TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Taaniel Kraavi 192926IVSB

Improving Ballot Privacy in the Estonian Internet Voting System

Bachelor's thesis

Supervisor: Ahto Buldas
PhD

Tallinn 2022

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Taaniel Kraavi 192926IVSB

E-häälte privaatsuse suurendamine Eesti e-hääletamise raames

Bakalaureusetöö

Juhendaja: Ahto Buldas
PhD

Tallinn 2022

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Taaniel Kraavi

16.05.2022

Abstract

Estonia has used internet voting (i-voting) for legally binding elections since 2005. In almost 20 years, the i-voting system has not suffered a breach or malfunction leading to the annulment of i-voting results. The current i-voting framework, called IVXV, was first used in 2017, and aimed to improve the security and verifiability of the previous scheme.

This work aims to address voter privacy concerns that existed in the previous scheme and continue to exist in IVXV. When casting their vote, a voter gives their digital signature to their encrypted ballot, a link which remains until ballots are anonymised for counting. The thesis illustrates how voter identities can be decoupled from ballots. In the first part of this work, the author introduces how IVXV works on a technical and organisational level. Then, the author proposes an alternative scheme that keeps the logic and flow of IVXV but enables the decoupling. In the proposed scheme, voter signatures are replaced and linked to pseudonyms instead of voter identities. The author describes two new services: a trust service independent from the i-voting infrastructure validates the process of signature replacement and an internal service manages the creation and publication of ballot revocation certificates, which are used to annul ballots superseded by re-voting or double-voting. The result is a system where ballot integrity remains auditable, but ballots are unlikable to voter identities without compromising the system on multiple levels or extensive collusion.

The thesis is written in English and is 46 pages long, including 6 chapters, 6 figures and 0 tables.

List of Abbreviations and Terms

BDOC	BDOC Digital Signature Format
BRS	Ballot Revocation Service
CA	Certificate Authority
CRL	Certificate Revocation List
E2E	End-to-End
EHS	Electronic Voting System (elektroonilise hääletamise süsteem)
eID	Electronic Identity
eIDAS	Electronic Identification, Authentication and Trust Services
NEC	National Electoral Committee
OCSP	Online Certificate Status Protocol
PKI	Public Key Infrastructure
PRE	Proxy Re-Encryption
QES	Qualified Electronic Signature
RVS	Re-Signing Verification Service
SK	SK ID Solutions AS
VIS3	Election Information System (valimiste infosüsteem)

Table of Contents

List of Figures

1 Introduction

Internet voting, also called i-voting, has been used for elections in Estonia since 2005, and the underlying scheme is continuously evolving in response to security and privacy concerns. While only 1.9% of voters voted over the internet at the 2005 municipal elections, the percentage of i-voters was 46.7% for the municipal elections of 2021 [1]. I-voting has also been used for local parliament elections since 2007, and for European Parliament elections since 2009. In English, the term ”i-voting” is used instead ”e-voting” to avoid confusion with other digital voting technologies, such as electronic voting machines or blockchain voting.

Electronic ballots follow the double envelope voting scheme, and as such, voters can be identified until the external envelope is discarded. For electronic ballots, a voter's encrypted choice represents the inner envelope, and their digital signature on the encrypted ballot represents the outer envelope. Ballot secrecy is guaranteed by the encryption and votes are anonymised before being decrypted for tallying. However, even though the personalised choices remain private, the author considers it a privacy concern that auditors and multiple parties involved in the i-voting process can identify voters.

The author believes that the security and integrity of the voting process can be maintained while increasing the degree of privacy for voters and ballots. The purpose of this work is therefore to determine whether the current Estonian internet voting system can be improved by restricting the number of parties that can identify and observe the connection between a voter and their ballot using the voter's electronic identity. While numerous privacy and security concerns exist regarding the i-voting system, many of which are being discussed or worked on, little is being mentioned regarding this identifiability concern, which is another reason behind the author's choice of topic.

1.1 Methodology and Data Sources

The author takes a qualitative and observational approach [2] by first performing descriptive analysis of the current system to understand its functioning and the design decisions at the root of privacy concerns. Understanding the latter is necessary because an apparent flaw may be the result of a trade-off that avoids deeper security, privacy, or practicality concerns. Moreover, the technical framework is subject to organisational and legal re-

quirements, both of which may impose restrictions on the system's design. The author's resulting hypothesis is that storing voter signatures alongside their ballots is not necessary to maintain the security of the i-voting system. The author thus seeks to propose a modified system of equivalent security but with enhanced privacy.

The results stem from an exploratory study that seeks alternatives to design decisions in the current system that the author does not agree with. This two-tiered approach helps the author make as little changes to the existing system as possible. It is also critical for the author to avoid introducing problems that do not already exist with the current system. Solutions can hence benefit from assumptions and existing research regarding the security of the system, while additionally being easier to implement. A disadvantage of this approach is that some advances in cryptology, electronic voting protocols, and computing cannot be used by proposed solutions. This trade-off is necessary to restrict the scope of this work, with a wider study best left for a master's thesis or separate research. A comprehensive study of security and privacy concerns of the system is not the author's goal, however the lack of discussion of certain shortcomings may be considered another limitation of this thesis.

The analysis is based on the publicly available general and technical documentation regarding the current i-voting framework. For questions about design decisions for which reasons aren't apparent or publicly available, the author reached out to parties involved with the framework's design and i-voting in Estonia, such as Sven Heiberg[1], Priit Parmakson[2], and Arne Koitmäe[3]. The author also considered third party research in assessing flaws withing the current system. To come up with solutions, the author took inspiration from general literature regarding electronic voting schemes and public key infrastructure (PKI).

1.2 Contributions

In this work, the author proposes a modification to the current Estonian i-voting system which keeps the general workflow of the current scheme, but provides additional privacy guarantees to voters. The modifications are designed in a way to keep the benefits of existing research regarding the current scheme, with any flaws in the proposed solution also existing in the current one. The proposed modifications remove the need for keeping voter-issued digital signatures on ballots until the counting of votes, thus removing the

[1] Co-author of the IVXV framework and product manager at Smartmatic-Cybernetica Centre of Excellence for Internet Voting
[2] Senior architect of VIS3 at the Estonian Information System Authority
[3] Head of service of the State Electoral Office

possibility for an attacker to collect identifiable i-ballots, albeit encrypted, by breaching the i-ballot box. As part of the modification, the author theorises the concept of a ballot revocation service, which functions in similar ways to certificate revocation lists (CRL) in PKI. The author also proposes the creation of a service that can be used for translating digital signatures from one party to another. While this is required by the modified i-voting scheme, the service may have more general use cases as well.

The resulting system is novel, at least in the context of the Estonian i-voting framework, and enables additional functionality such as undoing an i-vote, which is not specified in the current framework. While the author does not resolve the problem of the election organiser being able to completely breach ballot secrecy, the proposed system offers an additional degree of resiliency against such attacks by external parties when compared with the current system.

1.3 Organisation

The subsequent chapters are organised as follows. The current i-voting scheme is explained in Chapter 2, doubling as the background and literature review. The results and main contribution of the author are described in Chapter 3. Chapter 4 contains a discussion of the results, and Chapter 5 proposes topics for future work. Finally, Chapter 6 concludes the paper.

Schematics that disturb the flow of reading due to their size can be found in the annexes. References to the appropriate annex are then provided in text.

2 Estonian I-Voting System

In this chapter, the author provides an overview of the current Estonian i-voting framework and implementation in use. The author begins with a short overview of what enables i-voting in Estonia in Section 2.1, followed by a description of the i-voting process and main parties involved (Section 2.2). In the latter, some technical details are left out for the sake of brevity. Finally, Section 2.3 gives an evaluation of previous research on the topic.

2.1 Background

Part of what enables i-voting in Estonia is the existence of state-issued electronic identity (eID) and secure carriers for it, such as the ID-card, Mobile-ID, and Digi-ID [3]. eID relies on public key infrastructure[1] to operate, where cryptographic keys are used to perform actions, and certificates link keys to identities. In Estonia, the trusted party issuing those certificates, the Certificate Authority (CA), is SK ID Solutions AS (SK) [4], [5]. Being able to digitally identify oneself and digitally sign documents by means of eID is considered vital in Estonia and ensuring the continuity of both services is regulated in law [6].

The current Estonian i-voting system, named ”IVXV” [7], was first used in the 2017 local municipal elections and aimed to improve and answer concerns about the verifiability of the previous voting scheme [8], [9]. It significantly improved upon the preceding framework by reinforcing the digital ballot box integrity and improving the auditability of the correctness of vote decryption by third parties [9]. If i-voting is used, IVXV must be used, as stipulated by law [10, II (24)].

2.2 Main Processes

The organiser of an election (hereinafter *Organiser*) is in charge of appointing parties involved with the i-voting system. I-voting itself can be divided into the pre-voting stage, voting stage, processing stage and counting stage [7, p. 8]. During the pre-voting stage, lists of candidates and eligible voters are generated by the *Organiser* and made available to necessary parties of the i-voting system.

[1] https://www.ria.ee/en/state-information-system/electronic-identity-eid.html

2.2.1 Election Cryptosystem and Key Management

For each election, the *Organiser* uses the *Key Application* to generate election specific key-pairs. The *Key Application* is also used for tallying the votes and computing the results as part of the last stage of the i-voting process [7, p. 10]. Two keypairs are generated: one for the encryption and decryption of votes, and the other for signing the i-voting results. Both private keys are split into key-shares, each of which is loaded on a physical chip card [11, § 3.2]. The chip cards are sealed and distributed among members of the National Electoral Committee (NEC) and of the State Electoral Office. Because of the key-splitting requirement, a multi-party public key cryptosystem and signature scheme must be used. For IVXV, the Shoup RSA threshold signature scheme and the ElGamal cryptosystem with the Desmedt threshold scheme are used [12]. Once the election results are announced, the private key is destroyed [13, (6.5)], a concept known as cryptographic erasure [14, § 2.6] or "crypto-shredding" [15, § 3.4].

The ElGamal cryptosystem is also non-deterministic and partially homomorphic [16], both of which are desirable properties for IVXV. Because it is non-deterministic, encrypted votes for the same party do not result in equivalent cryptograms. Because it is homomorphic for multiplications, a Schnorr zero-knowledge proof, the *tally-proof*, can be created upon decryption. As such, the *Key Application* can prove to auditors that decryption and tallying were performed correctly, even after the decryption key is destroyed [17].

2.2.2 Voting Stage

Before a voter is allowed to digitally vote using the *Voter Application*, they must identify themselves to the vote collector (hereinafter *Collector*) for a preliminary check of their voting rights. This identification is carried out with the voter authenticating themselves via ID-card and Mobile-ID solutions [13, (7.2)]. If the voter is eligible, the *Collector* returns to the *Voter Application* the list of candidates available to the voter based on their electoral district, and whether or not the voter has already voted. A voter is allowed to re-vote, but only their latest vote is counted.

Once the voter makes their choice, the *Voter Application* encrypts the choice using the election's public key and a random number, the latter is required to ensure the non-deterministic outcome. Then, the encrypted ballot is signed either by means of ID-card, Digi-ID or Mobile-ID [18, § 4.3]. However, no certificate validation or time-stamping is performed during the signing process. Instead, the *Collection Service* of the *Collector* is tasked with qualifying the signature later on, both for additional auditability and due to the untrusted nature of the voter's device [19]. For this, the *Voter Application* sends the signed and encrypted ballot to the *Collection Service*, where the ballot is qualified

and registered according to requirements set forth by the *Organiser*. Once this process is complete, the voter is given the possibility to verify that their vote correctly reached the i-ballot box.

To ascertain the validity of ballots, excepting the validity of the choice contained therein, the *Collection Service* must, at minimum, check that

1. the signer of the vote is included in the eligible voters' list,
2. the signed vote respects the expected format (BDOC container),
3. the digital signature of the ballot is correct,
4. the signer's digital certificate was valid at the time of receiving the ballot [18, § 6.1].

The *Collection Service* can perform all checks except for the fourth. To verify the validity, the application uses the Online Certificate Status Protocol (OCSP) to request validation from the CA or a *Validity Service*, who responds with the validity status.

The IVXV framework also requires votes to be registered by a party independent of the Electronic Voting System (elektroonilise hääletamise süsteem—EHS). The party serves as a witness to the existence of votes so that the *Collection Service* cannot unnoticeably drop votes from the i-ballot box. This party, the *Registration Service*, also provides the registration timestamp to the *Collection Service* [18, ch. 5]. The current implementation in Estonia foresees that the vote registration and certificate validation are both handled at once by the CA, which is SK [20]. Given the timestamp and OCSP certificate status, the *Collection Service* can fully qualify the vote. Finally, the *Collection Service* returns to the *Voter Application* the elements qualifying the ballot along with a unique ballot identifier, and then stores the ballot with its qualifying elements in the i-ballot box [18, § 6.1].

2.2.3 Verification of Their Vote by the Voter

After casting their vote, the voter is given the option to verify whether their vote was registered and stored properly. Whether they chose to do so or not, the *Voter Application* verifies the vote qualifying elements, and so checks if the *Collection Service* properly registered the ballot and validated the voter's certificate. The *Voter Application* displays to the voter whether the checks were successful or not [18, § 6.2]. However, the voter cannot be sure that the *Voter Application* operates as intended and hasn't been compromised, by malware for example [21]. As such, the voter is given the option for additional verification using a smart device, separate from the device used to cast the vote [7, p. 15]. It is assumed that both devices are not compromised simultaneously.

After the *Voter Application* confirms the presumably correct casting of the vote, it displays

a QR code containing the ballot identifier provided by the *Collection Service* and the random number used for encryption. The voter can use the *Verification Application* on their smart device, which must have a working camera and network connection, to scan the QR code. The application makes a request to the *Collection Service* with the ballot identifier, and, if the voter is allowed to verify the vote, the service responds with the ballot from the ballot box and vote qualifying elements. The *Verification Application* performs necessary verifications and then uses the random number from the QR code to decipher the vote [18, § 6.3], which is possible for the ElGamal cryptosystem [22]. The application also verifies that the syntax of the decrypted vote is correct and displays to the voter their personal information and their plaintext vote if all checks pass [18, § 6.3].

Typically, the voter is only allowed to verify their vote within the hour following the casting of the vote. A restriction may also be set by the *Organiser* for the number of times the vote can be checked [7, p. 16]. These restrictions are of organisational nature, not technical, and are in part designed to protect a voter against coercion attacks.

2.2.4 Processing Stage

After the voting period—including physical voting—has ended, votes are processed by the *Processor* before they can be decrypted and counted. The processor verifies the digital signatures, the existence of a timestamp and the well-formedness for each vote in the ballot box. The *Processor* then verifies that the *Registration Service* has the same record of ballots as in the ballot box. In doing so, the *Processor* verifies the integrity of the i-ballot box [7, p. 17].

The *Processor* also checks whether all voters were in the list of voters at the time of voting. Then, based on the timestamps, the *Processor* makes a lists of only the latest votes, hence discarding the previous votes of re-voters [18, § 6.4]. Finally, an *Annulment List* containing the identifiers of voters who voted both physically and over the internet is drawn up, and the *Processor* discards votes of those figuring on the list [7, p. 17], [20]. How this list is compiled is further touched upon in Section 3.5. Then, the *Processor* groups the i-votes by electoral districts and removes the voters' signatures from them. Additionally, anonymous ballots are re-encrypted and passed through a mix-net to remove trailing links between personalised and anonymised ballots. Finally, the *Processor* passes the anonymised and mixed ballots to the *Key Application* who decrypts the ballots, sums the votes and signs the result before outputting it along with the *tally-proof* described in Section 2.2.1 [7, § 7.2].

2.3 Concerns and Related Research

Much of the research surrounding Estonian internet voting concerns the system in use prior to the implementation of IVXV in 2017. While IVXV answered a number of concerns, notably those by Springall, Finkenauer, Durumeric, et al. [8] in 2014, the resulting system is not without flaws as acknowledged by Heiberg, Martens, Vinkel, et al. [9] in their whitepaper presenting IVXV. More generally, shortcomings are of organisational or technical nature, with the technical part split into architectural and design concerns and cryptographic concerns.

In June of 2019, a workgroup for internet voting was called together by the Estonian Minister of Foreign Trade and Information Technology [23], [24]. In December of the same year, the workgroup published [25], a list of 25 suggestions to be worked on. Proposals of improving the transparency and education of the population concerning the functioning of i-voting received overwhelming support from workgroup members, as did proposals increasing the auditability and robustness of logging and monitoring. Proposals aiming to permit full end-to-end (E2E) verifiability of their votes by voters were opposed due to concerns of voter coercion, and adequacy of the current verification scheme.

In April of 2020, a support group for improving the transparency of i-voting was formed in the Parliament of Estonia [26]. In the same year, Heiberg, Krips, and Willemson [27] published a paper discussing the option to allow voting from mobile devices. Although mobile voting is the main topic of their paper, remaining weaknesses in IVXV are also discussed, as are potential solutions usable regardless of mobile voting, such as a proposal to introduce a feedback channel to notify voters about voting actions associated with them.

Research regarding IVXV has been completed also outside of Estonia. In 2021, Zhang Zhang, Li, and Willemson [28] published the first format systematic security analysis of IVXV, and further explored the E2E verifiability aspect of the system. Also in 2021, Oliver Pereira published a paper highlighting the lack of individual verifiability in IVXV, which allows for ballot manipulation even if a voter verifies their ballot with the *Verification Application* [29]. This concern was previously acknowledged in [27] by Heiberg et al. In 2022, Johannes Müller from the University of Luxembourg published a paper which shows how the malleability of a homomorphic cryptosystem can be exploited to breach the privacy of votes [30]. The author mentions that findings were presented to the Estonian election authorities in August of 2021 and discussed with members from Smartmatic-Cybernetica[1] in September.

[1] Smartmatic-Cybernetica Centre for Excellence for Internet Voting, a main author of IVXV, https://cyber.ee/resources/news/first-of-a-kind-global-centre-of-excellence-to-advance-internet-voting/

There is therefore sufficient existing and on-going research that aims to improve various aspects of IVXV, however, except for [30] which explores the concerns of vote-privacy, research focuses more on the security of the scheme and resilience to longevity concerns of cryptosystems than protecting the identity of voters, which is why the author focuses on the specific aspect of identifiability of voters, especially if the number of auditors is increased as proposed in [25].

3 Proposed I-Voting scheme

In this part, the author presents the proposed modifications to the IVXV scheme. The aim is to follow the framework as closely as possible and not introduce security, privacy, or integrity concerns that don't already exist for the current implementation in Estonia.

3.1 Setting

The author denotes a signature scheme $\mathcal{S} = (\text{Gen}_{sk}, \text{Sign}, \text{Verify})$ with functions for key-generation, signing and verification. Any party p that must give a digital signature is in possession of a signature keypair $(sk^p_{pub}, sk^p_{priv})$ generated by Gen_{sk} and certified by a Certificate Authority *CA*, who issues $Cert^p_{CA}$. There exist also a cryptographic hash function Hash and the public-key cryptosystem $\mathcal{E} = (\text{Gen}_{ek}, \text{Enc}, \text{Dec})$ with functions for key-generation, encryption and decryption.

IVXV uses the BDOC signature format [18], [31] and a signed BDOC container contains at minimum the BDOC basic profile, which is made up of the hashes of all files signed, the certificate of the signer and the signature given by the signer, i.e., the signature component [31, § 5]. As such, in the context of ballots, a signature is always accompanied by the signer's certificate. The signature can later be qualified with a timestamp and certificate validity confirmation to provide a signature in the BDOC-TS format which is compliant with eIDAS[1] requirements for qualified electronic signatures (QES) [31, § 6], [32]. Because a BDOC container contains also the data which is being signed, a well-formed and signed BDOC container contains all the information needed to access the signed data, verify the integrity of the data and identify the signer. If the signature is not qualified, additional verification may be performed to verify the signer's identity.

The term *signature* may be a source of confusion between the signature component and the pair made up of the signature component and signer certificate. When the meaning of a signature is unclear from context and the distinction is important, the author explicitly specifies whether BDOC containers or signature components are discussed.

[1] An acronym for "Electronic Identification, Authentication and Trust Services", and the common name for the European Union's regulation on eID and trust services. `https://digital-strategy.ec.europa.eu/en/policies/discover-eidas`

3.2 Privacy and Integrity Requirements

In IVXV, because ballots are signed using the BDOC format, the author of a ballot—i.e., the voter—can be identified. To prevent the connection between voters and their ballots, at least the voter certificates must be removed from signatures.

However, it must still be verifiable that all ballots were cast by eligible voters, no ballots were removed from the box, and that no ballot was modified to reflect anything else than the voter's original intent. Signature components can be used for monitoring the integrity of votes and a separate ledger for witnessing votes can be used to check for removals. Remains the problem of digitally signing a vote such that

- voter identities are not tied to their ballots,
- arbitrary signatures cannot be used to fabricate ballots.

In other words, it should be provable that a signer belongs to a group, i.e., the group of eligible voters, without making them identifiable. If members can be added to this group, e.g., someone reaching legal voting age, and only the final list of group members is considered before votes are counted, this approach holds. However, because a voter can lose their right to vote during the voting period, e.g., if a voter is criminally convicted or if their electoral district changes[1], group membership must be revocable as well. The final standing of a voter cannot be considered for eligibility since a vote does not lose validity if the voter was eligible at the time of voting and no other constraints apply [20], e.g., the vote itself is invalid or the voter re-voted. As such, auditors must be able to verify whether all counted votes were cast by voters eligible for voting at the time of voting. Hence, voters cannot remain truly anonymous and the group approach is insufficient, as there must be a way to check the eligibility of voters and track their votes for potential annulment even after voting.

3.3 Re-Signing Verification Service

To prevent voter identification using certificates, the author proposes the replacement of voter signatures by another party and use of an independent trust service to verify and certify the *re-signing* process. Alternative options are discussed in Section 4.1.

Figure 1 depicts a simplified re-signing procedure. If party B wishes to re-sign some data

[1] An example is in the case of municipal elections. If the voter's residency changes from one municipality to another during the elections, they will be able to vote for a new set of candidates, but can no longer vote for the previous municipality's candidates.

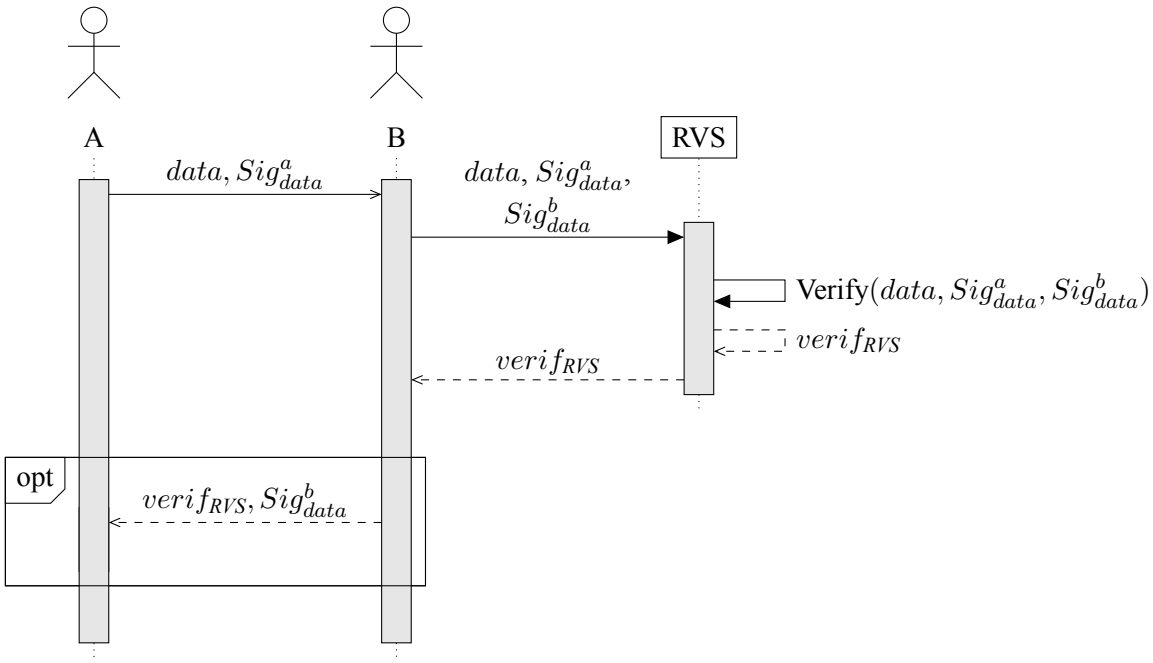

Figure 1. Simplified re-signing procedure.

on behalf of party A, B must obtain the data and the signature on it from A. B can then give its own signature to the data and provide both signatures and the data to the *Re-Signing Verification Service* (*RVS*). The *RVS* verifies that both signatures sign the same data and if so, returns a confirmation: the *RVS proof*.

Given the proof, the unsigned data, and B's signature on the data, anyone trusting the *RVS* can assert that A's signature on this data has existed. In practice, for this assertion to be possible, the *RVS* must additionally be provided with the certificates of A and B as shown in Figure 6 (Appendix 2), so that the service can verify the identity of the signers and include it in the proof. The verification component of the *RVS* response must thus be, at minimum,

$$verif_{RVS} = (id_a, id_b, Sig^a_{data}, Sig^b_{data}, Hash_{data}).$$

The *RVS proof* is the couple $(verif_{RVS}, Sig^{RVS}_{verif})$ made of the verification component and its *RVS*-issued signature. $Cert^{RVS}_{CA}$ must also be made available to verifiers.

If the identity of either party must be protected, the *RVS* can additionally be given a pseudonymisation function, which is assumed to be unidirectional. The *RVS* uses this function to process the identities of A and B to obtain pseudonyms to use instead of id_a and id_b. The verification component $verif_{RVS}$ must then additionally contain the chosen pseudonymisation function. Then, if B provides a third party C only sig^b_{data}, the data, and the *RVS proof*, C can be sure that sig^a_{data} exists, but not identify A without additional knowledge. If C knows that A is the member of a group, for example, if A's pseudonym

```
{
  "party0": "party_A_idcode",
  "party1": "party_B_idcode",
  "sig0": "party_A_signature",
  "sig1": "party_B_signature",
  "dataHash": "hash_of_data"
}
```

Figure 2. Minimal *RVS* verification component format.

appears on a trusted list of group members, i.e., the eligible voters list, then C can confirm A has signed data without having access to A's signature.

Figure 2 depicts a JSON object which could be used by the *RVS* for the verification component $verif_{RVS}$. This example does not use pseudonyms and hence does not protect the anonymity of signers. Only signature components are used however, certificates are not included in the response.

This model is impractical if multiple data files must be re-signed. For example, a BDOC container may contain multiple signed files. While only one file—the ballot—gets signed by the voter when casting a vote, a BDOC signature is given to data files and additional metadata regarding the signature [31]. Moreover, hashing the signed BDOC containers of A and B yields different results, since at least the signature component inside the containers is different.

Let D be the set of data objects to be signed, $n = |D|$ the number of objects to be signed, and $data_i$ a discrete data object, s.t., $\forall i, 0 < i \leq n, data_i \in D$. Moreover, Sig^a_D and Sig^b_D are of the form $Sig^x_D = \bigcup_{i=1}^{n} \text{Sign}(sk^x_{priv}, data_i)$, and $H = \{h \mid h = \text{Hash}(data_i)\}$. In this case, $verif_{RVS}$ is the tuple

$$(id_a, id_b, Sig^a_D, Sig^b_D, H).$$

For pseudonyms psd_a, psd_b obtained using the pseudonymisation function f_{psd}, $verif_{RVS}$ becomes

$$(psd_a, psd_b, Sig^a_D, Sig^b_D, H, f_{psd}).$$

For BDOC, Sig^x_D is obtained by signing an XML block (`ds:SignedInfo` in Appendix 3) that contains the hash values of all data files to be signed, and the hash of another XML block (`xades:SignedProperties`) containing metadata to be signed [31].

Figure 3 shows a redesigned data structure for when BDOC containers are used to pro-

```
{
  "parties": {
    0: {
      "id": "party_A_pseudonym",
      "sig": "party_A_signature_component"
    },
    1: {
      "id": "party_B_pseudonym",
      "sig": "party_B_signature_component"
    }
  },
  "data": "serialised_dsSignedInfo_block",
  "psd_algorithm": "id_of_pseudonymisation_function",
  "datetime": "xsDateTime_of_verification"
}
```

Figure 3. *RVS* verification component for BDOC signatures.

vide data and signatures. This structure can be extended to include additional information about signature qualifying elements, algorithms, metadata, etc. The `data` key's value must contain the hashes of data objects making up the signature for easy verification. For BDOC, the hashes are included in the serialised `ds:SignedInfo` block, but an additional key could be used to contain an array of hashes of data objects instead, to avoid having to parse the data block.

It is worth noting that for the pseudonymisation of signer identifiers to be possible, certificates accepted by the *RVS* must follow a standardised format. In Estonia, the ID code of an individual or the registry number of a company can be extracted from SK issued certificates. As such, the assumption for a standardised format holds for eID in Estonia.

3.4 Ballot Revocation Service

In the IVXV system, because only the last valid i-vote of a voter is kept, time-marks are used to establish the timeline of recurring ballots and signer certificates are used to verify if the ballots belong to the same voter, which is part of the reason why voter signatures must be preserved with ballots at least until the processing stage. In principle, only pseudonymity of ballots is required for annulment to be possible, i.e., it must be possible to identify ballots cast by a voter, but it is not necessary to identify the voter. Pseudonimity could be implemented by associating an election-specific identifier with each voter, from which identity could not be reconstructed without additional information. This pseudonym would then be bundled with the ballot instead of the voter's certificate, assuming that ballot integrity remains verifiable.

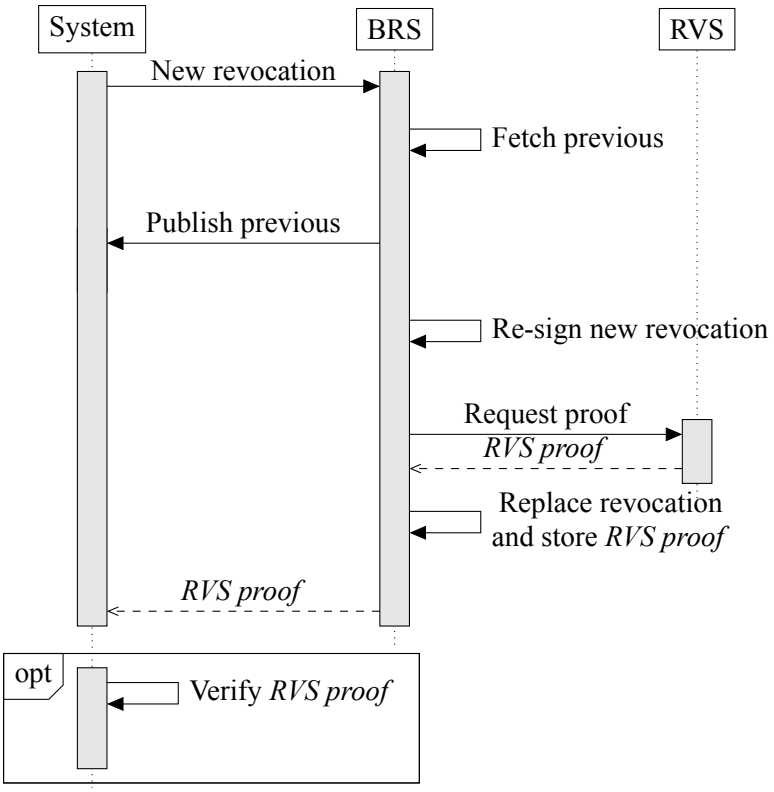

Figure 4. Ballot Revocation Service.

In practice, for IVXV, this implementation is flawed because in the case of parallel voting, the *Processor* must draw up a list of i-voters sorted by polling stations [7, § 5]. As such, the *Processor* must be able to determine the identity of a voter from the i-vote. Furthermore, without additional security checks, this approach may be susceptible to clash attacks [33] once the ballot verification period expires for the voter.

Instead, the author proposes an approach analogous to certificate revocation technologies such as certificate revocation lists (CRL). This approach requires the creation and use of the *Ballot Revocation Service* (*BRS*). Figure 4 illustrates this process.

The *BRS* maintains a list of ballot revocation certificates associated with voters, but does not create certificates, rather, it is only responsible for keeping and activating them. A ballot revocation certificate is activated once it is published by the *BRS*. A published certificate must carry the *BRS*'s signature so that revocation certificates intercepted before they reach the *BRS* cannot be used by other parties. Accepting only *BRS*-signed certificates is an organisational concern.

Each voter may only have one unpublished ballot revocation certificate associated with them at a time. When a voter re-votes, the *BRS* publishes the revocation certificate associated with the voter, which invalidates the previous ballot. Then, the *BRS* associates the

new certificate with the voter, replacing the published one. This process is repeated every time a voter re-casts a vote.

3.5 Annulment List Creation

The IVXV framework supports the concept of parallel voting which means that a voter who votes electronically is not prevented from later voting physically. Moreover, by convention, the physical ballot of a voter supersedes their electronic ballots [34, § 48^7 (2)]. As such, once both the i-voting and physical voting periods conclude, *double-voters* who have voted both physically and electronically must be identified and their i-votes annulled. This process is conducted using *Annulment Lists*.

To receive the *Annulment List*, the EHS sends a list of i-voters to VIS3, where the list of physical voters and i-voters are compared to determine the voters whose votes must be annulled. The VIS3 then returns the list of matches to the EHS [35], [36]. The lists use voter personal ID codes [37], [38]. Previously, because the lists of physical voters were also physical, it was easier to print out the list of i-voters by polling stations, send them to the respective stations, and have the polling stations mark the double voters manually [39]. Since 2021 however, physical polling stations use electronic voters lists which are managed by VIS3 [35], [40], which mitigates the workload factor.

The electronic lists enable two approaches for using pseudonyms instead of ID codes for the *Annulment List*. In the first way, the VIS3 sends the list of physical voters to the EHS, and the EHS handles pseudonymising the list. Then, the *Annulment List* can be compiled within the EHS. This reversed approach doesn't require changes to be made to the VIS3, however the EHS must then ensure the auditability of the list creation process. The second way is where the EHS forwards the list of pseudonymised i-voters to the VIS3, the VIS3 compiles the pseudonymised list of physical voters, makes the comparison, and returns the pseudonymised *Annulment List* to the EHS. If the second way is used, then the VIS3 should also provide pseudonymised eligible voter lists to the EHS duirng the voting stage.

3.6 Resulting I-Voting Scheme

In this subsection, the author describes the implementation of the newly introduced or modified concepts in the context of the IVXV i-voting scheme. Unchanged processes such as vote counting or preliminary voter identification are not described. The author assumes that a pseudonymisation function has been agreed upon. The choice of such a function is described in Section 4.3.

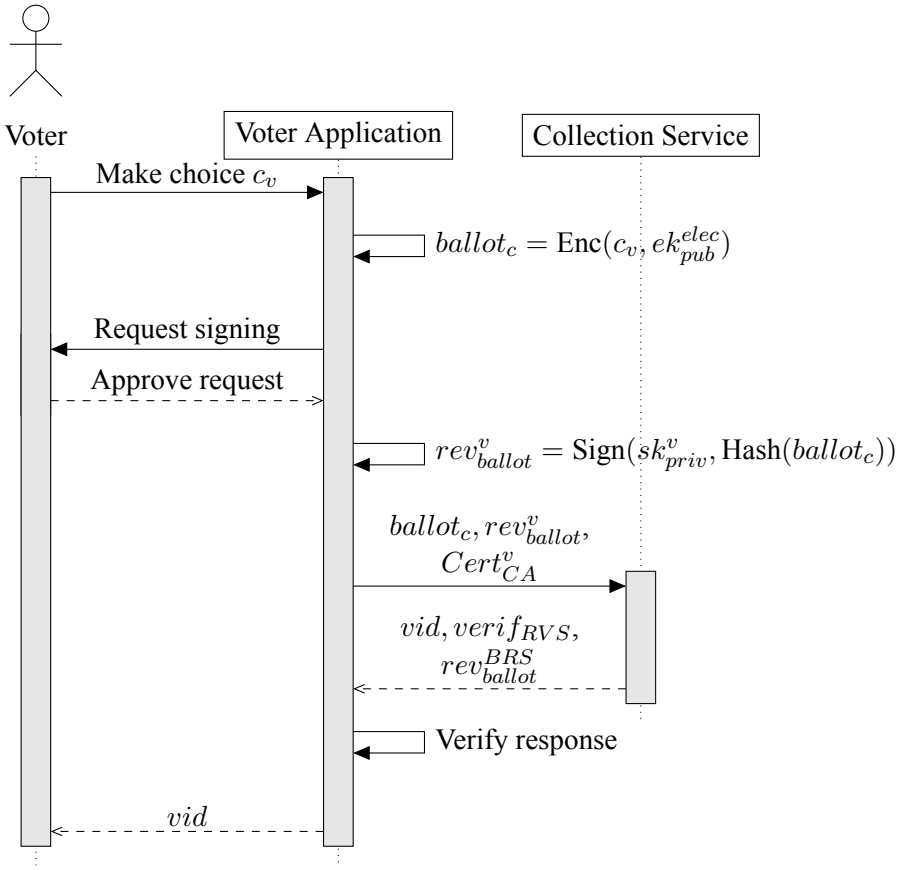

Figure 5. Revised vote casting.

3.6.1 Creating the Ballot Revocation Certificate

Let V represent the set of eligible voters and C the set of choices. An eligible voter $v \in V$ who wants to vote for a candidate $c_v \in C$ available to them uses the *Voter Application* to create a ballot and its revocation certificate using the process described below and shown in Figure 5.

The *Voter Application* creates the ballot $ballot_c$ by encrypting the voter's choice as in IVXV. It then generates brk_{ballot_c}, which is a data structure containing $\text{Hash}(ballot_c)$ and signs it using the voter's private key: $Sig^v_{brk} = \text{Sign}(sk^v_{priv}, brk_{ballot})$. The revocation certificate rev^v_{ballot} is the couple $(brk_{ballot_c}, Sig^v_{brk})$. Finally, the *Voter Application* forwards the ballot, the revocation certificate and the voter's certificate to the *Collection Service*.

The data structure brk_{ballot_c} is arbitrary[1] and can be as simple as a file containing the key-value pair ("Revoke:", $\text{Hash}(ballot_c)$). It serves as a link between the signature and the ballot and carries the intent of revoking a ballot. There is no cryptographic process of revoking a ballot, rather, revocation is based on convention.

[1] Arbitrary in the sense that the author does not define a format. A well-defined format must be used for a practical implementation.

3.6.2 Storing the Ballot and its Revocation Certificate

The *Collection Service* receives the ballot and its revocation certificate, generates an identifier vid for the ballot, and checks using the identity from the signer certificate if the voter is eligible to vote. The service then checks if the revocation certificate corresponds to the ballot. If so, the *Collection Service* forwards the ballot revocation certificate to the *Ballot Revocation Service*. It is in the *Collection Service*'s interest to verify the correspondence between the ballot and its revocation certificate since the service is responsible for the integrity of the ballot box. If it is later found that there are ballots for which no revocation certificate matches and the *BRS* can prove its own honesty, the blame will lie on the *Collection Service* either for manipulating ballots or for accepting a compromised ballot from the *Voter Application*.

The *BRS* re-signs the ballot revocation certificate by extracting the revocation file from the BDOC container and creating a new signed container with it. The *BRS* then forwards both the voter-signed and self-signed revocation certificates to the *Re-Signing Verification Service*. The *RVS* verifies the equivalence of data of both signed containers, the correctness of the signatures and the validity of the signer certificates. If all verifications are successful, the *RVS* extracts the identifiers from the signer certificates and obtains pseudonyms using the agreed-upon pseudonymisation function. It then compiles the data structure $verif_{RVS}$, signs it, and sends it back to the *BRS*. The *RVS* keeps a copy of all sent responses and the *BRS* keeps a copy of all received responses. This data can later be used by the *Processor* and *Auditor*.

The *BRS* verifies the signature on the response and the content of $verif_{RVS}$ with rev^{v}_{ballot} and rev^{BRS}_{ballot}. If successful, the *BRS* then stores the re-signed certificate and the *RVS proof* under the voter's pseudonym in its database and discards the voter-signed certificate. If not, the *BRS* alerts the system manager and the *Collection Service* of the discrepancy and keeps both self- and voter-signed certificates for subsequent investigation. Because the ballot itself is never given to the *BRS* and is discarded by the *Collection Service* when notified of the discrepancy, an investigator never gets hold of both the ballot and rev^{v}_{ballot}.

Finally, the *BRS* returns $verif_{RVS}$ to the *Collection Service*. The *Collection Service* checks Sig^{RVS}_{verif} and whether $verif_{RVS}$ matches with rev^{v}_{ballot}. If so, the service signs the vote, stores the self-signed vote and the *RVS proof* in the ballot box, and discards the voter-signed ballot revocation certificate. The service then returns to the *Voter Application* the ballot identifier vid and the re-signing proof $verif_{RVS}$. If however $verif_{RVS}$ does not match with the voter-signed certificate, the *Collection Service* does not store the ballot and alerts the *Voter Application*, the *BRS* and the system manager of the discrepancy.

3.6.3 Verifying the Process on the Client-Side

The *Voter Application* verifies Sig^{RVS}_{verif} and whether $verif_{RVS}$ corresponds with the ballot revocation certificate signed by the voter. It then displays the QR code to the voter containing the random number and the ballot identifier.

When the *Verification Application* makes a request to the *Collection Service* using the ballot identifier, the *Collection Service* responds with the ballot from the ballot box and $verif_{RVS}$. The *Verification Application* verifies Sig^{RVS}_{verif}, computes the unsigned ballot revocation certificate using the ballot, and compares it against $verif_{RVS}$.

3.6.4 Activating Revocation Certificates

The above steps omit the case where a voter votes again. There are two cases where revocation certificates must be activated: either because a voter re-votes, in which case their previous vote is annulled, or because the i-voting of a voter needs to be annulled, as is the case of double voters. The former case in handled by the *BRS* and activation is referred to as publishing. The latter case is handled by the *Processor* using the data from the *BRS* and not the *BRS* itself.

In the first case, before the *BRS* stores the re-signed ballot revocation certificate and the *RVS* proof under the voter's pseudonym, it must check whether an entry already exists for the voter. If no entry exists, the first entry is created and certificates stored as described in Section 3.6.2. If an entry exists, the *BRS* must publish the previous ballot revocation certificate to make place for the new one.

For this, the *BRS* sends to the *Collection Service* the revocation certificate of the previous ballot and the *RVS proof* associated with it. The *Collection Service* checks whether the identity of the new revocation certificate matches the pseudonymised identity in $verif_{RVS}$. Then, the *Collection Service* checks whether the old revocation certificate matches with $verif_{RVS}$. These steps are necessary to ensure that the *BRS* does not provide the revocation certificate of another voter. If checks pass, the *Collection Service* adds its signature to $rev^{BRS}_{ballot_{old}}$ and forwards it to the ballot box. It confirms the completion of the revocation procedure to the *BRS*. Only then does the *BRS* re-sign $rev^{v}_{ballot_{new}}$. Finally, the *BRS* replaces the voter's database entry with the new revocation certificate and *RVS proof*. The *BRS* stores the confirmation of the *Collection Service* for non-repudiation purposes.

The existence of an entry in the *BRS* database could also be used to notify the voter whether they have already voted if this information does not already come from the *List Service*, i.e., the service returning a list of choices for the voter and which may keep track of whether

the voter has already voted.

3.6.5 Processing Ballots

The *Processing Application* takes for input the contents of the i-ballot box, the contents of the *BRS* database, and the *RVS proofs* stored by the *RVS* itself. The application verifies that

- all *RVS proofs* are valid—their structure and *RVS*-issued signature—and that the number and contents of proofs from the *RVS* and *BRS* match,
- each revocation certificate has a corresponding proof,
- each revocation certificate was originally signed by an eligible voter,
- each ballot has a corresponding revocation certificate, either published or unpublished, and vice-versa,
- each revocation certificate is signed by the *BRS*,
- each published revocation certificate is also signed by the *Collection Service*.

Any discrepancy must be investigated to determine whether a technical error occurred or if there was foul-play. Section 3.7 where the general auditing of the system is described provides further insight into how the verifications are carried out.

Unlike in IVXV, the *Processing Application* does not have to annul superseded votes of voters since that is handled by the *BRS* during the voting period. However, votes of double-voters must still be annulled. As such, after the application performs the verifications listed above, it must annul the necessary ballots of double-voters according to the *Annulment List*. Either option for compiling the list described in Section 3.5 can be used, however if the list is compiled within the EHS, then the *Processor* must be observed when it pseudonymises the list of physical voters, similarly to the observation of vote counting. This is required to ensure pseudonymisation is done correctly and that the *Processor* does not misuse the list to map ballots to ID codes.

The *Processor* then inputs the *Annulment List* into the *Processing Application*. The application takes a snapshot of the ballot box for auditors and then signs and adds the ballot revocation certificates of voters in the *Annulment List* to the ballot box. Finally, the application outputs only the ballots that pass all validity checks and for which no revocation certificate is active—the ballots to be counted.

3.6.6 Reverting I-Voting for a Voter

An additional advantage of the *Ballot Revocation Service* approach is the easy implementation of vote cancelling by a voter. For example, if a voter is coerced into casting an i-vote, they can currently replace but not revoke their vote. As such, if the voter wishes to abstain from voting, which is their legal right, they can no longer do so. To counter this, an additional option could be given in the *Voter Application* to revoke a vote.

This option should only be shown in case the voter has previously cast an i-vote. Casting such a vote follows the same procedure as any other vote until the *BRS* has published the revocation certificate. Instead of replacing the old certificate with a new one, the *BRS* deletes the voter's pseudonym from its database. The *Collection Service* does not store the ballot issued to revert i-voting in the ballot box, and simply discards it after interacting with the *BRS*. Should the same voter choose to i-vote again, it is as if they were i-voting for the first time. Anyone knowing the exact time of voting could investigate log-files to obtain indirect proof that the voter did vote however.

3.7 Auditing the Scheme

The *Auditor* must be able to verify the activities of the *Collection Service*, of the *Ballot Revocation Service* and of the *Processing Application*. More specifically, the *Auditor* must be able to verify that

1. all revocation certificates belonged to valid voters during creation time,
2. the *BRS* properly re-signed and stored all revocation certificates using the *RVS*,
3. the *BRS* published correct certificates when requested by the *Collection Service*,
4. the *BRS* dropped no certificates,
5. published certificates reached the i-ballot box and were signed by the *Collection Service*,
6. all votes reached the i-ballot box,
7. all revocation certificates, both published and unpublished, have a matching vote and vice versa,
8. the *Processing Application* issued annulment certificates correctly.

To carry out the audit, the *Auditor* must have access to the pseudonymised form of all eligible voter lists used during the voting period, the pseudonymised *Annulment List*, the encrypted votes in the ballot box, the published and unpublished revocation certificates, and a snapshot of the ballot box taken before processing.

The *Auditor* begins by processing unpublished revocation certificates. For each certificate, it verifies that

- the *RVS proof* stored alongside the *BRS*-signed certificate is signed by the *RVS*, corresponds to the revocation certificate (condition 2),
- the pseudonym of the voter from the *RVS proof* figures in the list of eligible voters at the time of voting (condition 1),
- a matching ballot in the i-ballot box exists (conditions 6, 7).

Then the *Auditor* carries out a similar process, but for certificates published to the i-ballot box, with the only difference being that the signatures of both the *BRS* and of either the *Collection Service* or the *Processing Application* must figure on the certificates (conditions 5, 8). If for any revocation certificate, either published or unpublished, a vote cannot be found in the ballot box, condition 6 is unmet and the blame lies on the *Collection Service*.

The *Auditor* additionally verifies that for each pseudonym, only one unpublished revocation certificate is stored by the *BRS*. This is part of condition 2. If there remain ballots that have not been paired with a revocation certificate, the blame may lie either on the *Collection Service* or on the *BRS* (condition 6). For each unmatched ballot, the *Auditor* checks whether the *BRS* has a confirmation from the *Collection Service* regarding the publishing of the revocation certificate. If so, the blame is on the *Collection Service*, else, on the the *BRS*. The *Auditor* has the additional possibility of requesting re-signing records from the *RVS* and comparing them with records from the *BRS* to make sure that the *BRS* did not drop verifications (condition 4), however this would be partly caught during the voting process or with the previously described check.

The *Auditor* verifies condition 8 by using the *Annulment List* and the snapshot of the i-ballot box taken before double-votes are processed by the *Processor*. The *Auditor* can compare if the initial and final ballot box states correspond to annulments carried out following the *Annulment List*.

Checking for condition 3 follows from checking for 2, 4, and 7. If the *Collection Service* and *BRS* collude to publish wrong certificates, the *BRS* must shuffle its database and store certificates under the wrong pseudonym. This is caught by checking for condition 2. If to avoid this the *BRS* stores multiple certificates under the same pseudonym, the subterfuge is caught by checking that only one certificate is stored per pseudonym (condition 2). If the *BRS* drops certificates and the *Collection Service* drops votes to satisfy 7, this is caught by checking for condition 4.

4 Discussion

The proposed approach is not without its flaws and does make some fundamental modifications to the current i-voting scheme even though it attempts to use as much of the functionality supported by the IVXV framework as possible.

4.1 Cryptographic Approaches to Re-Signing

Because the use of the *RVS* introduces an additional party that requires organisational trust, approaches based on cryptographic trust could be considered for keeping the anonymity of signers, while preserving the integrity guarantees made by a signature.

4.1.1 Anonymous Signatures

The use of anonymous signatures could be considered, which are signatures that do not reveal the signer's identity but for the provenance of which verifiable claims can be made. Such signatures are openly verifiable and the signer can be associated with a group, but not identified. Two main approaches to anonymous signatures are group signatures [41] and ring signatures [42]. For either, any individual belonging to the group (or ring) can give signatures on behalf of the whole group. The difference is that for group signatures, there is a group manager who is able to verify who gave a signature. For ring signatures, no such entity exists [42, § 2].

The problem with such signatures is that the i-voting back-end cannot identify the voter at all without using a separate identification mechanism. If the identification mechanism of group signatures is used, this must be provided also to auditors, which defeats the purpose of protecting voter identities. If another identification approach is implemented, the problem of re-voting still cannot be solved without also implementing some form of vote-tracking tied to the identity, which is both complex and does not improve the ballot privacy of voters either. A second problem with such an approach is that there is no convenient integration with the existing PKI and eID infrastructure. For each election, eligible voters would need to be issued signing keys, or their existing keys managed, and then eligibility changes would also need to be monitored and managed.

4.1.2 Proxy Re-Encryption Schemes

Blaze, Bleumer, and Strauss [43] first described a PKI-based scheme where a conversion key is used to re-encrypt a ciphertext from one public-key to another, now called a proxy re-encryption (PRE) scheme. More generally, PRE schemes are designed to transform a ciphertext encrypted for one party such that another party may decrypt it, without knowing the first party's decryption key. If the scheme is unidirectional, the resulting ciphertext cannot be re-altered such that the first party's key may decrypt it again.

The reversed concept can be applied to PKI-based digital signatures, and could allow protecting the original signer's identity. A conversion key that re-signs data from one private-key to another can be created, and the signature can be verified using the latter party's public key. This reversed method for signatures is however impractical for IVXV, as a conversion key would need to be created for each eligible voter and for each election. Moreover, not any key and cryptosystem can be used for PRE.

4.1.3 Replacing Signer Certificates

Because a signature component on its own does not contain personally identifiable information, it could be argued that removing certificates anonymises the voters. In such a scheme, the *BRS* would be provided certificates that bind to the public keys of all eligible voters. Because the *BRS* does not have access to the private keys used for signing, the *BRS* cannot forge ballot signatures. Such an approach removes the need for the *RVS*, because the signature is never removed, and as such the integrity of the revocation certificate remains verifiable. To verify the validity of a revocation certificate, all public keys of the voter associated with it must be checked because the public keys of different signing means of a same person are different (ID-card, Mobile-ID, Digi-ID). As such, there must exist a verified list collating public keys of voters with their pseudonyms.

With such an approach, the *BRS* could be redundant, as the *Collection Service* could replace the certificates of ballots after receiving a confirmation from the *Validation Service* that the voter signature was valid. However, a mechanism must exist that binds the confirmation to the ballot without compromising the voter's identity. As such, the main problem points of this approach are the organisational concerns of issuing multiple certificates, the compilation of a list of keys and the implementation of voter eligibility verification.

More generally, maintaining the complete anonymity of voters once their vote is cast cannot currently be achieved because of the organisational requirements of allowing re-voting and double voting, and is why the author opted for the pseudonymity approach and the creation of an additional trust service. The approach where certificates are replaced instead

of signatures would be an interesting topic for future work.

4.2 Caveats of an External Trust Service

In the current IVXV implementation, a time-stamping service doubles as the *Registration Service*. This dual role is only possible due to the simplicity of the registration requirement, which overlaps with the principles of a time-stamping service [20]. Moreover, because the service issuing timestamps is already trusted, no additional trust needs to be placed in an independent party to handle vote registration.

While the additional *Ballot Revocation Service* can also fulfill the role of the *Registration Service* in the sense of being a witness to votes so that the *Collection Service* cannot drop them, the *BRS* is an internal service. To match the requirement for the *Registration Service* to be a party independent of the EHS, the *Re-Signing Verification Service* is used. However, its requirements are specific, and as such, an existing service cannot be used. One possibility is asking an already trusted party, such as the one providing the time-stamping service, to implement the custom functionality required. However, in practice, there are obstacles to this approach.

The organisations offering time-stamping or certificate validity confirmation services can be reasonably trusted because they are widely used and are not a component specific to elections. An example is SK who offers both services to both the government and private entities. Unless the *Organiser* colludes with such organisations, there is little benefit for them to implement a sporadically used functionality specific to one client only. It would not be cost-beneficial to the *Organiser* to solely pay for the upkeep of such a service either. The advantage that the *RVS* has over the *BRS*, is that it may have uses outside of i-elections, and as such, a private company may be more likely to implement the service if a market niche can be established. Exploring the concept of re-signing methods would be an excellent topic for future work and can have direct implications on the viability of the method proposed by the author.

Because the *RVS* is external and does receive the identities of signers, it has the capability of compiling a list of all i-voters. This however is not a new concern, as the *Validity Service* currently has the same capability as well. A difference is that the *RVS* learns also the ballot revocation certificates, while the *Validity Service* currently learns nothing other than a signer's certificate. To mitigate this, revocation certificates could also be re-signed by only sending the hash and two signatures on the hash to the *RVS* instead of the BDOC containers containing the revocation file. However, because the only information a revocation certificate should expose about a ballot is the ballot's hash, they cannot be

used to later link a voter to their choice without also obtaining copies of ballots, even by an attacker who is able to breach ballot encryption.

In essence, unless the *RVS* is handled by an organisation whose independence can be trusted and that is vetted by a trusted auditor, the proposed system weakens overall trust in the i-voting system. However, ballot privacy is still maintained as long as the *Collection Service* and *BRS* operate properly. Cryptographic measures avoid the organisational trust-related concerns, but as shown in Section 4.1, they are not practical.

4.3 The Problem of Pseudonymisation

The pseudonymisation function that translates personal identification codes to pseudonyms is a crucial component of the proposed solution. The two main ways of implementing one are either using a one-way function, or by using randomly defined mapping. For example, each voter could be assigned a randomly generated identifier that's stored in a database. The latter approach is however impractical if an external service is used, such as the *RVS*, because then the service would need to have access to the mapping database. Hence, one-way functions are the more practical choice.

Synchronising the pseudonymisation function with an external service is not trivial either, especially if the service is independent. If the EHS decides to use a certain function, it is arguably easier to push for implementation by the VIS3 than the *RVS*. A more practical approach is for the *RVS* to implement a set of common functions and then make the list of usable functions available to clients, similar to websites announcing supported cipher-suites for the TLS protocol. Clients then specify the identifier or name of the pseudonymisation function to the *RVS* when making their re-signing request.

Pseudonymisation functions must also be collision free, this is imperative for the integrity of the voting process. Cryptographic hash functions are therefore good candidates because they satisfy the requirement for low-collision rates and being irreversible. Salting may also be used with hash functions in order to make hash dictionary attacks more complicated, for example in the case where the *BRS* database is breached. For use with the proposed model, the salt must be the same for all voters, or else pseudonym comparison becomes infeasible. A salt can either be public or on a need-to-know basis. A need-to-know salt is made available only to parties involved in the system and who need to pseudonymise identifiers. At minimum, those parties are the *BRS*, the *RVS* and the VIS3, if the VIS3 provides pseudonymised lists. Otherwise, instead of the VIS3, the *Processor* and *Collection Service* must know the function to compute pseudonyms for the *Annulment List* and the eligible voters lists, respectively.

Keeping the salt from public access prevents some third party from computing a voter's pseudonym and looking it up, for example if the *Annulment List* is leaked. However, a non-public salt increases the complexity of making it available to required parties while protecting it from leakage. For example, if the *BRS* is breached, the attacker may get both the list of i-voters and the salt used for pseudonymising. The advantage of pseudonyms is that even if the attacker can compute them, there is no efficient way of looking pseudonyms up without also having a complete list of original identities, and pre-computing a correlation table.

4.4 Publishing the Detached Votes

Because personally identifiable information is no longer tied to the encrypted ballots themselves, it could be argued that the encrypted ballots should be published to a billboard, accessible by anyone. That is, making the i-ballot box public. However, this may end up lowering trust in i-voting in general.

While publishing the i-ballot box does not enable vote selling more than the current system, it may increase risks for a voter if they were coerced into voting. In a scenario where a voter is coerced into voting, and the coercer gets the vote's hash from the client side, the coercer can monitor the public bulletin board to check if a voter has re-voted. This is possible because upon re-voting, the revocation certificate of the previous vote is published to the bulletin board as well. The coercer is hence able to remotely detect if a coerced voter has re-voted, provided they know which vote to look for. By physically casting their vote, i.e., double-voting, the voter may still exercise free will however. The alternative of not publishing revocation certificates to the public bulletin board may raise questions when voting results and statistics are published, because the number of cancelled votes cannot be publicly counted.

If a bulletin board system was implemented, it would still need modifications to leak as little information about voters as possible. For example, the board should not be updated in real-time. Otherwise, it would be easier to single out voters by someone who knows the voting time. While more difficult at peak voting hours, this is trivial for a coercer who forces someone to vote at three in the morning, for example. A potential solution to this would be establishing a counter of incoming votes. The bulletin board can be updated after the count is reached, following which the counter is reset. Still, the potential for correlation may remain unless a mix-net and re-encryption are applied to the bulletin board, which is impractical, and serves no purpose other than displaying the total amount of votes and applied revocation certificates received.

5 Future Work

This paper only proposes the idea of modifications that can be made to IVXV to improve voter privacy. As such, the author does not establish concrete requirements for the newly created services, such as data formats or protocols. Before the proposal can be implemented, the protocols surrounding the services must therefore be defined. Moreover, the modified system would benefit of a formal security analysis, as enough changes are made that not all results established in [28] carry over to the modified system. Additionally, the proposed modifications could be discussed in the context of modifications proposed by other research, as the changes are major enough to warrant a new version of IVXV rather than a new iteration. For example, in case a feedback loop system is implemented, it could be interesting to establish how it would work with the Ballot Revocation Service.

There is also the topic of re-signing procedures. Both the IVXV and modified scheme could benefit from anonymous signatures (Section 4.1.1), for which the identification of a signer is possible before a certain moment, and the belonging to a group possible after that moment. The *Re-Signing Verification Service* may be a step in the right direction, however it relies on traditional trust rather than cryptographic trust. While the implementation of a nation-wide re-signing scheme (Section 4.1.2) may seem utopian, it may be possible to provide holders of eID signing keys and certificates that require an intermediary step before they can be identified. Different approaches to the pseudonymous signatures could therefore be researched. Instead of the *RVS*, the certificate replacement approach discussed in Section 4.1.3 may be suitable for achieving the same results as this work, but this approach would need to be more thoroughly researched. Finally, the use of the *RVS* in contexts other than i-voting could be explored, as the more areas the service is used in, the higher the trust that can be given to it due to the assumption that vulnerabilities would be discovered and patched sooner.

6 Conclusion

In this work, the author has shown that it is possible to improve the degree of privacy of voters while keeping the core flow of the IVXV internet voting system. The author addressed the following three problems:

1. *Identification of voters and their ballots is needed to annul votes of double-voters.* By proposing changes to the process of drawing up the *Annulment List*, the author removes the need for bundling voter identities with their ballots so that they can be identified and annulled.
2. *Bundling signatures with ballots is needed to annul votes of re-voters.* The author showed that ballots and identities can be separated while maintaining the integrity of ballots. By using the *Ballot Revocation Service*, ballots of re-voters can be annulled even if voter identification must be preserved. As such, an attacker cannot gain access to both a list of voters and their encrypted ballots without breaching two separate databases.
3. *Data cannot be re-signed by another party while protecting the identity of the original signer and also proving the integrity of data.* By introducing the *Re-Signing Verification Service*, the author enables a way to prove the re-signing of documents using an independent trust service. While this requires trust in an additional party, it is no different from trust placed in a CA, especially if the CA offers the verification service.

Organisational concerns regarding the solution proposed by the author can be argued for, and thus possible implementation is not an unreasonable prospect. Naturally, since implementation does require two significant modifications to the scheme, other research about the security and privacy of IVXV should also be factored for the next version of the i-voting system. Due to the author's solution retaining ideas from IVXV, it should be partly or completely compatible with many alternative modifications to the scheme. Moreover, before the solution can be implemented, some aspects need to be further fleshed out, such as the specification of a protocol for interfacing with the *Re-Signing Verification Service*, and a well-defined structure for ballot revocation certificates. Therefore, this work paves the way for future work regarding internet voting in Estonia and methods for replacing signatures while maintaining the integrity of the signed data.

References

[1] *Elektroonilise hääletamise statistika*. State Electoral Office of Estonia. URL: https://www.valimised.ee/et/valimiste-arhiiv/elektroonilise-haaletamise-statistika (visited on Apr. 26, 2022).

[2] Thomas W. Edgar and David O. Manz. "Part II. Observational Research Methods". In: *Research Methods for Cyber Security*. Ed. by Thomas W. Edgar and David O. Manz. Syngress, 2017, p. 93. ISBN: 978-0-12-805349-2.

[3] Riigikogu. *Identity Documents Act*. RT I, 15.10.2021, 3 (English translation). Riigi Teataja, 2021. URL: https://www.riigiteataja.ee/en/eli/501112021001 (visited on Apr. 25, 2022).

[4] *About SK*. skidsolutions.eu. URL: https://www.skidsolutions.eu/en/about/ (visited on Apr. 25, 2022).

[5] SK ID Solutions AS. *Terms and Conditions for Use of Certificates of Personal Identification Documents of the Republic of Estonia*. sk.ee. July 1, 2018. URL: https://www.sk.ee/upload/files/SK-TCU-ESTEID-EN-20180701.pdf (visited on Apr. 25, 2022).

[6] Estonian Minister of Entrepreneurship and Information Technology. *The description and requirements for ensuring the continuity of digital identification and digital signing as a vital service*. RT I, 15.01.2019, 11 (English translation). Riigi Teataja, 2019. URL: https://www.riigiteataja.ee/en/eli/510102019001 (visited on Apr. 25, 2022).

[7] *General Framework of Electronic Voting and Implementation thereof at National Elections in Estonia*. Tech. rep. Version IVXV-ÜK-1.0. State Electoral Office of Estonia, 2017. URL: https://www.valimised.ee/sites/default/files/uploads/eng/IVXV-UK-1.0-eng.pdf (visited on Apr. 25, 2022).

[8] Drew Springall, Travis Finkenauer, Zakir Durumeric, et al. "Security Analysis of the Estonian Internet Voting System". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS '14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 703–715. DOI: 10.1145/2660267.2660315.

[9] Sven Heiberg, Tarvi Martens, Priit Vinkel, et al. “Improving the Verifiability of the Estonian Internet Voting Scheme”. In: *Electronic Voting*. Ed. by Robert Krimmer, Melanie Volkamer, Jordi Barrat, et al. Cham: Springer International Publishing, 2017, pp. 92–107. DOI: 10.1007/978-3-319-52240-1_6.

[10] Vabariigi Valimiskomisjon. *Tehniliste nõuete kehtestamine elektroonilise hääletamise üldpõhimõtete tagamiseks*. RT III, 06.05.2017, 1 (in Estonian). Riigi Teataja, 2017. URL: https://www.riigiteataja.ee/akt/306052017001.

[11] State Electoral Office of Estonia. *IVXV architecture*. Tech. rep. Version IVXV-AR-EN-1.4.0. Jan. 18, 2019. URL: https://www.valimised.ee/sites/default/files/2021-05/IVXV%20architecture%20%E2%80%93%20overview%20of%20technical%20realisation.pdf (visited on Apr. 25, 2022).

[12] State Electoral Office of Estonia. *IVXV võtmerakendus*. (in Estonian). Tech. rep. Version IVXV-SVR-1.4.0. Jan. 18, 2019. URL: https://www.valimised.ee/sites/default/files/2021-05/IVXV%20key%20application%20%E2%80%93%20technology%20used%20in%20key%20management%20%28in%20Estonian%29.pdf (visited on Apr. 25, 2022).

[13] State Electoral Office of Estonia. *E-hääletamise süsteemi infoturbe poliitika*. (in Estonian). Tech. rep. Version IVXV-SVR-1.4.0. Riigi Valimisteenistus, Jan. 18, 2019. URL: https://www.valimised.ee/sites/default/files/2021-10/IVXV%20e-h%C3%A4%C3%A4letamise%20s%C3%BCsteemi%20infoturbe%20poliitika.pdf (visited on Apr. 25, 2022).

[14] Andrew Regenscheid, Larry Feldman, and Gregory Witte. *NIST Special Publication 800-88, Revision 1: Guidelines for Media Sanitization*. ITL Bulletin, National Institute of Standards and Technology, Gaithersburg, MD, 2015. DOI: 10.6028/NIST.SP.800-88r1.

[15] Canadian Centre for Cyber Security. *Guidance on cloud service cryptography*. Canadian Communications Security Establishment, Ottawa, 2020. URL: https://publications.gc.ca/pub?id=9.888411&sl=0 (visited on May 1, 2022).

[16] T. Elgamal. “A public key cryptosystem and a signature scheme based on discrete logarithms”. In: *IEEE Transactions on Information Theory* 31.4 (1985), pp. 469–472. DOI: 10.1109/TIT.1985.1057074.

[17] State Electoral Office of Estonia. *IVXV raamistiku nõuded krüptosüsteemile*. (in Estonian). URL: https://www.valimised.ee/sites/default/files/2021-10/IVXV%20raamistiku%20n%C3%B5uded%20kr%C3%BCptos%C3%BCsteemile.pdf (visited on Apr. 26, 2022).

[18] State Electoral Office of Estonia. *IVXV protocols*. Tech. rep. Version IVXV-PR-EN-1.6.0. May 31, 2020. URL: https://www.valimised.ee/sites/default/files/2021-05/IVXV%20protocols%20%E2%80%93%20data%20structures%20and%20data%20exchange%20protocols.pdf (visited on Apr. 25, 2022).

[19] Tarmo Hanga. *Kuidas allkirjastatakse e-hääli?* blog.ria.ee. Oct. 22, 2021. URL: https://blog.ria.ee/kuidas-allkirjastatakse-e-haali/ (visited on Apr. 25, 2022).

[20] S. Heiberg. private communication. Apr. 2022.

[21] Sven Heiberg, Peeter Laud, and Jan Willemson. "The Application of I-Voting for Estonian Parliamentary Elections of 2011". In: *E-Voting and Identity*. Ed. by Aggelos Kiayias and Helger Lipmaa. Berlin, Heidelberg: Springer, 2012, pp. 208–223. DOI: 10.1007/978-3-642-32747-6_13.

[22] Mark A. Will and Ryan K.L. Ko. "Chapter 5 - A guide to homomorphic encryption". In: *The Cloud Security Ecosystem*. Ed. by Ryan Ko and Kim-Kwang Raymond Choo. Boston: Syngress, 2015, p. 107. ISBN: 978-0-12-801595-7.

[23] *Valimiste küberturvalisus*. (in Estonian). Ministry of Economic Affairs and Communications. Dec. 12, 2019. URL: https://www.mkm.ee/digiriik-ja-uhenduvus/kuberturvalisus/valimiste-kuberturvalisus (visited on Apr. 25, 2022).

[24] Estonian Minister of Foreign Trade and Information Technology. *E-valimiste töörühma moodustamine*. 1.1-1/19-100 (in Estonian). Estonian Ministry of Economic Affairs and Communications, June 21, 2019. URL: https://adr.rik.ee/mkm/dokument/12268766 (visited on May 2, 2022).

[25] E valimiste turvalisuse töörühm. *E-valimiste turvalisuse töörühma koondaruanne*. (in Estonian). 2019. URL: https://docplayer.ee/203160304-E-valimiste-turvalisuse-t%C3%B6%C3%B6r%C3%BChma-koondaruanne-e-valimiste-turvalisuse-t%C3%B6%C3%B6r%C3%BChm.html (visited on Apr. 25, 2022).

[26] *Riigikogu liikmed moodustasid e-hääletamise läbipaistvaks muutmise toetusrühma*. (in Estonian). Parliament of Estonia Press. Apr. 21, 2020. URL: https://www.riigikogu.ee/pressiteated/muu-pressiteade-et/riigikogu-liikmed-moodustasid-e-haaletamise-labipaistvaks-muutmise-toetusruhma/ (visited on Apr. 25, 2022).

[27] Sven Heiberg, Kristjan Krips, and Jan Willemson. "Planning the next steps for Estonian Internet voting". In: *E-Vote-ID 2020*. TalTech Press Proceedings. 2020, pp. 82–97.

[28] Bingsheng Zhang, Zengpeng Li, and Jan Willemson. “UC Modelling and Security Analysis of the Estonian IVXV Internet Voting System”. In: *ArXiv* abs/2109.01994 (2021).

[29] Olivier Pereira. “Individual Verifiability and Revoting in the Estonian Internet Voting System”. In: *Cryptology ePrint Archive* Report 2021/1098 (2021).

[30] Johannes Mueller. “Breaking and Fixing Vote Privacy of the Estonian E-Voting Protocol IVXV”. In: *7th Workshop on Advances in Secure Electronic Voting* (2022).

[31] *BDOC - Format for Digital Signatures*. 2014. Estonian Centre for Standardisation and Accreditation: EVS821:2014. URL: https://www.evs.ee/en/evs-821-2014.

[32] “Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC”. In: *OJ* L 257 (2014-08-28), pp. 73–114.

[33] Ralf Kusters, Tomasz Truderung, and Andreas Vogt. “Clash Attacks on the Verifiability of E-Voting Systems”. In: *2012 IEEE Symposium on Security and Privacy*. 2012, pp. 395–409. DOI: 10.1109/SP.2012.32.

[34] Riigikogu. *Riigikogu Election Act*. RT I, 03.01.2020, 13 (English translation). Riigi Teataja, 2020. URL: https://www.riigiteataja.ee/en/eli/514122020002/.

[35] P. Parmakson. private communication. Apr. 2022.

[36] Priit Parmakson. *VIS3-EHS liideste spetsifikatsioonid*. github.com. URL: https://github.com/e-gov/VIS3-EHS (visited on Apr. 26, 2022).

[37] Priit Parmakson. *E-hääletanute nimekiri*. github.com. URL: https://github.com/e-gov/VIS3-EHS/blob/main/4_e_haaletanute_nimekiri/SPEC.md (visited on Apr. 26, 2022).

[38] Priit Parmakson. *5 Tühistus- ja ennistusnimekiri*. github.com. URL: https://github.com/e-gov/VIS3-EHS/blob/main/5_Tyhistusnimekiri/SPEC.md (visited on Apr. 26, 2022).

[39] A. Koitmäe. private communication. Apr. 2022.

[40] Riigikogu. *Euroopa Parlamendi valimise seaduse ja teiste seaduste muutmise seadus*. RT I, 09.07.2018, 1 (in Estonian). First point of §§1, 3 and 4 and second point of §2. Riigi Teataja, 2018. URL: https://www.riigiteataja.ee/akt/109072018001.

[41] David Chaum and Eugène van Heyst. “Group Signatures”. In: *Advances in Cryptology — EUROCRYPT ’91*. 1991, pp. 257–265. DOI: 10.1007/3-540-46416-6_22.

[42] Ronald L. Rivest, Adi Shamir, and Yael Tauman. “How to Leak a Secret”. In: *Advances in Cryptology — ASIACRYPT 2001*. 2001, pp. 552–565. DOI: 10.1007/3-540-45682-1_32.

[43] Matt Blaze, Gerrit Bleumer, and Martin Strauss. “Divertible protocols and atomic proxy cryptography”. In: *Advances in Cryptology — EUROCRYPT’98*. 1998, pp. 127–144. DOI: 10.1007/BFb0054122.

Appendix 1 – Non-exclusive License for Reproduction and Publication of a Graduation Thesis[1]

I Taaniel Kraavi

1. Grant Tallinn University of Technology free license (non-exclusive license) for my thesis ”Improving Ballot Privacy in the Estonian Internet Voting System”, supervised by Ahto Buldas
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive license.
3. I confirm that granting the non-exclusive license does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

16.05.2022

[1]The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 – Re-Signing Verification Service

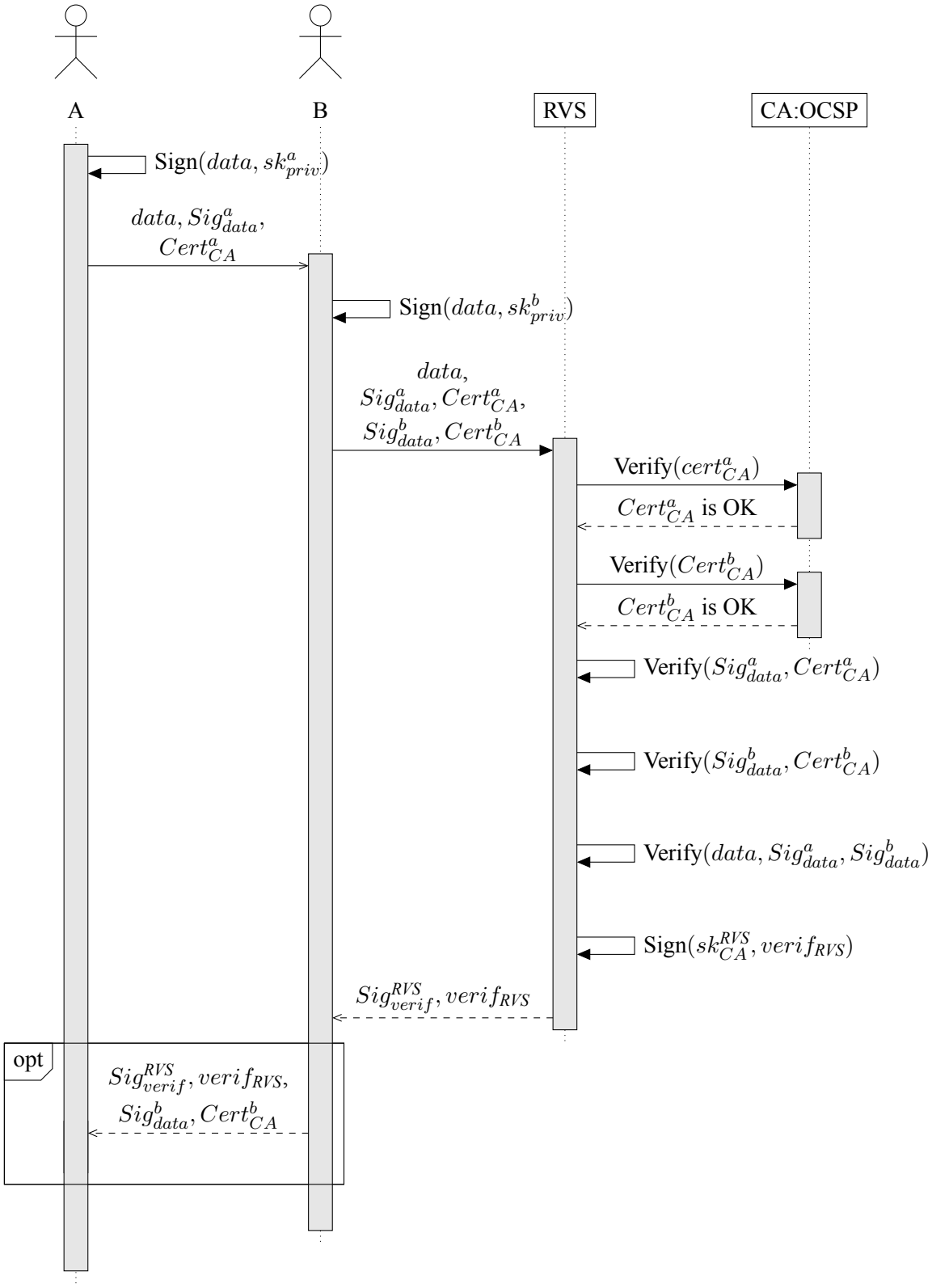

Figure 6. A successful re-signing procedure.

Appendix 3 – BDOC Signature Component

The following samples of XML blocks from a BDOC signature are taken from [31, Annex A].

A sample `ds:SignedInfo` XML block.

```
<ds:SignedInfo>
  <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2006/12/xml-
     c14n11"/>
  <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more
     #rsa-sha224"/>
  <ds:Reference Id="S0-RefId0" URI="document.doc">
    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256
       "/>
    <ds:DigestValue>5UyKB9ht94y6CZNvLdO1C7Z3MXaYc2Qol3Dt3Qp4Ajg=
    </ds:DigestValue>
  </ds:Reference>
  <ds:Reference Id="S0-RefId1" Type="http://uri.etsi.org/01903#
     SignedProperties" URI="#S0-SignedProperties">
    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256
       "/>
    <ds:DigestValue>YGDmd4GaWLgV4/hrEVV6/DvQ6uLhfnTSI0CQJX612KM=
    </ds:DigestValue>
  </ds:Reference>
</ds:SignedInfo>
```

`ds:SignatureValue` represents the cryptographic signature component and is given to the hash of `ds:SignedInfo`.

```
<ds:SignatureValue Id="S0-SIG">
YQs06u9ekMnZd2Jy+Won5VK0kIC9y5e2JPfraUItZOqwxd4rc4g3fiUnDkrf
iHIdD2xOGyszCZA/JAicqDPiFkmXbjkgpYYF8gY3NB/xFwoKv/zaWu7HEi+T
eq/OoSDlXVGi0H++27nI3xAl7P7Iz84xaji1aquZQVl5iOtWD8k=
</ds:SignatureValue>
```