, we used a beta distribution prior (conjugate to the binomial likelihood) with shape parameters a and b.

ClipAudit only has one version of its statistic. It is not optimized for sampling without replacement (for example, if you sample **all** of the ballots, it will not 'know' this fact), but the stopping thresholds are calibrated for sampling without replacement.

Election sizes and sampling designs. We explored combinations of election sizes $N \in \{500, 1000, 5000, 10000, 20000, 30000\}$ and maximum sample sizes $m \in \{500, 1000, 2000, 3000\}$. Most of our experiments used a sampling increment of 1 (i.e. check the stopping rule after each ballot is drawn). We also varied the sampling increment (values in $\{2, 5, 10, 20, 50, 100, 250, 500, 1000, 2000\}$) and tried sampling with replacement.

Benchmarking via dynamic programming. We implemented an efficient method for calculating the performance measures using dynamic programming.[18] This exploits the Markovian nature of the sampling procedure and the low dimensionality of the (univariate) statistics. This approach allowed us to calculate—for elections with up to tens of thousands of votes—exact values of each of the performance measures, including the tail probabilities of the sampling distributions, which require large sample sizes to estimate accurately by Monte Carlo. We expect that with some further optimisations our approach would be computationally feasible for larger elections (up to 1 million votes). The complexity largely depends on the maximum sample size, m. As long as this is moderate (thousands) our approach is feasible. For more complex audits (beyond two-candidate contests), a Monte Carlo approach is likely more practical.

5 Results

5.1 Benchmarking results

Sample size distributions. Different methods have different distributions of sample sizes; Figure 1 shows these for a few methods when $p_T = 0.5$. Some methods tend to stop early; others take many more samples. Requiring a minimum sample size might improve performance of some of the methods; see Section 5.3.

[18] Our code is available at: https://github.com/Dovermore/AuditAnalysis

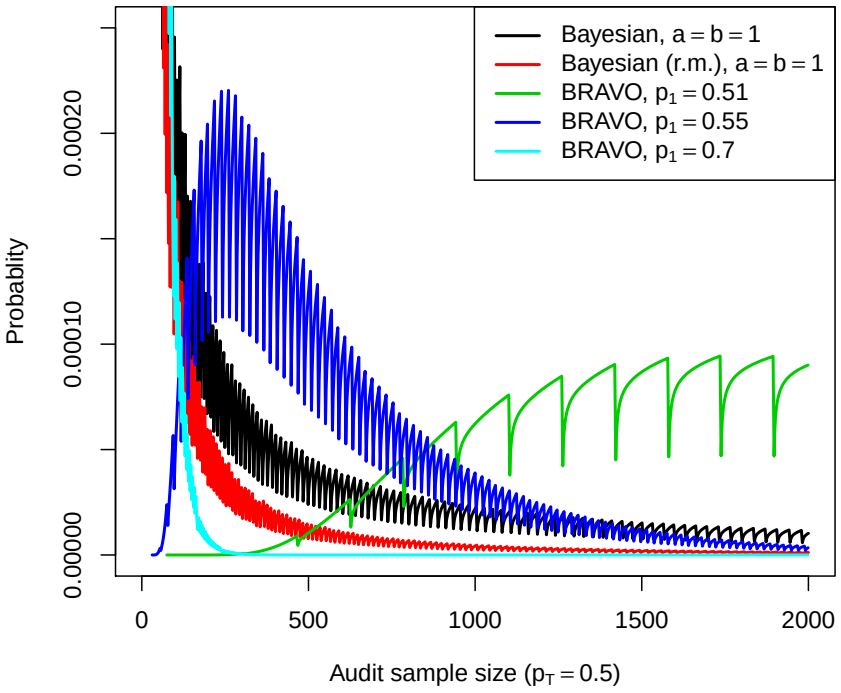


Fig. 1. Sample size distributions. Audits of elections with $N = 20,000$ ballots, maximum sample size $m = 2,000$, and true vote share a tie ($p_T = 0.5$). Each method is calibrated to have maximum risk 5%. The depicted probabilities all sum to 0.05; the remaining 0.95 probability in each case is on the event that the audit reaches the full sample size ($n = m$) and progresses to a full manual tabulation. 'Bayesian (r.m.)' refers to the Bayesian audit with a risk-maximizing prior. The sawtooth pattern is due to the discreteness of the statistics.

Mean sample sizes. We focus on average sample sizes as a measure of audit efficiency. Table 2 shows the results of experiments with $N = 20,000$ and $m = 2,000$. We discuss other experiments and performance measures below.

No method was uniformly best. Given the equivalence of BRAVO and Bayesian audits, the comparisons amount to examining dependence on the prior.

In general, methods that place more weight on close elections, such as BRAVO with $p_1 = 0.55$ or a Bayesian audit with a moderately constrained prior ($a = b = 100$) were optimal when p_T was closer to 0.5. Methods with substantial prior weight on wider margins, such as BRAVO with $p_1 = 0.7$ and Bayesian audits with the risk-maximizing prior, perform poorly for close elections.

Consistent with theory, BRAVO was optimal when the assumptions matched the truth ($p_1 = p_T$). However, our experiments violate the theoretical assumptions because we imposed a maximum sample size, m. (Indeed, when $p_1 = p_T = 0.51$, BRAVO is no longer optimal in our experiments.)

Table 2. Results from benchmarking experiments. Audits of elections with $N = 20,000$ ballots and a maximum sample size $m = 2,000$. The numeric column headings refer to the value of p_T; the corresponding margin of victory (MOV) is also reported. Each row refers to a specific auditing method. For calibrated methods, we report the threshold obtained. For easier comparison, we present these on the nominal risk scale for BRAVO, MaxBRAVO and ClipAudit (e.g. $\alpha = 1/h$ for BRAVO), and on the upset probability scale for the Bayesian methods ($\upsilon = 1/(h+1)$). For the experiments without calibration, we report the maximum risk of each method when set to a 'nominal' risk limit of 5%. We only report uncalibrated results for methods that are automatically risk-limiting, as well as ClipAudit using its 'best fit' formula to set the threshold. 'Bayesian (r.m.)' refers to the Bayesian audit with a risk-maximizing prior. The numbers in bold are those that are (nearly) best for the given experiment and choice of p_T. The section labelled '$n \geqslant 300$' refers to experiments that required the audit to draw at least 300 ballots.

		Power (%)			**Mean sample size**				
	p_T (%) →	52	55	60	52	55	60	64	70
Method	MOV (%) →	4	10	20	4	10	12	28	40
Calibrated	α or υ (%)								
Bayesian, $a = b = 1$	0.2	35	**99**	**100**	1623	637	**172**	**90**	46
Bayesian, $a = b = 100$	1.2	48	**100**	**100**	**1551**	616	232	150	97
Bayesian, $a = b = 500$	3.6	**53**	**100**	**100**	1582	709	318	219	149
Bayesian (r.m.), $a = b = 1$	6.1	19	94	**100**	1742	813	185	**89**	41
BRAVO, $p_1 = 0.7$	5.8	9	21	84	1828	1592	530	95	**37**
BRAVO, $p_1 = 0.55$	5.3	37	**99**	**100**	**1549**	**562**	196	129	85
BRAVO, $p_1 = 0.51$	22.7	**55**	**100**	**100**	1617	791	384	272	190
MaxBRAVO	1.6	30	**98**	**100**	1660	680	177	**91**	45
ClipAudit	4.7	33	**98**	**100**	1630	639	**169**	**89**	45
Calibrated, $n \geqslant 300$	α or υ (%)								
Bayesian, $a = b = 1$	0.6	45	**99**	**100**	1547	601	**311**	**300**	**300**
Bayesian (r.m.), $a = b = 1$	34.4	39	**99**	**100**	1554	587	**307**	**300**	**300**
BRAVO, $p_1 = 0.7$	100.0	0	6	83	1994	1900	708	309	**300**
BRAVO, $p_1 = 0.55$	6.0	38	**99**	**100**	**1545**	**583**	**309**	**300**	**300**
BRAVO, $p_1 = 0.51$	22.7	**55**	**100**	**100**	1617	791	392	313	**300**
MaxBRAVO	5.0	44	**99**	**100**	1546	595	**310**	**300**	**300**
ClipAudit	11.4	44	**99**	**100**	**1545**	595	**310**	**300**	**300**
Uncalibrated	Risk (%)								
Bayesian (r.m.), $a = b = 1$	3.7	17	93	**100**	1785	864	198	95	44
BRAVO, $p_1 = 0.7$	4.3	8	20	83	1846	1621	552	99	**38**
BRAVO, $p_1 = 0.55$	4.7	**37**	**98**	**100**	**1561**	**572**	200	131	86
BRAVO, $p_1 = 0.51$	0.029	6	89	**100**	1985	1505	760	542	377
ClipAudit	5.1	**34**	**98**	**100**	1618	628	**167**	**88**	45

Two methods were consistently poor: BRAVO with $p_1 = 0.51$ and a Bayesian audit with $a = b = 500$. Both place substantial weight on a very close election.

MaxBRAVO and ClipAudit, the two methods without a direct match to Bayesian audits, performed similarly to a Bayesian audit with a uniform prior ($a = b = 1$). All three are 'broadly' tuned: they perform reasonably well in most scenarios, even when they are not the best.

Effect of calibration on the uncalibrated methods. For most of the automatically calibrated methods, calibration had only a small effect on performance. BRAVO with $p_1 = 0.51$ is an exception: it was very conservative because it normally requires more than m samples.

Other election sizes and performance measures. The broad conclusions are the same for a range of values of m and N, and when performance is measured by quantiles of sample size or probability of stopping without a full hand count rather than by average sample size.

Sampling with vs without replacement. There are two ways to change our experiments to explore sampling with replacement: (i) construct versions of the statistics specifically for sampling with replacement; (ii) leave the methods alone but sample with replacement. We explored both options, separately and combined; differences were minor when $m \ll N$.

5.2 Choosing between methods

Consider the following two methods, which were the most efficient for different election margins: (i) BRAVO with $p_1 = 0.55$; (ii) ClipAudit. For $p_T = 0.52$, the mean sample sizes are 1,549 vs 1,630 (BRAVO saved 81 draws on average). For $p_T = 0.7$, the equivalent numbers are 85 vs 45 (ClipAudit saved 40 draws on average).

Picking a method requires trade-offs involving resources, workload predictability, and jurisdictional idiosyncrasies in ballot handling and storage—as well as the unknown true margin. Differences in expected sample size across ballot-polling methods might be immaterial in practice compared to other desiderata.

5.3 Exploring changes to the methods

Increasing the sampling increment ('round size'). Increasing the number of ballots sampled in each 'round' increases the chance that the audit will stop without a full hand count but increases mean sample size. This is as expected; the limiting version is a single fixed sample of size $n = m$, which has the highest power but loses the efficiency that early stopping can provide.

Increasing the sampling increment had the most impact on methods that tend to stop early, such as Bayesian audits with $a = b = 1$, and less on methods

that do not, such as BRAVO with $p_1 = 0.51$. Increasing the increment also decreases the differences among the methods. This makes sense because when the sample size is m, the methods are identical (since all are calibrated to attain the risk limit).

Considering the trade-off discussed in the previous section, since increasing the sampling increment improves power but increases mean sample size, it reduces effort when the election is close, but increases it when the margin is wide.

Increasing the maximum sample size (m). Increasing m has the same effect as increasing the sampling increment: higher power at the expense of more work on average. This effect is stronger for closer elections, since sampling will likely stop earlier when the margin is wide.

Requiring/encouraging more samples. The Bayesian audit with $a = b = 1$ tends to stop too early, so we tried two potential improvements, shown in Table 2.

The first was to impose a minimum sample size, in this case $n \geqslant 300$. This is very costly if the margin is wide, since we would not normally require this many samples. However, it boosts the power of this method and reduces its expected sample size for close contests.

A gentler way to achieve the same aim is to make the prior more informative, by increasing a and b. When $a = b = 100$, we obtain largely the same benefit for close elections with a much milder penalty when the margin is wide. The overall performance profile becomes closer to BRAVO with $p_1 = 0.55$.

6 Discussion

We compared several ballot-polling methods both analytically and numerically, to elucidate the relationships among the methods. We focused on two-candidate contests, which are building blocks for auditing more complex elections. We explored modifications and extensions to existing procedures. Our benchmarking experiments calibrated the methods to attain the same maximum risk.

Many 'non-Bayesian' auditing methods are special cases of a Bayesian procedure for a suitable prior, and Bayesian methods can be calibrated to be risk-limiting (at least, in the two-candidate, all-valid-vote context investigated here). Differences among such methods amount to technical details, such as choices of tuning parameters, rather than something more fundamental. Of course, upset probability *is* fundamentally different from risk.

No method is uniformly best, and most can be 'tuned' to improve performance for elections with either closer or wider margins—but not both simultaneously. If the tuning is not extreme, performance will be reasonably good for a wide range of true margins. In summary:

1. If the true margin is known approximately, BRAVO is best.
2. Absent reliable information on the margin, ClipAudit and Bayesian audits with a uniform prior (calibrated to attain the risk limit) are efficient.

3. Extreme settings, such as $p_1 \approx 0.5$ or an overly informative prior may result in poor performance even when the margin is small. More moderate settings give reasonable or superior performance if the maximum sample size is small compared to the number of ballots cast.

Choosing a method often involves a trade-off in performance between narrow and wide margins.

There is more to auditing than the choice of statistical inference method. Differences in performance across many 'reasonable' methods are small compared to other factors, such as how ballots are organized and stored.

Future work: While we tried to be comprehensive in examining ballot-polling methods for two-candidate contests with no invalid votes, there are many ways to extend the analysis to cover more realistic scenarios. Some ideas include: (i) more than two candidates and non-plurality social choice functions; (ii) invalid votes; (iii) larger elections; (iv) stratified samples; (v) batch-level audits; (vi) multi-page ballots.

References

1. Blom, M., Stuckey, P.J., Teague, V.J.: Ballot-polling risk limiting audits for IRV elections. In: Electronic Voting. pp. 17–34. Springer, Cham (2018)
2. Kulldorff, M., Davis, R.L., Kolczak, M., Lewis, E., Lieu, T., Platt, R.: A maximized sequential probability ratio test for drug and vaccine safety surveillance. Sequential Analysis **30**(1), 58–78 (2011). https://doi.org/10.1080/07474946.2011.539924
3. Lindeman, M., Stark, P.B., Yates, V.S.: BRAVO: Ballot-polling risk-limiting audits to verify outcomes. In: 2012 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE '12) (2012)
4. National Academies of Sciences, Engineering, and Medicine: Securing the Vote: Protecting American Democracy. The National Academies Press, Washington, DC (Sep 2018). https://doi.org/10.17226/25120
5. Rivest, R.L.: ClipAudit: A simple risk-limiting post-election audit. arXiv e-prints arXiv:1701.08312 (Jan 2017)
6. Rivest, R.L., Shen, E.: A Bayesian method for auditing elections. In: 2012 Electronic Voting Technology/Workshop on Trustworthy Elections (EVT/WOTE '12) (2012)
7. Stark, P.: Conservative statistical post-election audits. Ann. Appl. Stat. **2**, 550–581 (2008), http://arxiv.org/abs/0807.4005
8. Stark, P.: Sets of half-average nulls generate risk-limiting audits: SHANGRLA. Voting '20 **in press** (2020), preprint: http://arxiv.org/abs/1911.10035
9. Stark, P.B.: Risk-limiting postelection audits: Conservative P-values from common probability inequalities. IEEE Transactions on Information Forensics and Security **4**(4), 1005–1014 (Dec 2009). https://doi.org/10.1109/TIFS.2009.2034190
10. Stark, P.B., Teague, V.: Verifiable European elections: Risk-limiting audits for D'Hondt and its relatives. USENIX Journal of Election Technology and Systems (JETS) **1**(3), 18–39 (Dec 2014), https://www.usenix.org/jets/issues/0301/stark
11. Vora, P.L.: Risk-Limiting Bayesian Polling Audits for Two Candidate Elections. arXiv e-prints arXiv:1902.00999 (Feb 2019)
12. Wald, A.: Sequential tests of statistical hypotheses. Ann. Math. Statist. **16**(2), 117–186 (June 1945). https://doi.org/10.1214/aoms/1177731118

Towards Model Checking of Voting Protocols in UPPAAL

Wojciech Jamroga[1,2], Yan Kim[1], Damian Kurpiewski[2], and Peter Y. A. Ryan[1]

[1] Interdisciplinary Centre for Security, Reliability, and Trust, SnT, University of Luxembourg
[2] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
{wojciech.jamroga,yan.kim,peter.ryan}@uni.lu, kurpiewski@ipipan.waw.pl

Abstract The design and implementation of a trustworthy e-voting system is a challenging task. Formal analysis can be of great help here. In particular, it can lead to a better understanding of how the voting system works, and what requirements on the system are relevant. In this paper, we propose that the state-of-art model checker UPPAAL provides a good environment for modelling and preliminary verification of voting protocols. To illustrate this, we demonstrate how to model a version of Prêt à Voter in UPPAAL, together with some natural extensions. We also show how to verify a variant of receipt-freeness, despite the severe limitations of the property specification language in the model checker.
The aim of this work is to open a new path, rather then deliver the ultimate outcome of formal analysis. A comprehensive model of Prêt à Voter, more accurate specification of requirements, and exhaustive verification are planned for the future.

1 Introduction

The design and implementation of a good e-voting system is highly challenging. Real-life systems are notoriously complex and difficult to analyze. Moreover, elections are *social* processes: they are run by humans, with humans, and for humans, which makes them unpredictable and hard to model. Last but not least, it is not always clear what *good* means for a voting system. A multitude of properties have been proposed by the community of social choice theory (such as Pareto optimality and nonmanipulability), as well as researchers who focus on the security of voting (cf. ballot secrecy, coercion-resistance, voter-verifiability, and so on). The former kind of properties are typically set for a very abstract view of the voting procedure, and consequently miss many real-life concerns. For the latter ones, it is often difficult to translate the informal intuition to a formal definition that will be commonly accepted.

In a word, we deal with processes that are hard to understand and predict, and seek to evaluate them against criteria for which we have no clear consensus. Formal analysis can be of great help here: perhaps not in the sense of providing the ultimate answers, but rather to strengthen our understanding of both how the voting system works and how it should work. The main goal of this paper

is to propose that model checkers from distributed and multi-agent systems can be invaluable tools for such an analysis.

Model checkers and UPPAAL. Much research on model checking focuses on the design of logical systems for a particular class of properties, establishing their theoretical characteristics, and development of verification algorithms. This obscures the fact that a model checking framework is valuable as long as it is actually *used* to analyze something. The analysis does not have to result in a "correctness certificate". A readable model of the system, and an understandable formula capturing the requirement are already of substantial value.

In this context, two features of a model checker are essential. On the one hand, it should provide a *flexible model specification language* that allows for modular and succinct specification of processes. On the other hand, it must offer a *good graphical user interface*. Paradoxically, tools satisfying both criteria are rather scarce. Here, we suggest that the state of the art model checker UPPAAL can provide a nice environment for modelling and preliminary verification of voting protocols and their social context. To this end, we show how to use UPPAAL to model a voting protocol of choice (in our case, a version of Prêt à Voter), and to verify some requirements written in the temporal logic **CTL**.

Contribution. The main contribution of this paper is methodological: we demonstrate that specification frameworks and tools from distributed and multi-agent systems can be useful in analysis and validation of voting procedures. An additional, technical contribution consists in a reduction from model checking of temporal-epistemic specifications to purely temporal ones, in order to verify a variant of receipt-freeness despite the limitations of UPPAAL.

We emphasize that this is a preliminary work, aimed at exploring a path rather then delivering the ultimate outcome of formal analysis. A comprehensive model of Prêt à Voter, more accurate specification of requirements, and exhaustive verification are planned for the future. We also plan to cover social engineering-style attacks involving interactions between coercers (or vote-buyers) and voters. This will require, however, a substantial extension of the algorithms in UPPAAL or a similar model checker.

Structure of the paper. We begin by introducing the main ideas behind modelling and model checking of multi-agent systems, including a brief introduction to UPPAAL (Section 2). In Section 3, we provide an overview of Prêt à Voter, the voting protocol that we will use for our study. Section 4 presents a multi-agent model of the protocol; some interesting extensions of the model are proposed in Section 6. We show how to specify simple requirements on the voting system, and discuss the output of model checking in Section 5. The section also presents our main technical contribution, namely the model checking reduction that recasts knowledge-related statements as temporal properties. We discuss related work in Section 7, and conclude in Section 8.

2 Towards Model Checking of Voting Protocols

Model checking is the decision problem that takes a model of the system and a formula specifying correctness, and determines whether the model satisfies the formula. This allows for a natural separation of concerns: the model specifies how the system is, while the formula specifies how it should be. Moreover, most model checking approaches encourage systematic specification of requirements, especially for the requirements written in modal and temporal logic. In that case, the behavior of the system is represented by a transition network, possibly with additional modal relations to capture e.g. the uncertainty of agents. The structure of the network is typically given by a higher-level representation, e.g., a set of agent templates together with a synchronization mechanism.

We begin with a brief overview of UPPAAL, the model checker that we will use in later sections. A more detailed introduction can be found in [5].

2.1 Modelling in UPPAAL

An UPPAAL model consists of a set of concurrent processes. The processes are defined by templates, each possibly having a set of parameters. The templates are used for defining a large number of almost identical processes. Every template consists of *nodes*, *edges*, and optional local declarations. An example template is shown in Figure 2; we will use it to model the behavior of a voter.

Nodes are depicted by circles and represent the local states of the module. *Initial* nodes are marked by a double circle. *Committed* nodes are marked by circled C. If any process is in a committed node, then the next transition must involve an edge from one of the committed nodes. Those are used to create atomic sequences or encode synchronization between more than two components.

Edges define the local transitions in the module. They are annotated by selections (in yellow), guards (green), synchronizations (teal), and updates (blue). The syntax of expressions mostly coincides with that of C/C++. *Selections* bind the identifier to a value from the given range in a nondeterministic way. *Guards* enable the transition if and only if the guard condition evaluates to true. *Synchronizations* allow processes to synchronize over a common channel `ch` (labeled `ch?` in the receiver process and `ch!` for the sender). Note that a transition on the side of the sender can be fired only if there exists an enabled transition on the receiving side labeled with the same channel identifier, and vice versa. *Update* expressions are evaluated when the transition is taken. Straightforward value passing over a channel is not allowed; instead, one has to use shared global variables for the transmission.

For convenience, we will place the selections and guards at the top or left of an edge, and the synchronizations and updates at the bottom/right.

2.2 Specification of Requirements

To specify requirements, UPPAAL uses a fragment of the temporal logic **CTL** [14]. **CTL** allows for reasoning about the possible execution paths of the system by

means of the *path quantifiers* E ("there is a path") and A ("for every path"). A path is a maximal[1] sequence of states and transitions. To address the temporal pattern on a path, one can use the *temporal operators* $\bigcirc$ ("in the next moment"), $\Box$ ("always from now on"), $\Diamond$ ("now or sometime in the future"), and U ("until"). For example, the formula $\mathsf{A}\Box\big(\mathsf{has\_ballot_i} \rightarrow \mathsf{A}\Diamond(\mathsf{voted_{i,1}} \vee \cdots \vee \mathsf{voted_{i,k}})\big)$ expresses that, on all paths, whenever voter i gets her ballot form, she will eventually cast her vote for one of the candidates $1, \ldots, k$. Another formula, $\mathsf{A}\Box\neg\mathsf{punished_i}$ says that voter i will never be punished by the coercer.

More advanced properties usually require a combination of temporal modalities with *knowledge operators* K_a, where $K_a\phi$ expresses "agent a knows that ϕ holds." For example, formula $\mathsf{E}\Diamond(\mathsf{results} \wedge \neg\mathsf{voted_{i,j}} \wedge \neg K_c\neg\mathsf{voted_{i,j}})$ says that the coercer c might not know that voter i hasn't voted for candidate j, even if the results are already published. Moreover, $\mathsf{A}\Box(\mathsf{results} \rightarrow \neg K_c\neg\mathsf{voted_{i,j}})$ expresses that, when the results are out, the coercer won't know that the voter refused to vote for j. Intuitively, both formulas capture different strength of receipt-freeness for a voter who has been instructed to vote for candidate j.

3 Outline of Prêt à Voter

In this paper, we use UPPAAL for modelling and analysis of a voting protocol. The protocol of choice is a version of Prêt à Voter. We stress that this is not an up to date version of Prêt à Voter but it serves to illustrate how some attacks can be captured with UPPAAL. A short overview of Prêt à Voter is presented here; the full details can be found, for example, in [32] or [19].

Most voter-verifiable voting systems work as follows: at the time of casting, an encryption or encoding of the vote is created and posted to a secure public bulletin board (BB). The voter can later check that her encrypted ballot appears correctly. The set of posted ballots are then processed in some verifiable way to reveal the tally or outcome. Much of this is effectively a secure distributed computation, and as such is well-established and understood in cryptography. The really challenging bit is the creation of the encrypted ballots, because it involves interactions between the users and the system. This has to be done in a way that assures the voter that her vote is correctly embedded, while avoiding introducing any coercion or vote buying threats.

The key innovation of the Prêt à Voter approach is to encode the vote using a randomised candidate list. This contrasts with earlier verifiable schemes that involved the voter inputting her selection to a device that then produces an encryption of the selection. Here what is encrypted is the candidate order which can be generated and committed in advance, and the voter simply marks her choice on the paper ballot in the traditional manner.

Suppose that our voter is called Anne. At the polling station, Anne is authenticated and registered and she chooses at random a ballot form sealed in an envelope and saunters over to the booth. An example of such a form is shown

[1] I.e., infinite or ending in a state with no outgoing transitions.

(a)

Discard	Retain
Obelix	
Idefix	
Asterix	
Panoramix	
	7304944

(b)

Retain
X
7304944

Figure 1: (a) Prêt à Voter ballot form; (b) Receipt encoding a vote for "Idefix"

in Figure 1a. In the booth, she extracts her ballot form from the envelope and marks her selection in the usual way by placing a cross in the right hand column against the candidate (or candidates) of her choice. Once her selection has been made, she separates the left and right hand strips and discards the left hand strip. She keeps the right hand strip which now constitutes her *privacy protected receipt*, as shown in Figure 1b.

Anne now exits the booth clutching her receipt, returns to the registration desk, and casts the receipt: it is placed over an optical reader or similar device that records the string at the bottom of the strip and registers which cells are marked. Her original paper receipt is digitally signed and franked and returned to her to keep and later check that her vote is correctly recorded on the bulletin board. The randomisation of the candidate list on each ballot form ensures that the receipt does not reveal the way she voted, thus ensuring the secrecy of her vote. Incidentally, it also removes any bias towards the candidate at the top of the list that can occur with a fixed ordering.

The value printed on the bottom of the receipt is what enables extraction of the vote during the tabulation phase: buried cryptographically in this value is the information needed to reconstruct the candidate order and so extract the vote encoded on the receipt. This information is encrypted with secret keys shared across a number of tellers. Thus, only a threshold set of tellers acting together are able to interpret the vote encoded in the receipt. In practice, the value on the receipt will be a pointer (e.g. a hash) to a ciphertext committed to the bulletin board during the setup phase.

After the voting phase, voters can visit the Bulletin Board and confirm that their receipts appear correctly. Once any discrepancies are resolved, the tellers take over and perform anonymising mixes and decryption of the receipts. At the end, the plaintext votes will be posted in secret shuffled order, or in the case of homomorphic tabulation, the final result is posted. All the processing of the votes can be made universally verifiable, i.e., any observer can check that no votes were manipulated.

Prêt à Voter brings several advantages in terms of privacy and dispute resolution. Firstly, it avoids side channel leakage of the vote from the encryption device. Secondly, it improves on dispute resolution: ballot assurance is based on random audits of the ballot forms, which can be performed by the voter or independent observers. A ballot form is either well-formed, i.e. the plaintext order

matches the encrypted order, or not. This is independent of the voter or her choice, hence there can be no dispute as to what choice the voter provided. Such disputes can arise in Benaloh challenges and similar cut-and-choose style audits. Furthermore, auditing ballots does not impinge on ballot privacy, as nothing about the voter or the vote can be revealed at this point.

4 Modelling Prêt à Voter in UPPAAL

In this section, we present how the components and participants of Prêt à Voter can be modelled in UPPAAL. To this end, we give a description of each module template, its elements, and their interactions. The templates represent the behavior of the following types of agents: *voters*, *coercers*, *mix tellers*, *decryption tellers*, *auditors*, and the *voting infrastructure*. For more than one module of a given type, an identifier $i = 0, 1, \ldots$ will be associated with each instance.

The code of the model is available at `https://github.com/pretvsuppaal/model`. Here, we present in detail only the Voter template. The details of the other modules can be found in the extended version of the paper, available at `https://arxiv.org/abs/2007.12412`.

To facilitate readability and manageability of the model code, we define some data structures and type name aliases based on the configuration variables:

- **Ciphertext**: a pair (y_1, y_2). For the simplicity of modeling, we assume that ElGamal encryption is used.
- **Ballot**: a pair (θ, cl) of onion $\theta = E_{PK}(s, *)$ and candidate list $cl = \pi(s)$, where s is a seed associated with the ballot, and $\pi : \mathbb{R} \to Perm_C$ is a function that associates a seed with a permutation of the candidates. To allow absorption of the index of a marked cell into the onion, we use cyclic shifts of the base candidate order. This means that we just have simple ElGamal ciphertexts to mix.
- **Receipt**: a pair (θ, r) of onion θ and an index r of marked cell. It can be used to verify if a term was recorded and if it was done correctly.
- **c_t**: an integer with range [0, c_total), a candidate;
- **v_t**: an integer with range [0, v_total), a voter;
- **z_t**: an integer with range [0, z_total), an element of $\mathbb{Z}_p^*$.

4.1 Voter Template

The structure of the Voter template is shown in Figure 2. The idea is that while the voter waits for the start of election she might be subject to coercion. When the ballots are ready, the voter selects a candidate, and transmits the receipt to the system. Then she decides if she wants to check how her vote has been recorded, and if she wants to show the receipt to the coercer. If coerced, she also waits for the coercer's decision to punish her or refrain from punishment. The module includes the following private variables:

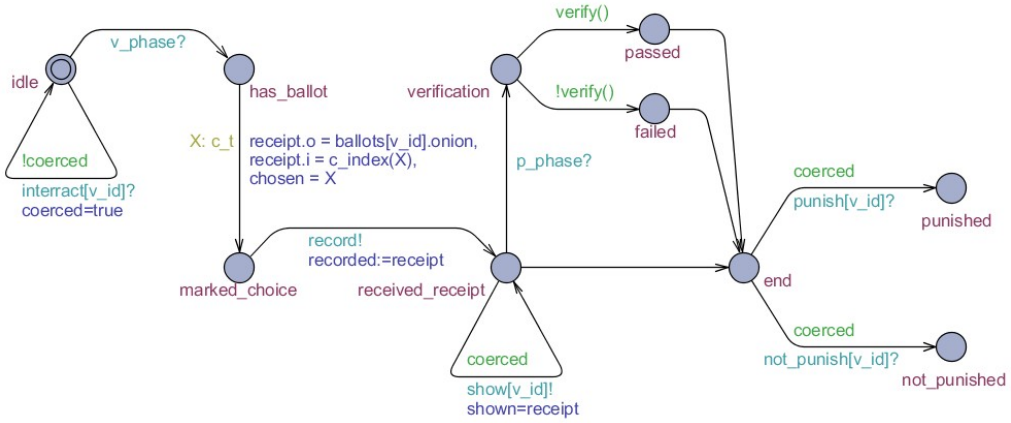


Figure 2: Voter template for the model of Prêt à Voter

- `receipt`: an instance of `Receipt`, obtained after casting a vote;
- `coerced[=false]`: a Boolean value, indicating if coercer has established a contact;
- `chosen`: integer value of chosen candidate.

Moreover, the following procedures are included:

- `c_index(target)`: returns an index, at which `target` can be found on the candidate list of a ballot;
- `verify()`: returns *true* if the voter's `receipt` can be found on the Web Bulletin Board, else it returns $false$.

Local states:

- *idle*: waiting for the election, might get contacted by coercer;
- *has_ballot*: the voter has already obtained the ballot form;
- *marked_choice*: the voter has marked an index of chosen candidate (and destroyed left hand side with candidate list);
- *received_receipt*: the receipt is obtained and might be shown to the coercer;
- *verification*: the voter has decided to verify the receipt;
- *passed*: the voter got a confirmation that the receipt appears correctly;
- *failed*: the voter obtains evidence that the receipt does not appear on BB or appears incorrectly;
- *end*: the end of the voting ceremony;
- *punished*: the voter has been punished by the coercer;
- *not_punished*: the coercer refrained from punishing the voter.

Transitions:

- *idle*→*idle*: if was not already coerced, enable transition; if taken, then set `coercion` to *true*;

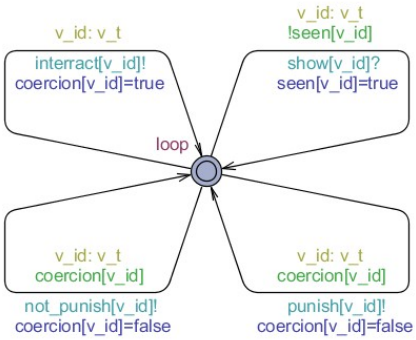


Figure 3: Coercer template

- *idle*→*has_ ballot*: always enabled; if taken, the voter acquires a ballot form;
- *has_ ballot*→*marked_ choice*: mark the cell with the selected candidate;
- *marked_ choice*→*received_ receipt*: send `receipt` to the Sys process over channel `record` using shared variable `recorded`;
- *received_ receipt*→*received_ receipt*: if was coerced, enable transition; if taken, then pass the `receipt` to the coercer using shared variable `shown`;
- *received_ receipt*→*verification*: always enabled; if taken, the voter decides to verify whether the receipt appears on the BB;
- *(received_ receipt* || *passed* || *failed)*→*end*: voting ceremony ends for the voter;
- *end*→*punished*: if was coerced, enable transition; if taken, then the voter has been punished by the coercer;
- *end*→*not_ punished*: if was coerced, enable transition; if taken, the coercer has refrained to punish the voter.

4.2 Coercer

The coercer can be thought of as a party that tries to influence t he outcome of the vote by forcing voters to obey certain instructions. To enforce this, the coercer can punish the voter. The structure of the Coercer module is presented in Figure 3; see the extended version of the paper at `https://arxiv.org/abs/2007.12412` for the technical details.

4.3 Mix Teller (Mteller)

Once the mixing phase starts, each mix teller performs two re-encryption mixes. The order of turns is ascending and determined by their identifiers. The randomization factors and permutation of each mix are selected in a nondeterministic way and stored for a possible audit of re-encryption mixes. When audited, the mix teller reveals the requested links and the associated factors, thus allowing Auditor to verify that the input ciphertext maps to the output. The structure of the mix teller is shown in Figure 4.

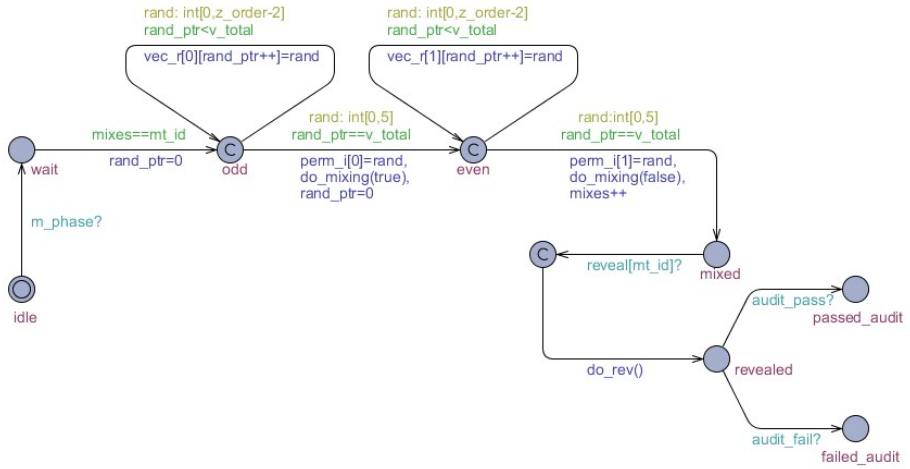


Figure 4: Mteller template

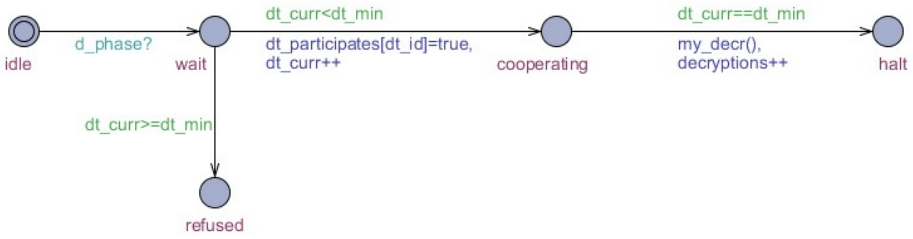


Figure 5: Dteller template

4.4 Decryption Teller (Dteller)

In this module, after the re-encryption mixes are done, a subset of cooperating decryption tellers is chosen nondeterministically. Note that if a subset has less than two elements (e.g. when two or more decryption tellers refused to cooperate), then they should not be able to reconstruct a secret key, which would lead to a deadlock. In order to avoid that, only subsets with cardinality of 2 are considered in our simplified model.

4.5 Auditor

In order to confirm that the mix tellers performed their actions correctly, the auditor conducts an audit. In this paper, we assume that the audit is based on the randomized partial checking technique, RPC in short [20]. To this end, each mix teller is requested to reveal the factors for the selected half of an odd-mix batch, and verify whether the input corresponds to the output. The control flow

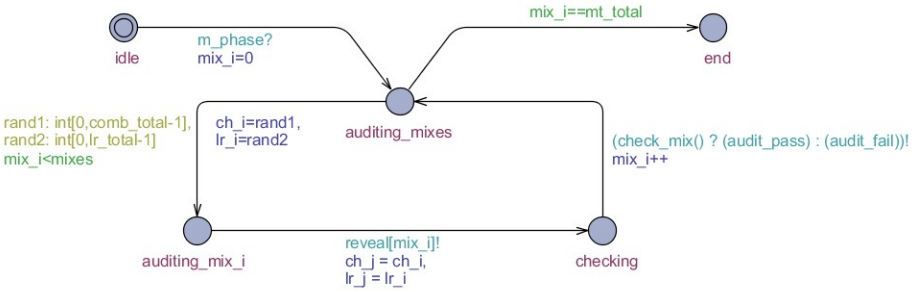


Figure 6: Auditor template

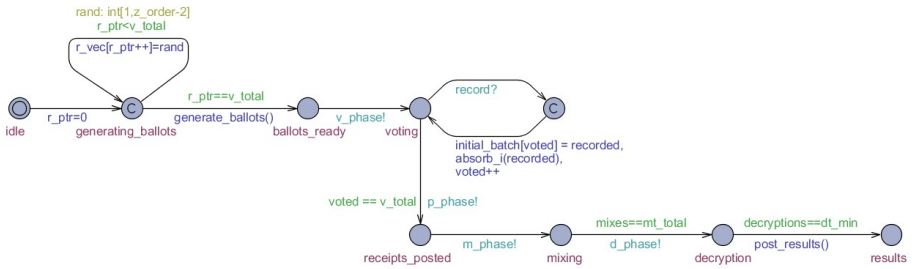


Figure 7: Module Sys

of the Auditor module is presented in Figure 6. In the future, we plan to extend the model with auditing techniques that rely on zero-knowledge proofs.

4.6 Voting Infrastructure Module (Sys)

This module represents the behavior of the election authority that prepares the ballot forms, monitors the current phase, signals the progress of the voting procedure to the other components, and at the end posts the results of the election. In addition, the module plays the role of a server that receives receipts and transfers them to the database throughout the election. We assume that all the ballots were properly generated and thus omit procedures (e.g. ballot audits) which can ensure that. Capturing related attacks and possible defences remains a subject for future work.

5 Verification

We chose UPPAAL for this study mainly because of its modelling functionality. Interestingly, the model checking capabilities of UPPAAL turned out rather limited for analysis of voting protocols, due to the limitations of its requirement specification language. First, UPPAAL admits only a fragment of **CTL**: it excludes the

"next" and "until" modalities, and does not allow for nesting of operators (with one exception that we describe below). Thus, the supported properties fall into the following categories: simple *reachability* ($\mathsf{E}\Diamond\mathsf{p}$), *liveness* ($\mathsf{A}\Diamond\mathsf{p}$), and *safety* ($\mathsf{A}\Box\mathsf{p}$ and $\mathsf{E}\Box\mathsf{p}$). The only allowed nested formulas come in the form of the p *leads to* q property, written $\mathsf{p} \rightsquigarrow \mathsf{q}$, and being a shorthand for $\mathsf{A}\Box(\mathsf{p} \rightarrow \mathsf{A}\Diamond\mathsf{q})$.

Nonetheless, UPPAAL allows to model-check some simple properties of Prêt à Voter, as we show in Section 5.1. Moreover, by tweaking models and formulas, one can also verify some more sophisticated requirements, see Section 5.2.

5.1 Model Checking Temporal Requirements

It is difficult to encode meaningful requirements on voting procedures in the input language of UPPAAL. We managed to come up with the following properties:

1. $\mathsf{E}\Diamond\mathsf{failed\_audit}_0$: the first mix teller might eventually fail an audit;
2. $\mathsf{A}\Box\neg\mathsf{punished}_i$: voter i will never be punished by the coercer;
3. $\mathsf{has\_ballot}_i \rightsquigarrow \mathsf{marked\_choice}_i$: on all paths, whenever voter i gets a ballot form, she will eventually mark her choice.

We verified each formula on the parameterized model in Section 4. Several configurations were used, with the number of voters ranging from 1 to 5. For the first property, the UPPAAL verifier returns 'Property is satisfied' for the configurations with $1, 2, 3$ and 4 voters. In case of 5 voters, we get 'Out of memory' due to the state-space explosion. This is a well-known problem in verification of distributed systems; typically, the blow-up concerns the system states to be explored in model checking and proof states in case of theorem proving. Formula (2) produces the answer 'Property is not satisfied' and pastes a counter-example into the simulator for all the five configurations. Finally, formula (3) ends with 'Out of memory' regardless of the number of voters.

Optimizations. To keep the model manageable and in attempt to reduce the state space, every numerical variable is defined as a bounded integer in a form of `int[min,max]`, restricting its range of values.[2] The states violating the bounds are discarded at run-time. For example, transition *has_ ballot→marked_ choice* of the Voter (Figure 2) has a selection of value `X` in the assignment of variable `chosen`. The type of `X` is `c_t`, which is an alias to `int[0,c_total-1]`, i.e., the range of meaningful candidate choices.

We also tried to keep the number of used variables minimal, as it plays an important role in the model checking procedure.

5.2 How to Make Model Checker Do More Than It Is Supposed To

Many important properties of voting refer to the knowledge of its participants. For example, receipt-freeness expresses that the coercer should never know how the voter has voted. Or, better still, that the coercer will never know if the

[2] Without the explicit bounds, the range of values would be [-32768,32768].

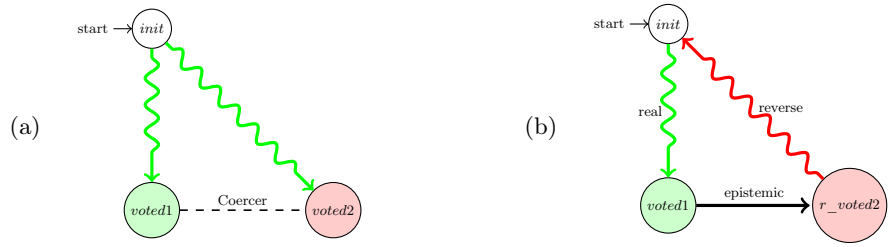


Figure 8: (a) Epistemic bisimulation triangle; (b) turning the triangle into a cycle by reversing the transition relation

voter disobeyed his instructions. Similarly, voter-verifiability says that the voter will eventually know whether her vote has been registered and tallied correctly (assuming that she follows the verification steps).

A clear disadvantage of UPPAAL is that its language for specification of requirements is restricted to purely temporal properties. Here we show that, with some care, one can use it to embed the verification of more sophisticated properties. In particular, we show how to enable model checking of some knowledge-related requirements by a technical reconstruction of models and formulas. The construction has been inspired by the reduction of epistemic properties to temporal properties, proposed in [17,21]. Consequently, UPPAAL and similar tools can be used to model check some formulas of **CTLK** (i.e., **CTL** + Knowledge) that express variants of receipt-freeness and voter-verifiability.

In order to simulate the knowledge operator K_a under the **CTL** semantics, the model needs to be modified. The first step is to understand how the formula $\neg K_c \neg \mathsf{voted}_{\mathsf{i,j}}$ (saying that the coercer doesn't know that the particular voter i hasn't voted for candidate j) is interpreted. Namely, if there is a reachable state in which $\mathsf{voted}_{\mathsf{i,j}}$ is true, there must also exist another reachable state, which is indistinguishable from the current one, and in which $\neg\mathsf{voted}_{\mathsf{i,j}}$ holds. The idea is shown in Figure 8a. We observe that to simulate the epistemic relation we need to create copies of the states in the model (the "real" states). We will refer to those copies as the *reverse states*. They are the same as the real states, but with reversed transition relation. Then, we add transitions from the real states to their corresponding reverse states, that simulate the epistemic relation between the states. This is shown in Figure 8b.

To illustrate how the reconstruction of the model works on a concrete example, we depict the augmented Coercer template in Figure 9.

In order to effectively modify the model and verify the selected properties according to the previously defined procedure, the model was first simplified. In the simplified version there are two voters and the coercer can interact only with one of them. Furthermore we removed the verification phase and the tallying phase from the model.

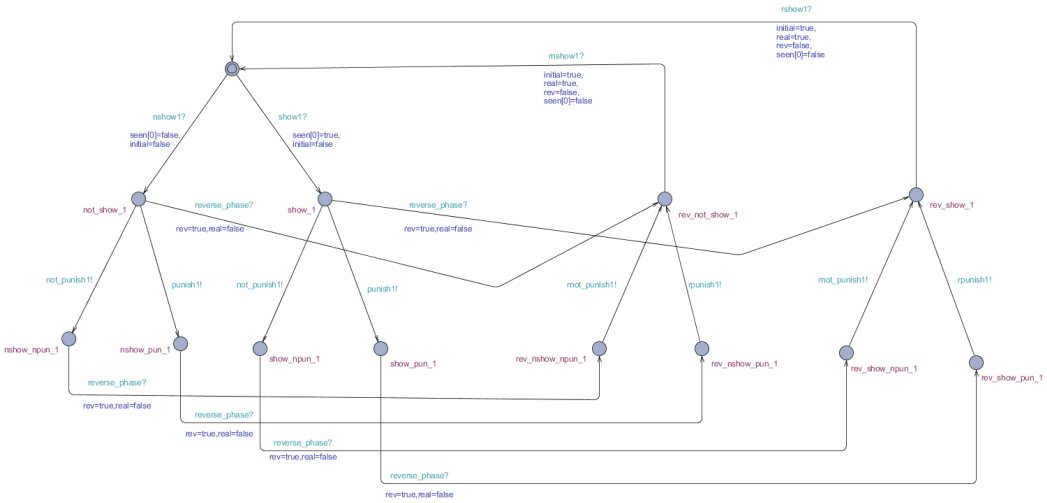


Figure 9: Coercer module augmented with the converse transition relation

The next step is the reconstruction of formulas. Let us take the formula for the weak variant of receipt-freeness from Section 2.2, i.e., $\mathsf{E}\Diamond(\mathsf{results} \wedge \neg\mathsf{voted}_{\mathsf{i,j}} \wedge \neg K_c \neg\mathsf{voted}_{\mathsf{i,j}})$. In order to verify the formula in UPPAAL, we need to replace the knowledge operator according to our model reconstruction method (see Figure 8 again). This means that the verifier should find a path that closes the cycle: from the initial state, going through the real states of the voting procedure to the vote publication phase, and then back to the initial state through the reversed states. In order to "remember" the relevant facts along the path, we use persistent Boolean variables $\mathsf{voted}_{\mathsf{i,j}}$ and $\mathsf{negvoted}_{\mathsf{i,j}}$: once set to true they always remain true. We also introduce a new persistent variable $\mathsf{epist\_voted}_{\mathsf{i,j}}$ to refer to the value of the vote after an epistemic transition. Once we have all that, we can propose the reconstructed formula: $\mathsf{E}\Diamond(\mathsf{results} \wedge \mathsf{negvoted}_{\mathsf{i,j}} \wedge \mathsf{epist\_voted}_{\mathsf{i,j}} \wedge \mathsf{initial})$. UPPAAL reports that the formula holds in the model.

A stronger variant of receipt-freeness is expressed by another formula of Section 2.2, i.e., $\mathsf{A}\Box(\mathsf{results} \rightarrow \neg K_c \neg\mathsf{voted}_{\mathsf{i,j}})$. Again, the formula needs to be rewritten to a pure **CTL** formula. As before, the model checker should find a cycle from the initial state, "scoring" the relevant propositions on the way. More precisely, it needs to check if, for every real state in which election has ended, there exist a path going back to the initial state through a reverse state in which the voter has voted for the selected candidate. This can be captured by the following formula: $\mathsf{A}\Box\big((\mathsf{results} \wedge \mathsf{real}) \rightarrow \mathsf{E}\Diamond(\mathsf{voted}_{\mathsf{i,j}} \wedge \mathsf{init})\big)$. Unfortunately, this formula cannot be verified in UPPAAL, as UPPAAL does not allow for nested path quantifiers. In the future, we plan to run the verification of this formula using another model checker LTSmin [23] that accepts UPPAAL models as input, but allows for more expressive requirement specifications.

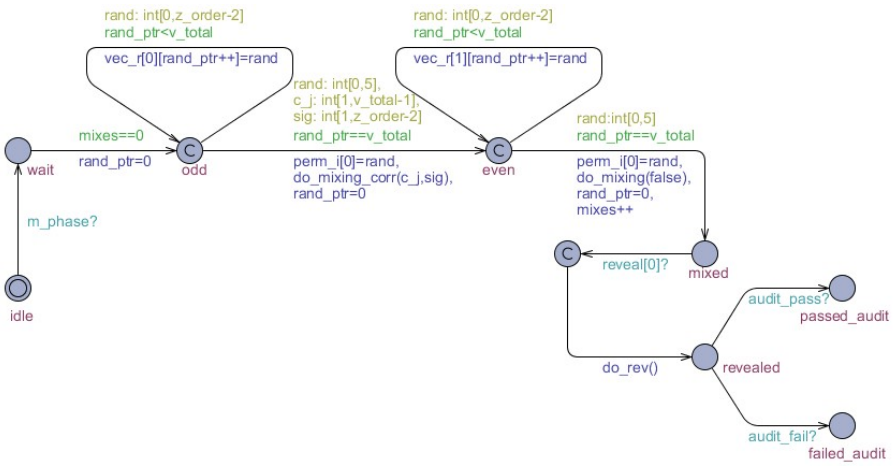


Figure 10: Corrupted Mix Teller module

6 Replicating Pfitzmann's Attack

A version of *Pfitzmann's attack* is known to compromise mix-nets with randomized partial checking [25]. It can be used to break the privacy of a given vote with probability 1/2 of being undetected. The leaked information may differ depending on both the implementation of the attack and the voting protocol.

The idea is that the first mix teller, who is corrupted, targets a ciphertext c_i from the odd mix input, and replaces some output term. c_j with c_i^δ, After the decryption results are posted, a pair of decrypted messages m and m' satisfying equation $m' = m^\delta$ can be used to identify the corresponding input terms.

Clearly, the model presented in Section 4 is too basic to allow for detection of the attack. Instead, we can examine attacker's behavior by a simple extension of the model. For that, we change the Mteller template as shown in Figure 10. The only difference lies in how the first re-encryption mix is done: the corrupted mix teller targets c_0, chooses a random non-zero δ, and uses c_0^δ instead of some other output term. We assume that the corrupt mix teller will always try to cheat. In all other respects, the teller behaves honestly.

Using UPPAAL, it can be verified that there exist executions where the corrupt mix teller's cheating behaviour is not detected during the audit. That is, both $\mathsf{E}\Diamond\mathsf{failed\_audit}_0$ and $\mathsf{E}\Diamond\mathsf{passed\_audit}_0$ produce 'Property satisfied' as the output. We note that, in order to successfully verify those properties in our model of Prêt à Voter, the search order option in UPPAAL had to be changed from the (default) `Breadth First` to either `Depth First` or `Random Depth First`.

7 Related Work

Over the years, the properties of *ballot secrecy*, *receipt-freeness*, *coercion resistance*, and *voter-verifiability* were recognized as important for an election to

work properly, see also [29] for an overview. More recently, significant progress has been made in the development of voting systems that would be coercion-resistant and at the same time allow the voter to verify "her" part of the election outcome [31,12]. A number of secure and voter-verifiable schemes have been proposed, notably Prêt à Voter for supervised elections [32], Pretty Good Democracy for internet voting [34], and Selene, a coercion mitigating form of tracking number-based, internet scheme [33].

Such schemes are starting to move out of the laboratory and into use in real elections. For example, (a variant of) Prêt à Voter has been successfully used in one of the state elections in Australia [9] while the Scantegrity II system [10] was used in municipal elections in the Takoma Park county, Maryland. Moreover, a number of verifiable schemes were used in non-political elections. E.g., Helios [1] was used to elect officials of the International Association of Cryptologic Research and the Dean of the University of Louvain la Neuve. This underlines the need for extensive analysis and validation of such systems.

Formal analysis of selected voting protocols, based on theorem proving in first-order logic or linear logic, includes attempts at verification of vote counting in [3,30]. The Coq theorem prover [6] was used to implement the STV counting scheme in a provably correct way [16], and to produce a provably voter-verifiable variant of the Helios protocol [18]. Moreover, Tamarin [28] was used to verify receipt-freeness in Selene [8] and Electryo [35]. Approaches based on model checking are fewer and include the analysis of risk-limiting audits [4] with the CBMC model checker [11]. Moreover, [22] proposed and verified a simple multi-agent model of Selene using MCMAS [27]. Related research includes the use of multi-agent methodologies to specify and verify properties of authentication and key-establishment protocols [26,7] with MCMAS. In particular, [7] used MCMAS to obtain and verify models, automatically synthesized from high-level protocol description languages such as CAPSL, thus creating a bridge between multi-agent and process-based methods.

In all the above cases, the focus is on the verification itself. Indeed, all the tools mentioned above provide only a text-based interface for specification of the system. As a result, their model specifications closely resemble programming code, and insufficiently protect from the usual pitfalls of programming: unreadability of the code, lack of modularity, and opaque control structure. In this paper, we draw attention to tools that promote modular design of the model, emphasize its control structure, and facilitate inspection and validation.

8 Conclusions

Formal methods are well established in proving (and disproving) the correctness of cryptographic protocols. What makes voting protocols special is that they prominently feature human and social aspects. In consequence, an accurate specification of the behaviors admitted by the protocol is far from straightforward. An environment that supports the creation of modular, compact, and – most of all – readable specifications can be an invaluable help.

In this context, the Uppaal model checker has a number of advantages. Its modelling language encourages modular specification of the system behavior. It provides flexible data structures, and allows for parameterized specification of states and transitions. Last but not least, it has a user-friendly GUI. Clearly, a good graphical model helps to understand how the voting procedure works, and allows for a preliminary validation of the system specification just by looking at the graphs. Anybody who ever inspected a text-based system specification or the programming code itself will know what we mean.

In this paper, we try to demonstrate the advantages of Uppaal through a case study based on a version of Prêt à Voter. The models that we have obtained are neat, easy to read, and easy to modify. On the other hand, Uppaal has not performed well with the verification itself. This was largely due to the fact that its requirement specification language turned out to be very limited – much more than it seemed at the first glance. We managed to partly overcome the limitations by a smart reconstruction of models and formulas. In the long run, however, a more promising path is to extend the implementation of verification algorithms in Uppaal so that they handle nested path quantifiers and knowledge modalities, given explicitly in the formula.

The model proposed here is far from complete. We intend to refine and expand it to capture a broader range of attacks, in particular coercion (or vote-buying attacks) that involve subtle interactions between coercer and voters. Prime examples include chain voting and randomisation attacks, where the coercer requires the voter to place an "X" in, say, the first position. Such an attack does not violate any privacy property – the coercer does not learn the vote – but it does deny the voter the freedom to cast her vote as intended. Still more subtle styles of attack have been identified against many verifiable schemes by Kelsey, [24]. Essentially any freedom the voter may have in executing the voting ceremony can potentially be exploited by a coercer.

A comprehensive discussion of coercion-resistance and its possible formalizations is also planned for future work. Another important line of research concerns data independence and saturation results. It is known that, to verify some properties, it suffices to look for small counterexamples [2]. It is also known that such results are in general impossible [15] or incur prohibitive blowup [13]. We will investigate what saturation can be achieved for the verification of Prêt à Voter.

Acknowledgements. The authors acknowledge the support of the Luxembourg National Research Fund (FNR) and the National Centre for Research and Development Poland (NCBiR) under the INTER/PolLux projects VoteVerif (POLLUX-IV/1/2016) and STV (POLLUX-VII/1/2019).

References

1. Ben Adida. Helios: web-based open-audit voting. In *Proceedings of the 17th conference on Security symposium*, SS'08, pages 335–348, Berkeley, CA, USA, 2008. USENIX Association.

2. M. Arapinis, V. Cortier, and S. Kremer. When are three voters enough for privacy properties? In *Proceedings of ESORICS*, volume 9879 of *Lecture Notes in Computer Science*, pages 241–260. Springer, 2016.
3. B. Beckert, R. Goré, and C. Schürmann. Analysing vote counting algorithms via logic - and its application to the CADE election scheme. In *Proceedings of CADE*, volume 7898 of *Lecture Notes in Computer Science*, pages 135–144. Springer, 2013.
4. B. Beckert, M. Kirsten, V. Klebanov, and C. Schürmann. Automatic margin computation for risk-limiting audits. In *Proceedings of E-Vote-ID*, volume 10141 of *Lecture Notes in Computer Science*, pages 18–35. Springer, 2016.
5. G. Behrmann, A. David, and K.G. Larsen. A tutorial on UPPAAL. In *Formal Methods for the Design of Real-Time Systems: SFM-RT*, number 3185 in LNCS, pages 200–236. Springer, 2004.
6. Y. Bertot, P. Casteran, G. Huet, and C. Paulin-Mohring. *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions*. Springer, 2004.
7. I. Boureanu, P. Kouvaros, and A. Lomuscio. Verifying security properties in unbounded multiagent systems. In *Proceedings of International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1209–1217, 2016.
8. A. Bruni, E. Drewsen, and C. Schürmann. Towards a mechanized proof of Selene receipt-freeness and vote-privacy. In *Proceedings of E-Vote-ID*, volume 10615 of *Lecture Notes in Computer Science*, pages 110–126. Springer, 2017.
9. C. Burton, C. Culnane, J. Heather, T. Peacock, P.Y.A. Ryan, S. Schneider, V. Teague, R. Wen, Z. Xia, and S. Srinivasan. Using Prêt à Voter in victoria state elections. In *Proceedings of EVT/WOTE*. USENIX, 2012.
10. D. Chaum, R.T. Carback, J. Clark, A. Essex, S. Popoveniuc, R.L. Rivest, P.Y.A. Ryan, E. Shen, A.T. Sherman, and P.L. Vora. Scantegrity II: end-to-end verifiability by voters of optical scan elections through confirmation codes. *Trans. Info. For. Sec.*, 4(4):611–627, 2009.
11. E.M. Clarke, D. Kroening, and F. Lerda. A tool for checking ANSI-C programs. In *Proceedings of TACAS*, volume 2988 of *Lecture Notes in Computer Science*, pages 168–176. Springer, 2004.
12. V. Cortier, D. Galindo, R. Küsters, J. Müller, and T. Truderung. SoK: Verifiability notions for e-voting protocols. In *IEEE Symposium on Security and Privacy*, pages 779–798, 2016.
13. W. Czerwiński, S. Lasota, R. Lazić, J. Leroux, and F. Mazowiecki. The reachability problem for petri nets is not elementary. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing STOC*, pages 24–33. Association for Computing Machinery, 2019.
14. E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier, 1990.
15. S.M. German and A.P. Sistla. Reasoning about systems with many processes. *Journal of the ACM*, 39(3):675–735, 1992.
16. M.K. Ghale, R. Goré, D. Pattinson, and M. Tiwari. Modular formalisation and verification of STV algorithms. In *Proceedings of E-Vote-ID*, volume 11143 of *Lecture Notes in Computer Science*, pages 51–66. Springer, 2018.
17. V. Goranko and W. Jamroga. Comparing semantics of logics for multi-agent systems. *Synthese*, 139(2):241–280, 2004.
18. T. Haines, R. Goré, and M. Tiwari. Verified verifiers for verifying elections. In *Proceedings of CCS*, pages 685–702. ACM, 2019.
19. Feng Hao and Peter Y. A. Ryan. *Real-World Electronic Voting: Design, Analysis and Deployment*. Auerbach Publications, USA, 1st edition, 2016.

20. M. Jakobsson, A. Juels, and R.L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In *USENIX Security Symposium*, pages 339–353, 2002.
21. W. Jamroga. Knowledge and strategic ability for model checking: A refined approach. In *Proceedings of MATES'08*, volume 5244 of *Lecture Notes in Computer Science*, pages 99–110, 2008.
22. W. Jamroga, M. Knapik, and D. Kurpiewski. Model checking the SELENE e-voting protocol in multi-agent logics. In *Proceedings of E-VOTE-ID*, volume 11143 of *Lecture Notes in Computer Science*, pages 100–116. Springer, 2018.
23. G. Kant, A. Laarman, J. Meijer, J. van de Pol, S. Blom, and T. van Dijk. LTSmin: High-performance language-independent model checking. In *Tools and Algorithms for the Construction and Analysis of Systems. Proceedings of TACAS*, volume 9035 of *Lecture Notes in Computer Science*, pages 692–707. Springer, 2015.
24. J. Kelsey, A. Regenscheid, T. Moran, and D. Chaum. *Attacking Paper-Based E2e Voting Systems*, pages 370—387. Springer-Verlag, Berlin, Heidelberg, 2010.
25. S. Khazaei and D. Wikstroem. Randomized partial checking revisited. In *Topics in Cryptology – CT-RSA 2013*, volume 7779 of *Lecture Notes in Computer Science*, pages 115–128. Springer, 2013.
26. A. Lomuscio and W. Penczek. LDYIS: a framework for model checking security protocols. *Fundamenta Informaticae*, 85(1-4):359–375, 2008.
27. A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: An open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer*, 19(1):9–30, 2017.
28. S. Meier, B. Schmidt, C. Cremers, and D.A. Basin. The TAMARIN prover for the symbolic analysis of security protocols. In *Computer Aided Verification, Proceedings of CAV*, volume 8044 of *Lecture Notes in Computer Science*, pages 696–701. Springer, 2013.
29. B. Meng. A critical review of receipt-freeness and coercion-resistance. *Information Technology Journal*, 8(7):934–964, 2009.
30. D. Pattinson and C. Schürmann. Vote counting as mathematical proof. In *Advances in Artificial Intelligence, Proceedings of AI*, volume 9457 of *Lecture Notes in Computer Science*, pages 464–475. Springer, 2015.
31. Peter Y. A. Ryan, Steve A. Schneider, and Vanessa Teague. End-to-end verifiability in voting systems, from theory to practice. *IEEE Security & Privacy*, 13(3):59–62, 2015.
32. P.Y.A. Ryan. The computer ate my vote. In *Formal Methods: State of the Art and New Directions*, pages 147–184. Springer, 2010.
33. P.Y.A. Ryan, P.B. Rønne, and V. Iovino. Selene: Voting with transparent verifiability and coercion-mitigation. In *Financial Cryptography and Data Security: Proceedings of FC 2016. Revised Selected Papers*, volume 9604 of *Lecture Notes in Computer Science*, pages 176–192. Springer, 2016.
34. P.Y.A. Ryan and V. Teague. Pretty good democracy. In *Security Protocols XVII*, volume 7028 of *Lecture Notes in Computer Science*, pages 111–130. Springer Berlin Heidelberg, 2013.
35. M-L. Zollinger, P. Roenne, and P.Y.A. Ryan. Mechanized proofs of verifiability and privacy in a paper-based e-voting scheme. In *Proceedings of 5th Workshop on Advances in Secure Electronic Voting*, 2020.

PhD Colloquium

Verifiable Public Credentials for Stronger End-to-End Verifiability

Sevdenur Baloglu

University of Luxembourg
sevdenur.baloglu@uni.lu

1 Introduction

Electronic voting systems have many advantages to be used in real-world elections. However, most of them are vulnerable to the attacks due to the security weaknesses in their nature coming from the fact that in electronic environment any participant of the election can be corrupted by an attacker. Therefore, any electronic voting system should have a strong design and provide some security guarantees even in the presence of a strong attacker. One of the security properties required in any electronic voting protocol is verifiability. Helios is one of the electronic voting protocol commonly used in university and IACR elections, which is claimed to satisfy end-to-end verifiability. However, ballot stuffing and clash attacks are the known attacks against verifiability of Helios. In this proposal, a solution resisting to clash attacks, i.e. a way to improve Helios against clash attacks, is presented.

2 Issue

As in many electronic voting protocols, in Helios, there are several participants as *trustees* deciding the election keys, *administrator* deciding eligible voters and candidates of the election, *voting server* managing the bulletin board to display ballots coming from the voters, and *voters with their voting platforms* performing all necessary procedures to vote for a candidate during the election. In general, the list of identities of the voters are not published due to concerns about privacy. Instead, another public credential, e.g. alias, is distributed to voter by a separate trusted participant called *registrar*.

The clash attacks against verifiability of Helios, presented in [**?**], is based on the clash on the ballots when there is a number of voters who are known to vote for the same candidate. The same alias is given to these voters by a corrupted registrar, in case they check the bulletin board which will display all cast ballots together with their alias. Moreover, the same ballot will be generated for them using a corrupted voting platform, and only one ballot will be added to the bulletin board by a corrupted server for the voters having the same alias. This will reduce a number of votes into one vote for the outcome of the election. Furthermore, the corrupted server may add many ballots encoding a different candidate, using different aliases, by aiming to equalise the number of votes

and number of voters who did cast a ballot. This attack manipulates the result without being noticed by any voters who are verifying the bulletin board nor by any election auditors.

There are solutions proposed in [**?**] to prevent clash attack by aiming to prevent corrupted behaviour of the voting platform which uses the same randomness to encode the votes. For that, randomness should consist of two parts; coming from the voter and selected by the voting platform. In that case, two voters will never have the same ballot. However, when revoting is allowed, as long as voters have the same alias, even if they have different ballots, it is possible to mount a clash attack depending on the verification procedures. The other solution is based on the audits of the voting platform several times by each voter to catch if a randomness is used twice. However, this procedure is heavy to apply by each voter, i.e. not usable.

3 Research Proposal: Verifiable Public Credentials

We see that whenever there is a corrupted registrar giving the same alias to a number of voters, we will have a clash attack affecting the outcome of the election. To achieve a stronger end-to-end verifiability for Helios, we have to find ways to prevent this corrupted behaviour of registrar. We have to focus on how we can achieve that every voter in the election has her/his own unique alias.

A solution to prevent clash attacks in Helios can be possible with the use of verifiable public credentials, i.e. verifiable aliases, which ensure that the aliases of any two voters who successfully verify their ballots are distinct. The aliases can be made by verifiable *by voters* with an additional step in the protocol. These aliases can be constructed by a one-way function F, which will take the input as the unique, private information of the voter in order to derive the public credential of this voter. With the property of one-way function, no one will be able to reach from the public credential to the identity of the voter. This private information can be a unique output constructed by the biometric data taken by the voter before the election. Whenever the voters want to vote, they can give the same biometric data to construct the alias. This solution is more usable and efficient against clash attack than the solutions sketched above.

Another way is making public credentials *publicly verifiable*. For that, registrar and voting platform can together act to generate alias, and any third party, election auditors, can verify the uniqueness of the aliases on the bulletin board with the published information by registrar and voting platform. If publicly verifiable aliases are achieved, this will be more usable from the perspective of voters.

References

1. Küsters, R., Truderung, T., Vogt, A.: Clash attacks on the verifiability of e-voting systems. In: Proceedings of the 33rd IEEE Symposium on Security and Privacy, 21-23 May 2012, San Francisco, California, USA. pp. 395–409. IEEE Computer Society (2012). https://doi.org/10.1109/SP.2012.32

Pin-Based JCJ Voting Scheme

Ehsan Estaji, University of Luxembourg

1 Introduction

One of the main threats in remote electronic voting is that they are inherently susceptible to shoulder-surfing and other coercion-attacks. In their seminal paper, Juels, Catalano and Jakobsson [2] gave a formal definition of coercion-resistance and further devised a protocol (JCJ) satisfying this strong security property. To achieve this, JCJ assumes a coercion-free setup phase where the voter gets a credential which is essentially a cryptographic key. To cast a valid ballot this key needs to be entered correctly together with the vote. However, in case of coercion, the voter can simply give a fake random credential to the coercer and even cast a vote together with the coercer using this fake credential – the corresponding vote will be removed in the tally process.

JCJ and similar constructions however also suffer from usability deficits, see also [3]. Also, the voter intrinsically cannot directly check if a cast ballot is valid and will be counted, see however [1]. Moreover the handling and storing of long credentials is not usable in practice, especially with at coercer present. This led Neumann et. al. [4] to use smart cards for handling voter's credentials. The stored credential is combined with a PIN code to produce the full credential which will be compared with the credential stored by the authorities on the bulletin board. However, the use of a smart card is not desirable in several ways:

- The smart card is trusted for correctly producing the ballot, and we cannot let the voter check if the ballot is correct without introducing coercion threats.
- The coercer can take the smart card away from the voter to force abstention.
- It is more expensive, less flexible and harder to update than a pure software solution.
- The coercer can use the smart card and cast ballots on his own. This not only endangers to overrule the coerced voter's real vote, but due to a leak of information in the weeding phase, the coercer can also detect, with non-negligible probability, whether the coerced voter has cast an independent ballot against his instructions.

Depending on the level of coercion, the voter can either fake the key length credential or, for stronger levels of coercion, the voter can reveal the digitally stored credential to the coercer, but fake the PIN. This, of course, allows the coercer to try to brute force the PIN space to create a valid vote.

Another problem with original construction is the high chance of PIN typo errors, which are not corrected. Note that naively giving feedback on the correctness of the pin is not possible for coercion-resistance as it would allow the coercer to check whether he got a fake pin or not. Instead we check at tally time whether the PIN is in the set of allowed PIN errors, without revealing this publicly.

2 Main Idea

Firstly we divide the full credential of a voter to two parts, namely long key and a PIN number. The long key will be publicly available but PIN is just in the voter's mind. Our main tool for tolerating

voter's errors during the entering of PIN's is a polynomial which is in charge of verifying the entered PIN is in some error list (which depends on the policy of the election authorities). Of course there are plenty of concerns about how this polynomial is verifiably issued to voter and how to announce that a ballot is accepted or not.

3 PIN space Analysis

We consider the attacker's view in the sense that the best strategy to use this accessibility in favor of himself. Since there are a lot of frequency analysis of PINs we strongly recommend that the PIN should be generated uniformly random and not by voter's choice. Hence there is not any guessing strategy for attacker and he should cover the PIN space with minimum attempts.

Suppose we denote the PIN by $p_1p_2\cdots p_k$. We compute the number of different numbers covered by each PIN. Let's start with the case $k = 2$. By $[p_1p_2]$, we mean the set of numbers covered by this PIN. Clearly $[p_1p_2] = \{p_1p_2, p_2p_1, p_1*, *p_2\}$, where $* \in \{0, 1, 2, \ldots, 9\}$. After removing the repeated cases we'll have $|[p_1p_2]| = 20$ for the case $p_1 \neq p_2$. This will be 19 for the case $p_1 = p_2$. Since the attacker is looking for covering the PIN space with the minimum attempts, we assume that he uses the distinct digits. If he tries the r distinct 2-digits numbers p_1p_2,$p_3p_4, \ldots$ and $p_{2r-1}p_{2r}$. One can verify that the number of PINs covered with this r attempts will be as follows:

$$|[p_1p_2] \cup [p_3p_4] \cup \cdots [p_{2r-1}p_{2r}]| = 20r - 2\binom{r}{2}$$

And it shows the attacker with 8 attempts could cover the entire PIN space (of all 2-digits numbers).

For the $k \geq 3$ for any two PIN, $p_1p_2\ldots p_k$ and $p'_1p'_2\ldots p'_k$, $[p_1p_2\ldots p_k]\cap[p'_1p'_2\ldots p'_k] = \emptyset$ provided that $\{p_1, p_2, \ldots, p_k\} \cap \{p'_1, p'_2, \ldots, p'_k\} = \emptyset$. This yields that if an attacker use different r PINs, the total number of covered PINs will be $10kr$. Hence we are looking for a minimum number of r such that $r \geq \frac{10^k}{10k}$.

References

1. Vincenzo Iovino, Alfredo Rial, Peter B Rønne, and Peter YA Ryan. Using Selene to verify your vote in JCJ. In *International Conference on Financial Cryptography and Data Security*, pages 385–403. Springer, 2017.
2. Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *Towards Trustworthy Elections*, pages 37–63. Springer, 2010.
3. André Silva Neto, Matheus Leite, Roberto Araújo, Marcelle Pereira Mota, Nelson Cruz Sampaio Neto, and Jacques Traoré. Usability considerations for coercion-resistant election systems. In Marcelle Mota, Bianchi Serique Meiguins, Raquel O. Prates, and Heloisa Candello, editors, *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems, IHC 2018, Belém, Brazil, October 22-26, 2018*, pages 40:1–40:10. ACM, 2018.
4. Stephan Neumann, Christian Feier, Melanie Volkamer, and Reto Koenig. Towards a practical jcj/civitas implementation. *INFORMATIK 2013–Informatik angepasst an Mensch, Organisation und Umwelt*, 2013.

Implementing Internet voting: does the type of regime matter?

Ekaterina Fedko

Tallinn University of Technology

katerina.fedko@gmail.com

1 Introduction

In September 2019, a mechanism of Internet voting was introduced during local elections to the City Duma of Moscow [1]. The possibility to cast vote online was provided to half a million of voters, available in 3 out of 45 electoral districts. Internet-voting was introduced as an additional voting channel, supplementing paper-based voting and was available to the voters exclusively on the day of the election for 12 hours. Developed on blockchain, the system of Internet-voting was promoted as a secure voting environment and open for public observation. Considering the previous experience on voting technologies of Estonia and Switzerland, the authorities advertised Internet voting as a tool to promote democracy and inclusivity [1]. Nevertheless, despite the declared transparency and security of the developed solution, the integrity of the voting system and obtained results were challenged by several experts and election candidates.

Attempting to analyze the Moscow 2019 elections through the lens of the existing studies, the author encountered certain challenges associated with limited research on Internet voting implementation in a non-democratic environment. Whilst previous studies have been majorly considering Internet voting in the democratic context, reinforcing its democratic potential [2], there is a limited research on implication of eDemocracy instruments in a non-democratic environment. To expand analysis on Internet voting in developing democracies, the present work considers the case of 2019 Local Elections in Moscow and aims to answer the following research question: "How is Internet voting implemented in a non-democratic country?"

2 Methodology and framework

The study is built upon OSCE/ODiHR framework [3]. The framework considers the context of Internet-voting implementation, including such aspects as decision making process, legislative framework, electoral system, and involvement of stakeholders. It further elaborates on technology specific aspects, analyzing the technological solutions per se, procurements process, compliance with the key principles of the election process, the process of Internet voting implementation. The study will deploy the OSCE/ODiHR framework to identify the major challenges and enablers of Internet voting in the Moscow elections. Seen as a benchmark for eVoting, the framework offers twelve aspects which will be consequently analyzed to detect the peculiarities of the Moscow 2019 case.

The research is designed as a single explanatory case study, aiming to explain implementation process of Internet voting during the local Moscow 2019 elections. The study relies on the document analysis of the secondary data, such as transcripts of stakeholders' meetings, public interviews, public reports, related legal acts, and information materials of the authorities regarding Internet voting. Additionally, for greater objectivity two interviews were conducted with a representative of the election administration and one of the oppositional candidates.

3 Findings

The multifaceted consideration of the study case revealed 29 enablers and 28 challenges for the Internet voting implementation in the Moscow 2019 elections. Associated with either of the OSCE/ODiHR framework aspect, the enablers and challenges were further grouped into two major categories: general and regime specific. The research identified several challenges intrinsic to a non-democratic environment, associated with political pressure, lack of political openness, mistrust and comparatively low level of civic engagement. Notably, weak development of political institutions facilitated the implementation process of Internet voting, allowing the associated decisions and legislations to be adopted significantly faster than in a democratic context.

In overall, the research likewise revealed a considerable number of regime-independent enablers and challenges which can be present in a democratic environment. For the government willing to visually comply with transparency and democracy principles, the Moscow Internet voting case possesses characteristics intrinsic to implementation of eVoting in a non-authoritarian environment. However, as follows from the case analysis, introduction of Internet voting in Moscow 2019 elections suffers from a certain set of challenges, associated with authoritarian nature of the regime. Despite the regime-specific challenges and enablers constitute only a limited number of the discovered characteristics, they pose a certain threat toward ensuring democratic principles of the elections.

Whilst currently there is no extensive research on implementation of eVoting technologies in a non-democratic environment, there is only a limited possibility to evaluate potential implications of the derived challenges and enablers onto further cases. However, the found set of characteristics could be used to assess possible threat of implementation of eVoting in other authoritarian regimes.

References

1. ———. 2019. "Jelektronnye Vybory v Moskovskuju Gorodskuju Dumu." https://www.mos.ru/city/projects/blockchain-vybory/
2. Goodman, Nicole, and Leah C Stokes. 2016. "Reducing the Cost of Voting: An Empirical Evaluation of Internet Voting's Effect on Turnout." SSRN Electronic Journal. https://www.cambridge.org/core/journals/british-journal-of-politicalscience/article/reducing-the-cost-of-voting-an-evaluation-of-internet-votings-effect-onturnout/6FF8DA77C59806F0175656D66DE66907
3. OSCE/ODIHR. 2013. OSCE Office for Democratic Institutions and Human Rights (ODIHR) Handbook For the Observation of New Voting Technologies. www.osce.org/odihr

Essays on Internet voting implementation in legally binding elections

Iuliia Krivonosova[0000-0001-7246-1373]

DigiGovLab, Ragnar Nurkse Department, TalTech University
iuliia.krivonosova@taltech.ee

1 Introduction

This is an abstract of the dissertation which looks at four cases of Internet voting implementation in legally binding elections: Estonia in 2017 [3] and 2019, the Aland Islands, Finland in 2019, and Moscow, Russia in 2019. This dissertation tries to generalize based on four cases and frame them into a broader theoretical discussion.

2 Framework and methodology

The small population of countries, which implement Internet voting in legally binding elections, allows us to focus on individual cases. The selection of cases is not random: the Aland Islands [2] and Moscow have proclaimed emulating some Estonian policy decisions during the process of implementation. Therefore, the case of Estonia could serve as a benchmark against which the cases of the Aland Islands and Moscow are compared. Therefore, the second step of the analysis, after the individual case-studies have been conducted, is the comparison of the cases.

In all cases, Internet voting is analyzed from the perspective of public administration, answering the question what it takes to deliver elections with new voting technologies. The analysis covers the following elements:

- financial (How much does an e-vote cost? and Whether Internet voting brings cost reduction?);
- contextual (What drivers and barriers for Internet voting implementation are in place?)
- procedural (How is Internet voting integrated into the existing electoral infrastructure? How does it co-exist with alternative voting channels? and Who delivers Internet voting?).

Each research question requires a distinct set of methods for data collection and analysis. For covering the financial aspects of Internet voting implementation, the methodology based on Time-Driven Activity Based Costing was developed. Procedural aspects of Internet voting implementation were studied by the means of modelling (Business Process Management and Notation) and legal analysis. All four aspects of analysis relied on the data derived from stakeholder interviews, document analysis and on-site observation. All cases except Russia involved on-site visits.

3 Findings

For the analysis of the financial side of implementation, this dissertation proposes a new approach for cost assessment of Internet voting implementation in multichannel elections [1].

The contextual analysis reveals that all four cases have very different contextual factors, ranging from the political environment (democratic regimes vs electoral authoritarianism) to the size of the electorate (around 2 000 voters in the Aland Islands trial vs 500 000 voters in the Moscow trial) to the stage of Internet voting implementation (Estonia held the 9th and 10th elections with Internet voting available to all eligible voters at all levels of elections, while both the Aland Islands and Moscow performed their first binding trials). The dissertation identifies drivers and barriers for Internet voting implementation in both democratic and non-democratic environments.

From the procedural side, in all cases, the implementation process shows similarities, however, the particular policy decisions vary significantly. All considered countries applied different approaches to the Internet voting system development, e.g. Election-as-a-Service (the Aland Islands), proprietary software developed by a private company (Estonia), and in-house development based on public-public partnership but with involvement of private actors (Russia). Furthermore, the implementation process in all three counties led to different outcomes: in Estonia, the implementation was smooth, while in the Aland Islands, it resulted in cancellation of Internet voting shortly before the Election Day, and in Moscow – in a system failure during the Election Day, and the electoral results being challenged.

References

1. Krimmer, R., Duenas-Cid, D., & Krivonosova, I. (2020). New methodology for calculating cost-efficiency of different ways of voting: is internet voting cheaper?. Public Money & Management, 1-10.
2. Krimmer, R., Duenas-Cid, D., Krivonosova, I., Serrano, R. A., Freire, M., & Wrede, C. (2019). Nordic Pioneers: facing the first use of Internet Voting in the Åland Islands (Parliamentary Elections 2019).
3. Krimmer, R., Duenas-Cid, D., Krivonosova, I., Vinkel, P., & Koitmae, A. (2018, October). How much does an e-Vote cost? Cost comparison per vote in multichannel elections in Estonia. In International Joint Conference on Electronic Voting (pp. 117-131). Springer, Cham.

E-stonia: from e-government to e-democracy. Social transformation towards a digital society.

Oliwia Kuban[1][0000-0001-5753-2022]

[1] Adam Mickiewicz University in Poznań, ul. Wieniawskiego 1, 61-712 Poznań, Poland
oliwia.kuban@amu.edu.pl

Keywords: e-government, e-democracy, e-society, transformation.

Dynamic technological development has brought rapid changes in data processing and information transfer in recent decades. Nowadays, technological progress plays an important function, which is manifested in the fact that it covers virtually every area of human life, bringing significant changes in social life. New information and communication technologies (ICT) are also increasingly used by politicians, political institutions and non-governmental organizations [1]. It is widely believed that the opportunities created by implementing modern technologies into political life have the potential to reduce the problems of modern democracies. It is worth noting that modern countries are primarily connected digitally rather than physically.

The implementation of ICT in democratic mechanisms has contributed to the creation of new concepts, such as electronic democracy or eGovernment [2]. E-democracy in relation to citizens refers primarily to the nature of government and decision-making in the state and the role of the individual in these processes. In e-democracy, there are problems that intervene with the e-government area. Those issues are based on customers experiences, thanks to that they become a part of development in e-administration area as well. The scope and nature of services provided as part of e-government is changing along with the progress taking place, both in the area of ICT and social technologies. A well-developed and effective e administration is often seen as the beginning of the creation of an electronic state in which both administration and democratic processes will be transferred to the electronic plane. It seems that the use of the Internet in political processes is a natural consequence of its dissemination. A pioneer in terms of implementing e-state-specific solutions is Estonia - one of the most advanced e-societies in the world. Success stories can be seen primarily in the advanced cooperation of the private and public sectors [1].

In the dissertation, the author will primarily analyze the experience of the Estonian state in the use of new technologies in political processes at various levels of the administrative division of the state. The goal will be the attempt of defining the role of modern technologies in shaping a new model of society and the fact of enablement.

Scientific studies detail the genesis and functioning of e-democracy and e-administration in the modern world. The proposed dissertation is to be a certain novelty in terms of the impact of technological development on the shape of the state, and above all on the preservation and development of the digital society. Available studies on the domestic and international arena focus rather on providing general definitions and assumptions regarding individual areas of electronic democracy, without taking into account the impact of ICT implementation on the development and involvement of society.

Due to the breadth of the research concept, a variety of research methods and techniques will be required. The effect of the work will be to show and evaluate the impact

of the use of modern technologies in administrative and democratic procedures on the society model.The main research method that will be used in the work is the analysis system method. To explore the relationship between electronic administration, electronic democracy and the shape of society, it should be described in the context of the functioning of the entire system. This method will be useful in the whole process of the dissertation, because the purpose of the work will be to examine the mutual relations between individual elements of the system. In the context of the proposed topic, it will valuable to examine the subject – legal regulations regarding the possibility of implementing modern technologies in administration and democratic procedures, as well as their functioning. It will also be important to use the technique of analyzing the content of sources, which will be used to examine publications and studies prepared to promote and spread Estonian experience in creating an e-state. The proposed dissertation plan assumes empirical research, therefore it will be necessary to use the quantitative and qualitative research method. Statistical data illustrating the usefulness and attractiveness of eGovernment and democratic processes in which new technologies are used will be examined [3]. The data obtained in the course of empirical research will be subject to analysis, thanks to which it will be possible to prepare conclusions regarding the impact of ICT implementation in administrative and democratic processes on the shape of public opinion, as well as the overall model of society in the state. The quantitative analysis will be complemented by qualitative research based on materials from political institutions in Estonia and public administration websites. This approach will help in achieving the assumed goal and it will be possible to assess the impact of modern technologies on the shape of the community. An analysis of Estonians' experience in the context of implementing ICT in public spaces can have a prognostic function and serve interested parties as a handbook of action or a warning against bad practices.

References

1. Solvak M., Vassil K., 2016, E-voting in Estonia: Technological Diffusion and Other Developments Over Ten Years (2005 - 2015), Tallinn.
2. Krimmer R., 2012, The Evolution of E-voting: Why Voting Technology is Used and How it Affects Democracy, Tallinn.
3. Statistics Estonia, https://www.stat.ee/en, dostęp 26.04.2019.

Development of a Model for a Secured Bimodal Voting Framework Using Timed Coloured Petri Nets

OLAOLUWA Adeoye Olayinka
Ladoke Akintola University of Technology, Nigeria

Elections seem simple as ordinary counting but practically complicated with a challenging combination of security and privacy requirements. Electorates demand convincing assurance their votes are counted, result is correct and privacy assured. Independent National Electoral Commission (INEC), the body responsible for elections in Nigeria has reformed the voting system used in Nigeria severally from the first election in 1959 to the 2019 election [1]. Unfortunately elections in Nigeria have been characteristically marred with massive rigging, ballot snatching, and increasing violence [2]. There is resultant voter apathy in Nigeria as evident during 2019 national election with highly populated voters register (84 million registered voters) but deserted voting booths having only 29 million voters exercising their suffrage. Without trust or confidence that elections will produce fair outcomes, voters may choose to stay home, thereby compromising the legitimacy of the government [4]. This research therefore seeks to develop an executable model of a framework using Coloured Petri Net formalism. Coloured Petri Nets (CPN)[5] are one type of high-level nets consisting of places, transitions, and arcs with state and action orientation. The CPN modelling language is a general-purpose modelling language with application domains including communication protocols, data networks, distributed algorithms, and embedded systems. CPN models are executable models that can be structured into a set of modules which interact with each other through a set of well-defined interfaces. Formal analysis of the model is done through simulations to investigate different scenarios using CPN Tools [6] which is a general-purpose verification tool for modelling and analyzing CPN.

This framework is a communication protocol for different forms of data from the voter as source during online registration through verification, accreditation, voting, tallying and display of result on the board to storage by INEC as sink. The plan is structured hierarchically to accommodate the complexity brought in by added security features to meet security requirements and win voters trust. The proposed framework is composed of five main modules. These modules are based on the legal structure dictated by the 1999 Constitution of the Federal Republic of and presented by INEC on 12th of January 2019 [7]. Functionalities are included in the framework to specifically address the short comings (accessibility, verifiability, voter privacy and trustworthiness, staff security) observed during the 2019 National election in Nigeria. The main components are conceptualized on paper and electronic processes and records. These include the online registration component which accepts voter's bio data as input; encrypt it to be forwarded to central server for validation, update or storage. It will produce as output both paper-based and electronic voter register for INEC, and Permanent Voter Cards; Voter Record Update Unit which is incorporated on the on-

line platform to allow update of voter information in order to allow migration from polling unit or to delete records of dead voters; Enhanced Distributed Accreditation (EDA) component which is the voter authentication unit that produces validated voters for the next stage of election; the voting module which accepts validated voters and supplies the ballot. This was further divided to choice of desired mode of voting and capturing of the voter's intention in ballot box as marked ballots. The last module is the Secured Result Collation (SRC) Component where voters' intentions captured as marked ballots are extracted publicly within the polling unit while stakeholders observe as demanded by the electoral law. This unit oversees conversion of paper records of vote tally to electronic form, its encryption and its transmission to collation levels. Tallying will be carried out using Paillier Cryptosystem, which gives the ability to sum up votes even though they have been encrypted. The result collation component will deliver paper and electronically transmitted result for display on E-board.

Summarily, the objectives of the research are to:

1.develop a secured voting framework with improved accessibility, verifiability, voter privacy and result integrity for Nigerian elections.

2. model the developed framework using Timed Coloured Petri Nets; and verify the CPNs model using CPN Tools in order to assess its overall security and performance.

Upon successful completion, our main deliverable will be a Timed Coloured Petri Nets (TCPN) model of a secure and verifiable voting framework that should provide a viable solution which can be studied as a referential model to promote future modifications and reinvigorate public participation in democratic life by making voting more accessible and secured in emerging democracies.

References

1. European Union Election Observation Mission Nigeria General elections 2019: first preliminary statement, 25 February (2019)
2. Sakue-Collins Yimovie.: Rethinking Electoral Democracy: A Critical Analysis of Nigeria's 2015 General Election. Global Journal of HUMAN-SOCIAL SCIENCE 17(4) (2017)
3. Almustapha A. J., Olaniyi O. M., Abdullahi I. M. and Abdulsalam,Y. S (2018): "Towards the Use of Bpann Technique For Mitigating Layer 4 DDOS Attack In Electronic Voting" AICTTRA 2018 Proceeding. Pg 41-48
4. Duruji, Moses; Ayo, Charles; Oni, Samuel; Oni, Aderonke (2015): "Making a Case for e-Voting in Nigeria, European Conference on e-Government" 100-106, Kidmore End: Academic Conferences International Limited
5. Christensen. S and Mortensen. K. H. (2017): "Teaching Coloured Petri Nets- A Gentle Introduction to Formal Methods in a Distributed Systems Course". In ICATPN, pages 290–309, 1997.
6. The Department of Computer Science, University of Aarhus, Denmark. CPN Tools: Computer Tool for Coloured Petri Nets. [Online]. Available: http://wiki.daimi.au.dk/cpntools/cpntools.wiki, 2004.
7. INEC (2019): "Regulations and Guidelines for the Conduct of Elections": Independent National Electoral Commission, 2019. Abuja.

Verifying the Security of Electronic Voting Protocols

Morten Rotvold Solberg

Norwegian University of Science and Technology, NTNU, Norway

Abstract. We briefly describe the proof assistant EasyCrypt and how we aim to use EasyCrypt to verify different security properties for different electronic voting protocols and important sub-protocols.

Keywords: EasyCrypt · Formal Verification · E-Voting

1 Introduction

Traditionally, cryptographic protocols have been analyzed by designing a protocol and then trying to break it. In the later years, however, it has become tradition to provide a security proof along with the proposed cryptographic protocol. A typical approach in such proofs is to use reductionist arguments based on the hardness of different well-studied mathematical problems, such as factoring integers and computing discrete logarithms.

Even though certain techniques such as game hopping [6] have been developed to ease the notorious complexity of security proofs, such proofs are still error prone. An encryption system (or voting system) might also be perfectly secure on paper, but become insecure due to implementation errors (such as, for example, the Scytl/SwissPost system [4]).

To address these issues, different computer-based proof assistants have been developed. Some are designed to assist at the design level (e.g. to verify security proofs) and some are designed to assist at the implementation level (e.g. to raise assurance that implementations behave according to their specifications). One proof assistant designed to raise assurance at the design level, by verifying cryptographic security proofs, is EasyCrypt.

2 EasyCrypt

EasyCrypt [1] is a framework designed for the verification of game-based security proofs in the computational model. EasyCrypt uses an underlying logic known as *Hoare logic* [5]. The core component of Hoare logic is the *Hoare triple* $P\{Q\}R$, where P is a *precondition*, R is a *postcondition* and Q is some algorithm or program. The Hoare triple is to be understood as "if P is true, then R will be true after executing the program Q". This has been extended further to *relational Hoare Logic*, which is designed to reason about relations between probabilistic programs, or games.

EasyCrypt uses a set of built-in *proof tactics* as well as external software (SMT solvers) to transform the goal, or the theorem that you want to prove, into simpler claims. A proof is a sequence of such transformations that eventually results in claims that follow directly from axioms or assumptions. Proofs can be checked step by step, by EasyCrypt's *trusted computing base* (which includes the SMT solvers used).

EasyCrypt has previously been used to verify the security of both voting protocols and protocols not related to voting. Two notable examples where EasyCrypt is used for voting protocols, are the verification of the ballot privacy property for Helios [2] and the ballot privacy and verifiability properties for Belenios [3]. Both of these systems were proven secure under the assumption that the adversary is unable to tamper with the bulletin board.

3 The Aim of Our Project

There are several important security notions for voting protocols that are not modeled in EasyCrypt. In addition, some of the properties that are modeled seem to be suitable only for the protocols in question. Thus, we aim to develop a "tool box" in EasyCrypt for security properties related to voting. Some examples of what we aim to include in such a tool box are ballot privacy (where we also allow for a malicious bulletin board), integrity and coercion resistance. We also aim to model security notions for widely used sub-protocols, such as re-encryption mixnets and different zero knowledge constructions.

References

1. Barthe, G., Dupressoir, F., Grégoire, B., Kunz, C., Schmidt, B., Strub, P.Y.: EasyCrypt: A Tutorial, pp. 146–166. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-10082-1_6, https://doi.org/10.1007/978-3-319-10082-1_6
2. Cortier, V., Dragan, C.C., Dupressoir, F., Warinschi, B.: Machine-checked proofs for electronic voting: Privacy and verifiability for belenios. In: 2018 IEEE 31st Computer Security Foundations Symposium (CSF). pp. 298–312 (2018)
3. Cortier, V., Dragan, C.C., Strub, P.Y., Dupressoir, F., Warinschi, B.: Machine-checked proofs for electronic voting: privacy and verifiability for belenios. In: Proceedings of the 31st IEEE Computer Security Foundations Symposium (CSF'18). pp. 298–312 (2018). https://doi.org/10.1109/CSF.2018.00029
4. Haines, T., Lewis, S., Pereira, O., Teague, V.: How not to prove your election outcome. In: 2020 IEEE Symposium on Security and Privacy (SP). pp. 1042–1058. IEEE Computer Society, Los Alamitos, CA, USA (may 2020). https://doi.org/10.1109/SP40000.2020.00048, https://doi.ieeecomputersociety.org/10.1109/SP40000.2020.00048
5. Hoare, C.A.R.: An axiomatic basis for computer programming. Communications of the ACM **12**(10), 576–580 (1969)
6. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332 (2004), https://eprint.iacr.org/2004/332

Posters and Demo Session

Vocdoni - Making governance sovereign

Roger Baig Viñas[1][0000−0001−9813−7678] and Pau Escrich Garcia[2]

[1] Universitat Politècnica de Catalunya
rbaig@ac.upc.edu
https://dsg.ac.upc.edu/
[2] DVote Labs OÜ
pau@vocdoni.io
https://vocdoni.io

Abstract. The Vocdoni project is building a toolkit (software and infrastructure) to empower organisations and civil society in governance and decision making processes, in a self-sovereign, censorship-resistant and decentralised way.

Keywords: e-Governance·Sovereign Governance·e-Voting

1 Introduction

Based on Distributed Ledger Technology (DLT), Vocdoni[3] combines third-party open source tools and new developments to deliver features such as cryptographic sovereign identity and claims management, anonymous and secure voting, automation of governance models, secure Communication channels and cryptocurrency wallets to realise a full stack (software and hardware) for sovereign governance.

Vocdoni aims to be 1) *user-centric* by being mobile-first, 2) *privacy-centric* by adopting a Self-Sovereign Identity (SSI) approach and leveraging on the latest cryptographic technologies (e.g. zk-SNARKS), 3) *universally verifiable* through the publication of the datasets, the source code and the documentation, 4) *censorship-resistant* building on a decentralised architecture, and 5) *self-sustainable* through the development of affordable solutions, meaningful business models (collaborative economy, tokenomics, etc.), and opening new markets.

2 e-Voting, and Minimal Viable Product (MVP)

After establishing the principles of the project (summarised in Section 1) and an intensive research on the available solutions, either as components or integrated toolkits, the bulk of the efforts focused on developing the e-Voting primitive and achieving an MVP. e-Voting is a cornerstone of e-Governance. However, none of the other existing solutions fulfils the project's requirements in terms of privacy, auditablility, affordability and resilience. MVPs are crucial to have access to financing and to showcase the project's value proposition. At the time of this writing, significant

[3] https://vocdoni.io/

progress has been made in e-Voting and a MVP is being evaluated through an early access programme involving around 20 organisations with diversified profiles (unions, NGOs, political parties, coops, etc.)

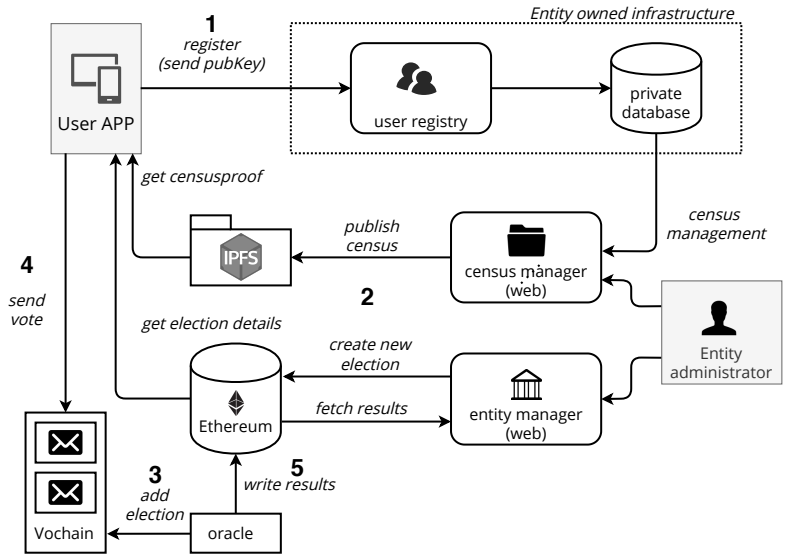


Fig. 1. Vocdoni's e-Voting architecture.

Figure 1 shows the e-Voting primitive overall architecture. First, after downloading the Vocdoni's App (available for Android and iOS in the official app stores) and creating their identities (multiple identity is supported), the users send a registration request with the public key they want to use to the organisation they want to register. The organisations, also referred as entities, build their users' registries by accepting these requests. These registries are private data bases that must be managed confidentially. Secondly, the entity administrator, on the one hand, creates the voting census by applying the corresponding filters to the user database and publishes it as a Merkle tree in a public repository (IPFS), and on the other hand, sets the election specifications (several types of voting schemes are supported and more will be integrated soon) and publishes them in a general purpose decentralised open source blockchain (Ethereum). Thirdly, a set of oracles monitoring Ethereum fetch the election details (the census Merkle root, duration, type, etc.) and publishes them to a specialised blockchain (Vochain). In a fourth step, eligible voters cast their vote as a transaction in the Vochain. Once the election finishes, a set of cryptographic data needed for te scrutiny is automatically revealed and the oracles compute the result and upload it back to Ethereum. Any third Party can also fetch the data and compute the same results.

In the demo we will elaborate on the principles of Vocdoni and the solution given to the e-Voting challenges and will show live how to create an election, vote and reach an outcome.

Polys Blockchain-Based Voting System

Aleksandr Korunov[1], Aleksandr Sazonov[1] and Petr Murzin[1]

1 AO Kasperski Lab

Abstract. Voting is an integral part of a democratic society. Today's modern voting systems have evolved from the counting of raised hands to a wide range of complex electronic systems. The current version of paper voting is a tested model, but it is far from ideal: the process is organizationally complex and expensive, vulnerable to manipulation, and, in many respects, depends on the human factor. There are various options for applying technologies to optimize the process, from digital counting machines to online voting solutions. While online voting solutions are very attractive, they introduce new problems to be solved, in particular, that of ensuring trust. The blockchain based system Polys simplifies the entire voting process, reduces organizational costs, increases voter turnout, and ensures the legitimacy of election results. It also has functionality that allows independent observers to monitor online voting and confirm it was conducted securely and correctly. Here we take a brief look at how Polys works, its features, and its advantages.

Keywords: Blockchain, e-Voting, Online Election, Transparency, Polys

1 Applications and process

Polys includes a remote online voting system and digital polling station, so it can be used in uncontrolled and controlled environments. The whole ecosystem works in a single blockchain network, so the two environments can be used together or separately in elections at all levels. To vote online, the voter can use a personal digital device, such as a smartphone or tablet; for voting at a polling station, we provide special voting machines that have the same simple interface as the online version.

1.1 Voting algorithm overview

A. Creating a vote
This is the stage where a ballot is created, the voting options are entered and the voting access criteria is defined. During this stage voting, organizers generate keys for the signing and encryption of ballots, and public keys with voting options are published to the blockchain. The most important element at this stage is the selection of so-called trusted representatives who may be members of the participating parties or other authorized persons. They validate the blockchain blocks and sign them using their personal keys.

B. The voting process
At the second stage, individual votes are received from voters and recorded in the blockchain. During this process, the system fulfills the following tasks:
* Ensures voter anonymity
* Provides protection against ballot stuffing
* Enables voters to check that their votes have been recorded in the blockchain.

C. Voting results
After voting has ended, the organizers publish decryption keys for the blockchain. Smart contracts are responsible for the tallying process, meaning anyone can check its legitimacy.

2 Reliability and immutability of voting results

During the vote creation stage, a pair of keys is formed – one public for encryption and the other secret for decryption. The secret key remains with the vote organizer (or is divided into parts between several organizers); the public key is published on the blockchain. To cast their votes, voters request an encryption key, create encrypted ballots and send them directly to the blockchain anonymously – voter anonymity is guaranteed by the blind signature scheme. After voting is complete, the decryption key is published on the blockchain, and each participant in the process has the opportunity to verify the correctness of entire decryption process. At the same time, the voters have data on their own transaction with the ballot, and they can make sure their vote that was correctly counted by the system.
The blockchain technology works in such a way that once data is recorded, it cannot be changed without the consent of the majority of network participants (more than twothirds) who own nodes to validate blocks. And even if such a change is made, it will not go unnoticed – all transactions receive a cryptographic signature of the nodes that validate blocks, which allows us to reliably prove that such a transaction was processed by the blockchain.

3 Polys use cases

The Polys system is actively used by educational institutions and student organizations around the world. During the COVID-19 pandemic quarantine period alone the platform was used for voting by the European Youth Parliament in Germany, the International Institute of Debate in Tunisia, Aktief Slip, a study association at Wageningen University & Research (Netherlands), the Rokiskis Juozas Tumas-Vaizgantas gymnasium (Lithuania) and several other institutions. Polys has also been successfully used in participatory budgeting, as well as for votes in non-profit organizations and private companies.

4 References

1. Polys. Online voting system, Whitepaper. [online] Available: https://polys.blob.core.windows.net/site/Polys_whitepaper.pdf
2. B. G. a. J. T. Jordi Barrat i Esteve, «International Experience with E-voting,» June 2012. [online]. Available: https://www.parliament.uk/documents/speaker/digital-democracy/IFESIVreport.pdf.
3. «Voting Equipment — Voluntary Voting System Guidelines,» 2019. [online]. Available: https://www.eac.gov/voting-equipment/voluntary-voting-system-guidelines.