TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Software Science

ITC70LT
Tarmo Oja 182499IVCM

X-ROAD TRUST MODEL AND TECHNOLOGY THREAT ANALYSIS

Master's Thesis

Supervisor: Ahto Buldas
PhD

Co-Supervisor: Mari Seeba
MSc

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutisüsteemide instituut

ITC70LT

Tarmo Oja 182499IVCM

X-ROAD USALDUSMUDEL JA TEHNOLOOGILISTE OHTUDE ANALÜÜS

Magistritöö

Juhendaja:	Ahto Buldas
	PhD
Kaasjuhendaja:	Mari Seeba
	MSc

Tallinn 2020

Author's declaration of originality

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

Author: Tarmo Oja

2020-08-04

Acknowledgments

I wish to express my appreciation to my supervisor, Professor Ahto Buldas, who introduced me and clarified some dark corners of the research and its objectives with some simple and downright principles and remarks.

Co-supervisor Mrs Mari Seeba for helping drill down to essence of the models, showing the World behind words and her invaluable consistent drive to complete my thesis writing and moving towards graduation.

I wrote my thesis being part of a amazing team of Cybernetica AS. I am grateful for opportunity to do so and supporting me with the reviews and insights.

Mr Arne Ansper, one of the X-Road architects and founders of the Cybernetica AS, providing background information and validating the results.

I would also like to thank Information System Authority (*RIA*) *X-tee* and CERT-EE teams for validating the findings and providing necessary feedback.

My utmost gratitude for patience belongs to my daughter Mari, who has suffered most from my endless late hours studying.

Abstract

The X-Road technology and framework is used to provide secure interoperability functionality between organizations and information systems. The concepts and requirements have evolved since 2001 when the first version of X-Road (Estonian *X-Tee*) was developed in Estonia. With international adoption and its constant usage growth the need to explain the concept and analyze security properties has been growing. This thesis provides comprehensive overview how and why X-Road has evolved from early years, what were the related requirements and changes. The X-Road trust relationship model is introduced to provide abstract overview of relations between participants and external providers. Different threat modeling techniques are used to provide systematized information about current X-Road components, protocols, asset threat profiles. The method how to use provided layered model information for identifying potential weaknesses is outlined. Validation uses Estonian deployment *X-tee* and current X-Road version artifacts for identifying potential weaknesses using systematized information composed and analysis method proposed. As the result of conducted threat analysis, the Estonian Information System Authority (RIA) has already started risk reduction actions. Summary of identified issues is demonstrated. The work is a good starting point for further, deeper research on the X-Road. Provided content can be used for further security analysis, to evolve requirements or for identifying additional research topics on X-Road. Results are also usable as a informational material for explaining X-Road properties. Methods used to decompose and provide information on X-Road elements can be used for other cyber domain sophisticated systems as well.

This thesis is written in English and is 62 pages long, including 11 chapters, 10 figures, and 8 tables.

Annotatsioon
X-Road usaldusmudel ja tehnoloogiliste ohutude analüüs

Tehnoloogia ja raamistik X-Road on kasutusel organisatsioonide ja infosüsteemide vahelise turvalise ristkasutuse võimaldamiseks. X-Road (Eesti X-tee) esimene versioon loodi aastal 2001. Pideva kasutajaskonna kasvu ja rahvusvahelise kasutuselevõtuga seoses on vajalik X-Road põhimõtteid aina rohkem tutvustada ja analüüsida, sh turvalisuse seisukohalt. Lõputöö pakub põhjaliku ülevaate, kuidas ja mis põhjustel X-Road on arenenud, millised on seotud nõuded ja muutused. Töö esitab X-Road kokkuvõtva usaldusmudeli. Mudel kirjeldab raamistiku osapoolte suhteid ja sõltuvusi. Usaldusmudelile täiendavalt esitatakse ohu mudeldamise meetoditega kokkuvõtlik, süstematiseeritud informatsioon praegustest X-Road seostest, protokollidest ning varade ohuprofiilidest. Eelneva põhjal pakutakse välja meetod info kasutamiseks võimalike nõrkuste tuvastamiseks. Nõrkuste tuvastamise meetodi valideerimiseks kasutati loodud struktureeritud informatisooni ja metoodikat X-Road ja Eesti X-tee võimalike nõrkuste kaardistamiseks. Kaardistuse tulemusena on Eesti X-tee haldur Riigi Infosüsteemi Amet (RIA) plaaninud tegevusi, mis vähendavad nõrkuste võimalikku mõju. Töös sisaldub tuvastatud leidude kokkuvõte. Töö tulemeid saab kasutada X-Road nõuete täiendamiseks ja arendamiseks. Loodud sisu on kasutatav ka koolituste ning tutvustavate seminaride ülevaatlike materjalidena. Loodud tehised on hea alguspunkt edasisteks uurimistöödeks ja analüüsideks. Esitatud meetodid on rakendatavad ka teiste kübervaldkonna keerukate süsteemide informatsiooni süstematiseerimiseks ja analüüsimiseks.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 62 leheküljel, 11 peatükki, 10 joonist, 8 tabelit.

List of abbreviations and terms

ACL Access Control List

AES Advanced Encryption Standard

CA Certificate Authority

Central Server (CS) X-Road Central Server manages and distributes global configuration and its changing requests

CVE Common Vulnerabilities and Exposures

Global Configuration (GC) Global Configuration consists of X-Road instance member/server identity mappings, trusted providers, etc, used by all instance participants

Governing Authority (GA) Governing Authority. Running central components: Cenrtal Server, management security server, central monitoring

HSM Hardware Security Module

Member X-Road member registered or to-be-registered on instance.

NIIS Nordic Institute of Interoperability Solutions

OCSP Online Certificate Status Protocol

PKI Public Key Infrastructure

RIA Estonian Information System Authority (RIA)

SDSB Secure Distributed Service Bus. R&D project led by Cybernetica AS to modernize X-Road 2011-2014. Later renamed to Unified eXchange Platform(UXP). Basis of X-Road version 6.

Security Server (SS) X-Road Security Server mediates messages between organizations based on information included in Global Configuration. Provides confidentiality, integrity, non-repudiation qualities for the exchanged messages

Software Provider (SP) X-Road software provider/developer

SSCD Secure Signature Creation Device

STRIDE-LM Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privileges, Lateral Movement

Subsystem Logical unit within organization registered on X-Road instance as separate identifier. Frequently represents one information system or logical role.

TLS Transport Layer Security

Trust Authority (TA) General identifier for all required trust services: CA, OCSP, TSA

TSA Timestamping Authority

UXP Unified eXchange Platform. Technology and product family implementing X-Road protocols by Cybernetica AS.

X-tee X-Road instance in Estonia. Project name for modernizing Estonian state databases 2000-2001.

X-Road Intra-organization interoperability technology

Table of Contents

List of Figures

List of Tables

1 Introduction

X-Road (original name *X-tee* in Estonian) distributed architecture concept has been used since 2001 to establish secure connectivity between Estonian Government institutions, private companies. During this period X-Road has become the most used backbone technology in Estonia. Usage of X-Road for governmental institutions in Estonia is mandatory as stipulated by Public Information Act[Avt].

More than 1.3 billion transactions were performed during year 2019 alone[1]. This averages 1000 request per one resident (total: 1.3 million[2]) of Estonia per year. More than 600 organizations are relying on X-Road technology to establish secure, trusted communications.

According to the X-Road world map[3] there are 34 countries which are using or at least have evaluated X-Road or Cybernetica's amplified version – UXP[4], technology. The number of unique installations is definitely larger. Active users maintain more than one environment (e.g development and staging environments), also there are private/planned installations which are not announced.

The biggest and the most dedicated adopter of the X-Road technology outside Estonia has been the Finnish Government. Governments of Estonia and Finland established a non-profit Nordic Institute for Interoperability Solutions (NIIS)[5] to coordinate X-Road joint development and popularization.

Together with X-Road usage in critical information exchanges, popularity and wide-spread recognition the systems will be targeted by adversaries. With increase in number of installations the probability of human error, targeted attack, manipulation attempt will increase exponentially. Besides direct damage to data exchange parties and data subjects, each such event will effect the general image of X-Road and through its origin – Estonian image as well.

This thesis introduces X-Road trust dependency model/graph which can be used for analyzing entity relationships and select focus for further research.

Current work systematizes and analyzes X-Road most current version 6 main assets, core protocols and components. Based on analysis and empirical observations vulnerabilities and possible weaknesses are identified. Due to limited time and scope of the thesis this is

[1] `https://x-tee.ee/factsheets/EE/`

[2] `https://www.stat.ee/pressiteade-2020-007`

[3] `https://x-road.global/xroad-world-map`

[4] `https://cyber.ee/uxp`

[5] `https://niis.org`

not a full security audit and does not provide possible solutions for identified issues.

Estonian X-Road deployment, *X-Tee*, and Estonia specific configuration is used as reference, as it is the oldest and has the highest usage. Finnish specifics are referred when needed.

1.1 Motivation

X-Road connectivity has been integrated to critical information systems, directly or indirectly. Handled data and connected registries, information systems contain critical and sensitive information. Lack of availability, confidentiality or integrity may result in inconvenience for business processes, leaking private information or distrust in collected data and systems as such. Lateral movement should be considered as a possible attack method also.

Number of governments, municipalities, private companies have recognized X-Road as a technology which enables to break down data/process silos. Requiring automated processes with better turnaround times provides services to citizens, partners and clients supported by secure data exchange.

While legal consequences may impact organizations registered on production environment as each exchanged message carries legally binding value, the security servers, regardless of the environment they're registered on, are connected to organization internal networks and information systems.

Any vulnerability on X-Road technology or breach in deployment may affect more than one organization. Understanding X-Road requirements and its relationships external environment is crucial to mitigate current risks and design improvements for future releases.

The Author of this thesis has been working at Cybernetica AS as deployment engineer since 2013, responsible for supporting SDSB/X-Road/UXP in-house software development process. Also consulting and providing expert support to RIA and other Estonian government institutions during the process of deploying and rolling out X-Road v6 from development to test and production environment.

1.2 Research Problem

There is lack of systematized security analysis which covers intra-organization communication distinctive feature for X-Road. The X-Road documentation repository[1] contains general description of the architecture, use case models and protocol specifications.

Analyzing X-Road is a complex task due to its extent. Complexity of the X-Road model can be summarized as follows:

- X-Road spans and connects multiple organizations.
- Usage of sophisticated protocols – protocol flow spans different trust domains (crossing component and organization boundaries) and are loaded with assumptions and dependencies.
- Exchanged messages must have long-term evidentiary property while providing availability and confidentiality qualities in transit.
- Third-Party trusts – software provider, governing authority, trust (PKI) service providers.

General research problem is the systematization and representation of the X-Road information for threat analysis and function evaluation purposes.

Research Questions

1. How to systematize trust relationships in X-Road architecture?
   (a) Who are the main actors of the X-Road environment?
   (b) What are the critical trust relations between actors?
2. How to analyze X-Road multimessage, multiparty complex trust relationship models efficiently?
   (a) How to represent different information reusable, renewable and graspable manner?
   (b) How to use resulting systematized information for threat analysis?
3. What weaknesses can be identified from X-Road using created trust relationship descriptions?
   (a) What is the current security posture of the X-Road instances in Estonia?

[1] `https://github.com/nordic-institute/X-Road/tree/develop/doc`

1.3 Research Method

This research aims to build knowledge, propose information systematization formats and analysis method for complex multiparty systems like X-Road. Results of the thesis can be used as input for further research, improvement on X-Road state or designing new properties. Thus the methodology has characteristics of design science. [VK04]

1.4 Contribution

This research contributes to the scientific world by proposing an approach to decompose sophisticated systems using different methods to systematize and present information for threat analysis.

Additionally, strategic know-how is provided about X-Road technology, evolving the requirements through timeline and implementation. Significant conclusions on Estonian X-Road instance *X-tee* are made as result of validation.

1.5 Structure of the Thesis

Chapter 2 introduces the building blocks behind this thesis and describes information decomposition, different classifications used in the structured threat information. Chapter 4 describes actors and the trust relationship model used in the X-Road environment. Cross dependent relationships are briefly explained with references to relevant protocols. Chapter 3 provides detailed history of the evolving X-Road, milestones are described with explanation why and how it was changed. Chapter 5-7 organizes the information about X-Road technical solutions to be used in threat analysis. Information on main asset threat profiles, visual representation of protocols used and component internal data flows are demonstrated. General method using provided structured data in context to find potential weaknesses is provided in chapter 8. Chapter 9 summarizes threat analysis findings for the Estonian X-Road deployment and technology itself. In the final chapter the conclusion and possible future work is provided.

2 Theory

Thesis author's 20+ years of the experience on IT and IT security has taught – good security posture consists of three components: policy, process and technical measures. Lack of or over enforcing one or multiple components puts the system under stress and weaknesses may emerge opening way for vulnerabilities. Filling all the corners equally is very exhausting and frequently wasting of the resources. The goal is to find the balance between them.

Figure 1 provides graphical idea of the balanced policy-process-technical relationship. The concept is inspired from the project management triangle which is looking for quality balance in project's cost, scope and time.[Pmt]

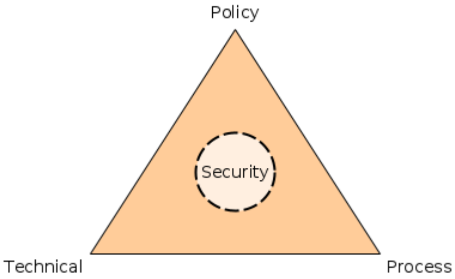


Figure 1. Balance of the security.

It is important to acknowledge that given balance components are not comparable in complexity nor volume. Main idea is to keep all components in relevant focus and benefit from complementing each other. In other words, to assess system security posture, all three angles must be considered.

The X-Road instance deployment, like every other system, must keep all 3 components under control to maintain good posture. The base architecture is highly technical which must be supported by other two components. Each instance must create policies, processes to have balanced posture.

Using the security balance categories, X-Road framework can be divided into different topics. For X-Road following items are identified for each component:

- Policy
  - Global level: general instance policy. Requirements, rules, governance, etc.
  - Identity: certificate and trust service policy.

  - Member level: data exchange/service usage policies, stipulations from legalization.
- Process
  - Global level: approving trust services, membership management.
  - Identity: certificate issuance and revoking processes.
  - Member level: authorizing data usage. Service Agreements.
- Technical
  - Software, components and environment.
  - Protocols and measures to create, transform, protect and exchange assets.

The thesis does not analyze the instance specific policies and processes due to limited scope and time. The focus is on the technical component. Some references to Estonian *X-tee* policies and processes are made when necessary and relevant.

2.1 Science and Threat Analysis

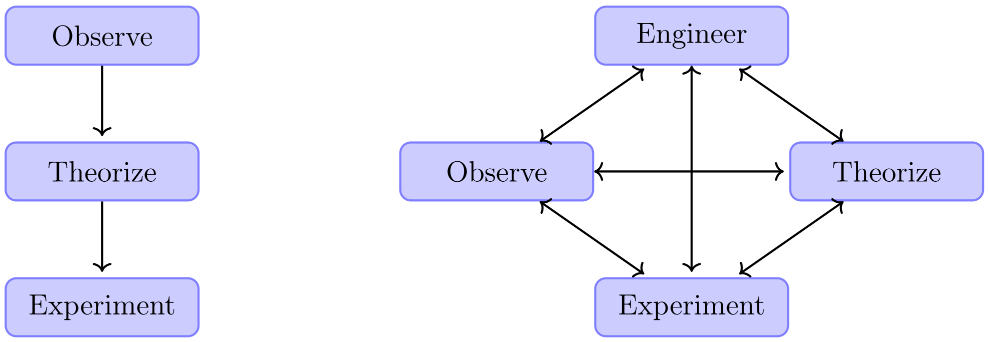
Threat analysis cannot be linear work, there is no "right" spot to start or follow same steps on every case. Each step needs to be verified back and forth before we can conclude the result.

We can see similarity with scientific methods. There is no single path or sequence to produce the result. Figure 2a shows the waterfall approach, which seems to be more streamlined but it lacks loopback and back-communication opportunities. This severely limits the quality of the output produced. To get most of research the ways to learn and refine knowledge must be accepted. Figure 2b shows a more "agile" approach where we can move between phases freely. It allows the researcher to prove or invalidate observations/theories in most efficient way [EM17, pp. 68–71].

To allow moving between methods it is required that information is organized in a way which allows to quickly grasp the essence of the object under research. Current work in major part focuses on finding methods to organize information about X-Road for efficient threat analysis of the platform.

2.2 Analyzing System

In such complex systems as X-Road it is not possible to grasp details, dependencies and risks in one go. It is necessary to systematize and make abstract models for repetitive



(a) How research is taught.

(b) How research is in the real World.

Figure 2. Idealized vs Actual Research Progression[EM17, p. 68].

reviews. Models or descriptions should not be overloaded with details and must be understandable without a deep theoretical background. Each model and representation should focus communicating the most essential, viable information. Otherwise the models are unusable for explaining or reviewing system requirements and properties.

Trying or forcing to fit everything from a system into the one model or method counteracts with the prior essential requirement.

One way to decompose the sophisticated system is to split it to logical views. Idea is to move from general to more detailed one as each system works in some larger environment. Each outer system provides the context, limitations and expectations to inner system under question.

From experience the author proposes layered decomposition of the information about systems for threat analysis:

Environment The surrounding legalization, policies and general rules.

Trust model Who are the actors and what are their (trust) relationships?

Threat profile What can happen to assets the actors are relying on?

Protocol How the assets are handled through larger trust domain boundaries. What are the critical inputs?

Component flow How component handles the assets and protocols? Sub-component boundaries and assumptions.

Implementation The actual implementation, product itself.

This thesis does not cover the first (Environment) nor the last layer (Implementation) information representations. In general they are observable in written documentation or implementation artifacts.

2.3 Threat Modeling and Classification

Threat modeling directs thinking about risks using abstractions [Sho14]. Chapter introduces threat modeling techniques used in thesis.

STRIDE is a threat modeling mnemonic to present threat categories for individual entities like systems, assets and interactions. It helps follow through different aspects of the element under review, and ask question 'what can go wrong'. Table 1 lists the threat categories and protected properties [MF19; Sho14, pp. 61–86]. The original STRIDE is augmented with an additional threat category Lateral Movement, as proposed in [MF19]. The lateral movement is definitely important in X-Road context as it connects different systems and trust domains.

Table 1. STRIDE-LM threat categories and properties[MF19].

STRIDE-LM	**Threat**	**Protected Property**
S	Spoofing	Authenticity
T	Tampering	Integrity
R	Repudiation	Non-Repudiation
I	Information Disclosure	Confidentiality
D	Denial of Service	Availability
E	Elevation of Privilege	Authorization
LM	Lateral Movement	Segmentation

To distinguish root causes and possible mitigation points threats are separated into three issue categories. These categories are derived from Microsoft's Security Development Lifecycle (SDL) SD3[Mss] principle for designing product security:

- Secure by Design;
- Secure by Default;
- Secure by Deployment.

Transforming security balance provided on Figure 1 with SD3 principles we get an idea of the balanced design-default-development relationship shown on Figure 3.

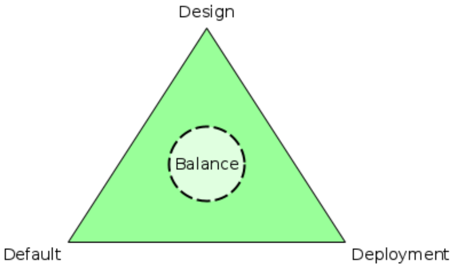


Figure 3. SD3 component balance.

Failure to secure the system from one or multiple categories we might open a vector for weaknesses leading to vulnerabilities.

2.4 Threat Actor Classification

Important part is to understand adversary classifications – each threat actor class has its own capabilities, intents. Threat classes are used while evaluating assets and potential vulnerabilities. The possible motivations are added for reference. These descriptions may vary on the environment where X-Road is used.

Government-Sponsored/APT Government run/used services are always in focus for government-level adversaries. Intent can include: data collection, system manipulation for gaining access to data, discrediting government or stability.
Sophisticated attacks, advanced and persistent attackers may use X-Road as a lateral movement gateway to/from internal systems in the same or connected organizations.
Infected internal systems or social engineering of the staff may be used to manipulate or gain access to X-Road components.

Insiders Due to mandated privileges or collected knowledge the insiders may use X-Road to collect and manipulate data in the same or other organizations.
Critical are the insiders of the high trust actors – software provider, trust service provider, governing authority.

Cyber Terrorists/Hacktivists/Cybercriminals The main intent could be disrupting or discrediting the systems. Collected data may be used for identity theft or extortion purposes.
Criminals are focused on the data collection or manipulation for profit.

Script Kiddies Using widespread easy-to-use tools script kiddies may gain access to

components via un-patched environment or software. Also they may unintentionally get access or collect resources from X-Road administrator workstation.
Probability is higher on non-maintained/out-dated components and may be benefit from mistakes on environment (firewalls, access rules, etc) configuration.

Internal User Errors Not classic threat actor but hazard, there is no intent involved. Unintended deviations from system or environment configuration, procedures or policies. On threat modeling involves tightly with 'Secure by Default' principle. This class can contribute to success of other threat actor classes.

3 Related Work: History and Evolution of the X-Road

Evolution of the distributed inter-organizational data exchange system X-Road has been long and constantly evolving. To provide reasoning and details about design decisions made, extensive overview of the history and processes is provided.

3.1 The Need and Way to X-Road Principles

After regaining independence Estonia needed to build up organizations and processes to govern the republic. Information technology was desirable and a modern way to support state processes. Ambitions were driven by young specialists and lack of money.

Notable pre-conditions and milestones which contributed emerging X-Road and its services during this time were:

1989 Personal Code[1] Personal unique identifier was introduced – helping cross-reference different data sets.

1996 Personal Data Protection Act[2] Stipulated right to request information about him/herself. Requirement for consents, right to be forgotten, etc

1997 Databases Act[3] How governmental institutions may create, manage, use and share data sets.

2000 Digital Signatures Act[4] Foundation for using digital signatures, stipulated that digital signatures are legally equal to handwritten ones.

2000 Public Information Act[5] Right to access public information.

2001 Identity Documents Act[6] Updated to add digital keys and certificates to ID-card

It is clear that the lack of state owned legacy systems provided a good starting point to select emerging technologies. It was not necessary to spend resources to migrate old systems but just align new systems with up-to-date principles at that time. Co-operation and reusing existing resources was also in the focus. Fine example is commercial banks

[1] https://et.wikipedia.org/wiki/Isikukood

[2] https://www.riigiteataja.ee/akt/862756

[3] https://www.riigiteataja.ee/akt/32230

[4] https://www.riigiteataja.ee/akt/71878

[5] https://www.riigiteataja.ee/akt/26643

[6] https://www.riigiteataja.ee/akt/73019

which already provided internet based banking solutions and its authentication mechanisms were used later for *X-tee* citizen portal.

By the year of 2000 the project for modernizing state databases, document exchange was initiated by the Department of State Information Systems. Descriptive document included short vision to describe the goals.

> ***Program to modernize state databases (X-tee). Vision to 2003.***[Ris]
>
> *State has access to databases acting as a integral whole 7 days a week and 24 hours a day, which ensures:*
>
> - ***Citizen*** *receives and provides information within the law;*
> - ***Civil servant*** *can use state databases for decision making within their mandate;*
> - ***Entrepreneur*** *can use information from state databases for business procedures within their mandate;*
> - ***State*** *has become more transparent, consistent and understandable for citizens;*
> - *Using* ***unified databases*** *the state administrative capacity has been improved and resources needed for management reduced. Usage of the databases is improved by using homogenized user interfaces.*

One of the pilot projects for achieving interconnections between information systems proposed a central service layer (Estonian: *teeninduskiht*) which was implemented and demonstrated in the second half of 2000. The solution was exchanging XML-RPC messages through a central dispatching server. For evaluation purposes code samples and example interfaces were published. Services allowed to request information about validity of the passport and public information on vehicles from respective registries. [Kad+00]

XML usage was proposed for the service layer and document management systems due to simplicity, available libraries for different programming languages, independence from platform, possibility to secure with standard protocols and components (HTTP/web servers) [Tam00b; Tam00a]. XML-RPC was selected over SOAP due to maturity of the earlier one[Xrh].

Cybernetica AS was tasked to analyze *X-tee* initiative goals and program status in early 2001. The goals were to assess objective feasibility and required changes in legal situation, provide principles how to implement security functions for databases and their connections.

Additionally the authentication methods and possible protocols/interface properties were assessed against security requirements. An important part of the report focused analyzing the interoperability vision and improving it to meet security requirements. Principles laid out in report and later implemented in X-Road were[Ans+01a]:

- Use distributed approach instead of centralized one. Dispatcher systems impose risks for availability, scalability, confidentiality. Using central directory services and direct data exchange was proposed.
- Solution must not contain a single point of failure/bottleneck. Data exchange should not be interrupted if the directory service is unavailable for some period of time. Internet facing data exchange services should have protection from denial of service attacks.
- Service usage agreements are between organizations (bilateral agreement between provider and client), access of the civil servant to remote database must be controlled by service client organization internal procedures/mechanisms. Communicating all personnel changes to all service providers is not feasible.
- Each transaction must be logged for auditing and evidential value (for resolving disputes and deterring effect) using verifiable method. Transactions are signed with server keys not end-user keys.
- Message exchange over public/external networks must be encrypted.
- Addressing must not use handlers which may change (e.g IP-addresses). The end-point should not deal with 'real' location of the counterpart.

In parallel, a strategic plan to introduce digital signatures in state institutions was compiled [Ans+01b]. The report analyses all important state initiatives connected to usage of digital signatures. Gap analysis and action plan for legalization, technical and process matters was provided. *X-tee* initiative was seen as a good corner stone of the complementary structure using and surrounding digital signatures.

Wider picture on principles for developing e-state strategy together with the e-state architecture model can be found from Arne Ansper master's thesis E-State From a Data Security Perspective [Ans01].

Next phase was to provide working a interoperability solution.

3.2 X-Road Implementation

Immediately after analysis detailing of the proposed principles begun. It resulted in an architecture which was the basis for the procurement to develop X-Road first version. Figure 4 shows original system architecture and protocols.[Ans+01a]. All the principles and component roles are used for modern X-Road as well.

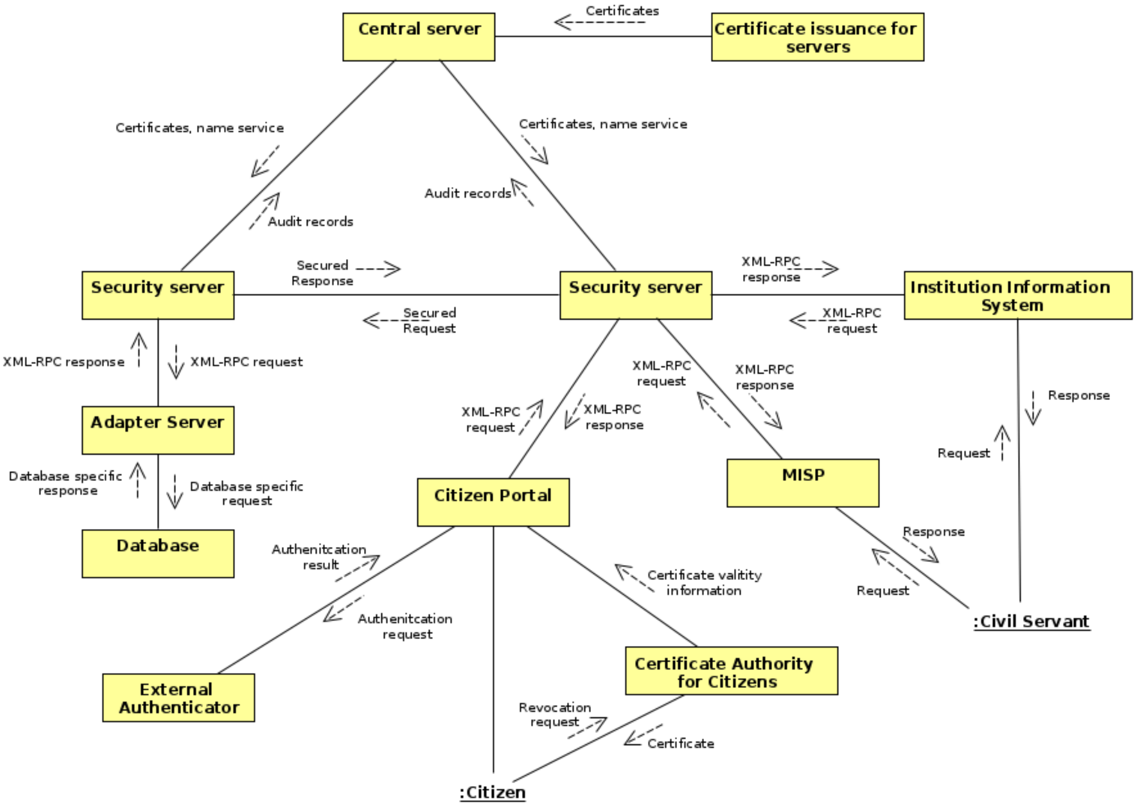


Figure 4. Specified X-Road protocols and system architecture[Ans+01a, translated].

AS Assert managed consortium (Cybernetica AS – architecture, protocols; AS Andmevara – integration with the population register, Estonian Registry of Buildings; Reaalsüsteemide AS – integrations with the Commercial Register; AS Datel – integration with the Land Register;) won the public procurement. On 17th December 2001 X-Road 1.0 installation was handed over to state agencies, ready for piloting above mentioned registries [Xrh].

Rationales for the design decisions and technical description of the implemented X-Road solution with relevant background information were presented at the 19th Computer Security Applications Conference 2003 [Ans+03].

On first years of the *X-tee* each data cross-usage was required to have Data Protection Agency approval. It severely affected *X-tee* adoption and spread. The national information

Table 2. X-Road major versions and features[Kal+13].

Version	Year	Changes
1	2001	XML-RPC, DNSSEC-based directory service
2	2003	SOAP, WSDL
3	2004	Asyncronous queries
4	2006	Log encryption
5	2010	WSDL Document/Literal, upgrading used crypto primitives, Data Encoding service

security baseline (ISKE)[1] was established 2003-2004. Common and comparable security levels allowed asses the database requirements without extra work.

The following years did not change the core principles of the X-Road. Table 2 provides the information about X-Road major versions and the notable improvements. [Kal+13] X-Road core component development and maintenance was carried out by Cybernetica AS.

3.3 Modernizing Principles

Continuously working on development and improvement of the X-Road and seeing its rapid growth, Cybernetica's researches started to look into international deployment options and working with the X-Road federation idea. Three different possible solutions were discussed: (a) using higher level instance as root of trust to connect existing instances; (b) parallel usage of separate national and international instances; (c) using bilateral agreements between central agencies to trust each other infrastructure [AW06; WA08].

Using experiences from the X-Road deployment projects outside of Estonia, Cybernetica AS initiated internal analysis and research project mid 2010. Internal documents[Ans10] proposed the following improvements:

- Using external PKI services (X-Road used specific, private certificate issuing up to version 5);
- Using standard time stamping services (The messages were chain linked and chain checkpoints were submitted to central audit server);
- Improve signature structure to be compatible with emerged standards, legal frameworks and common utilities;

[1] https://iske.ria.ee/

- Improving signature creation performance using batch signatures.

One of the drivers on focusing PKI and digital signatures was upcoming adoption of European Union digital signature directives due to Estonia joining EU in 2004.

Following years 2011-2012 the legal frameworks, international standards and technical resources were analyzed. One of the focus was enabling X-Road transactions with full legal digital signature power[Pwr]. The new X-Road vision was outlined, adding and refining requirements[Xte]:

- Digital signatures must follow Estonian and EU legalization. Usage of signature creation devices (SSCDs) must be supported.
- Using external Trust Service providers for certificates and timestamps.
- Improving fault-tolerance for central services (global configuration).
- Removing distinction between service provider and client (Up to version 5 it were different registration processes).
- One subsystem must be able to use multiple adapter servers (service endpoints).
- X-Road must be usable outside of Estonian legal and technical environment.
- Federation of independent X-Road instances. Supporting International federations.
- Analyze the service to monitor personal data usage.
- Analyze using other protocols next to existing SOAP protocol.

Cybernetica AS initiated a research and development project within ELIKO [1] Competence Centre in Electronics-, Info- and Communication Technologies to validate proposed architecture and develop Secure Distributed Service Bus (SDSB) first version. Input for validation and piloting environments were provided by Estonian Information System Authority (RIA) and Estonian eHealth Foundation (eTervis).[Sds]

It must be noted, that SDSB was complete software rewrite considering all requirements and foreseeable changes in legalization and environment.

Introducing new requirements initiated research activities to solve issues ahead resulting in multiple publications and theses:

- **Signature creation performance and fault-tolerance:**

[1] `http://eliko.ee`

  - Publication: Arne Ansper et al, Batch signatures High-Performance Qualified Digital Signatures for X-Road [Ans+13a]
  - Report: Margus Freudenthal, Using Batch Hashing for Signing and Time-Stamping [Fre13]
  - Report: Margus Freudenthal, Profile for High-Performance Digital Signatures [Fre17]
- **Designing trust federation:**
  - Report: Margus Freudenthal and Jan Willemson, Challenges of Federating National Data Access Infrastructures [FW17]
  - Thesis: Riin Saarmäe, Analysis of Configuration Management in Federated X-Road Systems [Saa15]
- **Analysing availability:**
  - Publication: Ansper et al, Protecting a Federated Database Infrastructure against Denial-of-Service Attacks [Ans+13b]

The development of pilot-ready solution was planned 2nd half of 2013. Objectives for the project code named *X-tee 5.5* or *X-tee* $5\frac{1}{2}$ (referring for being in half way from version 5 to version 6) were:

- Integrating SDSB and X-Road 5.0 versions to the same server so it can process both protocol families, translating from one to other if needed.
- Provide functional requirements and technical preparedness using fully qualified external TA services for certificates and timestamping.

By the end of the year 2013 the project was completed by Cybernetica AS and the piloting in development environment with Estonian institutions begin in January 2014.[Hanb]

Next major feature to be introduced for production was the federation. The development was finished by the end of 2014.[Hana]

From the 2014 the X-Road technology is shared with Finnish Government under agreement signed in 2013. After two years the source code was published in GitHub as open source which development coordination is governed by Estonian and Finnish state institutions. Notable change from this period since 2014 is the development model change – development is now coordinated by non-profit Nordic Institute for Interoperability Solutions (NIIS) founded by Estonia and Finnish government from 2018. [Xrh]

From functionality side, the X-Road version 6.21.0 introduced REST protocol support for messages [X pb].

3.4 X-Road Main Principles

X-Road current principles (as of 2020) can be found from X-Road architecture[X aa], with authors emphasis on security items:

1. *X-Road is decentralized – the **data exchange happens directly between organizations**. There are no intermediaries. If the two organizations have established secure connection, the continuous data exchange depends only on availability of the organizations and the network between them.*
2. *Ownership of data – X-Road does not change ownership of data. The data owner (service provider) controls who can access particular services.*
3. *Availability is a central concern – the protocols are designed so that there is **no single bottleneck** in the system. Additionally, **no component should become a single point of failure**.*
4. *All the messages processed by the X-Road are usable as **digital evidence**. The technical solution must comply with requirements for digital seals according to eIDAS [EIDAS]. This implies support for secure signature creation devices (SSCDs).*
5. *All the communication is implemented as SOAP or REST service calls. SOAP services are described using the WSDL language and REST services are described using the OPENAPI Specification v3.*
6. *Cross-border services – it is possible for an organization to invoke services provided by an organization belonging to a different instance of X-Road.*
7. *Encapsulating the security protocol – the **security measures and the security protocol are encapsulated in standard components**. The organizations are not required to implement security-related functionality for data exchange.*
8. *Standardization – X-Road aims to standardize the communication protocol between organizations. This enables the organizations to connect to any number of service providers without implementing additional protocols. X-Road core does not perform protocol and data conversion. If necessary, these conversions can be performed by the organization's information system.*
9. *No predetermined roles – once an organization has joined the X-Road infrastructure, it can act as both service client and service provider without having to perform any*

additional registration.

10. *Two-level authentication – X-Road core* ***handles authentication and access control on the organization level****. End-user authentication is performed by information system of the service client.*

3.5 Future principles

Future principles are yet to be discovered. Meanwhile the NIIS has published plans to update UI and usability and supporting newer operating systems.

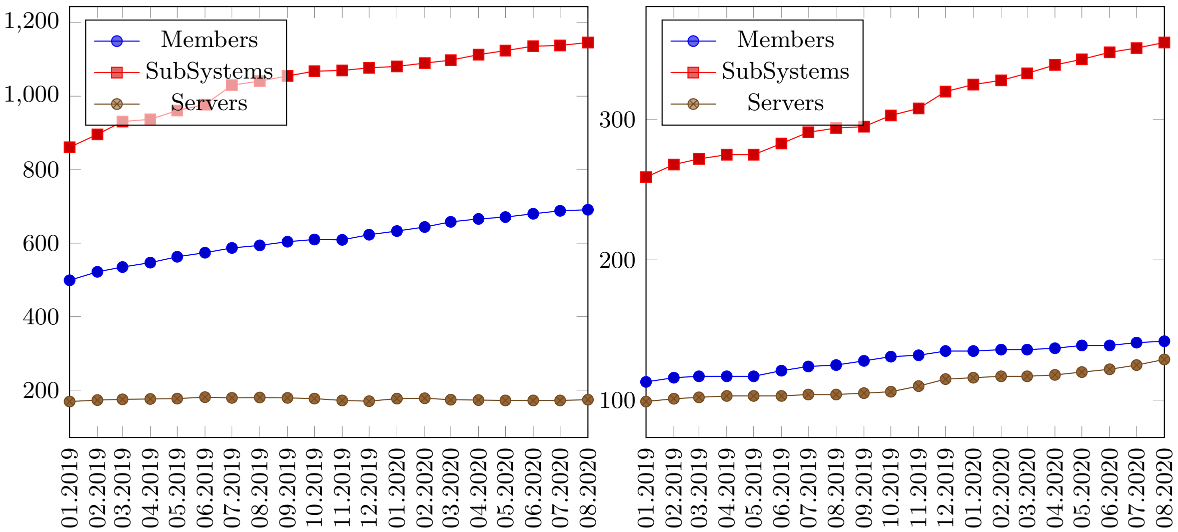
From research field Marten Kask defended Master's thesis Blockchain-based Members Management for the Unified eXchange Platform in June 2020 [Kas20]. Thesis provides analysis about requirements and possible solution to change the trust models on UXP platform to enable auditability.

3.6 Current Usage in Estonia and Finland

To assess the potential impact of the weakness or breach, number of impacted organizations and information systems can be used.

Statistics indicate that there are 833 (as of 01.08.2020) organizations with 1501 subsystems present on Estonian and Finnish production environment. Six environments have 821 security servers registered in total.[Sta]

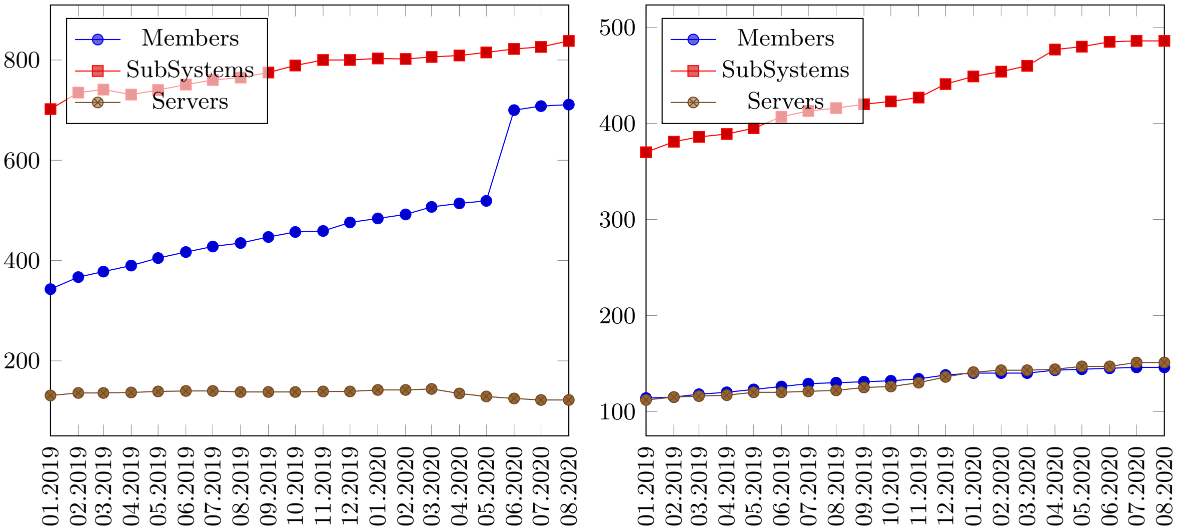
Figures 5a, 5b show Estonian and Finnish production environment statistics. Figures 5c, 5d and 5e, 5f show Estonian and Finnish test and development environment statistics respectively.



(a) Estonian production: EE.

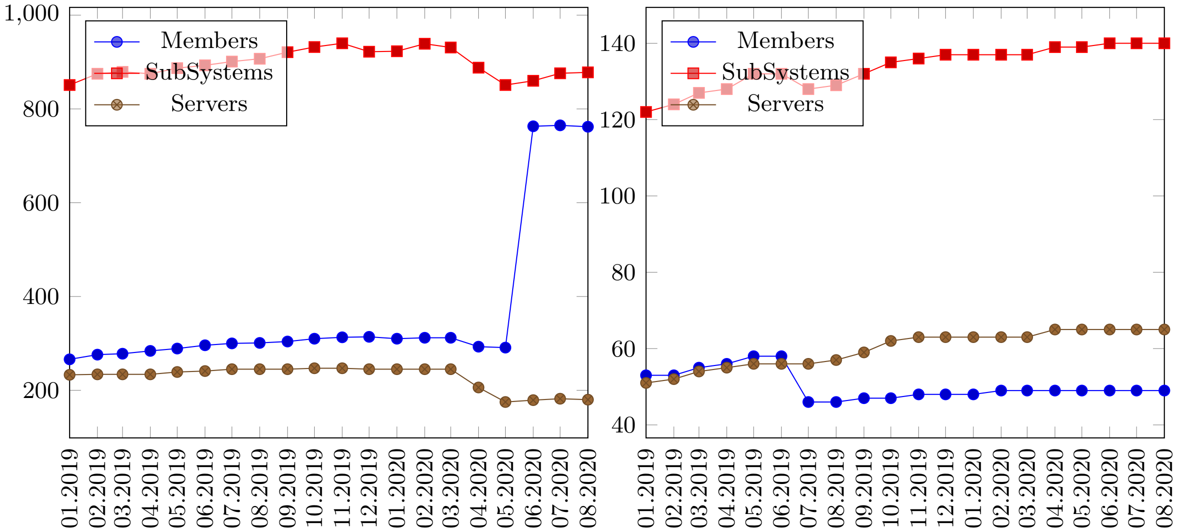
(b) Finnish production: FI.

Figure 5. Estonian and Finnish X-Road environment instance members, subsystems and servers[Sta].



(c) Estonian testing: ee-test.

(d) Finnish testing: FI-TEST.



(e) Estonian development: ee-dev.

(f) Finnish development FI-DEV.

Figure 5. (contd.) Estonian and Finnish X-Road environment instance members, subsystems and servers[Sta].

The statistics show constant growth and expansion of the X-Road within two main X-Road dependent countries. Additionally, X-Road usage is in expansion mode Internationally and security is a more critical topic than ever before.

4 Trust on X-Road

Analyzing X-Road architecture[X aa], relevant protocols and use-cases the trust dependency model was compiled. The chapter is presenting results of the analysis. Trust model and description of the cross dependencies can be used to assess potential weaknesses and breaches.

4.1 Actors

Following participants are present in X-Road trust dependency model:

GA **Governing Authority**. One per instance.
Acting as the root of trust, providing global configuration for all participants.
Providing general management services for the instance.

TA **Trust Authority**. One or few per instance.
Providing PKI services – certificates and their validity information, timestamping service.

SP **Software Provider**. One or few per instance.
Providing approved software for GA and Members

M1,M2 **Members**. Two to hundreds or thousands per instance.
Organizations which are joined the platform for data exchange.
Members have sub-dependents (which are not part of X-Road core):

- **IS** **Information System**, acting as a client/requestor. None to hundreds per member.
- **SE** **Service**, service provider. None to hundreds per member.

Governing Authority and Members form the core of the X-Road instance. Trust Authority and Software Provider are external dependencies.

IS and SE may belong to and/or managed by respective member itself or it might be different entity which is using Member as a service provider to access X-Road infrastructure services. It depends on the management model – is the security server managed by organization itself or if it is hosted/used as a service.

4.2 Trust Relationship Model

Figure 6 provides graphical representation of X-Road environment trust relationship model. The model is followed by numbered legend explaining each trust-dependency relationship. If specific protocol is used, it is referenced.

Graph-like representation was selected for its directed edges and readability. Usage of different line types and colors help grasp the essence and put important elements (like core actors) to focus. Numbered edges are reference to descriptive legend and also order the relationships in X-Road communication.

Reading the model:

- Blue – X-Road core actor. Running X-Road component(s).
- Black – external actor.
- Dashed line – one-time or seldom communication. Possibly human interaction at some level.
- Solid line – periodic, frequent automatic communication and excahge of assets.

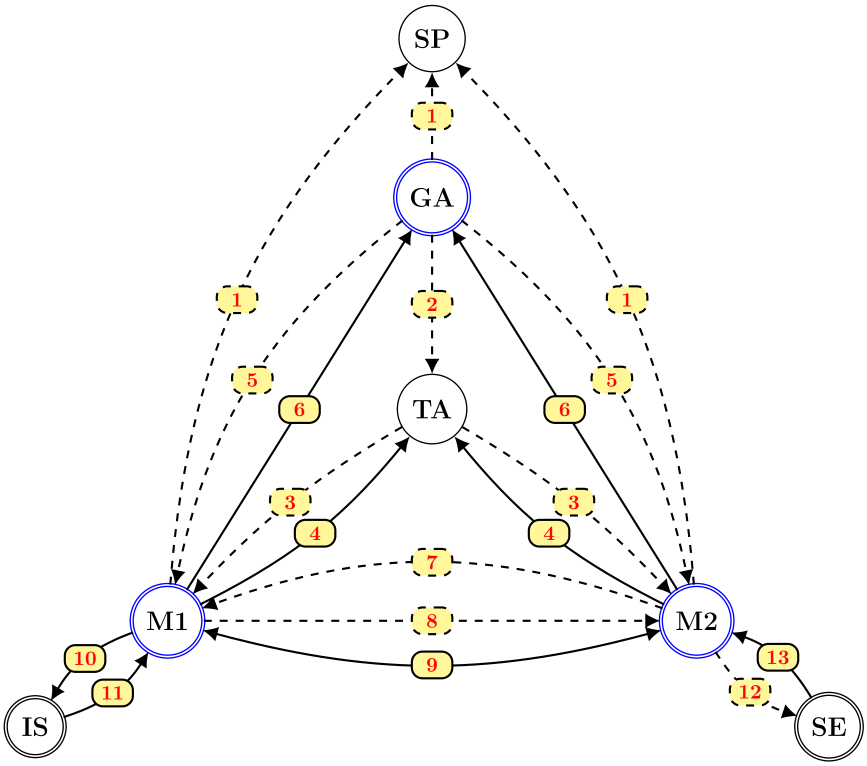


Figure 6. X-Road Trust Dependency Model.

IMPORTANT OBSERVATION

Described trust relationships are ordered transitive cross dependencies – a relationship uses prior one(s) as trusted computing base.

① Governing Authority (GA) and members M1, M2 are depending on Software Provider (SP):

(a) Developing and distributing software securely.

(b) Providing software updates.

Occurence Installing and updating application software

Result Installed components are secure and updated in timely manner

② GA depends on Trust Authority (TA):

(a) TA processes – accordance to regulations, audit reports, etc

Ref: eIDAS[1], CA/Browser Forum Baseline Requirements [2], TA Practice Statements

(b) Certificate Profiles for signing and authentication certificates

Ref: TA Practice Statements and Certificate/Service Policies

(c) OCSP endpoints, certificates and profiles

Ref: TA Validation Service Policy

(d) Timestamping endpoints, certificates and profiles

Ref: TA Time-stamping Service Policy

Occurence Before including TA information to Global Configuration (GC)

Result GA acknowledges CA/OCSP/TSA as authorized trust service provider. Information is included in Global Configuration.

③ TA depends on members M1 and M2:

(a) Providing correct information for issuing certificates.

Ref: CA regulations and policies, PKCS#10 CSR syntax RFC2986[3]

(b) Protecting private keys

Ref: TA Certificate Profiles

(c) Notifying TA about key breaches

Ref: TA Practice Statements and Certificate/Service Policies

[1] http://data.europa.eu/eli/reg/2014/910/oj

[2] https://cabforum.org/baseline-requirements-documents/

[3] https://tools.ietf.org/html/rfc2986

Occurence Before issuing certificates. When key breach is suspected.

Result TA issues certificates for Member. Provides correct OCSP/TSA service. Revokes certificates in timely manner.

4 Member M1 and M2 depends on TA:

(a) Issuing, revoking correct certificates to rightful owners

Ref: TA Practice Statements

(b) Providing correct OCSP responses

Ref: OCSP RFC6960[1], X-Road Arhitecture[X aa]

(c) Providing correct TSA responses

Ref: Timestamping RFC3161[2], X-Road Arhitecture[X aa]

Occurence Periodic

Result Certificates are issued/revoked correctly. OCSP responses are available for Member. Correct timestamp service available.

5 GA relies on members M1 and M2

(a) Providing correct information for registering member, subsystems, security servers

Ref: GA policies, X-Road Arhitecture[X aa]

(b) Providing correct information for subsystem <-> security server relations.

Ref: GA policies, X-Road Arhitecture[X aa]

Occurence Before adding/modifying/removing item.

Result Member relevant information is known to GA.

6 Member M1 and M2 relies on GA:

(a) Providing correct configuration anchor

Ref: GA policies, Global Configuration [X pa]

(b) Providing member registration/modification/removing management services.

Ref: X-Road Arhitecture[X aa], Global Configuration [X pa]

(c) Providing correct TA,member, global group, trusted federation, etc information

Ref: X-Road Arhitecture[X aa], Global Configuration [X pa]

Occurence Periodic. Seldom on management services

Result Correct Global Configuration is available.

[1] `https://tools.ietf.org/html/rfc6960`

[2] `https://tools.ietf.org/html/rfc3161`

7 Member M2 (service provider) relies on M1 (service client)

(a) Provide correct identification and information for agreement.

Ref: GA policies, bilateral agreement

(b) Provide correct information (client and service pair) for adding M1 to ACL

Ref: Bilateral agreement

(c) Provide information when M1 should be revoked from ACL.

Ref: Bilateral agreement

Occurence Before adding/removing M1 to/from service ACL

Result Access rights are granted/revoked.

8 Members M1 (service client) relies on M2 (service provider)

(a) Provide correct identification and information for agreement.

Ref: GA policies, bilateral agreement

(b) Provide correct WSDL/OpenAPI description

Ref: GA policies, bilateral agreement, Service Metadata Protocol [X pe], Web Services Description Language (WSDL), OpenAPI [1]

Occurence Before implementing client side.

Result Service message structure is correct.

9 Member M1 and M2 relaying on each other

(a) Message request and responses are following the protocols.

Ref: Message Transport Protocol [X pd], Message Protocol [X pc], Message Protocol for REST [X pb], X-Road Architecture[X aa]

(b) Follow the SLA and data-transmission bi-lateral agreement.

Ref: Instance policies, bi-lateral agreement

Occurence Each message exchanged.

Result Message exchange is possible. Terms of agreement are followed.

10 Member M1 is relaying Information System (IS)

(a) Message request are well formed.

Ref: Message Protocol [X pc], Message Protocol for REST [X pb]

(b) Message request are correctly addressed.

Occurence Each message exchanged.

[1] `https://swagger.io/specification/`

Result Message exchange is possible.

(11) Information System (IS) is relaying on Member M1

(a) Message is transported to correct service

(b) Message confidentiality is ensured.

(c) Unauthorized usage of IS identity is not possible

Occurence Each message exchanged.

Result Message spoofing is not possible.

(12) Member M2 is relaying on Service (SE)

(a) Provide correct description of endpoints.

(b) Provide correct client IS ACL information.

Occurence On setup. Seldom on client ACL change.

Result Information for forwarding requests is available.

(13) Service (IS) is relaying on Member M2

(a) Correct ACL is used on service.

(b) Correct client identifiers are provided with request to service.

(c) Requests are forwarded to correct endpoint.

(d) Message confidentiality is ensured.

Occurence Each message exchanged.

Result Message spoofing is not possible.

4.3 Summary

Analyzing the trust relations between actors reveals that any pair of actors have multiple relationships which differ from their objective and characteristics. A mapped relationship may consist more than one technical protocol. The trust dependencies tend to build up as each prior relationship is frequently used as part of trusted computing base.

5 Asset Threat Profiles

Threat profile write up on the important assets created, used or exchanged by X-Road. Threats to assets are categorized using STRIDE-LM threat classification, the larger letter marks the potential threat. Mnemonic is described in Section 2.3, Threat Modeling and Classification.

Adopting threat profile template[MF19] to X-Road assets with the items contributing to asset threats are presented. Attack and threat details are not exhaustive, but providing ideas, directions, ideas and influences how assest may be manipulated or misused. Listed items are result of brainstorming and the experience of the author.

For the reference the example profile is described. Threat Profiles for the main X-Road assets are provided in Appendix 5.

5.1 Example Profile: Global Configuration

Specification: Protocol for Downloading Configuration [X pa]

Protocol Flow: GC generation and distribution

Global Configuration (GC) provides directory service information for the dependent participants. GC is cached after verification in components for later usage. Following key items are included within GC:

- Details of approved TAs
- Security server details
- Registered instance members and their mapping to security servers
- Approved federation information

Table 3 provides summary of the GC threat profile.

Table 3. Threat profile: Global configuration.

	Description
Asset	**Global configuration**
Threat types	`STRIdE-LM`
Ownership	Governing authority
Attack surface	Administration activities Management requests Configuration database GC signing key GC generation Restoring from backup files Environment
Attack vectors	Gain permissions: Central server administrator, OS admin or database. Compromised GC signing key w/w-o MitM. Insecure fetching GC or its verification information Gain permissions: Member's X-Road component administrator or OS admin Modified GCA w/w-o GC MitM. Modifing GC on system restore Including extra GC via federation Manipulation of the configuration parts Manipulating server or its environment
Threat actors	APT on X-Road GA/member server or environment Crooked or uncaring GA X-Road / system administrator Crooked or uncaring member's X-Road system or system administrator Knowledgeable attacker

5.2 Summary

Asset threat profiles provide quick overview of the asset. Pre-determined threat types in STRIDE-LM menomonic provides information about suspected threats. Fields about attack surface, vector and actors allow record found information during threat analysis. Attack info is to give ideas and list general attack attributes which may affect the asset.

Threat profiles can be used to check information about specific asset, record new knowledge about the asset or use provided information synthesize new hypothesis.

6 Protocol Flows

Protocols create, transform or transport trusted assets. For getting good overview of the protocol inputs and flow, reading the specification documents is overwhelming and often too detailed for first analysis. The fish-bone diagrams are used for visual representation of protocol essence, it can be seen as ordered mindmaps for protocol flows. This provides good platform for quickly updating or reviewing flows when threat analysis is performed.

Used graphical methodology is derived from Ishikawa cause-effect discovering diagrams. Ishikawa diagrams are often used in quality management disciplines to determine root (and contributing) causes of a problem. It provides way to systematize and clarify issues in focused manner. [JD10, pp. 551–552]

Similar fish-bone/mindmap diagrams have been used by the author for few years. Mainly for explaining protocol and system flows during training sessions.

Reading the fish-bone protocol flow diagrams:

- Desired outcome is rightmost centerline element in a box shape.
- The elements (or topics) contributing to (successful) outcome are ovals above and below centerline
- Order of elements is read from left to right. Exact order is indicated by point of joining centerline.
- Each element has multiple inputs or dependencies which contribute to element. Order is determined by reading from outside moving towards cernterline.
- Input coding and boundaries:

  red Color red. Critical inputs. Breach cannot be compensated on other component

  | Blue lines. System or trust boundary

  underline Underlined items. Human interaction.

For the reference the example protocol flow is explained in Chapter 6.1. Protocol flows for the main X-Road protocols are provided in Appendix 2.

6.1 Example Flow: Global Configuration

Specification: Protocol for Downloading Configuration [X pa]

Figure 7 shows the GC protocol flow. The GC is generated on Central Server from stored configuration, the expiration time is set and GC signed. Web server provides signed global configuration download. The Security Server has Global Configuration Anchor (Ref: Threat profile: Global configuration anchor) which is used to determine the download URL and the certificate for signature verification. Expiration Time is checked comparing expiration time on GC and system time. Successfully verified GC is stored for usage by other components.

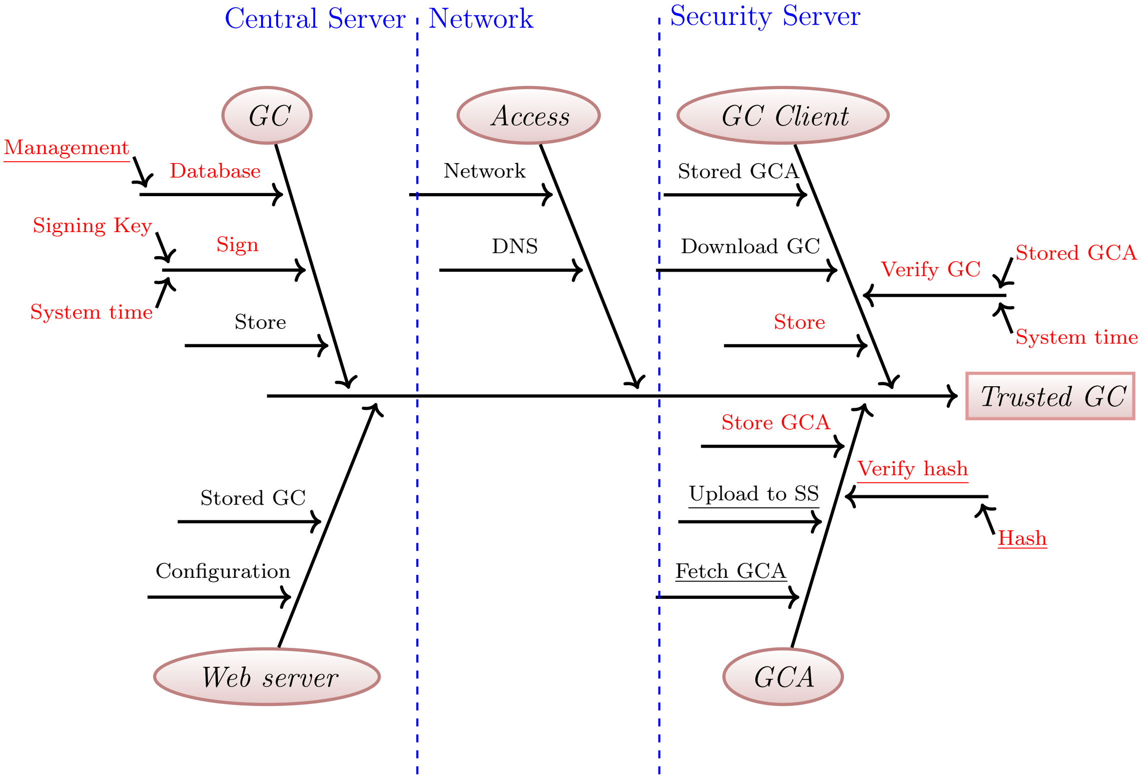


Figure 7. Protocol Flow: GC generation and distribution.

6.2 Summary

Protocol flows provide ordered, renewable, expandable way to keep and rewiew protocols during hypothesis validation, detailing relationships in general trust model. Flows itself can be used to synthesize new hypothesis to be verified on other models or on implementation. Protocol flows should be updated when new knowledge is gained about inputs or dependencies.

7 Component Data Flows

Previous sections have provided detail on trust model and protocol flows which are more for communication between external parties. For deeper analysis we need to observe also internal model of components. Protocol flows do not give enough insight to component or sub-component level.

X-Road architecture documentation component provides overview of data flows and interfaces provided. But this is not sufficient for security analysis – information on separation of contexts and sub-components is necessary to find answers to questions like 'what sub-components have access to other sub-component or asset' or 'what sub-components run in same trust domain'.

Trust boundaries method was used to map component interfaces and flows between trust domains. Diagrams below are drawn with open-source tool OWASP Threat Dragon[1]. Same tool can be used to record element and flow properties for further usage and analysis.

7.1 Central Server Data Flow

Simplified data flow with trust boundaries based on Central Server Architecture [X ab] and observations on deployed component is shown on Figure 8.

The external and internal network boundaries expose few communication protocols to external components. Internally the sub-comonents can be divided into 3 zones - general OS, system user and database domains.

Additionally the boundary and the component of organizational processes is shown – it is important input for management/registration processes.

[1] `https://threatdragon.org`

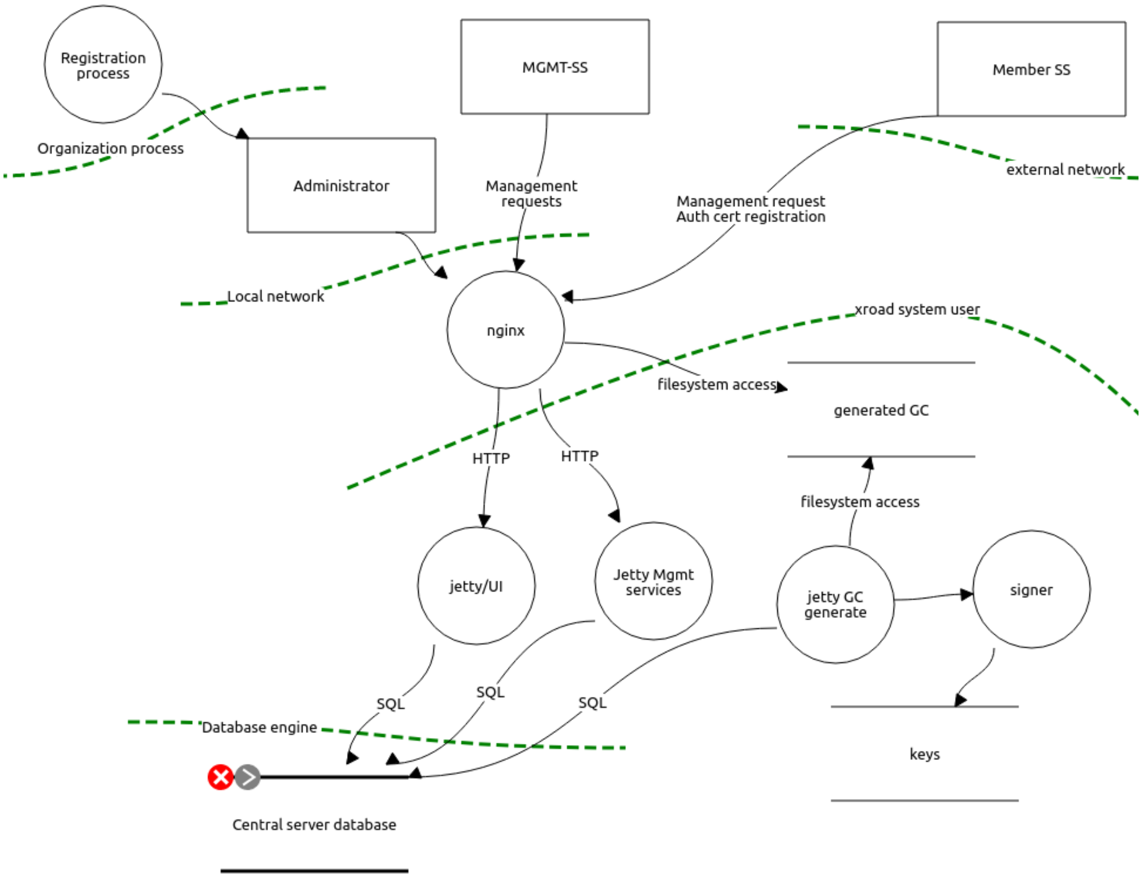


Figure 8. Flows and boundaries in Central Server.

7.2 Security Server Data Flow

Simplified data flow with trust boundaries based on Security Server Architecture [X ac] and observations on deployed component is shown on Figure 9.

The division into sub-components and trust domains is similar to central server.

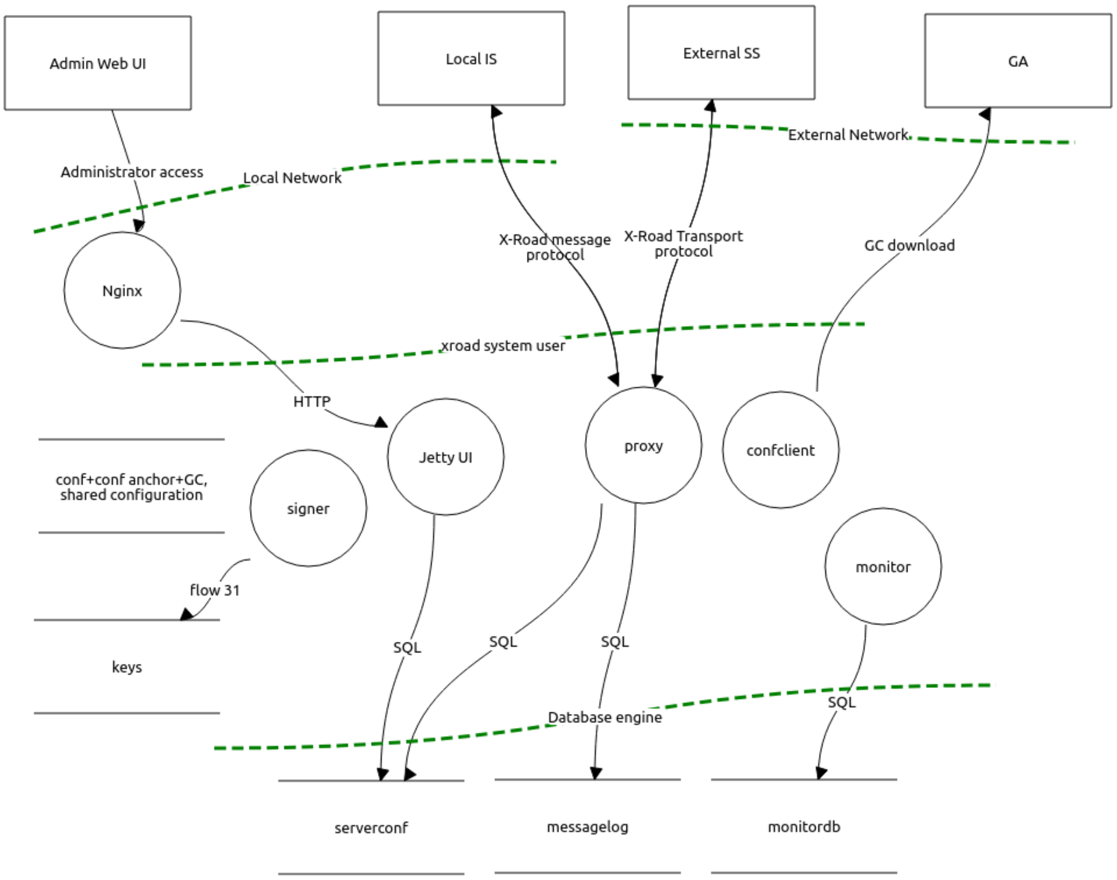


Figure 9. Flows and boundaries in Security Server.

7.3 Summary

Component data flows with trust boundaries provides investigator information about interactions and trust assumption changes on subcomponent level. When data flow is augmented with references to protocols, the protocol flows can be used to check if assumptions are correct and there is no excessive privileges or unwanted effects to components in the same trust boundary.

8 Method for Identifing Potential Weaknesses

Chapter proposes a method how to use previously compiled information to identify potential weaknesses. Figure 10 contextualizes models, it has abstract layer outside and more detailed towards center.

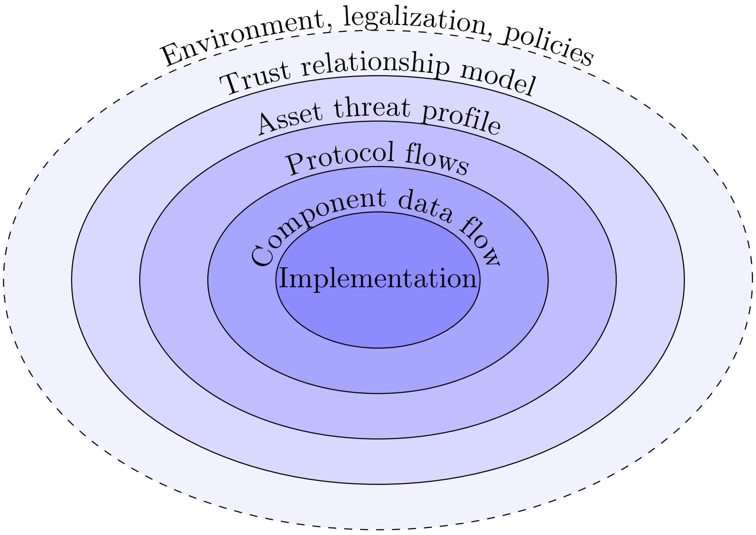


Figure 10. Layered contextualization of the composed models.

The proposed method to identify and validate potential weakness from these models is:

1. Create hypotheis or question to be validated. Source can be just an idea of weakness or detail from specification or implementation.
2. Starting from the layer the hypothesis is connected to, fill in the details contributing or counteracting to potential issue.
3. Develop attack scenario for experimentation, later verification and discussion
4. If possible, verify findings and scenario with the environment, artifacts or implementation in hand.
5. Moving to upper or lower adjacent layers allows contextualize and verify if and how the issue is mitigated or not.
6. The breach impact can be assessed using trust relationship model (asking: what trust relationship assumptions are breached?). Outer layer, environment with legalization determines the magnitude of the potential issue.

Using proposed method the example process is explained with example in Chapter 8.1.

8.1 Example: Process of Identifying Potential Weakness

All previously compiled information is used to refine the knowledge:

Trust Model	Explained in Chapter 4 Trust on X-Road
Protocol Flows	Explained in Chapter 6 Protocol Flows and main flows in Appendix 2 Protocol flows
Asset Threat Profiles	Example provided in Chapter 5 Asset Threat Profiles and main profiles in Appendix 1 Asset Threat Profiles
Component Data Flows	Explained in Chapter 7 Component Data Flows

The summary of the process:

Hypothesis Malicious Central Server administrator can manipulate Global Configuration which is un-noticeable for dependent parties.

Details The issue is related to trust model, GC protocol flow and implementation. GC is trusted by all Members, it is periodically updated. GC is used in most protocols as a trusted input.

Scenario Administrator modifies configuration item, GC is distributed and accepted. After the fact contents of the GC is reverted.

Verification From implementation: software logs fact about downloading modified GC, but does not save the state or provide details on changes. Central server has user action audit log functions available, but they are not verifiable by default without external measures. Audit logs are not visible to Members.

Impact All participants of the instance, the GC is input for most protocols.

Summary Temporary change of GC may alter the behavior of platform without participants notifying.

8.2 Example: Reporting Findings

Table 4 shows the example finding from the report, refernced in Appendix 3. The report summary is discussed in more detail in chapter 9 Validation.

Issue Classificaton: Design
Issue Identifier: T1.1

Table 4. Example finding from weakness report provided in Appendix 3.

	Description
Asset	**X-Road central server**
Vulnerability	Changes in global configuration are not externally auditable.
Attack surface	Insider
Scenario	CS admin - modifies GC, - executes queries/actions, - reverts GC modifications.

The reporting format was selected to be simple but providing crucial items for reference. Additional details were discussed during workshop.

8.3 Summary

Using structured information previously compiled, understanding of layered context and method proposed, the hypothesis and questions can be validated or invalidated. Process is explained using a hypothesis on example case.

9 Validation

To validate approach on decomposition and identifying potential weaknesses, the study on current X-Road implementation, artifacts and Estonian X-Road deployment *X-tee* was conducted.

Using previously compiled information presented in this thesis, standard tools for network scan, dependency checks and few self developed utilities the limited scope analysis was performed in May 2020.

Chapter provides summary of the findings. Risk ratings are dependent on exact environment and therefore not calculated by the author. Mitigation options and improvements discussions are undergoing and not in the scope of this work.

9.1 Reference System

Validation of the trust model and proposed analysis approach, following parameters were used:

- X-Road version 6.23.0 (released 19th Feb 2020) source code and documentation
- Packages/default configuration for Ubuntu (Estonian flavor)
- Deployed test environment operating system: Ubuntu 18.04 LTS
- Scope: Estonian environment with federated instances (i.e. Finnish environments). Relevant public documentation about instance.[1]

9.2 Summary of the Identified Weaknesses

Using trust model, test instance, protocol specification and flows 40 potential weakness were identified. 5 weaknesses are specific to Estonian X-tee instance setup, 35 issues are bound to X-Road technology itself.

Identified weaknesses were released to Estonian Information System Authority (RIA) for evaluation in June 2020. Further explanations were provided during seminar held with RIA *X-tee* and RIA CERT-EE teams. The report is provided in Appendix 3.

Table 5 provides categorization of the findings. Weaknesses are categorized by thesis author

[1] `https://www.ria.ee/et/riigi-infosusteem/x-tee/liitumine.html`

subjectively, using question 'what properties may be affected by the weakness'. Note: one weakness may appear under multiple threat category.

Table 5. Weakness category breakdown.

Category	Total	Spoofing	Tampering	Repudiation	Information Disclosure	DoS	Elevation of Privilege	Lateral Movement
Design	**12**	6	3	6	7	2	7	3
Default	**17**	6	8	2	8	4	7	5
Deployment	**11**	6	4	2	5	5	7	5

Initial response from RIA and X-Road development coordinator NIIS teams was acknowledging the issues and confirming the improvements for mitigating weaknesses. Answers breakdown is provided in Table 6.

Table 6. RIA responses for identified weaknesses.

Response	**Count**
Mitigated in current version	10
Fix planned for next minor release	6
Considered for next major release	19
More information needed	5

9.3 Deployment Issues: Network Scan Results. Estonia and Finland

To assess current instance posture, the network scan with standard utilities (nmap) and custom made shell and python scripts was conducted. Network scan can reveal deployment issues and also confirm hypothesis about issues

Table 7 lists single scan results of the Internet connected security servers from 20th of May 2020. Identified potential issues were classified as violations of 'secure by default' and 'secure by deployment' principles.

Issues were communicated to CERT-EE/RIA and were clarified with the security server owners.

Table 7. Estonian and Finnish summarized deployment issues.

X-Road Environment	**Count**
Production	37
Test	38
Development	53

9.4 Deployment Issues: Network Scan Results. World wide deployments

Using OSINT methods with network scan tools (shodan, nmap) and custom developed shell and python scripts, potentially mis-configured deployments were searched in May 2020.

In total 272 unique servers belonging to 36 different instances with some configuration issue were identified.

It must be acknowledged, that the search was not exhaustive and some of the registered issues might be mitigated by other means or even be deliberate acts for other reasons.

In conclusion – the technology is widely adopted, mistakes are possible everywhere.

9.5 Software Dependency Vulnerabilities Scan

X-Road uses 3rd party software libraries and dependencies extensively, as any modern software project. Published vulnerabilities are usable for any knowledgeable adversary and rises the risk of security breach considerably. Therefore dependency analysis was performed to assess overall health of X-Road vulnerability management.

CVE (Common Vulnerabilities and Exposures) list is providing information on the potentially vulnerable versions of the software components. The scoring system is established to provide information how critical the vulnerability potentially is. Scoring is on the scale of 0 to 10, the 10 is most severe. By CVSS v3 score over 9.0 is considered critical. [Cvs]

Using software components with known vulnerabilities might allow adversaries to use known attack methods against the system or software. The assessment of the software different versions against known vulnerabilities allows to gain knowledge about:

- How quickly known, potential vulnerabilities are managed by software developer.
- How vulnerable are the users running old releases without upgrading.
- Is software release and patch cycle frequent enough to limit the time running software with known issues.

Java and Ruby dependencies were checked with Dependecy-check[1] and bundler-audit[2] comparing X-Road version release dates and reported CVE publishing dates.

Note: CVE criticality is not assessed in context of X-Road, i.e conditions of the vulnerability may not be exposed in the X-Road release exploitable manner.

Table 8 lists the results of the dependency scan. For 6.23.0 hypothetical release 6.24.0 is added with a date 2020-06-01 (meaning: number of CVEs on that date if release would be made).

Column description for reading the Table 8:

1. Version number
2. Release date
3. CVEs published up to the release date
4. CVEs published up to the release date rated >=9
5. Next major release version
6. Next major release date
7. CVEs published up to next major release date for version on 1^{st} field.
8. CVEs published up to next major release date for version on 1^{st} field rated >=9.

[1] https://jeremylong.github.io/DependencyCheck/

[2] https://github.com/rubysec/bundler-audit

Table 8. External dependency CVE findings per version.

Version	**Release Date**	**CVE (all)**	**CVE (>=9)**	**Next Version**	**Next Release date**	**CVE (all)**	**CVE (>=9)**
6.17.0	2018-02-19	16	2	6.18.0	2018-05-29	18	3
6.18.0	2018-05-29	18	3	6.19.0	2018-09-27	38	6
6.18.1	2019-02-13	48	13	6.19.0	2018-09-27	38	6
6.19.0	2018-09-27	38	6	6.20.0	2019-01-24	48	13
6.19.1	2019-02-13	48	13	6.20.0	2019-01-24	48	13
6.20.0	2019-01-24	36	11	6.21.0	2019-04-29	41	11
6.20.1	2019-02-13	36	11	6.21.0	2019-04-29	41	11
6.20.2	2019-10-23	8	1	6.21.0	2019-04-29	6	0
6.21.0	2019-04-29	35	9	6.22.0	2019-10-23	50	18
6.21.1	2019-05-23	36	9	6.22.0	2019-10-23	50	18
6.21.2	2019-10-23	7	1	6.22.0	2019-10-23	7	1
6.22.0	2019-10-23	7	1	6.23.0	2020-02-19	17	6
6.22.1	2019-11-11	9	2	6.23.0	2020-02-19	17	6
6.23.0	2020-02-19	12	3	(6.24.0)	(2020-06-01)	31	8

For conclusion – through releases the dependencies are updated to exclude vulnerable components. The quality of the dependency management is somewhat improved.

10 Conclusions

Thesis provides methods to decompose sophisticated systems, like X-Road, and organize information and properties suitably for further analysis, like threat analysis. Proposed methods were used to identify potential weaknesses through using compiled information.

From validation phase it is concluded that described method allows identify potential weaknesses which were not discovered or handled. Provided details allowed to present issues in understandable manner and plan proper controls.

Answers to Research Questions

1. How to systematize trust relationships in X-Road architecture?
   Answer: Chapter 4 proposes the X-Road trust relationship model. Graphical model connects the actors and their relations. Relations are ordered transitive cross dependencies as each relationship is based on prior ones.
   (a) Who are the main actors of the X-Road environment?
   Answer: Chapter 4.1 lists the actors and their roles.
   (b) What are the critical trust relations between actors?
   Answer: Chapter 4.2 contains the model with relationship essence explanations.
2. How to analyze X-Road multimessage, multiparty complex trust relationship models efficiently?
   Answer: System is decomposed and abstract using to trust relationship model, asset threat profiles, protocol flows and component data flows. Method using decomposed information is provided.
   (a) How to represent different information reusable, renewable and graspable manner?
   Answer: Chapters 4, 5, 6, 7 provide X-Road information in decomposed manner, usable for analysis. Appendices 1 and 2 provide information in graspable and renewable form.
   (b) How to use resulting systematized information for threat analysis?
   Answer: Chapter 8 proposes context and method for identifying weaknesses using compiled information.
3. What weaknesses can be identified from X-Road using created trust relationship descriptions?
   Answer: Most recent X-Road version 6.23.0 artifacts were used for validation the

method and summary of the report regarding identified weaknesses is in Chapter 9.

(a) What is the current security posture of the X-Road instances in Estonia? **Answer:** Chapter 9.3 contains summary of the network and dependency vulnerability scans.

10.1 Future Work

Structured information provided in thesis can be used for security or functional analysis and plan mitigation measures in deployments or improvements for future releases.

Thesis limitations allowed only to present main flows and asset profiles about X-Road. Decomposing system further allows better coverage for deeper analysis.

Findings summarized in thesis are already disclosed to Estonian Information System Authority (RIA) and NIIS teams. Some of the findings have resulted changing the environment already. Other issues need additional analysis and fixing the software or research for better solutions.

Thesis author estimates in-depth security analysis for the X-Road to 3-6 man-months. Each instance specific, with covering other dependencies (i.e TA, internal procedures, policies), security analysis is estimated to 2-3 month work.

Methods presented in thesis can be used for other system decomposition and security analysis.

11 Summary

The objective of this thesis was systematize numerous elements of the X-Road to provide good foundation for further security analysis to help eliminate potential weaknesses. Using provided information informed decisions can be made to select proper counter measures for existing and future instances.

The X-Road is a multiparty, technology loaded system which security analysis should be performed using decomposed method. Using models and systematic representation of information allows researcher quickly move between different phases to prove or invalidate questions and hypothesis. The models can be used for exploration and introducing system properties.

Validation of the method revealed potential improvement areas within X-Road concept, design and implementation. The work on the risk reduction has already started but has long list of issues which will get update from further security analysis yet to be conducted.

The conceptual development of X-Road principles must use scientific research and co-operation between state and research institutions. The World requires convenient, yet secure solutions which follow modern concepts.

References

[Ans01] Arne Ansper. "E-State From a Data Security Perspective". Master's Thesis. Tallinn University of Technology, 2001. URL: https://cyber.ee/research/theses/arne_ansper_msc.pdf (visited on 2020-07-28).

[Ans+01a] Arne Ansper, Ahto Buldas, Eva Einama, Monika Oit, and Kaidi Oone. "Riigi Andmekogude Moderniseerimise Projekt: Püstitatud ülesannete juriidiline ja turvaanalüüs. [Project to modernize state databases: legal and security analysis of the project objective]". Estonian. Cybernetica AS. Internal ref: DO-LU-T-05-0301. 2001-04-26.

[Ans+01b] Arne Ansper, Ahto Buldas, Sven Heiberg, Monika Oit, Kaidi Oone, Olev Sepp, and Jan Willemson. *Digitaalallkirja juurutamine riigiasutustes: stateegiline plaan [Introducing digital signatures in state inistutions: stategic plan]*. report. Cybernetica AS, 2001. URL: https://cyber.ee/research/reports/Digital-signatures-strat-plan-2001.pdf (visited on 2020-08-28).

[Ans+03] A. Ansper, A. Buldas, M. Freudenthal, and J. Willemson. "Scalable and efficient PKI for inter-organizational communication". In: *19th Annual Computer Security Applications Conference, 2003. Proceedings.* 2003, pp. 308–318.

[Ans10] Arne Ansper. "Uus X-tee [New X-Road]". Estonian. Cybernetica AS. Internal ref: J-105-16. 2010-08.

[Ans+13a] Arne Ansper, Ahto Buldas, Margus Freudenthal, and Jan Willemson. "High-Performance Qualified Digital Signatures for X-Road". In: *Secure IT Systems.* Ed. by Hanne Riis Nielson and Dieter Gollmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 123–138. ISBN: 978-3-642-41488-6.

[Ans+13b] Arne Ansper, Ahto Buldas, Margus Freudenthal, and Jan Willemson. "Protecting a Federated Database Infrastructure against Denial-of-Service Attacks". In: *Critical Information Infrastructures Security.* Ed. by Eric Luiijf and Pieter Hartel. Cham: Springer International Publishing, 2013, pp. 26–37. ISBN: 978-3-319-03964-0.

[Avt] *Public Information Act.* 2001. URL: https://www.riigiteataja.ee/akt/113032019036?leiaKehtiv (visited on 2020-05-08).

[AW06] Arne Ansper and Jan Willemson. *X-Road – A Complete Solution for Inter-organizational Information Exchange.* Internal ref: T-4-1. Cybernetica AS, 2006-12-06. URL: https://cyber.ee/research/reports/T-4-1_X-Road_complete_solution_for_inter-organizational_information_exchange.pdf.

[Cvs] *X-Road technologies.* NIST. URL: https://nvd.nist.gov/vuln-metrics/cvss# (visited on 2020-07-28).

[EM17] T.W. Edgar and D.O. Manz. *Research Methods for Cyber Security.* 2017. ISBN: 978-0-12-805349-2.

[Fre13] Margus Freudenthal. *Using Batch Hashing for Signing and Time-Stamping.* report. Internal ref T-4-20. Cybernetica AS, 2013. URL: https://cyber.ee/research/reports/T-4-20-Using-Batch-Hashing-for-Signing-and-Time-Stamping_3.pdf (visited on 2020-08-28).

[Fre17] Margus Freudenthal. *Profile for High-Performance Digital Signatures.* report. Internal ref T-4-23. Cybernetica AS, 2017. URL: https://cyber.ee/research/reports/T-4-23-Profile-for-High-Performance-Digital-Signatures.pdf (visited on 2020-08-28).

[FW17] Margus Freudenthal and Jan Willemson. “Challenges of Federating National Data Access Infrastructures”. In: *Innovative Security Solutions for Information Technology and Communications*. Ed. by Pooya Farshim and Emil Simion. Cham: Springer International Publishing, 2017, pp. 104–114. ISBN: 978-3-319-69284-5.

[Hana] *Development of the cross-border X-Road.* Public Procurement. URL: https://riigihanked.riik.ee/rhr-web/#/procurement/576227/general-info (visited on 2020-08-01).

[Hanb] *Development of X-tee 5.5.* Public Procurement. URL: https://riigihanked.riik.ee/rhr-web/#/procurement/515436/general-info (visited on 2020-08-01).

[JD10] Joseph M. Juran and Joseph A. De Feo. *Juran’s Quality Handbook. The Complete Guide to Performance Excellence.* 6th ed. 2010. ISBN: 978-0-07-162972-0.

[Kad+00] Vello Kadarpik, Hannu Krosing, Tanel Tammet, and Ain Järv. *Riigi andmebaaside sidumise pilootprojekt: eesmärgid, tulemuste ülevaade, soovitused [Pilot project connecting state databases: goals, result overview, recommendations].* Estonian. 2000. URL: https://web.archive.org/web/20010709133923/http://www.riik.ee/teeninduskiht/ (visited on 2020-05-08).

[Kal+13] A. Kalja, J. Pold, T. Robal, U. Vallner, and V. Viies. “Estonian eGovernment services: Lesson learned”. English. In: *2013 Proceedings of PICMET 2013: Technology Management in the IT-Driven Services.* 2013, pp. 562–568.

[Kas20] Marten Kask. “Blockchain-based Members Management for the Unified eXchange Platform”. Master’s Thesis. Tallinn University of Technology, 2020. URL: https://digikogu.taltech.ee/et/Item/ca50bea0-f72c-41c2-a656-703e6ff11bef (visited on 2020-07-28).

[MF19] Michael Muckin and Scott C. Fitch. *A Threat-Driven Approach to Cyber Security.* Lockheed Martin Corporation. 2019. URL: https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Threat-Driven-Approach.pdf (visited on 2020-05-08).

[Mss] *Secure simply by SD3 way!* 2007. URL: https://docs.microsoft.com/en-us/archive/blogs/stronglinks/secure-simply-by-sd3-way (visited on 2020-07-30).

[Pmt] *The project triangle.* URL: https://support.microsoft.com/en-us/office/the-project-triangle-8c892e06-d761-4d40-8e1f-17b33fdcf810 (visited on 2020-07-30).

[Pwr] “Digitaalallkirja jõud X-teele [Legal Digital Signature for X-Road]”. Estonian. Cybernetica AS. Internal ref: M-41-1. 2011-11-11.

[Ris] *Riigi andmekogude moderniseerimise programm (X-tee) [Program to modernize state databases (X-tee)].* Estonian. 2001. URL: https://web.archive.org/web/20011126191007/http://www.riik.ee/ristmik/lyhitutvustus.htm (visited on 2020-05-08).

[Saa15] Riin Saarmäe. “Analysis of Configuration Management in Federated X-Road Systems”. Bachelor’s Thesis. Tallinn University of Technology, 2015. URL: https://digikogu.taltech.ee/et/Item/af59c9da-4047-4e07-ab0c-669dfb1dac91 (visited on 2020-07-28).

[Sds] “Cooperation on Secure Distributed Service Bus (SDSB).” [Contract between Cybernetica AS, ELIKO, RIA, eTervis]. 2012-08.

[Sho14] Adam Shostack. *Threat Modeling: Designing for security.* 2014. ISBN: 978-1-118-80999-0.

[Sta] *X-Road statistics API. Estonia and Finland.* URL: `https://app.swaggerhub.com/apis-docs/NIIS/x-road-statistics/1.0.0` (visited on 2020-07-30).

[Tam00a] Tanel Tammet. *Soovitused riigiasutuste elektrooniliseks asjaajamiseks: XML põhine dokumendihaldus [Recommendations for state electronic procedures: XML-based document managament].* Estonian. 2000. URL: `https://web.archive.org/web/20030926155044/http://www.riik.ee/dh/ylevaade/xmldokhaldus.pdf` (visited on 2020-07-28).

[Tam00b] Tanel Tammet. *XML rakendused ja XML-RPC põhine näidissüsteem [XML applications and XML-RPC based example system].* Estonian. 2000. URL: `https://web.archive.org/web/20010709133923/http://www.riik.ee/teeninduskiht/xmlrpcandmebaasid.pdf` (visited on 2020-05-08).

[VK04] Vijay Vaishnavi and B Kuechler. "Design Science Research in Information Systems". In: *Association for Information Systems* (2004-01).

[WA08] J. Willemson and A. Ansper. "A Secure and Scalable Infrastructure for Inter-Organizational Data Exchange and eGovernment Applications". In: *2008 Third International Conference on Availability, Reliability and Security.* 2008, pp. 572–577.

[X aa] *X-Road Architecture.* Cybernetica AS and NIIS. 2019-04-16. URL: `https://github.com/nordic-institute/X-Road/blob/6.23.0/doc/Architecture/arc-g_x-road_arhitecture.md` (visited on 2020-05-07).

[X ab] *X-Road: Central Server Architecture.* Cybernetica AS and NIIS. 2018-03-02. URL: `https://github.com/nordic-institute/X-Road/blob/6.23.0/doc/Architecture/arc-cs_x-road_central_server_architecture.md` (visited on 2020-05-07).

[X ac] *X-Road: Security Server Architecture.* Cybernetica AS and NIIS. 2019-04-17. URL: `https://github.com/nordic-institute/X-Road/blob/6.23.0/doc/Architecture/arc-ss_x-road_security_server_architecture.md` (visited on 2020-05-07).

[X pa] *Protocol for Downloading Configuration.* Cybernetica AS and NIIS. 2018-11-08. URL: `https://github.com/nordic-institute/X-Road/blob/6.23.0/doc/Protocols/pr-gconf_x-road_protocol_for_downloading_configuration.md` (visited on 2020-04-28).

[X pb] *X-Road: Message Protocol for REST.* NIIS. 2019-04-25. URL: `https://github.com/nordic-institute/X-Road/blob/6.23.0/doc/Protocols/pr-rest_x-road_message_protocol_for_rest.md` (visited on 2020-05-07).

[X pc] *X-Road: Message Protocol v4.0.* Cybernetica AS and NIIS. 2018-03-06. URL: `https://github.com/nordic-institute/X-Road/blob/6.23.0/doc/Protocols/pr-mess_x-road_message_protocol.md` (visited on 2020-05-07).

[X pd] *X-Road: Message Transport Protocol.* Cybernetica AS and NIIS. 2019-03-04. URL: `https://github.com/nordic-institute/X-Road/blob/6.23.0/doc/Protocols/pr-messtransp_x-road_message_transport_protocol.md` (visited on 2020-05-07).

[X pe] *X-Road: Service Metadata Protocol.* Cybernetica AS and NIIS. 2019-10-09. URL: `https://github.com/nordic-institute/X-Road/blob/6.23.0/doc/Protocols/pr-meta_x-road_service_metadata_protocol.md` (visited on 2020-05-07).

[Xrh] *X-Road History.* URL: `https://x-road.global/xroad-history` (visited on 2020-05-08).

[Xte] “X-tee2 vision [Vision of the X-tee2]”. Estonian. Cybernetica AS. Internal ref: Y-743-2. 2012-04.

Appendix 1 – Asset Threat Profiles

1.1 Environment

Environment is not shown on trust relation model, but it has influence over X-Road functions.

Table 1. Threat profile: X-Road server with OS and environment.

	Description
Asset	**X-Road server with OS and environment**
Ownership	X-Road participant
Threat types	STRIDE-LM
Attack surface	Operating System and services Required external services DNS, time Network Extra repositories included on X-Road servers
Attack vectors	Gain permissions: OS privileged/unprivileged user, X-Road API user or UI admin, database access Dependencies: vulnerability is discovered Inject replacement packages through extra repositories configured Alter information about repository/keys: in public sources or configured on X-Road server MitM on insecure fetching repository details or software External service manipulation (DNS, time etc) Denial of Service
Threat actors	APT on X-Road GA/member server or environment Crooked or uncaring X-Road component or system administrator Knowledgeable attacker Malicious software provider Configuration mistakes Lack of maintenance and monitoring

1.2 Software and Distribution

X-Road participants usually rely on software built and distributed by the Software Provider.

Table 2. Threat profile: X-Road software.

	Description
Asset	**X-Road software assets (development and deployment summarized)**
Ownership	Software provider
Threat types	STRIDE-LM
Attack surface	Source code with external dependencies Build environment and external dependencies Repository signing key Repository server Repository access information Extra repositories included on X-Road servers X-Road component interfaces and protocols
Attack vectors	Gain permissions: on source code, build system, X-Road repository or key Dependencies: vulnerability is discovered, modify existing dependency or include new Repository: replace or add contents to repository Inject replacement packages through extra repositories configured Alter information about repository/keys: in public sources or installed on X-Road server MitM on insecure fetching repository details or software Insecure default settings Outdated software
Threat actors	APT on X-Road development/distribution/GA/member server Crooked or uncaring developer, X-Road component or system administrator Knowledgeable attacker Malicious dependency provider

1.3 Global Configuration Anchor

Global Configuration Anhor is the file generated by the governing authority to provide X-Road participants global configuration download URL(s) and certificate(s) for integrity verification. GCA is uploaded to security server via UI, for verification purpose the generation time and the hash of the GCA is provided out of band.

Specification: Protocol for Downloading Configuration [X pa]

Table 3 lists the profile for global configuration anchor.

Table 3. Threat profile: Global configuration anchor.

	Description
Asset	**Global configuration anchor**
Ownership	Governing authority
Threat types	STriDe-lm
Attack surface	GCA on generation or re-keying Distribution of GCA GCA on dependent server disk Restoring from backup files
Attack vectors	Gain permissions: Central server administrator or OS admin. Member's X-Road component administrator or OS admin Modify GCA or verification information on distribution. Insecure fetching GCA or its verification information Modifing GCA on system restore
Threat actors	APT on X-Road GA/member server or environment Crooked or uncaring GA X-Road / system administrator Crooked or uncaring member's X-Road system or system administrator Knowledgeable attacker

1.4 Global Configuration

Global Configuration (GC) provides directory service for X-Road identifiers and mappings, lists approved Trust Authority details and privileged identifiers (e.g monitoring and management system identifiers).

Specification: Protocol for Downloading Configuration [X pa]

Following key items are included within GC:

- Details of approved TAs
- Security server details
- Registered instance members and their mapping to security servers
- Approved federation information

Table 4 provides threat profile on GC.

Table 4. Threat profile: Global configuration.

	Description
Asset	**Global configuration**
Threat types	`STRIDE-LM`
Ownership	Governing authority
Attack surface	Administration activities Management requests Configuration database GC signing key GC generation Restoring from backup files Environment
Attack vectors	Gain permissions: Central server administrator, OS admin or database. Compromised GC signing key w/w-o MitM. Insecure fetching GC or its verification information Gain permissions: Member's X-Road component administrator or OS admin Modified GCA w/w-o GC MitM. Modifing GC on system restore Including extra GC via federation Manipulation of the configuration parts Manipulating server or its environment
Threat actors	APT on X-Road GA/member server or environment Crooked or uncaring GA X-Road / system administrator Crooked or uncaring member's X-Road system or system administrator Knowledgeable attacker

1.5 Asymmetric Keys: Summary

X-Road relies heavily on asymmetric cryptography and PKI.

Following key pairs provide critical properties of the X-Road, breakdown by owner:

- GA: Global Configuration signing
- GA: Management service authentication
- Member: Message signing
- Member: Security Server authentication
- Member: User interface
- Member: Internal connections (connections to/from information systems)
- IS/SE: Internal connections (connections to/from security servers)
- SP: Software repository signing
- TA: CA and OCSP/TSA services

In the context of X-Road components, the message signing keys can be stored on PKCS#11 compatible SSCD. Other keys are softkeys, stored in filesystem.

Table 5 lists the profile asymmetric keys.

Table 5. Threat profile: Keys.

	Description
Asset	**Keys**
Ownership	X-Road participant
Threat types	`STRIDE-LM`
Attack surface	Key generation or re-keying Key usage interface Keys in memory Key storage Keys in backup files Restoring from backup files
Attack vectors	Gain permissions: component administrator, OS privileged/unprivileged user or database. Theft of the keys from backup Brute forcing key storage encryption Adding/Replacing/Removing keys during restore from backup file. Dual usage keys
Threat actors	APT on X-Road GA/member server or environment Crooked or uncaring X-Road / system administrator Knowledgeable attacker

1.6 Instance Configuration

Central server contains input for the global configuration creation and signing process.

Table 6 lists the profile for central server contained configuration.

Table 6. Threat profile: Central Server / Instance configuration.

	Description
Asset	**Central Server /Instance Configuration**
Ownership	Governing Authority
Threat types	`sTriDE-LM`
Attack surface	Administrative interface Instance Management Requests Database Configuration on disk Management service Restoring from backup files
Attack vectors	Gain permissions: component administrator, OS privileged/unprivileged user or database. Adding/Replacing/Removing configuration elements during restore from backup file. Social engineering Human error
Threat actors	APT on X-Road GA/member server or environment Crooked or uncaring X-Road / system administrator Knowledgeable attacker

1.7 Security Server Configuration

Security server mediates requests between service and its client. SS is connected to external and internal networks simultaneously. In simplified view the SS is a firewall with a limited protocol support.

Configuration has following key items, including:

- backend service details, incl access list information
- configuration for authenticating internal clients
- message logging configuration
- asymetric keys
- stored GCA/GC
- service/daemon/application configuration

Table 7 lists the profile for security server configuration.

Table 7. Threat profile: Security Server Configuration.

	Description
Asset	**Security Server Configuration**
Ownership	X-Road security server owner
Threat types	`STRIDE-LM`
Attack surface	Administrative interface Database Configuration on disk Restoring from backup files
Attack vectors	Gain permissions: component administrator, OS privileged/unprivileged user or database. Adding/Replacing/Removing configuration elements during restore from backup file. Social engineering Human error
Threat actors	APT on X-Road GA/member server or environment Crooked or uncaring X-Road / system administrator Knowledgeable attacker

1.8 Messages

Messages are payloads which are exchanged between service and its client. Scope is two fold:

- messages in transit;
- messages stored in message log for long term evidental value.

Tampering threat applies only for unsigned messages in transit (Information system <-> SS).

Table 8 lists the profile for messages.

Table 8. Threat profile: Security Server Configuration.

	Description
Asset	**Messages**
Threat types	sTrIDE-LM
Ownership	X-Road member
Attack surface	Information System request interface Service calls Configuration Database Connection to database Message log Message log archiving/archive Metaservice to fetch signed container
Attack vectors	Gain permissions: component administrator, OS privileged/unprivileged user or database. MitM on database connection Human error Exposed interfaces
Threat actors	APT on member server or environment Crooked or uncaring X-Road / system administrator Knowledgeable attacker Malicious insider

1.9 Monitoring Information

X-Road has implemented functions and protocols to collect meta-data about transactions and health statistics. Monitoring can be central (GA collects information from all participants) or local (Member monitors owned security server).

Table 9 lists the profile for monitoring subsystem.

Table 9. Threat profile: Monitoring information.

	Description
Asset	**Monitoring information**
Threat types	sTrIde-lm
Ownership	X-Road member
Attack surface	Database Connection to database Central Monitoring Local Monitoring
Attack vectors	Gain permissions: component administrator, OS privileged/unprivileged user or database. MitM on database connection GC configuration Human error Exposed interfaces
Threat actors	APT on member server or environment Crooked or uncaring X-Road / system administrator Knowledgeable attacker Malicious insider

Appendix 2 – Protocol flows

2.1 Global Configuration

GC is generated on central server based on information from the instance management processes and protocols. Dependent parties use Global configuration anchor (GCA) as a input for downloading and verifying GC signature.

On Figure 1 the most critical parts are generating GC and its verification with its storage for safe usage later.

Specification: *Protocol for Downloading Configuration* [X pa]

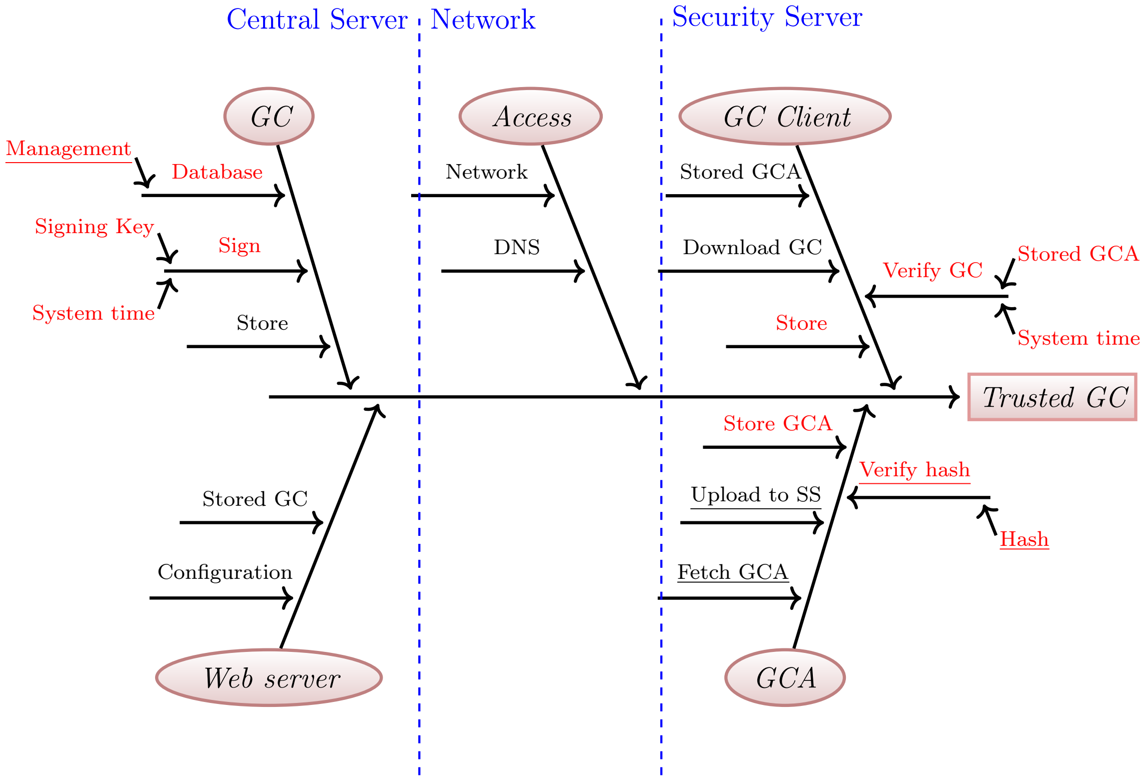


Figure 1. Protocol Flow: GC generation and distribution.

2.2 Certificates, Validity

TA protocols are not integral part of X-Road but they're providing critical elements for trust. Understanding the processes/protocols for issuing/validating certificates and time stamping is mandatory.

The certificate revoking is omitted, as it happens usually out of band. The certificate validity input in validity checking protocol is tied to revoking fact.

2.2.1 Certificate Issuance

Certificate issuance is acknowledging the public key and ensuring the subscriber identity by issuing certificate signed with CA key.

Basically all the flow is critical, as any breach may result wrongful issuing.

Specification: *X-Road: Security Server Architecture*, CA specific documentation.

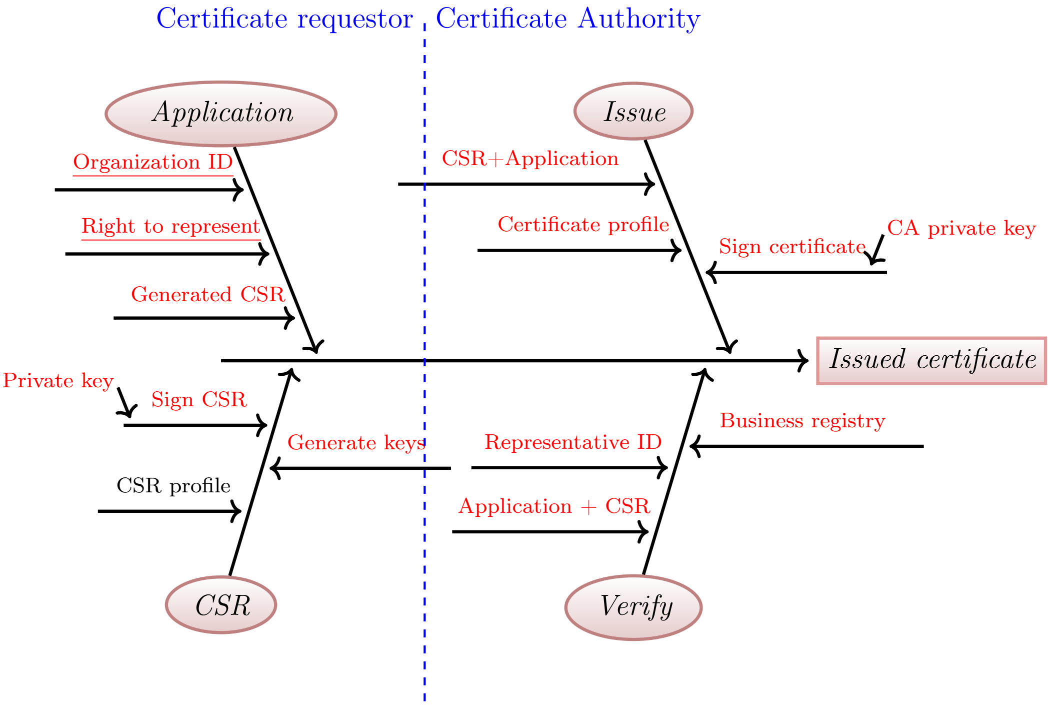


Figure 2. Protocol Flow: Issuing subscriber certificate.

2.2.2 Certificate Validity

Validity token attested by certificate issuer is additional, fresh information to ensure that the certificate is trustworthy by the knowledge of the CA.

On Figure 3 the critical part is the checking OCSP response based on information distributed via GC. OCSP service itself is not integral part of the X-Road.

Specification: *X-Road Architecture*, RFC6960.

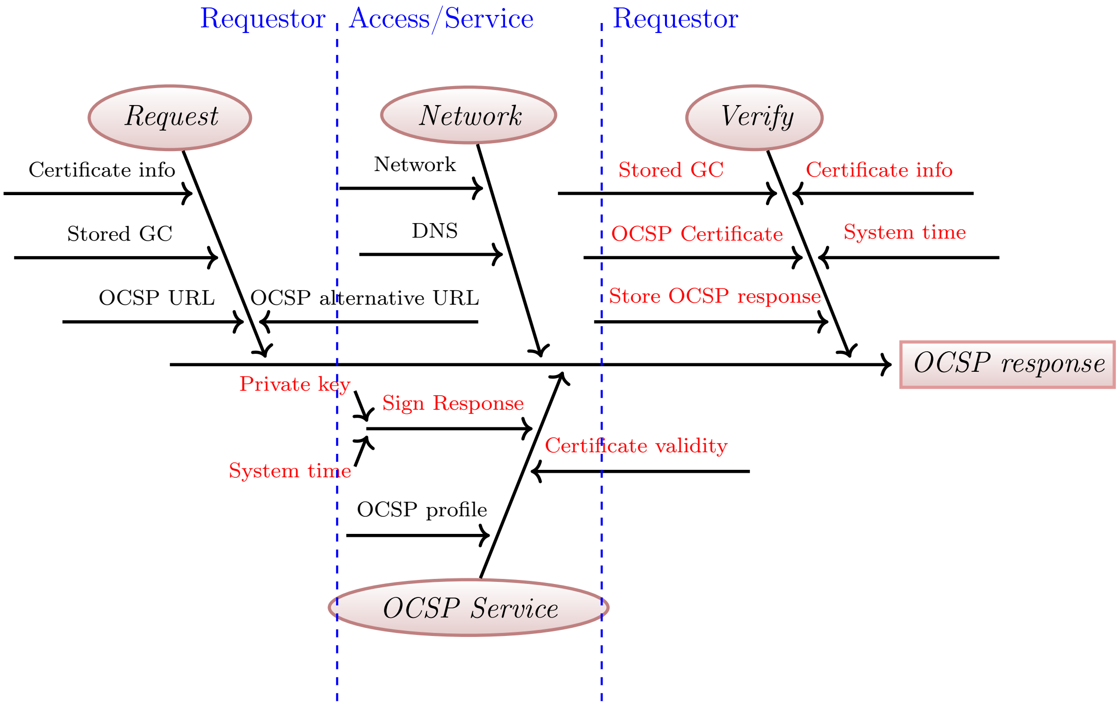


Figure 3. Protocol Flow: Certificate validity check with OCSP.

2.3 Timestamping

Timestamping is the 3rd party attestation that message existed before time presented in timestamp response signed by TSA.

On Figure 4 the input hash can be the hash of the message itself or top hash of the Merkle tree containing more than one messages.

Specification: *X-Road Architecture*, RFC3161.

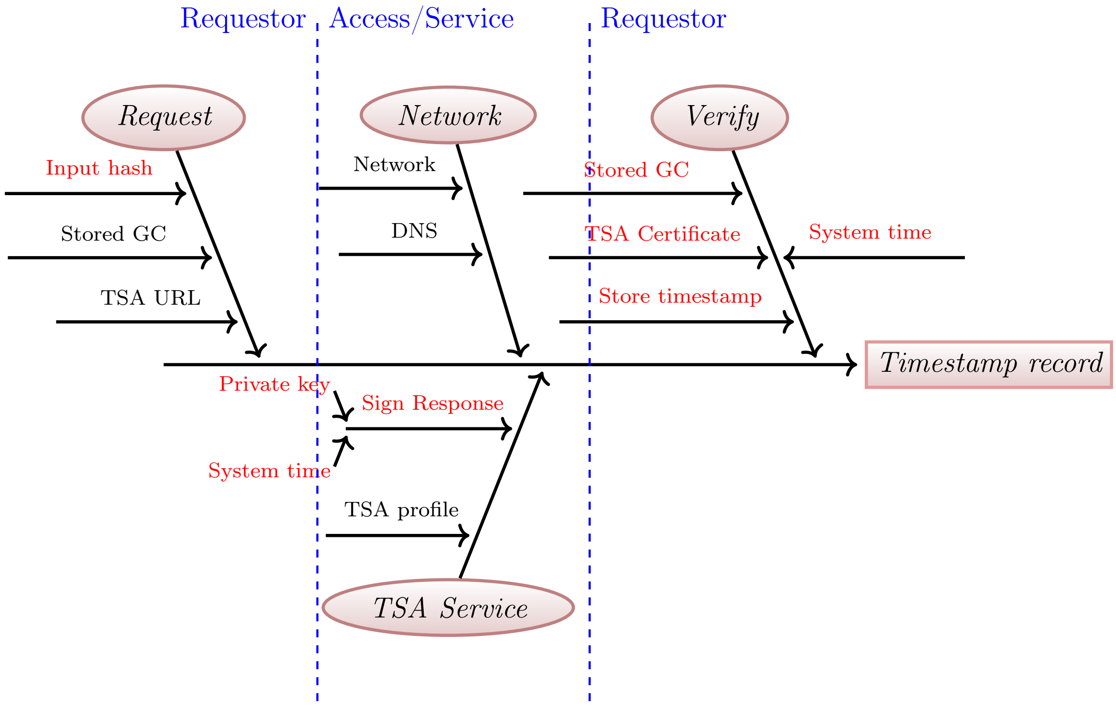


Figure 4. Protocol Flow: Timestamping.

2.4 Message Transport

Message transport is happening between two security servers. Standard TLS with AES encryption is used for message confidentiality and integrity properties during transport. TLS mutual authentication is based on information present in GC.

Figure 5 shows the main flow from starting with the Request and completing it with returning the response to client process within same HTTP connection.

Specification: *X-Road: Message Transport Protocol*.

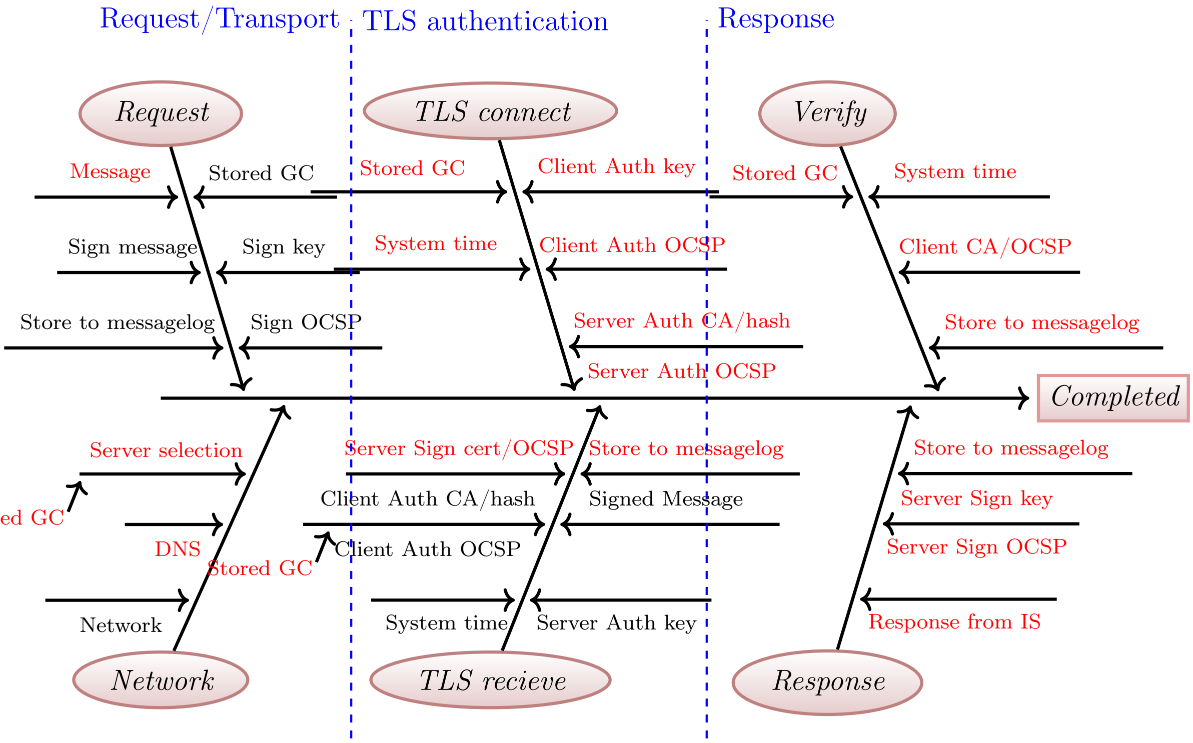


Figure 5. Protocol Flow: Message transport.

2.5 X-Road Protocol

X-Road protocol is used between security server and connected information system. Protocol transport is using HTTP.

Flow is split into two parts: the client (requestor) and service (responder) side. Between client and service parts is the X-Road transport protocol, function provided by Security Server.

2.5.1 Client-side

Information system acts as a client/requestor. The diagram 6 shows functions inside of the security server before and after transport protocol. The response is returned to client within same HTTP connection.

Specification: *X-Road: Security Server Architecture*, *X-Road: Message Protocol v4.0*, *X-Road: Message Protocol for REST*, *X-Road: Service Metadata Protocol*

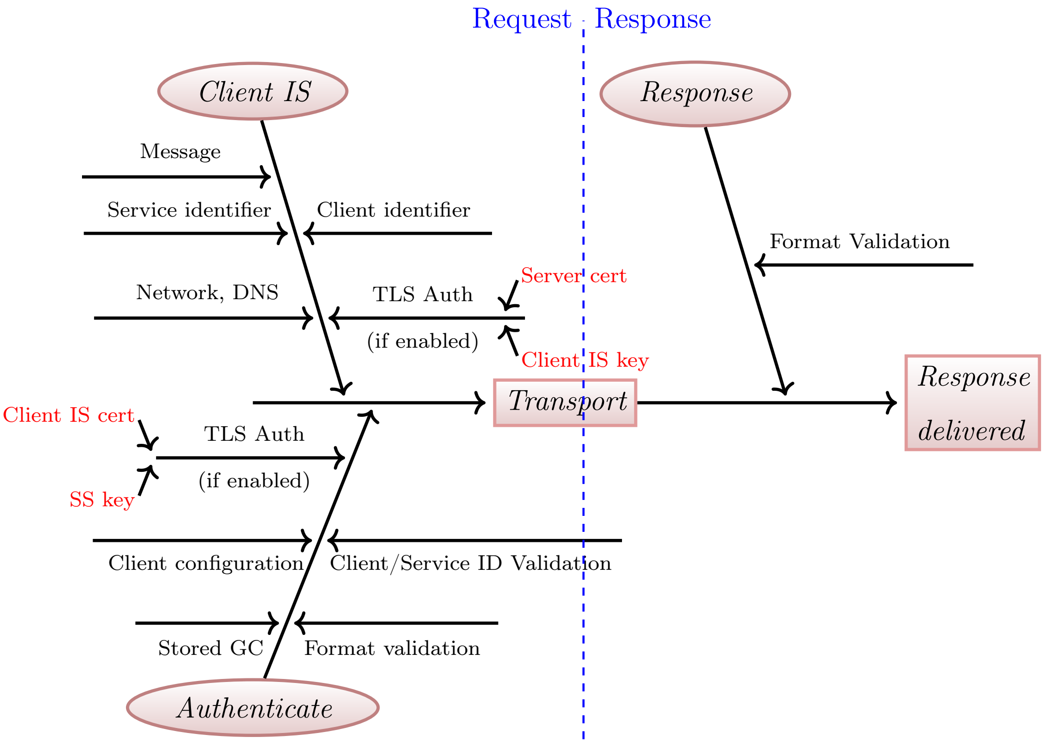


Figure 6. Protocol Flow: Client side X-Road protocol.

2.5.2 Service-side

Flow for communication between service side security server and backend is provided in Figure 7.

The request is received from X-Road transport protocol, the local configuration about service, including ACL and backend information, is used. After reaching the external service, the response is validated for formating information and returned to transport protocol within same HTTP connection.

Specification: *X-Road: Security Server Architecture*, *X-Road: Message Protocol v4.0*, *X-Road: Message Protocol for REST*, *X-Road: Service Metadata Protocol*

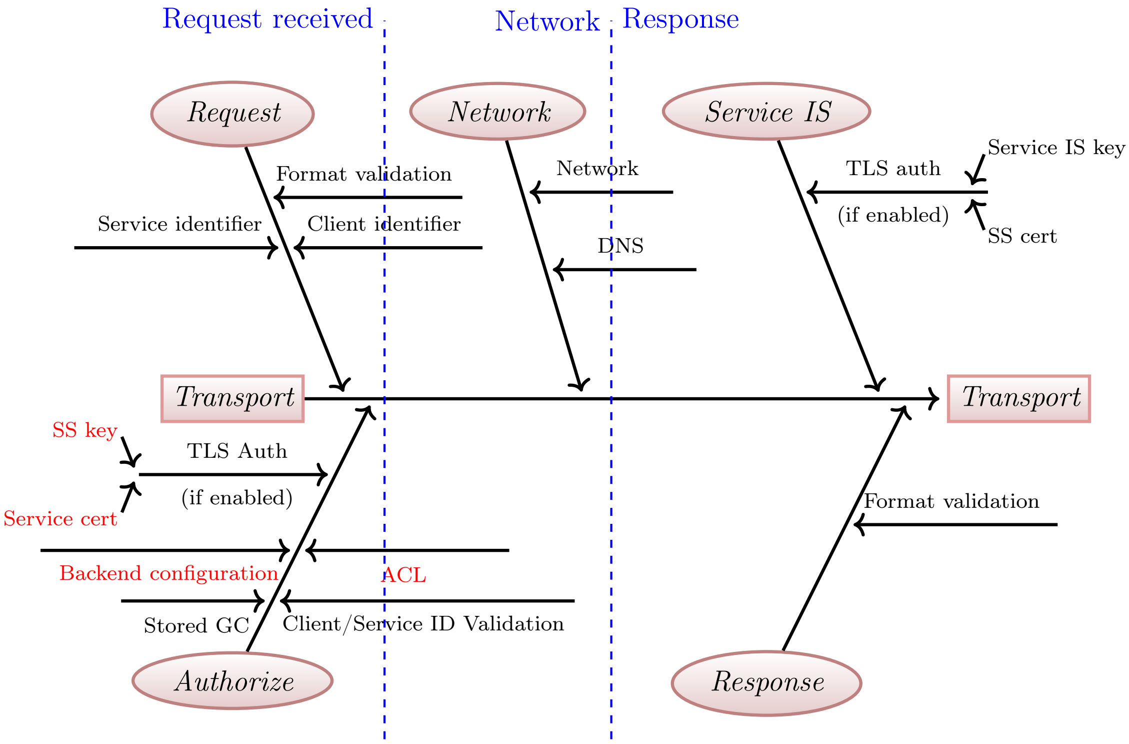


Figure 7. Protocol Flow: Service side of X-Road protocol.

Appendix 3 – X-Road Threat Analysis Findings

The report (total 14 pages) about findings released for analysis to RIA.

[Findings are not yet public due to disclosure agreement with RIA.]

X-Road threat analysis findings

Tarmo Oja
supervisor: **Ahto Buldas, PhD**

Cybernetica AS
Tallinn University of Technology

June 11, 2020

Preface

These findings are supplemental material for the Masters Thesis to be defended August 2020.

Document is to inform about potential issues or questions raised. Issues are split into 3 principle categories:

- **Secure by design** – general trust model or architecture issues, probably cannot be fully removed without major changes. Mitigation is possible through procedures and policies.
- **Secure by default** – software or asset issues which are introduced by insecure defaults or assumptions.
- **Secure by deployment** – deployment issues introduced by non-proper management or practices.

NOTE: The findings may not result direct vulnerability or risk but may contribute to such.

NOTE2: It is NOT complete security audit due to limited scope and resources available to author.

NOTE3: Issues and risks are not rated – they are dependent on procedures and practices used in specific environment.