DISSERTATIONES
INFORMATICAE
UNIVERSITATIS
TARTUENSIS
24

ARNIS PARSOVS

Estonian Electronic Identity Card and its Security Challenges

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS

24

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS
24

ARNIS PARSOVS

Estonian Electronic Identity Card and its Security Challenges

UNIVERSITY OF TARTU
Press

Institute of Computer Science, Faculty of Science and Technology, University of Tartu, Estonia.

Dissertation has been accepted for the commencement of the degree of Doctor of Philosophy (PhD) in computer science on February 26, 2021 by the Council of the Institute of Computer Science, University of Tartu.

Supervisors

Dr.	Jan Willemson Cybernetica AS Tartu, Estonia
Prof. Dr.	Dominique Unruh University of Tartu Tartu, Estonia

Opponents

Prof. Dr.	George Danezis University College London London, United Kingdom
Assoc. Prof. Dr.	Petr Svenda Masaryk University Brno, Czech Republic

The public defense will take place on April 9, 2021 at 14:15 via Zoom.

The publication of this dissertation was financed by the Institute of Computer Science, University of Tartu.

ISSN 2613-5906
ISBN 978-9949-03-570-0 (print)
ISBN 978-9949-03-571-7 (PDF)

University of Tartu Press
http://www.tyk.ee/

To Estonia – the world's most advanced digital society

ABSTRACT

For more than 18 years, the Estonian electronic identity card (ID card) has provided a secure electronic identity for Estonian residents. The public-key cryptography and private keys stored on the card enable Estonian ID card holders to access e-services, give legally binding digital signatures and even cast an i-vote in national elections.

This work provides a comprehensive study on the Estonian ID card and its security challenges. We introduce the Estonian ID card and its ecosystem by describing the involved parties and processes, the core electronic functionality of the ID card, related technical and legal concepts, and the related issues. We describe the ID card smart card chip platforms used over the years and the identity document types that have been issued using these platforms. We present a detailed analysis of the asymmetric cryptography functionality provided by each ID card platform and present a description and security analysis of the ID card remote update solutions that have been provided for each ID card platform. As yet another contribution of this work, we present a systematic study of security incidents and similar issues the Estonian ID card has experienced over the years. We describe the technical nature of the issue, mitigation measures applied and the reflections on the media. In the course of this research, several previously unknown security issues were discovered and reported to the involved parties.

The research has been based on publicly available documentation, collection of ID card certificates in circulation, information reflected in media, information from the involved parties, and our own analysis and experiments performed in the field.

CONTENTS

1. INTRODUCTION

For more than 18 years, the Estonian electronic identity card (ID card) has provided a secure electronic identity for Estonian residents. It has been the technological cornerstone for secure electronic authentication and legally binding digital signatures. The ID card has enabled secure identification in various e-services, the most ambitious of them being internet voting in national elections. The Estonian ID card roll-out started in 2002 and is considered to be one of the most successful deployments of smart card-based national ID card systems in the world with respect to dissemination and active use. From the 1.3 million Estonian residents, 67% have used the ID card electronically at least once in the second half of 2018 [1].

The purpose of this work is to provide a comprehensive study on the Estonian ID card[1] and its security challenges. The topic is very broad as the security of the ID card depends not only on the cryptographic functionality embedded in the chip, but also on the way it is used and the security of the whole ecosystem built around it. Nevertheless, we believe that this work has achieved its aim as it covers the main aspects of the ID card and its security in a historical and multidisciplinary perspective. The central thesis of this work is: *Despite its success, the Estonian ID card and its ecosystem has experienced and still experiences technical, organizational and managerial deficiencies that affect its security.*

This work is based on publicly available documentation, information reflected in media, information from the involved parties, and our own analysis and experiments performed in the field. An important dataset that was used for our own validation and analysis was the collection of ID card public-key certificates that have been accumulated over the years from the public ID card certificate repository.

The thesis follows the monograph style, referring to and briefly summarizing our original publications listed at the end of the thesis. Due to the historical nature of the research, the thesis is rich with the precise dates of when events took place. The short ISO 8601 ("YYYY-MM-DD") date notation is used throughout the document when referring to dates.

1.1. Research questions and tasks

The main research question that this work aims to answer is: *What are the security issues that the Estonian ID card and its ecosystem has experienced over the years?*

To answer this question, we first define the field (the ID card ecosystem) and then study it in a systematic manner. During this process, we enumerate

[1] This work does not cover aspects related to the mobile phone-based electronic identity solutions used in Estonia, such as Mobile-ID and the recently introduced Smart-ID.

previously known and unknown security issues and the corresponding technical and procedural measures that were put in place to address them. To achieve this goal, we define and solve five research tasks:

1. *Provide a comprehensive study of the Estonian ID card ecosystem.* A wide variety of information about the Estonian ID card and its ecosystem is available. However, it has been rather dispersed, not being able to provide a holistic overview of the ecosystem. This task aims to map the main components of the ID card ecosystem, describe their role and purpose, use cases and interrelations and the historical and legal context, with a particular focus on the related security aspects.
2. *Study the ID card chip platforms that have been in use.* Over the years, the ID card's electronic functionality has been implemented using five smart card chip versions that rely on different software and hardware implementations. However, comprehensive research on these platforms and their differences is nonexistent. This task seeks to identify these different platforms and study their functionality, mainly focusing on cryptographic functionality, performance characteristics, and compliance to standards.
3. *Study and analyze ID card remote update solutions.* At times the Estonian ID card has employed unique technical solutions that have enabled ID card cardholders to update data and software on their cards remotely over the internet. However, the description of the protocols used to implement this functionality and their security analysis has been missing. The goal of this task is to document the protocols used and analyze their security.
4. *Investigate previously known security issues.* Over the years, information about several ID card related security incidents has become publicly known. However, the information that has reached the public space has been rather limited and fragmented, unable to provide the full picture of the known security issues the Estonian ID card has experienced. This task aims to put these publicly available bits and pieces together, placing them into chronological context, validating them and extending on the findings when possible.
5. *Uncover previously unknown security issues.* Since this field has not been heavily researched before, we can assume that only some of the security issues have emerged to the public. Systematic work on the research tasks listed above should lead to a discovery of security issues and other findings that were not previously known.

After the main research question of this work is sufficiently answered, we extend this work by proposing the answer to an additional research question: *How sufficient are the technical, organizational and managerial mechanisms at addressing the ID card related security challenges?* The answer to this question is provided in the last two chapters of this work.

1.2. Methods and data sources

This work falls under the field of observational research [2]. First, we use the descriptive study to gather in-depth qualitative data about the Estonian ID card, its ecosystem and the security issues it has experienced, and then we conduct an exploratory study to establish reasons behind these issues and to infer common traits in their handling.

This work is mostly based on publicly available information. We have analyzed the smart card interface specifications for the Estonian ID card [3–13], information available in the Certificate Authority's (CA) documentation repository [14], security certification documentation of the ID card components, and hundreds of news articles and other online resources referenced in this work. We have gathered additional information from the involved parties and by performing our experiments with actual ID cards.

An enormously useful data source for this research was the ID card certificate dataset described in detail in Section 2.17.1. We used it in conjunction with other data, such as certificate revocation data, to validate the scale of the publicly known ID card incidents and performed data analysis to discover previously unknown artefacts and anomalies.

1.3. Contributions

The main contributions of this work can be summarized as follows:

1. *We provide a comprehensive overview of the Estonian ID card and its ecosystem.* This is the first work to provide a broad overview of the Estonian ID card and related aspects. This work discusses practical aspects, insights and nuances not covered elsewhere. An important contribution of this work is the provided legal and historical context around the topic. Throughout this work we have discovered several legal non-compliances and have highlighted shortcomings of the Common Criteria security certification scheme. This work can serve as a starting point for the scholars interested in the Estonian ID card and its ecosystem.
2. *We provide several in-depth studies on the ID card related aspects.* In separate academic publications, we have published our research on the practical security aspects of using the ID card for authentication to e-services [15] and the use of the ID card for authenticaton to machines [16]. In separate publications we have also published our interdisciplinary research on the compliance of the Estonian digital signature scheme to the legal requirements in the context of the time of signing [17] and the legal challenges involved in solving the ID card security incident in 2017 [18].
3. *We provide a comprehensive study of the ID card technological platforms and identity document types used throughout the years.* This is the first

study to provide a detailed overview of the Estonian ID card technological platforms and the corresponding identity document types. For each platform, the description of the technological solution used to implement the electronic functionality of the ID card is provided together with the photos of the smart card microcontroller and images of the corresponding identity document specimens. We have analyzed the cryptographic functionality provided by each platform, its performance characteristics and compliance to standards. By applying various black box analysis methods we have partially recovered the RSA key generation algorithms implemented by the platforms. We have found that for several ID card platforms the chips supplied by the ID card manufacturers do not correspond to the certified version as defined in the official specification.

4. *We document and analyze the ID card remote update solutions used in Estonia.* In this work we have documented and analyzed the protocols used to implement the ID card remote update functionality. As a result, we have found that the ITSEC-certified secure messaging protocol had a cryptographic flaw that allowed card impersonation attacks to be performed and the implementation of the GlobalPlatform secure messaging protocol had a padding oracle vulnerability, allowing the decryption of communication. These findings have helped to improve the security of the remote update solution and we hope that this work serves as a useful reference for the development of future remote smart card update solutions.
5. *We present a comprehensive study of publicly known security incidents and other issues the Estonian ID card has experienced.* This is the first study providing a detailed overview of incidents the Estonian ID card has experienced. The events that took place have been reconstructed based on bits and pieces covered in public resources. When it was possible, we have gathered additional information by contacting the parties involved, analyzing our certificate dataset and conducting our own analysis and experiments. We have put forth the effort to determine the cause and impact of the incident, the incident response taken by the involved parties and the public communication. The study provides several previously unknown insights and can serve as a basis to draw further lessons from the incidents. The experience from the Estonian ID card provides other countries implementing nation-wide PKI a comprehensive overview of the issues that they may encounter.
6. *We discover previously unknown ID card security flaws and related issues.* In the course of this work we have discovered various security issues of different severity. The most significant findings related to the security flaws in key management have been published as a separate academic publication [19]. These findings have resulted in real-world implications –

the recall of the affected ID cards and a litigation process against the ID card manufacturer Gemalto.

7. *We derive broader findings for improving the security of the Estonian ID card and its ecosystem.* We have drawn broader findings and have proposed recommendations that can be used as an input by the state when making policy changes related to the organization of ID card security.

1.4. Structure of the thesis

The subsequent chapters of this work have been divided as follows:

- Chapter 2 introduces the Estonian ID card ecosystem, among other things, providing an overview of: the parties that are involved in the ID card manufacturing process, issuance and supervision; the core electronic functionality of the ID card and the related legal concepts; the use of authentication, decryption and digital signing, and the related issues; the related support components such as smart card readers and the ID card software; the certificates, the public certificate repository and the `@eesti.ee` email address; and the lifecycle of the ID card and its certificates.
- Chapter 3 proceeds by chronologically introducing the ID card platforms (smart card chip versions) used over the years and the identity document types that have been issued using these platforms.
- Chapter 4 describes in detail the asymmetric cryptography functionality provided by each ID card platform. By analyzing the properties of the keys and the timing information of the cryptographic operations, we attempted to recover the implementation details of the asymmetric cryptography algorithms used on each platform. We found that on each platform the cryptographic algorithms were implemented with slight differences.
- Chapter 5 provides a description and security analysis of the ID card remote update solutions. We outline the secure messaging protocols that are used to implement card management operations and describe the remote update solutions that have been used for each ID card platform.
- Chapter 6 presents a list of security incidents and similar issues that the Estonian ID card has experienced over the years. Some of the issues listed in this chapter were found by us and reported in the course of this research.
- Chapter 7 discusses the broader findings of this work, providing a list of recommendations that, in our opinion, could strengthen the security of the Estonian ID card and its ecosystem.
- Chapter 8 provides the concluding statements for this work.

2. ESTONIAN ID CARD ECOSYSTEM

This chapter provides a comprehensive overview of the Estonian ID card and its ecosystem. We start by providing the historical background of how the ID card was introduced and continue by describing the involved parties and processes, the core electronic functionality of the ID card, related technical and legal concepts, and the related issues.

2.1. Historical background

In 1992, after Estonia obtained full independence from the Soviet Union, the Citizenship and Migration Board (CMB) started to issue passports. The first generation of passports were valid for 10 years and hence had to be renewed in 2002. The Estonian government decided to use this as a chance to introduce a new type of identity document in the form of a national identification card (ID card). The main purpose for introducing the ID card was to provide Estonian residents with the means for digital signing under the Digital Signatures Act whose drafting process started in 1997 and completed in March 2000. [20, 21]

In 1997, CMB published their internal plans on the development of the ID card and soon after several public and private entities became interested and involved [21]. In May 1998, the Ministry of the Interior formed a committee for the development and preparation of the issue of the identification card and its technical specifications. In June of the same year, the committee issued a call for tenders for the initial research on the ID card. In July, the committee selected AS Aprote to perform the preliminary survey of the ID card and Küberneetika AS to perform the preliminary research on the ID card technologies. By the end of 1998 both reports were completed and the committee set out the starting points for future activities. [22]

The preliminary ID card survey [23] envisioned the ID card as a multi-functional card in which private and public entities would include data about the cardholder. This, however, did not materialize as the ID card is merely an authentication tool, with the information about the cardholders stored in the respective databases of these entities.

The preliminary study on ID card technologies [24] provided an overview of smart card technologies, standards and ongoing pilot projects worldwide. This study and further reports [25–27] recommended the Estonian ID card profile to be based on the Swedish SEIS standards that were significantly further developed by the Finnish FINEID national ID card standardization activities. As a result, it was decided to base the electronic functionality of the Estonian ID card on the Finnish FINEID specifications, using two public-key certificates for the purpose of authentication and digital signature, and adding improvements mainly in the form of additional card management functionality.

While the idea of a digital signature was new, the problem of online authentication already had a solution. In 1996, Estonian banks started to introduce online banking using password cards and PIN calculators to authenticate their clients [20]. As the banks already had an authentication infrastructure in place that could be used to authenticate a large part of the Estonian population, in the late 1990s, banks started to provide a federated authentication service (the so-called bank link) to third parties [28]. The provision of such service was not legally regulated, but as banks were considered trustworthy, several governmental e-services, such as e-Taxation and Citizen Portal, relied on bank authentication to provide e-services to citizens [20]. Today, this authentication option is still used in Estonia but is being slowly deprecated for governmental e-services [29].

The final decision to issue ID cards was made in March 2000 and the first cards were issued in 2002 [21]. However, the final decision to make the ID card a mandatory identity document for all Estonian residents aged 15 and above was only made at the end of 2001 [20, 30]. As we know it now, this decision was the key factor in the success story of the Estonian ID card [21].

2.2. Main parties

In the manufacturing and issuance of the Estonian ID card, several private and public parties are involved. There is a smart card manufacturer that produces the smart card chip microcontroller and, usually, also the core software (operating system) for it. Then there is an ID card manufacturer that incorporates the chip into a plastic card, prints cardholder information on the card and personalizes the chip by loading the electronic information on the card. During the process of personalization, cryptographic keys are generated and public-key certificates are loaded into the card. The public-key certificates are issued by a trusted party called Certificate Authority (CA). The purpose of a certificate is to establish a trusted binding between the generated public key and the cardholder's identity, allowing relying parties to later link the actions performed with the cryptographic key to the corresponding cardholder. The system governing certificate issuance and management is called a public key infrastructure (PKI). The reliance on certificates means that the security of the ID card depends on the PKI and the related cryptography as much as it depends on the security of the smart card chip.

The process described above produces an identity document that can be issued to the cardholder. The document issuer is a government entity that ensures that the ID card is distributed to the rightful person. In addition, there are government entities that are responsible for the development and oversight of the electronic identity (eID) field in Estonia.

Figure 1 shows the interaction between the main parties involved in the ID card manufacturing and issuance process. In the sections below we describe the involved parties and the related components in more detail.

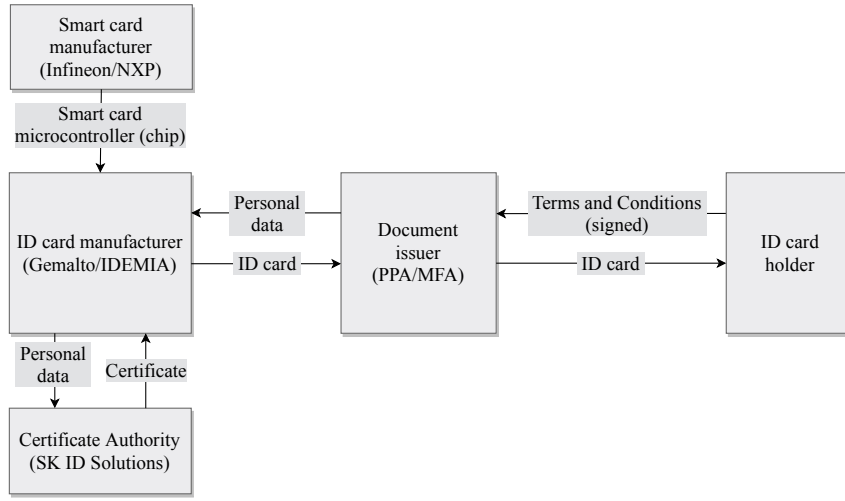


Figure 1: ID card manufacturing and issuance process

2.3. Document issuance

The ID card is an identity document issued by the Estonian state as a public service. The issuance of identity documents is regulated by the Identity Documents Act [31] (IDA) and related government regulations [32–35]. There are several types of identity documents that we refer to in this work as the ID card. These document types will be introduced and described in detail in Chapter 3.

Initially the authority issuing the ID cards was the Citizenship and Migration Board (CMB). In 2010, CMB became a part of the Police and Border Guard Board (Politsei- ja Piirivalveamet – PPA) and currently all types of ID cards are issued by PPA, except the *diplomatic identity card* which from the legal viewpoint is issued by the Ministry of Foreign Affairs (MFA). PPA is the authority responsible for the procurement of ID card manufacturing services on behalf of the Estonian state.

The customer service points of PPA provide several ID card-related services: applying for, receiving and revoking an ID card; suspending, terminating suspension of, revoking and updating ID card certificates; and receiving a new PIN envelope. Some of these services were also previously provided in the branch offices of Swedbank and SEB banks. IT support for managing ID card issuance and the related activities is provided to PPA by the IT and Development Centre at the Estonian Ministry of the Interior (Siseministeeriumi infotehnoloogia- ja arenduskeskus – SMIT).

PPA is also the national authority responsible for operating the Country Signing Certification Authority (CSCA). CSCA's purpose is to ensure the cryptographic authenticity of ICAO-compliant electronic machine-readable travel documents (eMRTDs) issued by Estonia [36], i.e., ePassports and *residence permit cards* (see Section 3.3.4).

2.4. ID card manufacturing

In 2000, after the concept of ID cards began to materialize, the Ministry of the Interior issued a public bid where six companies submitted an offer to manufacture the ID cards: Bundesdruckerei GmbH, TRÜB AG, Setec OY, Maurer Electronics GmbH, Gemplus and the Estonian company ID Süsteemide AS [37]. At the end of 2000, a five-year ID card service purchase agreement was signed with the Swiss company Trüb AG [21]. A separate contract to provide qualified certificates for the ID cards was concluded with Sertifitseerimiskeskus AS (SK). In addition, the Estonian company ID Süsteemide AS was contracted to develop the EstEID smart card interface specification [3] to run on top of the MICARDO smart card operating system (OS) platform.

Due to the contractual requirement that ID cards must be personalized inside the country [38], in 2001 Trüb AG established subsidiary company Trüb Baltic AS with a card personalization center in Tallinn. The personalization center was located in the high-security cellar of Hansapank (now Swedbank) head office and the ID card production line was able to produce 350 ID cards per hour [39] with the full capacity of 5 000 cards per day [40].

In January 2002, the first cards were issued [40]. The standard state fee for the card was 150 kroons (10 EUR), while the actual cost of production was 350 kroons (24 EUR) [30, 41]. The initial ID card agreement faced sharp criticism as being unfavorable for the state. It was claimed that the state overpaid by several millions of euros for the manufacturing equipment [38, 42] and that the same production line was actually also used to produce payment cards for banks [43, 44].

In 2007, a new contract with Trüb came into force. The contract was concluded in 2006 without a public procurement process. According to CMB, this was done to avoid a possible security risk and to maintain the same infrastructure for ID card issuance. Starting with this contract the provision of certification services was not a separate contract but rather a part of the ID card manufacturing contract with Trüb purchasing certification services from its subcontractor SK. [45]

In 2010, a new contract was signed with Trüb. In this procurement only Trüb submitted an offer [46]. In 2011, new ID cards with an updated design were introduced. The state fee was increased from 150 kroons (10 EUR) to 24 EUR [47]. In February 2015, Trüb AG was acquired by Gemalto (formerly Gemplus), with Gemalto taking over the contractual liabilities of Trüb.

Over the years, Trüb has outsourced tasks that are related to the development of chip functionality and personalization to several local contractors, such as RAULWALTER OÜ, Proekspert AS and Ideelabor OÜ.

At the beginning of 2015, PPA proclaimed a classified procurement for ID card manufacturing for the period 2018–2023, sending invitations to four selected companies: Giesecke & Devrient, Morpho, Gemalto and Oberthur [46]. From the four companies Morpho, Gemalto and Oberthur made an offer [46]. In spring of 2016, the procurement committe announced Morpho as the winner, but

due to the complaints of Gemalto and Oberthur the decision was annulled [48], the technical specification was updated and a new procurement process was initiated [49]. In spring 2017, PPA signed a contract with Oberthur for ID card manufacturing beginning from 2019 [50]. The decision of this procurement was appealed again – this time by Gemalto and Morpho. At the end of 2017, the first instance court dismissed Gemalto's complaint, ruling Oberthur the winner [51]. Oberthur's victory was overshadowed by the news that the World Bank imposed a two year ban on Oberthur for corruption charges in Bangladesh [52]. Gemalto later appealed the decision in the court of second instance [53] and later to the Supreme Court [54]. PPA and Gemalto had additional litigations that are discussed later in Section 6.7.4.

The current agreement with Gemalto was valid until the end of 2018 [49], with an additional 5-year after-contract warranty period. At the beginning of 2019, Gemalto ceased their operations in Estonia [55], providing PPA with a money-back guarantee for the warranty period.

As a result of the merger of Oberthur Technologies and Safran Identity & Security (Morpho), PPA's contractual partner Oberthur is now known as IDEMIA. IDEMIA chose SK as its subcontractor for provision of certification services, and the Estonian company Hansab AS for performing card personalization in Estonia. The first ID cards manufactured by IDEMIA were issued at the end of 2018.

It is important to note that currently there are only 5 major companies in the world that manufacture smart card security microcontrollers. These are Infineon, NXP, Samsung, STMicroelectronics and EM Microelectronic [56]. The other companies in the smart card business write software for these chips and build products to run on top of these chips. The ID cards manufactured by Gemalto were built on the chips produced by Infineon, while the ID cards manufactured by IDEMIA uses the chip manufactured by NXP.

The manufacturing of ID cards has always been a closed, non-transparent activity, not open to scrutiny even to the manufacturer's contracting partner – the Estonian state. The personalization protocols and procedures have never been publicly documented, leaving the security aspects of this process to be determined by the competency of the ID card manufacturer. As described in Section 6.8 of this work, this lack of supervision oversight allowed the ID card manufacturer to engage in activities that compromised the ID card security without it being detected for years.

2.5. Certificate Authority

From the introduction of the ID card in 2002 until today, the ID card certificates are issued by the privately-owned Estonian company SK ID Solutions AS (formerly AS Sertifitseerimiskeskus – SK).

SK was established in 2000 to create eID solutions to enable authentication

and digital signatures [57]. SK is owned with equal shares by two of the biggest banks in Estonia – Hansapank (now Swedbank) and Eesti Ühispank (now SEB), and two Estonian telecom operators – AS Eesti Telefon (later Elion) and AS Eesti Mobiiltelefon (later EMT). In 2016, Elion and EMT were merged under the Telia name, which gave Telia 50% of the SK shares [58].

Estonian banks have played an important role in promoting the use of the ID card. For example, the Estonian Banking Association from 2007 to 2009 gradually decreased the password card daily transaction limit to 3 000 kroons (191 EUR), requiring the use of an ID card or PIN calculator for transactions above that limit [59]. In the years 2013–2015, online banking has been responsible for around 65-75% of all the ID card transactions [60].

Historically, SK has been an eID competence center in the private sector. Before the state actively engaged in eID development, SK privately financed the development of ID card software, designed digital signature file formats and promoted development of e-services [20].

Legally, SK is a qualified trust service provider (QTSP) that provides qualified certificates for electronic signatures and seals, qualified electronic time stamps and other trust services. This means that digital signatures created using SK certificates can provide equivalent legal effect as a handwritten signature.

2.5.1. Legal framework

On the European Union level, requirements for qualified trust service providers are regulated by Regulation (EU) No. 910/2014 [61] (eIDAS). The aspects not regulated by eIDAS are regulated by the Estonian national law – Electronic Identification and Trust Services for Electronic Transactions Act [62] (EITSETA). In the context of trust services, PPA acts as a registration authority (RA) of SK, performing identity verification for certificate applicants. Before eIDAS, the requirements for certification service providers (pre-eIDAS terminology) were prescribed in the national Digital Signatures Act [63] (DSA), which implements EU eSignature Directive 1999/93/EC [64] (Directive).

The compliance to DSA requirements had to be confirmed in annual information systems audits. Under eIDAS, the QTSP has to be audited every two years by an accredited conformity assessment body. SK has been audited from 2002 – 2014 by KPMG Baltics AS, and later, to be in compliance to eIDAS requirements, by TÜV Informationstechnik GmbH and TÜV AUSTRIA CERT GMBH. As we will see later in this work, the assurance level provided by these audits is rather limited in practice.

The contractual relationships between the cardholder and SK are regulated in the document “Terms and Conditions for Use of Certificates of Personal Identification Documents of the Republic of Estonia” [65] (Terms and Conditions) and other documents referenced therein, such as the Certification Practice Statement (CPS) and certificate policies (CPs). The Terms and Conditions is also considered to be legally binding towards any relying party that

relies on the trust services provided by SK. Without accepting the Terms and Conditions, the ID card is not issued [66]. As the ID card is a mandatory identity document in Estonia, the contractual relationships with SK are forced upon every Estonian resident age 15 and above.

Starting from 2017-07-01, the lawmaker has declared SK a provider of vital services [67] as the continuity for services provided by SK are of crucial importance to the Estonian state. In 2019, the government issued a regulation specifying the requirements for ensuring the continuity of digital identification and digital signing [68].

The perception of SK's sovereignty in the ID card context has varied between SK being an independent CA providing certification services for the security token (ID card) issued by the Estonian state, versus SK being the government's CA that is only technically run by SK as a legal entity. With the introduction of IDEMIA-issued ID cards this has visibly shifted to the latter: (1) the certificates for IDEMIA-issued ID cards are issued under a separate SK root CA named EE-GovCA2018; (2) the activation of a private key requires the presence of a PPA representative (Section 6.2.2 in [69]); and (3) the authorship of the certificate policy [70] now belongs to PPA. The state's wish to infringe upon the decisions of SK was strongly visible in 2017 when the ID card crisis had to be solved (Section 6.7.3).

2.5.2. Business model

SK is paid by the ID card manufacturer for every certificate issued for the Estonian ID card. The conditions of the ID card manufacturing contract are confidential, therefore it is not publicly known how much revenue SK receives from ID card certificate issuance. However, we know that CMB paid SK 60 kroons (4 EUR) for each ID card (which contained two certificates) in the years preceeding 2007 when the state had a separate contract with SK [41].

Additionally, e-service providers who wanted to enable ID card authentication in their web services had to pay SK for each query made to the OCSP (Online Certificate Status Protocol) service that establishes whether a certificate has been suspended or revoked. The revocation information was freely available in CRLs (Certificate Revocation List), but the information there is not up-to-date. Some e-service providers also opted to use the public LDAP service to establish certificate validity free of charge (see Section 2.17). After eIDAS came into force, SK was required to provide free access to the OCSP service. However, SK continues to also offer a commercial OCSP package with improved support [71].

As every digital signature is expected to be accompanied with validation data consisting of OCSP confirmation and a cryptographic timestamp, SK was paid for every digital signature created with the ID card. In practice, the cost for digital signing is covered by the e-service providers who offer digital signing in their web environments, as every ID card user is only allowed to create up to 10 digital

signatures in a month free of charge on their computers for personal purposes[1]. For digital signature formats DDOC and BDOC (timemark profile), the OCSP response also served as a timestamp (see Section 2.10.2). The ASiC-E format (BDOC timestamp profile) prescribed by eIDAS requires the use of a separate timestamp, which alternatively could be purchased from another QTSP, thereby bypassing the need to use SK commercial services for digital signature creation.

Currently, SK's dependency on revenue from the ID card and related services has decreased. For several years SK has been providing Mobile-ID service in Estonia and Lithuania [58], as well as the recently launched Smart-ID service, which has experienced huge growth in the Baltics [73].

Additional SK revenue comes from operating the Estonian Country Signing Certificate Authority (CSCA) that PPA had outsourced to SK since the introduction of ePassports in 2007. This service is provided for PPA under a separate contract and the operation of the Estonian CSCA is not audited in the context of trust services.

2.5.3. Customer service points

Because of the owners, SK has had a special relationship with Swedbank and SEB, which allowed the bank branch offices to be used as SK customer service points. The customer service points of SK provided services to: receive a new PIN envelope; suspend, terminate suspension of, revoke and (in some cases) update ID card certificates. From 2002 to 2007-01-15, the state had a cooperation agreement with SK which allowed the ID card applicants to collect their ID cards in branch offices of Swedbank and SEB [74]. From 2019-03-01 ID card-related services are provided only in PPA customer serivice points [75]. SK, however, still provides 24/7 phone service for certificate validity suspension requests.

2.6. Oversight and development of eID field

The Ministry of Economic Affairs and Communications (Majandus- ja Kommunikatsiooniministeerium – MKM) creates national development plans and policies and performs general coordination in the field of ICT. Under eIDAS (and previously under the Directive) MKM has been a supervisory body but has delegated some of the duties to subordinate government organizations. Under the Directive, MKM also acted as the conformity assessment body that evaluated the compliance of signature creation devices to the security requirements (see Section 2.10.1).

The Estonian Information System Authority (Riigi Infosüsteemi Amet – RIA) is the state agency under MKM that is responsible for the coordination and development of electronic identity (eID) and cyber security. Estonian national CERT (CERT-EE) runs under RIA's cyber security branch. RIA is considered the

[1] This has been a soft limit which has been enforced based only on an IP address [72].

eID competence center in the public sector. RIA organizes the development of ID card software (see Section 2.13) and as of 2016 is responsible for the digital elements on the ID card [76].

Starting in 2019, RIA was given the role for supervision of trust service providers, to ensure that their services comply to the requirements set forth in eIDAS. Before 2019, this function was performed by the Technical Regulatory Authority (previously known as Technical Surveillance Authority) (Tehnilise Järelevalve Amet – TJA). TJA did not have the technical competence in this field and it performed its supervisory function rather formally – just receiving the compulsory conformance audit reports from the trust service providers and updating the registry [20].

While RIA has the competence, its involvement in the development of the eID field conflicts with its supervisory function. Unfortunately, considering the state's direct involvement in the development of the eID field, the question is whether the state can at all objectively supervise the trust service providers that provide trust services for the state-backed eID solutions.

2.7. Electronic functionality of the ID card

From its introduction in 2002 until now, the core electronic functionality provided by the Estonian ID card has stayed the same. The ID card contains two asymmetric (RSA or ECC) keys with the corresponding X.509 public-key certificates, and symmetric keys to perform card management operations.

Authentication key. The authentication key is used to log into e-services by providing a signature in the TLS client certificate authentication process (see Section 2.8). This key can also be used to decrypt documents encrypted for the cardholder (see Section 2.9). Cryptographic signature and decryption operations with this key have to be authorized using the 4-digit PIN1 code. The PIN verification mechanism is described in detail in Section 2.11.

Digital signature key. The digital signature key is used to give legally binding digital signatures that under eIDAS are recognized as qualified electronic signatures. Each signature operation with the key has to be authorized using the 5-digit PIN2 code. The digital signature function and the related concepts are further discussed in Section 2.10.

Key usage counters. The publicly readable key usage counters show how many private key operations have been performed with a particular private key (Section 12.4 in [9]). The counters can be used to determine how active the cardholder is in regards to using the ID card for authentication and digital signatures. This feature is not present on the IDEMIA-powered ID cards introduced in December 2018.

Personal data file. The ID card chip contains a publicly readable personal data file, which consists of 16 records containing the same information as is printed on the card. The use of this functionality is discussed in Section 2.12.

Card management operations. The cards are preloaded with symmetric keys that can be used by the manufacturer to perform various card management operations in the post-issuance phase. This provides a method to reset PIN codes in the event the cardholder forgets them, generate new keys, write new certificates, and even reinstall the whole smart card applet if needed. The card management keys and secure messaging are discussed in Section 5.1.

2.8. Authentication function

The most popular use case of the authentication key is to authenticate to web services over the internet using the TLS client certificate authentication protocol. Less common use cases for the authentication certificate include signing email with S/MIME, authenticating over SSH and VPN, and logging on to a workstation [77].

The legal status of the authentication certificate is regulated by the Identity Documents Act (IDA). IDA refers to the authentication certificate using the term "the certificate enabling digital identification". According to IDA clause 18^1 (2), "the digital verification of the identity of the holder of a document is carried out through the certificate enabling digital identification". IDA clause 18^1 (3) gives the right for the service providers in the public sector to require the use of authentication and digital signature certificates in the process of providing e-services. The providers of public services, however, are not restricted and may also decide to provide e-services using less secure options, such as bank link authentication [28] or passwords. The exception here is i-voting, where the electronic voting procedure described in Riigikogu Election Act [78] up to 2017 required the use of the authentication certificate for voter identification.

eIDAS and EITSETA do not specify requirements for the authentication certificate, therefore it does not have to be issued by a QTSP. However, since the authentication certificate is issued under the same CA and the same Terms and Conditions, the certificate policy documents and other CA statements are applicable to both certificates.

Starting from 2018-11-07, the authentication functionality of the ID card became part of the notified eIDAS electronic identification scheme with assurance level "high" [79], requiring it to be recognized by other EU member states.

In practice, all noteworthy public and private service providers in Estonia support the ID card authentication mechanism for accessing their systems. At the end of 2020, we compiled a list of more than 200 Estonian e-services [80] that support the ID card authentication. These systems usually allow people to see the data collected about them. The abuse potential for unauthorized use of ID card authentication has been nicely demonstrated by the artistic installation "Memopol" [81], which gathered and displayed various details about the user once the cardholder inserted their ID card and PIN1 in the machine.

2.8.1. TLS client certificate authentication

The TLS protocol provides a client certificate authentication (CCA) feature that is supported by all major browsers and TLS server implementations, and is used by e-service providers in Estonia to implement ID card authentication in their services. In the TLS CCA process the client sends their authentication certificate to the server and proves access to the corresponding private key by signing the TLS protocol handshake. In our paper "Practical Issues with TLS Client Certificate Authentication" [15] we studied this use case and the related practical issues in detail by analyzing the TLS CCA implementations of 87 Estonian e-service providers.

TLS CCA provides very strong protection against man-in-the-middle (MITM) attacks, as the attacker who is impersonating the server cannot reuse the client's signature on the handshake to authenticate to the legitimate server (Section III-K in [15]). However, we found that in practice only 7% of Estonian service providers (mostly banks) also require the presence of a TLS CCA connection after the login phase. Other providers rely on an HTTP session cookie that can be stolen in a successful server impersonation attack (Section IV-B-4 in [15]).

At the time of the study, most of the service providers were using the Apache HTTP server with the module `mod_ssl` to implement TLS CCA. Most of the service providers (87%) were checking the revocation status of the client's certificate. A large part of service providers (47%) had superfluous CAs in their trust store and 45% of service providers had a longer chain verification depth than needed. While these misconfigurations are not directly exploitable, we found that the two biggest Estonian banks were using F5 BIG-IP SSL load balancer with a misconfiguration that allowed the TLS CCA to be bypassed using a fake authentication certificate whose CA signature was not valid [82, 83]. At the end of 2020, Semjon Kravtšenko (supervised by the author of this work) repeated the experiment and found 4 other e-services (`cooppank.ee`, `elisa.ee`, `printincity.ee` and `arved.ee` [84, 85]), whose ID card authentication implementation had a similar flaw.

The use of TLS CCA introduces a privacy issue, as the client certificate is sent over the network in plaintext unless TLS CCA is requested in the TLS renegotiation process. We observed that 33% of the e-service providers request TLS CCA in the initial handshake, resulting in the client's certificate being sent unencrypted. TLS v1.3 has introduced mandatory encryption of the TLS `Certificate` message. However, support for TLS v1.3 is not yet widespread.

There are plans currently in development to introduce an alternative ID card authentication method by performing authentication on the application level using a browser extension that signs a challenge with the authentication key [86].

2.9. Decryption function

The authentication key[2] of the Estonian ID card can also be used to decrypt data encrypted for the corresponding cardholder. The public keys of the cardholders' can be found in the public LDAP certificate repository (see Section 2.17) and the authentication certificate contains an `@eesti.ee` email address assigned to the cardholder (see Section 2.18). In theory, this allows a person to initiate encrypted communication with an ID card holder without having prior contact. The requirement, however, is that the personal identification code of the recipient is known and the recipient has enabled email forwarding for their `@eesti.ee` email address.

The encryption and decryption functionality is implemented in the ID card software [87] (see Section 2.13). By specifying the recipient's personal ID code, the encryption utility retrieves the authentication certificate from LDAP and encrypts data using the corresponding public key. The encrypted data is stored using the XML-based Estonian-specific CDOC encryption file format. The utilization of CDOC is the most popular ID card encryption use case, although the encryption function can also be used to encrypt emails with S/MIME[3] [77].

CDOC encryption is widely used in the private and public sector for sending sensitive email attachments. The government's regulation for document transmission in offense proceedings [88] requires delicate personal data to be sent using ID card encryption. A popular use case is sending CDOC encrypted fines for traffic violations to natural persons [89]. Since the encryption requirements for transmission of state secrets are classified, it is not publicly known for which levels of classification the ID card encryption functionality can be used (see page 50 in [90]).

ID card encryption can only be used for short-term transmissions, as the renewal or replacement of the ID card would render the CDOC unopenable. Since the CDOC encryption scheme does not provide forward secrecy, the compromise of the private key would put the confidentiality of all CDOCs encrypted for that key at risk, as it was in the case of the ID card crisis in 2017 (see Section 6.7). There is, however, an ongoing effort to address these concerns [91].

We note that the current ID card encryption scheme is not secure if the adversary is the Estonian state, as the state can decrypt the files by confiscating the recipients ID card (which is the property of the state) and resetting the PIN codes using the PIN replacement procedure (see Section 2.11.4). The encryption scheme, however, is secure against covert attacks by the state, assuming that the ID card certificates are not mis-issued and the state does not store the private keys of the cardholders.

[2] Technically, the ID card platforms issued up to fall 2014 (MICARDO, MULTOS and jTOP SLE66) also provided decryption functionality for the digital signature key. This, however, was not compliant with the certificate's key usage extension and was not used in practice.

[3] In practice, we have not seen anyone using the ID card for the S/MIME encryption or signing.

2.9.1. CDOC format

The CDOC [92] file format (`.cdoc` file extension) was established in 2005 and consists of a single XML file based on the XML Encryption specification [93].

The data is encrypted with a random symmetric 128-bit AES transport key and stored as a base64-encoded value in the XML. The symmetric transport key is encrypted with RSA PKCS#1 v1.5 encryption scheme[4] using the public key from the recipients authentication certificate. A single CDOC file can contain encrypted data for several recipients (it is a common practice for the sender to also add themselves as a recipient). In this case, the symmetric transport key is encrypted using the public keys of several recipients. The encrypted transport key is placed in a recipient-specific `EncryptedKey` element along with the authentication certificate of the recipient.

To encrypt more than one file, the contents of the files are embedded in an XML structure as base64-encoded values and then the XML structure is encrypted as if it was a single plaintext file. The base64 encoding of the plaintext introduces some noticeable overhead.

The document-wide `EncryptionProperties` element contains plaintext metadata about the encrypted files: the filename, size (in bytes) and the MIME type of the plaintext file.

The AES block cipher is used in CBC (Cipher Block Chaining) mode. The IV (Initialization Vector) is randomly generated and prepended to the ciphertext. The CBC mode does not provide integrity for the ciphertext, therefore an attacker can modify some parts of the ciphertext in a possibly meaningful manner. To protect against such ciphertext malleability attacks and to provide authenticity of the data and metadata, the sender can manually sign the CDOC file.

AES 128-bit cipher provides a 128-bit security level[5]. However, since the symmetric transport key is encrypted using a 2048-bit RSA key which provides a 112-bit security level, the overall security level of the encryption scheme is limited to 112 bits (or 80 bits for the ID cards using 1024-bit RSA keys).

2.9.1.1. Elliptic Curve (EC) support

Due to the flaw in the Infineon's RSA key generation algorithm (see Section 6.7), starting from November 2017, the public key algorithm used in the Estonian ID card was switched from 2048-bit RSA to 384-bit Elliptic Curve Cryptography (ECC) using NIST P-384 curve. The CDOC format specification was updated to enable ECC support [95].

The ECC-enabled ID cards provide ECDH (Elliptic Curve Diffie-Hellman) key agreement operations with the authentication key. Thus, for ECC recipients the transport key is encrypted using a 256-bit AES key which is derived from a 48-byte ECDH shared secret. The use of ECDH key agreement requires the sender

[4] RSA PKCS#1 v1.5 decryption is vulnerable to padding oracle attacks (see Section 6.3).

[5] If considering multi-target attacks, AES-128 may not provide 128-bit security, e.g., a 2^{80} attack will break an AES-128 key out of a batch of 2^{48} keys [94].

to generate an ephemeral EC key pair using the same P-384 curve and embed the public key in the CDOC. Hence, in case of ECC, the sender has to not only generate a random transport key, but also a random ECC key for each recipient.

In addition, the updated specification improves the security level of the transport encryption, replacing 128-bit AES that operates in CBC mode with 256-bit AES that operates in GCM (Galois/Counter Mode) mode. The authenticated GCM encryption mode prevents ciphertext malleability attacks, although the authenticity for the complete ciphertext and metadata has to be provided by other means.

While 256-bit AES provides a 256-bit security level, the overall security level for a CDOC that contains only ECC recipients is limited to 192-bits of security, since the 384-bit ECC provides a 192-bit security level. If the CDOC contains any 2048-bit RSA recipients, the security level of the scheme is further limited to 112 bits. It is important to note that this is not so in the case of quantum computer attacks as data encrypted using 384-bit ECC would be broken earlier than data encrypted using 2048-bit RSA (see Section 2.1 in [96] for estimates). Hence, if we consider quantum computer attacks, the switch to ECC actually makes the current ID card encryption scheme less secure against future adversaries.

2.10. Digital signature function

The digital signature key of the ID card can be used to give legally binding digital signatures. Under eIDAS this type of electronic signature is called a qualified electronic signature (QES) and under the Directive it was called an advanced electronic signature based on a qualified certificate and which is created by a secure-signature-creation device. The Estonian law uses the term "digital signature" to denote this type of legally binding electronic signature. EU regulation gives this type of signature the same legal effect as handwritten signatures, meaning that this signature should satisfy requirements for data in electronic form in the same way as a handwritten signature satisfies signature requirements for paper form documents. This, however, does not force the EU member states to accept digital signatures in every field as the member states are free to apply specific form requirements.

The public sector is required to accept digitally signed documents. Document exchange between government institutions is mainly performed electronically using digitally signed documents [97]. Court decisions and legal acts are signed digitally and the state provides various e-services that rely on the digital signature – from changing vehicle ownership online [98] to casting a vote over the internet in national elections [99]. The digital signature is also widely used in the private sector, with the most popular use cases being the signing of contracts and online banking transactions. According to `id.ee` statistics, more than one billion digital signatures have been created as of October 2020.

Technically, a signature given using the digital signature key is no different

from a signature given using the authentication key and they both can be used as evidence in legal proceedings. However, signatures given using the digital signature key have the presumption of authenticity established in subsection 277(3) of the Code of Civil Procedure [100]. This means that when a document signed using the digital signature key is presented as evidence, burden of proof is on the signatory to prove that it was not the signatory who signed the document. Furthermore, there are a few types of legal transactions that are required to be in written form. These transactions will be valid in electronic form only if given using the digital signature key (see page 44 in [17]).

The digital signature certificate contains certificate policy references and the `nonRepudiation`[6] flag in the key usage extension denoting that the key can be used to give legally binding signatures. While this key should not be used for other purposes, for several years the Firefox browser had a bug which allowed the user to select a digital signature certificate for authentication [101]. It turned out that some TLS server implementations such as Microsoft IIS for client authentication would also accept the digital signature certificate (see Section IV-B7 in [15]).

Today there are mainly two ways by which digital signatures are given. One is using the ID card software and signing the document on a user's device, and the other is signing in a web environment using the ID card browser extension [102]. When signing in the web environment, e-service providers use JavaScript function calls to obtain a signatory's signature on a hash value prepared by the service provider. This type of digital signing in a web environment provides a very poor evidentiary value in practice as the signatory is not able to see what is being signed and thus has to blindly trust that the service provider is not asking them to sign something that they did not intend to sign.

2.10.1. Signature creation devices

For an electronic signature to have equivalent legal effect as handwritten signatures, the eSignature Directive 1999/93/EC [64] required the signature to be created by a secure-signature-creation device (SSCD). The technological solution of the SSCD had to satisfy security requirements set out in Annex III of the Directive. The Directive required the conformity of SSCDs to be determined by a public or private body designated by member states, and Decision 2000/709/EC [103] specified the minimum criteria to be taken into account when designating such conformity assessment bodies. The conformity assessment bodies were not required to assess the conformance based on certifications such as ITSEC or Common Criteria, but Decision 2003/511/EC [104] stated that products compliant to technical standard CWA 14169 shall be presumed to be compliant with the SSCD requirements of the Directive.

Historically, the assessment of the Estonian ID card platforms' conformance to the security requirements has been done rather poorly. In Estonia the Ministry of Economic Affairs and Communications (MKM) acted as an SSCD conformity

[6] In the recent editions of X.509 standard the flag has been renamed to `contentCommitment`.

assessment body under the Directive. While SSCD conformance assessments had to be performed before introducing each ID card platform and after making any changes to the platform, only on 2016-03-04, after the eIDAS regulation appeared on the horizon, did MKM confirm SSCD compliance of the ID card platforms used in Estonia. The assessment [105] was based on the conclusions of the compliance assessment committee formed by four RIA employees. The assessment relied on security certificates of ID card components, available technical information and expert opinions (Section 6 in [106]).

The above-mentioned conformance assessment had purely formal qualities as the authorities have not been able to even determine the actual components that are used in some of the ID card platforms. While for some smart card components the committee could rely on security certifications, the EstEID applet has never been formally certified[7]. As we found from the 2011 ID card incident (Section 6.4), the applet did not even receive sufficient security testing before being put in production[8], and as we found from the incident of key generation outside the ID card (Section 6.8), the ID card manufacturer made unauthorized modifications to the applet without informing the authorities.

On 2016-07-01, eIDAS [61] regulation came into force, which requires a qualified electronic signature to be created by a qualified electronic signature creation device (QSCD). The security requirements for QSCDs largely stayed the same as for SSCDs, however, article 30 of eIDAS introduced a requirement for security certification of QSCDs. The Commission Implementing Decision (EU) 2016/650 [107] specifies Common Criteria evaluation standards and Protection Profiles according to which the QSCDs must be certified.

The ID card platforms used in Estonia before eIDAS came into force, are recognized under eIDAS as QSCDs not through the certification, but through the transitional measures (page 10 in [108]) laid down in article 51 of eIDAS, which states that signature creation devices, which under the Directive were recognized as SSCDs, are deemed to be QSCDs under eIDAS.

The discovery of the security flaw in 2017 (Section 6.7) showed that the affected ID card platform in practice does not satisfy the security requirements of SSCDs. While the affected platform was updated to work around the flaw, the reassessment of SSCD conformity was not done [18].

The IDEMIA-powered ID card platform introduced at the end of 2018 is the first ID card platform that has received a decent SSCD conformity assessment both in form and substance – unfortunately, not without compliance issues (see Section 3.5.5.1). In theory, the legal noncompliances of the platforms could be used to challenge the SSCD/QSCD status of the platforms and hence the validity of the digital signatures created with them.

[7] We note that the EstEID applet implements logical protection of private keys and hence is subject to conformity assessment (see Decision 2003/511/EC [104] and recital 56 of eIDAS).

[8] The compliance assessment committee, in its 2016-03-04 decision, also confirmed SSCD status for the vulnerable ID card platform issued in 2011 (the vulnerable and the fixed platform in the conformity assessment is recognized as one platform).

2.10.2. Signature file formats

In 2002, SK developed the digital signature file format DDOC [109] (DigiDoc) which served as the de facto Estonian digital signature file standard for more than a decade. The DDOC format is based on the XML Signature specification and the XAdES (XML Advanced Electronic Signatures) extension. It consists of a single XML file with the `.ddoc` file extension, where the signed files are base64-encoded and stored inside the XML structure. In addition to the signatory's signature and certificate, DDOC stores additional validation data to enable signature validation after the signatory's certificate has expired or has been revoked. According to standards, the validation data consists of a cryptographic timestamp over the signatory's signature proving the time of the signature's existence, and an OCSP response proving that at the time the signature existed, the signatory's certificate was still valid.

DDOC format relies on a clever non-standard solution that removes the need to have a separate timestamp: having the OCSP response also serve as a timestamp. The OCSP response's nonce extension is used to carry the hash of the timestamped data (signatory's signature). The SK OCSP responder that produces such OCSP timestamps is also audited as a timestamping service. The added functional requirement is that OCSP responses issued by the service have to be logged for evidentiary purposes. To improve the integrity of the logs, SK implemented the SeqLog hash chain system and they periodically publish hash values in the newspaper [110].

Today the DDOC file format is deprecated as it is not internationally recognized and is limited to the RSA cryptosystem and SHA-1 hash algorithm.

The development of the BDOC signature file format (successor of DDOC) started in 2008 and finished with the completion of BDOC version 2.1.2 [111] in 2013. The main difference compared to DDOC is that the signed files are now detached and stored together with signature XML files (separate file for each signature) in a single ZIP container following the ASiC (Associated Signature Containers) standard. In addition to the non-standard OCSP timestamps (so-called timemarks), the BDOC specification also supports a timestamp profile, enabling the use of a separate RFC 3161 standards-compliant timestamp. The signatures created using the BDOC timestamp profile are fully compliant with the XAdES file format specification stipulated by eIDAS [112]. eIDAS also stipulates two additional signature file formats – CAdES (CMS Advanced Electronic Signatures) and PAdES (PDF Advanced Electronic Signatures). However, these formats are not supported by the Estonian ID card software.

Signatures created using the BDOC timemark profile are usually stored using the `.bdoc` file extension, while the preferred file extension for signatures using the internationally recognized BDOC timestamp profile is `.asice` (ASiC Extended – ASiC-E). We note that the timestamps issued by the SK OCSP timestamp responder are not recognized under eIDAS as qualified electronic time

stamps, therefore the timestamp of the BDOC timemark profile does not have the presumption of authenticity provided in article 41(2) of eIDAS.

The support for creating BDOC files was already introduced in the ID card software v3.5 released on 2011-11-10. However, only in v3.10, released on 2015-03-05, was the BDOC using timemark profile set as the default format and the option for creating signatures using the timestamp profile was introduced. The support for creating DDOC signatures was dropped in v3.12 released on 2016-02-01. ID card software v18.12 released on 2018-12-03 removed the option to choose between BDOC profiles, making the internationally recognized BDOC timestamp profile the only format supported for signature creation[9]. [113]

Versions 1.0 and 2.1 of BDOC were established as Estonian standards [114, 115]. However, since no Estonian legislation has ever stipulated their use, the use of DDOC and BDOC signature formats in Estonia has been more of a social agreement. The Commission Implementing Decision (EU) 2015/1506 [112] now sets the BDOC format with the timestamp profile as one of the formats that has to be accepted by public sector bodies in the EU.

2.10.3. Signature validation

Unfortunately, there is no binding standard that would precisely describe the validation rules of a digital signature as yet. The general algorithm is described in ETSI EN 319 102-1 [116], but for its input parameters it relies on a signature validation policy that has to be provided by the party validating the signature. ETSI TS 119 172-4 [117] (currently in standardization process) plans to provide such a common policy for validation of eIDAS qualified electronic signatures.

In Estonia, the validation rules implemented in the state-provided ID card software are the de facto rules that are used to determine what is technically considered to conform to the legal term “digital signature”. The validation rules are implemented as understood by software developers and have changed over the years leading to a situation where signatures recognized as valid by older software versions are no longer recognized as valid by newer versions (and vice versa). For some file format errors introduced in previous versions, newer versions have backwards compatibility, returning the validity status “valid with warnings” [118].

From the introduction of the digital signature, Estonian law has required a digital signature to provide the ability to establish the time when the signature was created. However, in our article “Time of signing in the Estonian digital signature scheme” [17], we showed that contrary to this legal requirement, the Estonian digital signature scheme does not provide reliable proof of the time when a document was digitally signed. The “Signed on” field reported by the

[9]BDOC timestamp profile signatures created using the ID card software still use SK OCSP timestamp responder for OCSP responses, thereby providing an extra non-qualified timestamp that benefits from SK SeqLog integrity features.

ID card software shows the time from the cryptographic timestamp. However, the timestamp of the signature can only prove that the signed document existed at that particular time, while the document might have actually been signed much earlier, as the timestamp can be later added by anyone. To demonstrate this problem, we published a video [119] showing how the "Signed on" time of a digitally signed legal act downloaded from the Riigikogu website could be updated. Technically, the modification was trivial. The original cryptographic timestamp and OCSP response was removed and replaced with a new one. The timestamp could be updated in this manner by anyone at any time as long as the certificate of the signatory, used to sign the document, was still valid. This came as a great surprise to the public as it was widely believed that the signing time of a digital signature could be reliably established.

In the article, we also pointed out a more fundamental signature validation issue rooted in the validation algorithm's assumption that the signatory's certificate is valid from the moment of its issuance until its expiration or revocation. Since the ID card certificates, after their issuance, are in a suspended state and can also be suspended later in their lifetime (see Section 2.15), the validation process cannot provide assurance that the digital signature was given when the signatory's certificate was valid. This allows the validity of any digital signature created with the Estonian ID card to be challenged.

2.10.4. Long-term validity

The timestamp on a signatory's signature provides the theoretical ability to verify the validity of the signature as long as: (a) the second-preimage resistance of the hash function used to sign the document is secure; (b) the CA records of issued and revoked certificates are available; and (c) the records of issued timestamps are maintained by the timestamping authority (TSA).

To validate a signature without requesting offline evidence from the CA and TSA, the validation scheme must assure that at the time of validation the private keys of the CA, OCSP responder and TSA have not been compromised. During the validity period of the corresponding certificates the revocation information of compromised keys is available through PKI. However, after the certificates have expired, there is no source to establish whether the signatures made using the corresponding keys can still be trusted. The current validation algorithms ignore this issue, assuming that after the certificate expired, the corresponding private key was destroyed and hence cannot be compromised as long as the cryptographic algorithms are considered strong.

2.10.4.1. Collision attacks against SHA-1

Due to the weaknesses in the cryptographic hash function SHA-1, the Estonian digital signature scheme has already experienced risks of outdated cryptography. The first cryptanalytic attack against the collision resistance of SHA-1 was

discovered in 2005 [120]. In 2011, the cryptographic algorithms life cycle study ordered by RIA recommended abandoning the use of SHA-1 in favor of the SHA-2 hash function family (Section 3.4.4 in [121]). In October 2015, it became known that a practical attack against SHA-1 collision resistance using the current technology costs much less than previously estimated [122]. In February 2017, the first SHA-1 collision was published [123] proving that SHA-1 was broken not only in theory, but also in practice. The collision blocks published by the researchers allowed anyone to trivially construct two different PDF files that would produce the same SHA-1 hash [124].

While the issuance of ID card certificates switched from SHA-1 to SHA-256 in January 2015 [125], a complete deprecation of SHA-1 in digital signatures has still not occurred, as SHA-1 signatures (including recently created signatures) can still be successfully validated.

The main issue concerns the DDOC signature format which only supports the SHA-1 algorithm. The BDOC format also supports SHA-1, but the ID card software has always created BDOC signatures using SHA-256[10]. For BDOC containers that contain any SHA-1 signature, ID card software v3.8, released on 2013-12-10, introduced a warning preventing the addition of signatures to such BDOC containers. Support for creating signatures in DDOC format was removed from the ID card software starting with v3.12 released on 2016-02-01. [113]

Since SHA-1 resistance against second-preimage attacks is still strong, the SHA-1 signatures created before the collision attacks became feasible can still be trusted, while signatures created today can be exploited in collision attacks. In practice, SHA-1 collision attacks could be exploited by tricking a victim into signing a specially crafted document that would result in the signature also validating another document with a colliding hash value which the victim did not intend to sign[11].

While the users with up-to-date ID card software cannot be tricked into signing documents that exploit SHA-1 collisions[12], the acceptance of SHA-1 signatures allows cardholders, who have intentionally created such SHA-1 colliding signatures, to later repudiate the signature based on the claim that they have fallen victim to a SHA-1 collision attack. To be on the safe side, the current validation algorithm should be updated to discard SHA-1 signatures created after January 2017. This, however, will also invalidate some legitimate signatures created using the outdated software, which in turn shows that the Estonian authorities have been late in deprecating SHA-1 usage in digital signatures.

[10] Except for signatures created using MICARDO-powered ID cards where an RSA signature with SHA-224 algorithm was used (see Section 4.1.2 for the reasons).

[11] For the DDOC format the collision blocks published by the researchers cannot be directly used, as the hash reference is calculated over the XML structure containing base64-encoded files.

[12] The current technological solution enabling digital signing in web environments cannot prevent such attacks, but as we discussed before, this solution is prone to a much more straightforward attack by a service provider.

2.10.4.2. TeRa (Tembeldamisrakendus)

As a response to the weaknesses of the SHA-1 algorithm, in July 2017 RIA released a new version of the ID card software containing the timestamping application TeRa (TembeldamisRakendus) and urged everyone to timestamp their DDOC files by July 2018 at the latest [126].

The application searches for DDOC files in the user's file system and obtains a SHA-256 timestamp over the whole file, storing the timestamp along with the file in an ASiC-S (ASiC Simple) container using the file extension `.asics`.

The timestamp is issued by the Lithuanian qualified trust service provider (QTSP) BalTstamp UAB using an RSA 2048-bit key whose validity will expire on 2022-05-15. The QTSP promises to maintain the records of all issued timestamps for at least 10 years after the timestamp has been issued. [127, 128]

Contrary to RIA's announcement, the risk of SHA-1 forgery is not present as there are no signs of the second-preimage resistance of SHA-1 being broken in the foreseeable future. We were unable to find an analysis that could describe the potential benefits for the timestamping of DDOC files.

A timestamp over the whole content of a DDOC file could help to verify (without requesting offline records from the CA) that the signatory's certificate was issued during the validity period of the CA's certificate and that the OCSP timestamp response was created during the validity period of the OCSP responder's certificate. However, to provide such continuity, the timestamping should be done before the corresponding certificates expire. The OCSP certificates for DDOC signatures created before 2011 have already expired. The OCSP certificate "SK OCSP RESPONDER 2011" used since 2011 will only expire on 2024-03-18, which is two years after the BalTstamp's TSA certificate would have expired.

The timestamping of the whole signature file can provide some level of trust in the event the CA's or TSA's private key gets compromised in the future. However, these benefits are not specific to DDOC files.

2.11. PIN verification mechanism

The Estonian ID card enforces cardholder verification using PIN and PUK codes, similar to the usage in EMV payment cards and mobile phone SIM cards. The PIN security mechanism ensures that without the knowledge of the PIN codes, an unauthorized person who has access to the cardholder's ID card cannot perform cryptographic operations with the card.

The ID cards are preconfigured with three random security codes – PIN1 (4 digits), PIN2 (5 digits) and PUK (8 digits). In general, PIN1 is used for authorizing cryptographic operations with the authentication private key, PIN2 for authorizing cryptographic operations with the digital signature private key, and PUK to unblock and change the values of PIN1 and PIN2. The cardholder is

expected to preserve the secrecy of the PIN codes until the expiration or revocation of the ID card, or even after, if the confidentiality of the documents encrypted for the ID card has to be maintained.

The signature operations with PIN2 reset the card's active security state, meaning that each signature given using the digital signature key has to be confirmed with a separate input of PIN2. This is not enforced for signing and decryption operations with PIN1. The reason is mainly convenience, as the TLS protocol may require frequent re-authentication to the server (see Section II-D in [15])[13].

Estonian legislation does not regulate or even mention the PIN verification mechanism. The use of this mechanism results from the security requirements of QSCDs laid out in eIDAS ANNEX II 1 (d), namely that the electronic signature creation data used for electronic signature creation must be reliably protected by the legitimate signatory against use by others. The eIDAS technical standard EN 419 211 for QSCDs prescribed by [107] describes the PIN verification mechanism, but does not set any specific security requirements for the retry counter or the length of the PIN. The decision to use this particular PIN configuration was made by the authors of the first EstEID specification [4]. The use of this PIN mechanism in the ID cards is described in Section 6.2.8 of SK Certification Practice Statement (CPS) [129] and Certificate Policy (CP) [130] as a method of activating private keys.

We note that the ID cards issued up to fall 2014, as an alternative to the PIN verification mechanism, also provide a legally unregulated passphrase authentication feature (see Section 6.1).

2.11.1. PIN envelope

The security envelope containing the PIN codes is handed over to the cardholder together with the ID card when the cardholder visits a PPA customer service point to receive the card. The card manufacturer configures PIN codes in the ID card personalization phase and prints them on a security envelope, which is then delivered together with the ID card to the document issuer. In this process the card manufacturer learns the PIN codes, but is required to not store them. The employees of the document issuer cannot learn the codes (and hence abuse the ID card) without opening the security envelope. When receiving the ID card, the cardholder has to sign the confirmation which states: "I confirm that I have received the above mentioned document and PIN codes in an intact envelope.". The SK CPS (Section 6.4.1.1 in [131]) requires the PIN codes to be protected in such a way that it is impossible to read them without breaking the security element and that the cardholder has the prerogative to refuse any PIN codes with an altered security element. In the history of the ID card there have been two

[13] As an alternative, the PIN code could be cached and resent automatically by the software running on the user's computer. This, however, will not work when a smart card reader with a PIN pad is used (see Section 2.14.1).

cases where it was found that in practice the security envelopes did not satisfy this requirement, allowing the codes to be viewed through the envelope (see Section 6.11). We note that in the eID schemes used in other countries (e.g. Finland and Germany) the PIN envelopes are delivered to the cardholders over a separate channel, thereby reducing the risk of abuse before the ID card has reached the cardholder.

2.11.2. Preventing PIN guessing

To prevent the brute-forcing of PIN codes, the card maintains a verification retry counter which is decreased after each incorrect PIN verification try. The current value of the counter for each PIN code is publicly readable from the card. A verification with the correct PIN restores the retry counter to its initial value. If PIN1, PIN2 or the PUK is entered incorrectly 3 consecutive times the corresponding code gets blocked. If PIN1 or PIN2 gets blocked, it can be unblocked or changed using the PUK code. If the PUK gets blocked, it can only be unblocked by the document issuer using the card management operations (see Section 2.11.4).

It is interesting to note that knowledge of the PUK effectively allows the recovering of the PIN1 and PIN2 values by brute-forcing the respective PINs and resetting the retry counter after every 3 consecutive wrong guesses. The card issuer can also perform such a brute-force attack against the PUK code using the card management operations.

While the probability of guessing the 4-digit PIN1 or 5-digit PIN2 in 3 tries is very small (0.03% and 0.003%, respectively), opportunistic attacks in environments where a large number of ID cards are inserted in potentially compromised terminals are practical. As we have discussed in Section 3.3 of [16], a malicious terminal can brute-force PINs by attempting one PIN try per ID card and continue the attack when the cardholder returns with the PIN retry counter reset. It is unlikely that after using the ID card in such a terminal, the cardholder would notice that the PIN retry counter has decreased.

The PIN length requirements for the Estonian ID card are rather low compared to, for example, German Act on Identity Cards and Electronic Identification [132] which requires the use of 6-digit PIN codes.

2.11.3. PIN change

Cardholders can change the original PIN and PUK codes and increase their length up to 12 digits. Technically PIN codes can contain any byte value, but since the standard PIN entry device only accepts digits, the convention is to construct PIN codes from only digits. Non-blocked PIN1, PIN2 and PUK codes can be changed by authenticating with the correspond code. Blocked PIN1 and PIN2 codes can be changed using the PUK. On a PIN change, the card requires the value of the new code to be different from the current value. The client-side support for changing the codes has been implemented in the official ID card client-side

software (see Section 2.13). The ID card application DigiDoc4 (released in July 2018) introduced several PIN quality requirements. Namely, the new PIN code cannot be: an increasing or decreasing sequence of numbers; a sequence of a repeated number; part of the personal ID code or the birthdate of the cardholder (YYYY, MMDD or DDMM) [133].

Cardholders are not urged to change the original codes after receiving their ID card. The portion of cardholders who do change them is unknown. Anecdotal evidence suggests that more advanced cardholders change the codes to match the codes of their other eID tools which they have already memorized. The PUK codes, however, are rarely changed.

We note that eID schemes in other countries (e.g., Finland and Germany) enforce the change of the PIN code by implementing the so-called transport PIN codes. The card requires the transport PIN codes to be changed before the cryptographic functionality of the card can be used.

2.11.4. Issuance of new PIN envelopes

A PIN replacement service is provided for cardholders in the event they forget their PIN codes or the PINs get blocked. At the end of 2018, it was reported that nearly 4 000 cardholders apply for a new PIN envelope each month [134]. Currently the service is only provided in PPA customer service points, while before 2019-03-01 the service was also provided in SK customer service points for a small fee. As of 2020-01-01, PPA has also introduced a fee of 5 EUR for a new PIN envelope [135].

The PIN replacement procedure is implemented using card management operations (see Section 19.6 in [9]). To replace PIN codes, the cardholder has to fill and sign the PIN replacement application. The PIN replacement service includes technical constraints that do not enable the replacement of PIN codes for non-valid ID cards and certificates [136].

After visual authentication of the cardholder, the person providing the service takes the PIN envelope from a heap of pre-printed security envelopes. The unique envelope identification number is entered into the system and over the end-to-end encrypted channel between the card and the manufacturer's backend the PIN codes corresponding to the specified envelope are written to the ID card.

The SK CP (Section 6.4.1.1 in [70]) requires the mechanism for replacing the PIN codes to ensure, by technical means, the impossibility of the PPA employee viewing or storing the replacement PIN codes during the whole process. The incidents of transparent PIN envelopes (see Section 6.11) showed that this requirement in practice is not always satisfied.

In 2016, in the context of the remote ID card update process, the concept of a virtual PIN envelope was introduced. The new PIN codes were delivered to the user's computer by encrypting them using the cardholder's authentication key. The decrypted codes were then shown on the screen and the user was asked to write them down (see Section 5.4).

2.12. Personal data file

The ID card chip contains a publicly readable personal data file, which consists of 16 records containing the same information as printed on the card (see Table 1).

Table 1: Contents of a personal data file stored on an ID card [16]. IDEMIA-powered ID cards have introduced some slight differences (see page 16 in [13]).

No.	Content	Example	Length (bytes)
1	Surname	ŽAIKOVSKI	Max 28
2	First name line 1	IGOR	Max 15
3	First name line 2		Max 15
4	Sex	M	1
5	Nationality code	POL	3
6	Date of birth	01.01.1971	10
7	Personal ID code	37101010021	11
8	Document number	X0010536	8 or 9
9	Expiry date	13.08.2019	10
10	Place of birth	POOLA / POL	Max 35
11	Date of issuance	13.08.2014	10
12	Permit type		Max 50
13	Notes line 1	EL KODANIK / EU CITIZEN	Max 50
14	Notes line 2	ALALINE ELAMISÕIGUS	Max 50
15	Notes line 3	PERMANENT RIGHT OF RESIDENCE	Max 50
16	Notes line 4	LUBATUD TÖÖTADA	Max 50

In practice, the personal data file is read by various physical systems to identify the cardholder. In our paper "Using the Estonian Electronic Identity Card for Authentication to a Machine" [16] we studied this use case in detail. We found that several large merchants in Estonia allow the ID card to be used as a customer loyalty card, providing access to rewards once the ID card is inserted in the merchant's terminal. Similarly, the ID card can be used to authenticate to self-service printing machines and self-checkout machines in libraries. Pharmacies use the ID card chip to look up the drugs prescribed using the digital prescription system. In some public and less public security installations the ID card can be used as an entrance card to unlock the door and gain access to restricted areas. [16]

We found that many chip terminals read more data from the personal data file than necessary to identify the cardholder (Section 5 in [16]). It is, however, not known whether this data is stored and processed by the service providers.

It is important to note that this use case provides very little security guarantees as we were able to build an ID card emulator that is accepted as a genuine ID card by all the chip terminals tested. We note that the verification of the ID card's physical security features did not help as we were able to successfully transplant our programmed chip onto a real ID card without damaging any of the card's physical security features (Section 4.3 in [16]).

We analyzed the possible technological improvements that could provide cryptographic security and improve usability, therefore enabling wider use of the ID card as a physical authentication token (Section 6 in [16]). With the introduction of NFC-capable IDEMIA-powered ID cards in 2018, the ID card has the potential to be used as a secure and convenient physical authentication token, but not in its current configuration (see Section 3.5.3)

2.13. ID card software

Standard operating systems do not contain support for the Estonian ID card, therefore additional ID card software has to be installed to make full use of the ID card. Historically, the main components of the ID card desktop software have been: drivers and middleware for communication with the ID card; the ID-card Utility to change PIN codes and update the ID card; the DigiDoc Crypto application for file-based encryption and decryption; the DigiDoc Client application for digital signature creation and validation; and browser extensions for authentication and digital signing in a web environment.

The development of the first ID card software started in 2002 and was financed by SK until the Estonian Informatics Centre (Riigi Infosüsteemide Arenduskeskus – predecessor of RIA), in 2008 with support from the European structural funds, announced a tender for the development of ID card software [20].

At the end of 2008, Smartlink OÜ was contracted to develop the ID card software using an open source development model. The contract consisted of an 8-month development period and a 36-month software support period. Support for Linux, Mac and different browsers had to be developed. After repeated extensions of the deadline, a working software was not delivered and in July 2010 the development of the ID card software was given over to SK. [137, 138]

At the end of 2010, a new ID card desktop software version DigiDoc3 was available to users (November 2010 for Linux and Mac, January 2011 for Windows) [113]. At the end of 2014, the software development migrated from `svn.eesti.ee` to GitHub [139]. In July 2018, the DigiDoc4 client was introduced. The main change was its visual design and integration of the ID-card utility, DigiDoc Crypto and DigiDoc Client into a single application.

The main distribution point for the ID card software is the website `id.ee`, where installers for Windows and Mac can be downloaded. On Mac computers the DigiDoc4 client can also be installed from Apple's App Store. In Windows, the minidriver component, which enables ID card authentication using Internet Explorer and Google Chrome, is automatically installed on the computer through the Windows update mechanism the first time the ID card is plugged into the computer. For Ubuntu users, the ID card software is distributed using a package repository maintained by RIA. [140]

Support for automatic software updates was implemented starting in ID card software version 3.5, released on 2011-11-10. In version 3.9, released on 2014-07-01, a "kill switch" functionality was implemented, meaning that the

software would fail to run if the currently installed software version was not supported or if the automated software version check had not succeeded during the past 12 months [113].

As of 2019, the ID software is used in approximately 600 000 computers [141]. Since 2018, the DigiDoc client application has also been provided for Android and iOS mobile operating systems. The name of the mobile app is RIA DigiDoc and it can be installed from Google Play and Apple's App Store, respectively [142].

The ID card software has not managed to completely avoid closed-source software dependencies. With the introduction of the IDEMIA-powered ID cards on 2018-12-03, the ID card software started to ship IDEMIA's AWP software package whose source code is not available [143]. This continued until open source support for communication with the IDEMIA-powered ID cards was implemented in OpenSC. On Linux and Mac the switch to OpenSC was done starting with the ID card software version 19.10, released on 2019-11-05. For Windows OS the proprietary AWP software and minidriver are still used to communicate with the IDEMIA-powered ID cards. [113]

2.13.1. Vulnerabilities

Over the years, the ID card software has experienced several security vulnerabilities, however, only a few of the flaws have gained public attention.

2.13.1.1. Certificate leakage in ID card browser extension

In November 2010, Antti Andreimann published proof-of-concept code [144] demonstrating that malicious JavaScript code served by a website can abuse the ID card browser extension to read a user's ID card certificate without the user's consent. The flaw had been present since the introduction of the browser extension and had been known for at least 5 years [145]. Interestingly, if there were several certificates in the Windows certificate store, the risk was not present as the certificate selection window was displayed [145].

In their response, RIA downplayed the impact [146, 147] and SK responded that it was a legal problem, as an illegal collection of personal data was forbidden by law [148]. Nevertheless, the flaw was fixed in the ID card software released on 2011-01-23 by introducing a mandatory certificate selection window [149].

2.13.1.2. Directory traversal vulnerability

In July 2013, in the process of auditing the i-voting server-side source code, Renee Trisberg discovered a directory traversal vulnerability in the code handling BDOC files [150]. The same flaw was also present in the ID card software, allowing an attacker to overwrite any files on the victim's computer (with their system user privileges), if the attacker was able to persuade the user to open a specially crafted BDOC[14] or DDOC file [151].

[14]The BDOC attack vector is not mentioned in the release notes as the BDOC format was not actively used at that time. The updated ID card software version removed the support for BDOC but reintroduced it in a later version.

On 2013-08-22, RIA and SK published an announcement on their websites urging the public to install ID card software version 3.7.2 that fixes the vulnerability [152, 153]. Criticism was expressed in the media towards the authorities regarding the insufficient public announcement, the critics demanding responsibility be taken for the technical solutions of the ID card [151, 154].

On 2013-08-27, there were nearly 30 000 users who had not accepted the automatic update to version 3.7.2 that fixed the vulnerability [155]. At that time there was also a number of users of older versions that did not even support automatic updates and hence had to install the new version manually [151].

2.13.1.3. ID card authentication man-in-the-middle attack using browser signing extension

At the end of 2020, we noticed that several Estonian e-services (`swedbank.ee`, `coopbank.ee`, `bigbank.ee`, `inbank.ee`, `unicredit.ee`, `jetoil.ee`, `rahvaalgatus.ee` and `portal.smart-id.com`) were not using TLS client certificate authentication to authenticate their users, but the ID card browser extension instead. To our surprise, the ID card browser signing extension was quietly extended in 2017 [156] to allow e-service providers to request raw signatures using the authentication key (rather than only using the digital signature key). Since the e-service provider's identity is not included under the signature, a malicious e-service provider could use this feature to ask a user to sign a value that would allow them to impersonate the user in any other e-service that enables ID card authentication, regardless of whether it was using TLS or the browser extension.

On 2020-12-17, we shared the attack's proof-of-concept video [157] with RIA. RIA decided to remove the option to sign using the authentication key from the ID card browser extension. However, an agreement was made to wait until the most prominent e-service `swedbank.ee` had moved back to using the TLS client certificate authentication feature. Swedbank made this move on 2021-01-14 and later explained that the browser extension was used for authentication because they considered it to be more reliable [158].

On 2021-01-28, a new ID card software, version 20.12, was released, and on 2021-02-03 RIA published a press release [159] urging users to update their software.

2.13.1.4. Other vulnerabilities

The release notes of the ID card software [113] contain several security related fixes. Most of them are due to the complexity of XML parsing, resulting in an invalid signature being recognized as valid. The latest security issue in the release notes is dated 2017-08-15. However, it is likely that security bugs are still regularly found but fixed quietly, as our recently reported vulnerabilities in the digital signature validation code [160, 161] have not been marked as a security issue in either the release notes or the commit message.

2.14. Smart card readers

A smart card reader is an essential component for the electronic use of the ID card, but only very few computers have a built-in smart card reader.

In 2002, the cheapest smart card reader available in the market was priced at 354 kroons (24 EUR) [41]. In 2003, Elion stores started to distribute an ID card starter package with a smart card reader and a CD containing the ID card software installer. The price of the package was 20 EUR which was still above the expectations of the average consumer [20]. In 2007, Elion made a bulk deal with OMNIKEY GmbH which brought the Omnikey CardMan 1021 smart card reader to the Estonian retail market for around 6 EUR, which was below the average market price of the reader [20, 162].

In 2014, an Estonian designer brought the +iD smart card reader to the market. The reader was the smallest and lightest device of its kind available for full size smart cards [163]. Currently there are several versions of the reader, some of them selling for as low as 10 EUR.

2.14.1. Smart card readers with PIN pad

In 2010, malware analysts observed a modification of the banking trojan Zeus. The modified version was able to use a smart card connected to the victim's computer to make fraudulent bank transactions in several Russian online banks [164]. The attacks against the ID card and Estonian online banks were believed to be just a matter of time, therefore RIA recommended that ID cards should only be placed in the reader when the ID card functionality was actually used [165].

Later, as a solution to the malware problem, RIA made the recommendation that on high-risk computers which are used by several persons, a smart card reader with a PIN pad should be used [166]. Readers with a PIN pad allow PIN code entry on the PIN pad, which is then sent directly to the smart card, thereby preventing a potentially infected computer from learning the code and using it in the future without the cardholder's consent.

The malware can still abuse the ID card once the cardholder has entered the PIN on the PIN pad, but this attack is more complicated. In the case of PIN2 it only allows the forging of a single signature after the cardholder has entered the code. Another problem of standard smart card readers with a PIN pad is that they also work in a so-called pass-through mode, allowing the PIN verification commands to be also received from the computer. Malware can abuse this to slowly brute-force the PIN by performing one try after each successful user authentication that resets the PIN retry counter. Alternatively, the malware can execute a phishing attack, asking for the PIN code to be entered from the computer and using it later without the cardholder's consent.

In 2011, Martin Paljak discovered that a secure PIN entry on the HP USB keyboard with a built-in smart card reader (model KUS0133) actually did not

provide the claimed security as the entered PIN code was transmitted to the computer [167]. These HP keyboards with a built-in smart card reader were a very popular product in the private and public sector in Estonia.

In 2012, RIA initiated talks with Gemalto for a bulk purchase of the smart card reader with PIN pad, IDBridge CT710 [168] (also known as Ezio Shield), personalized for Estonia. RIA convinced Elion to make a bulk purchase from Gemalto and in fall 2013 the readers were available in the retail market for as little as 20 EUR, which is below the average market price of the reader. It was promised that the public authorities would buy the readers in a centralized public procurement, but this never happened, leaving Elion with a large stock of readers. [169, 170]

The product was graphically customized for the Estonian market: the Gemalto logo was relaced with the ID card help line number and the user interface was provided in Estonian [171]. An important security feature provided by this reader is the so-called PIN firewall, which blocks PIN verification and PIN change commands received from the computer [171]. This means that even if malware gains access to the cardholder's PIN codes, the malware cannot pass them to the smart card. However, the card is still open to an attack once the cardholder enters the code on the PIN pad.

We found that the passphrase authentication feature on the ID cards issued up to fall 2014, in practice, allows the security advantages provided by the smart card readers with PIN pad to be bypassed (see Section 6.1.1). Fortunately, as of today there have been no public records of malware that attempted to abuse the Estonian ID card connected to an infected computer.

2.15. Validity lifecycle of the ID card and its certificates

ID cards and the certificates therein are issued with a specific validity period, depending on the identity document type and the right of residence, but most frequently for the validity period of 5 years. Before 2007 ID cards were issued with a longer validity period than the certificates therein (see Section 3.1.1), but from 2007 the ID cards and certificates were issued with the same validity period. Nevertheless, there are some slight differences in the validity life cycle of the ID card and the certificates, which we desribe below.

Before the ID card is handed out to the cardholder, the certificates are legally not yet valid (see clause 16 (4) of EITSETA). After handing out the ID card, the PPA employee registers the ID card certificates as valid (clause 4.4.1.1 of [172]). If the cardholder already has an ID card of the same type, the PPA employee revokes the previous ID card and the certificates therein. This ensures that the cardholder can only have one electronic identity document of a particular type valid at a time.[15]

[15] In practice, we have observed several cases where this workflow has failed, resulting in the previous ID card certificates remaining valid.

The relying parties are expected to verify the validity of the certificates before relying on the certificate (see clause 7 in [65]) and the law only gives the digital signature a legal effect if at the time of signing the certificate was valid (article 32 (1)(b) of eIDAS)[16].

During the validity period of certificates, the validity of certificates can be temporarily suspended or permanently revoked. While technically possible, the suspension or revocation of only one certificate from the pair (i.e., authentication or digital signature) is not practiced.

The suspension of certificates is useful in cases when the ID card has been lost or stolen, as validity can be suspended instantly without the need to submit a signed application. To request certificate suspension, the cardholder has to call the ID card helpline that is available around-the-clock and identify himself using basic personal data (name and personal identification code). To restore the validity of the certificate (e.g., in case the ID card is later found) the cardholder has to submit a signed application. The law allows certificates to also be suspended by the CA and other authorities. The validity of the certificates, however, can only be restored by the party who requested the suspension.[17]

To revoke the certificates, a signed application is required from the cardholder or other eligible party. Revocation or suspension of the certificates does not have an effect on the validity of the identity document. The document issuer does not provide a service for replacing revoked certificates, therefore to renew revoked certificates, the cardholder has to apply for a new ID card. However, historically there have been cases where the replacement of expired (Section 5.2) and revoked (Section 6.4.1) ID card certificates has been provided. There has also been a precedent for extending certificate validity and hence the validity of the *digital identity card* beyond the date printed on the card (Section 3.4.4).

The life cycle of identity documents does not allow the validity of the ID card to be temporarily suspended. To revoke an ID card, the cardholder has to submit a signed application. With the revocation of the ID card the certificates therein are also revoked. The ID card is automatically revoked in cases where the person dies or a resident obtains citizenship (see Section 6.10 for the issues in applying this mechanism in practice).

In the event a person changes their name (e.g., due to marriage), the ID card with the previous name remains valid until the ID card with the new name is received or until the end of its validity. This conflicts with the standard practice of CAs revoking the certificate if the personal details specified therein become inaccurate.

[16] See Section 2.10.3 for the technical problems fulfilling this legal requirement.

[17] The problems with the suspension mechanism when applied by a party other than the cardholder has been analyzed by us in [18].

2.16. Certificates and personal data therein

By the definition of public-key certificate, the purpose of a certificate is to bind a public key to an entity. To achieve this purpose, the certificate has to contain enough information to unambiguously identify the entity to whom it has been issued. In the case of the Estonian ID card, the certificates contain the cardholder's full name and personal identification code (personal ID code). The personal ID code is a unique 11-digit number that generally remains fixed for the lifetime of the person and therefore is widely used in public and private databases to identify persons. The personal ID code is also usually used in civil contracts to identify the contracting parties. The ID code is not purely a serial number, since the first 7 digits of the code encode the gender and date of birth of its holder.

The data contained in the certificate allows additional personal data about its holder to be inferred. The validity period of the certificate usually corresponds to the validity period of the identity document in which the corresponding private key resides. In some cases the validity period may be used to deduce the cardholder's right of residence.

The Organization Name (O) field of the certificate's subject name can be used to determine the identity document's type. The value "ESTEID" corresponds to *identity card* and *residence permit card*, "ESTEID (DIGI-ID)" corresponds to *digital identity card* and *diplomatic identity card*, "ESTEID (DIGI-ID E-RESIDENT)" corresponds to *e-resident's digital identity card*, and "ESTEID (MOBIIL-ID)" corresponds to *digital identity card in a mobile-ID format*.

From the end of 2018, with the introduction of IDEMIA-powered ID cards, the abovementioned document type was removed from the certificate's subject name. However, an additional certificate policy field was introduced, which now encodes not only the type of identity document, but also provides quite detailed information about the certificate holder's right of residence (see Section 1.2 in [70]).

The certificate validity services can be used to obtain additional information about the certificate's life cycle and hence about its holder. For example, the certificate revocation information accumulated in CRLs can be used to deduce the time when the cardholder visited the document issuer to receive their new ID card and the old one was revoked.

This information and also some other pecularities of the ecosystem allowed us to deduce many interesting details, some of which played a crucial role in finding the answers to the important research questions of this study.

2.17. LDAP certificate repository

With the introduction of the ID card in 2002, all valid certificates issued to the ID card holders have been made available for lookup in the public LDAP directory service `ldap://ldap.sk.ee` maintained by SK [173]. Initially, it was possible

to search for certificates using a first name, last name or personal ID code. The only restriction applied was the maximum number of responses returned in one query to protect against server overload [174][18].

The ability to find a personal ID code using the person's name led to controversy. On 2005-01-19, TV investigative programme Pealtnägija (Eyewitness) showed that the personal ID code found in the LDAP repository could be used to not only find out the date of birth for prominent Estonian persons, but that it could also be used to authenticate to the two biggest Estonian banks over the phone and ask the bank to block the person's payment card [175].

In 2006, the Chancellor of Justice published an opinion [176] on whether the publication of personal ID codes on the internet is lawful. The Chancellor found that the technical solution for certificate verification via LDAP did not comply to the applicable law and that the recipients of ID cards were not clearly informed that their personal ID codes and names would be made available via the internet to everyone. The Chancellor suggested the modification of the system, requiring a first name, last name and personal ID code for certificate lookup.

As a response, on 2006-06-07 the parliament of Estonia amended the Identity Documents Act adding clause $9^4(6)$, which states that the certificates are connected to the personal data of the certificate holder and are publicly verifiable through the personal ID code. In addition, on 2006-12-05 the LDAP service was reconfigured to allow certificate lookup if at least the full personal ID code is known [177]. The Chancellor of Justice considered the solution to be an acceptable short-term compromise, but suggested that looking for a long-term solution should continue.

We note that despite the added restrictions, as the search space for all possible personal ID codes is small, it is possible to retrieve all certificates and then perform a reverse search by the person's name, thereby circumventing this restriction. In later years, to protect against such crawlers additional restrictions were added to the LDAP service limiting the number of certificates that can be queried in a particular time frame (see Section III-B in [178]).

The existence of an LDAP certificate repository has been publicly motivated by the need to verify certificates [175]. We find this motivation questionable, as the authenticity of the certificate is verified by verifying the CA's signature on the certificate and the validity of the certificate is verified using CRLs and OCSP validity services, where the validity of a certificate is verified by querying the serial number of the certificate. To some extent, the LDAP repository could be used as a validity service, as SK's Certification Practice Statement [129] states that only valid and unexpired certificates are published in LDAP.[19] The data connection to the LDAP service, however, is not cryptographically protected, therefore the use of LDAP for certificate validity checking comes with risks.

[18] The LDAP service returned a maximum of 50 entries per query.

[19] Unfortunately, we (and others [179]) have frequently observed the LDAP repository to be out of synchronization, containing revoked certificates and not containing the valid ones.

Before eIDAS came into force requiring SK to provide the OCSP service free of charge, some service providers did indeed use the LDAP service for certificate validity checks (see Section IV-B9 in [15]).

Traditionally, LDAP in a PKI is not used as a validity service, but to distribute certificates and CRLs to the relying parties. In the context of the Estonian ID card, the problem of certificate distribution is largely solved, because certificates are attached to each digital signature and authentication transaction, thereby making it available to the corresponding relying parties for verification. The only use case where the certificate lookup service provides a benefit is encryption, giving convenient means for the sender to obtain a recipient's public key without requesting it from the recipient.

The consideration of the name and personal ID code of a person as public data in Estonia [174] quite sharply contrasts with the European notion of privacy. A similar approach to privacy is also observable in other fields in Estonia, for instance, making the data about the real estate persons' own publicly available [180].

On 2018-11-14, with the introduction of IDEMIA-powered ID cards, a new LDAP directory service `ldaps://esteid.ldap.sk.ee` was made available over a TLS connection [173].

2.17.1. Certificates analyzed in this study

A significant part of analysis in this work is based on an ID card certificate dataset that we collected over the years by crawling the LDAP certificate repository. While our dataset of more than 7 million ID card certificates is not complete, we believe that it contains a representative sample of ID card certificates issued throughout the years. Figure 2 shows the distribution of ID card certificates in our dataset by issuance month (based on the certificate's `notBefore` field) for different ID card platforms. Due to the crawling process, the dataset lacks certificates issued from 2002 to 2007 and certificates which have been valid for a short period of time. Therefore, in general, our findings provide only a lower bound for the observations.

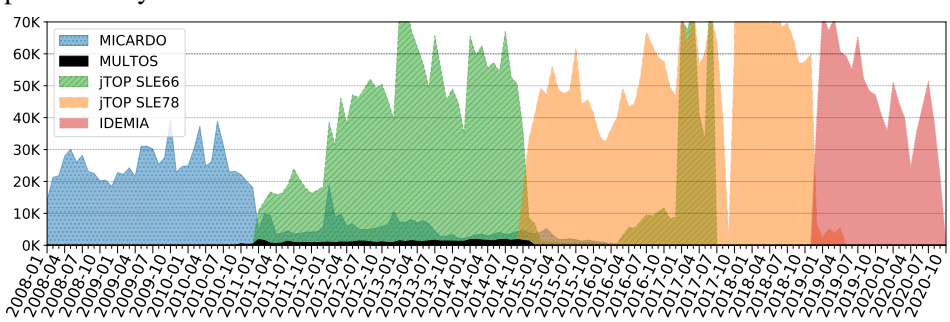


Figure 2: ID card certificates analyzed in this work (by month of issuance)

2.18. @eesti.ee email address

Two email addresses on the @eesti.ee domain are automatically assigned to each Estonian resident. One address is in the form personal_ID_code@eesti.ee and the other is in the form name.surname@eesti.ee. The eesti.ee SMTP server does not store emails and works only as an email forwarder. To receive emails to the assigned addresses, the person has to log into the citizen portal eesti.ee and configure email forwarding to their personal email address (or several addresses).

For the recipients who have not configured the forwarding, the eesti.ee SMTP server will reject the email right after the RCPT TO command, before the body of the message is even transmitted. The VRFY command supported by the SMTP server allows a check to be performed, without sending an email, to determine whether the person has configured the forwarding. As of August 2020, from around 1.4 million persons with an Estonian personal ID code, more than 389 000 have configured email forwarding for their @eesti.ee email addresses.

To avoid email address collisions for namesakes, in the first years the addresses were generated in the format firstname.lastname_NNNN@, where NNNN corresponds to four random numbers [174]. From the end of 2005, to make the addresses easier to remember and use, the addresses were assigned without the _NNNN postfix. To avoid collisions for namesakes, an address name.surname.1@ was assigned to the second namesake, name.surname.2@ to the third, and so on. Persons who were assigned a new email address as a result of this reform retained the old address with the _NNNN postfix and hence have three addresses in total. In case the name of the person changes, a new address is automatically assigned to the person and the old address is removed. According to RIA, the old address is marked reserved and is never assigned to any other person (namesake), not even to the same person if the person decides to switch back to the previous name.

The email address personal_ID_code@ is restricted such that only authorized institutions can send email to this address. The owner of the address, however, cannot separately disable email forwarding for potential spam emails addressed to the unrestricted name.surname@ address. From 2019-02-19, for the persons who have configured email forwarding, the emails sent to the personal_ID_code@ address are also stored on a virtual "mailbox" that can be read through the citizen portal eesti.ee [181].

The email address in the form name.surname@ has been embedded in the subjectAltName extension of the cardholder's authentication certificate. However, starting with IDEMIA-powered ID cards the address in the form personal_ID_code@ is now included in the certificate. According to RIA, the email addresses in the form name.surname@ will not be assigned for new cardholders anymore, while those who had it assigned in the past will be able to continue using them. This decision has most likely been motivated by the fact that the correct assignation of name.surname@ email addresses has turned out to be a challenging task in practice. We discuss this further in Section 6.5.

3. CHIP PLATFORMS AND IDENTITY DOCUMENT TYPES

The Estonian state issues several types of identity documents that contain contact-type smart card chips that provide cryptographic functionality. These are the *identity card*, the *digital identity card*, the *residence permit card*, the *e-resident's digital identity card* and the *diplomatic identity card*. We will use the common term "ID card" to denote all of these identity document types. The Estonian state also issues the *digital identity card in a mobile-ID format* (Mobile-ID), which also contains a smart card chip that provides cryptographic functionality, but as it implements a different protocol and is also different on the architectural level, we will not cover Mobile-ID in this work.

Over the years, the ID cards have been largely issued using five different smart card chip platforms: MICARDO, MULTOS, jTOP SLE66, jTOP SLE78 and IDEMIA. In this chapter we will document each of these chip platforms and consecutively introduce the identity document types that have been issued using these platforms. The timeline of identity documents and ID card platforms used is shown in Figure 3.

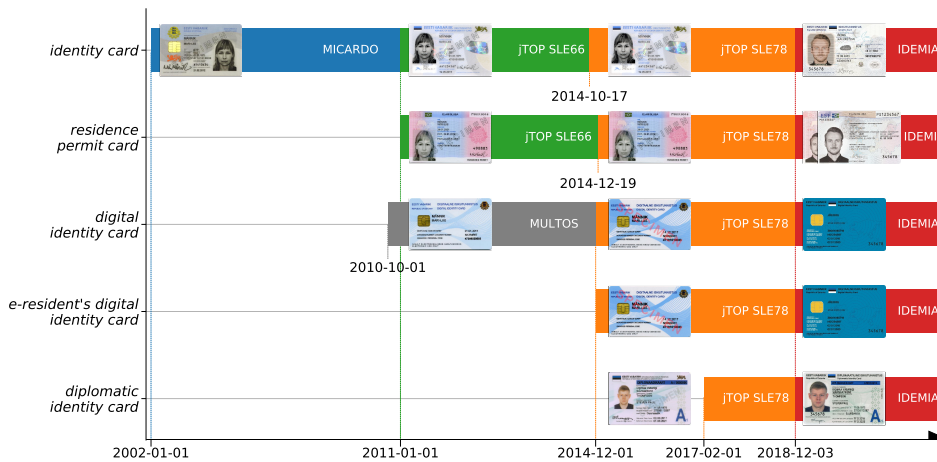


Figure 3: Timeline of identity documents and ID card platforms used in Estonia

3.1. MICARDO platform

The first ID card type, introduced in January 2002, was the *identity card* implemented on top of several slightly different MICARDO platform versions. We will describe the *identity card* and the MICARDO platform in the subsections below.

3.1.1. *Identity card*

The *identity card* is issued to Estonian citizens and citizens of the European Union. The first *identity cards* were issued in January 2002. The cards were distributed to their cardholders at a public ceremony held on 2002-01-28 [21].

The electronic functionality was implemented using the MICARDO smart card operating system. MICARDO-powered *identity cards* had been produced for 9 years until the JavaCard platform jTOP SLE66 was introduced in 2011 (see Section 3.3).

Figure 4: A MICARDO-powered *identity card* issued from 2002-01-01 to 2010-12-31 [182]. Cards issued before 2007-09-03 contain the line "omaniku allkiri" (owner's signature) instead of "kasutaja allkiri" (user's signature).

The visual appearance of the card is shown in Figure 4. Over the 9 years, the visual appearance of the *identity card* only changed when on 2007-09-03: (1) the Estonian translation of "holder's signature" was changed from "omaniku allkiri" (owner's signature) to "kasutaja allkiri" (user's signature); (2) the document's expiration year was removed from the variable laser image on the top left corner of the back of the card; and (3) the laser engraved contour of the Estonian map was moved to cover the facial image. The description of the visual security features for all Estonian identity documents are available in the EU Council PRADO website [183]. An unknown number of MICARDO-powered *identity cards* were also issued with a chip that had a squared contact layout (see Figure 5).

Figure 5: A MICARDO-powered *identity card* with a squared contact layout designed chip [174]

Initially, the *identity cards* were valid for 10 years, but their certificates were only valid for 3 years. The cardholders were therefore provided with a remote certificate renewal solution (further discussed in Section 5.2) to ensure cryptographic functionality for the entire document validity period. However, starting from 2007-01-01[1], the validity of the *identity cards* and certificates were both set to five years [45, 184], hence the remote certificate renewal solution was not needed for these cards. The last of MICARDO-powered ID cards, as seen from our certificate dataset, expired on 2017-03-22.

3.1.2. MICARDO-powered ID cards

The electronic functionality of the card was implemented using the smart card operating system "MICARDO Public Version 2.1 64/32 Release 1.0" produced by "ORGA Kartensysteme GmbH" (later known as "Sagem Orga GmbH", "Morpho Cards GmbH", OT-Morpho, and most recently known as IDEMIA) [4]. The MICARDO operating system provides a defined set of functionality which can be configured by the card issuer. This is done by creating different types of file objects and setting access rules for the files and operations performable with PIN codes and cryptographic keys. The functionality of the MICARDO card operating system is fully documented in the MICARDO User Manual [185].

The specification of the electronic functionality and its implementation on MICARDO Public 2.1 was developed by the Estonian company "ID Süsteemide AS" in close co-operation with ORGA [186]. The resulting functionality and the communication interface was documented in the EstEID card specification that was first published on 2002-11-20 (v2.01) [4]. The EstEID card specification later became the Estonian standard EVS 827:2004 [6] which is now withdrawn.

According to the EstEID specification (Section 3.1 in [9]), starting from January 2006 the chip platform was upgraded to "MICARDO Public 3.0". From the ID cards issued we see that the switch was actually done later on 2007-09-03. According to a presentation [187] by a CMB employee, the upgrade was done because the old chip was not available anymore, hence the MICARDO Public 2.1 OS had to be ported to a new microcontroller.

The MICARDO-powered ID card chips support both the T=0 and T=1 transmission protocols as defined by the ISO/IEC 7816-3 standard [188] and can be identified by their cold and warm ATR (answer to reset) bytes. The cards issued before the upgrade have cold[2] and warm[3] ATRs, which are different from the cold[4] and warm[5] ATRs of the upgraded cards (Section 3.1 in [9]), therefore the systems had to be updated in 2007 to recognize the new cards.

[1] For cardholders who applied for the document in 2006, but the decision was made and the card issued in 2007, the validity of the card was still 10 years. Therefore, there are ID cards with 10-year validity also issued in 2007 (the latest was issued in May 2007).

[2] 3B FE 94 00 FF 80 B1 FA 45 1F 03 **45 73 74 45 49 44 20 76 65 72 20 31 2E 30 43**

[3] 3B 6E 00 FF **45 73 74 45 49 44 20 76 65 72 20 31 2E 30**

[4] 3B DE 18 FF C0 80 B1 FE 45 1F 03 **45 73 74 45 49 44 20 76 65 72 20 31 2E 30 2B**

[5] 3B 5E 11 FF **45 73 74 45 49 44 20 76 65 72 20 31 2E 30**

The asymmetric keys for a cardholder's authentication and digital signature certificates were generated inside the card using 1024-bit RSA with a random public exponent chosen by the RSA key generation algorithm as implemented by the MICARDO operating system (see Section 4.1.1).

3.1.2.1. MICARDO platform versions

By analyzing several MICARDO-powered ID cards issued from 2002 to 2011[6], we found that over the years, ID cards were actually issued using three slightly different MICARDO platform versions. First, we made an extensive comparison using all readable information from the card by scanning the whole smart card file system. The metadata was read from the FCP (file control parameters) and FMD (file management data) fields of all DF (dedicated file) and EF (elementary file) files and the contents of all EF files were read. The differences found are listed below.

1. Cards issued in 2002:
    - These are the only cards in which the FMD of the MF (master file) report the chip identifier as Infineon 30 (`0x1E`), which matches the one specified in the MICARDO User Manual (Section 8.4.4. in [185]), and hence should correspond to Infineon's SLE66CX320P microcontroller. Other cards have a different chip identifier.
    - These are the only cards that comply to the MICARDO User Manual by having the `EF_ATR` file be a transparent file (Section 4.10 in [185]). The others have a formatted `EF_ATR` file that contains a single record.
    - The Image ID specified in the FMD of the MF is set to `0x000000` while the MICARDO User Manual specifies `0x000001` (Section 8.4.4. in [185]).
    - The `EF_Rule` file in the PKCS#15 directory (`MF/5015`) has the maximal record length of 40, while in other cards it is decreased to 26 (most likely to save space).
    - The RSA public key exponent length specified in the key generation template is set to 4 random bytes, while in other cards it has been decreased to 3 bytes.[7]
    - The FMD of the `EF_TIN` file contains the customer identifier `548-a002` and customer specific version and release `001548CS01.V02`.
    - The cards have 21 765 bytes of free space reported in the FCP of the DFs, which may suggest that the chip in these cards has a smaller EEPROM size.

[6] ID cards issued in: 2002-01, 2003-04, 2004-04, 2005-04, 2007-05, 2009-04 and 2010-09.

[7] The reason for avoiding 4-byte public exponents was that Microsoft Windows 98 SE was not able to handle certificates with correctly ASN.1-encoded 32-bit public exponents.

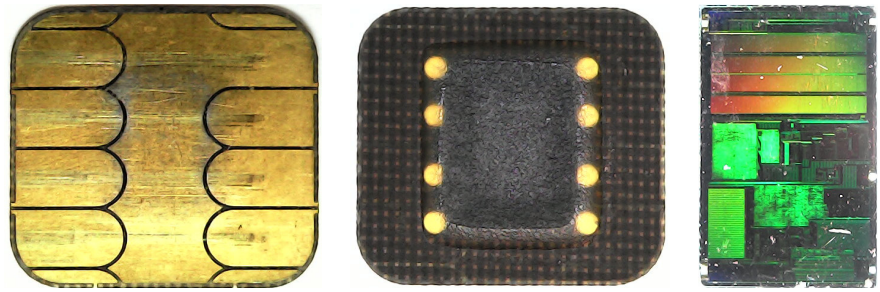
2. Cards issued from 2003 to 2007-09-03:
    - The cards in the FMD of the MF report the chip identifier as Infineon 34 (`0x22`).
    - The Image ID specified in the FMD of the MF is set to `0x020000` while the MICARDO User Manual specifies `0x000001`.
    - The FMD of the `EF_TIN` file contains the customer identifier `548-a003` and customer specific release `001548CS02.V01`.
    - The cards have only 5 570 bytes of free space reported in the FCP of the DFs.
3. Cards issued from 2007-09-03:
    - The cards in the FMD of the MF report the chip identifier as Infineon 45 (`0x2D`).
    - The Image ID specified in the FMD of the MF is set to `0x030000` while the MICARDO User Manual specifies `0x000001`.
    - The FMD of the `EF_TIN` file contains the customer identifier `548-a004` and customer specific release `001548CS03.V01`.
    - The cards have 22 334 bytes of free space reported in the FCP of the DFs. This is almost 16KB more than in the MICARDO cards issued from 2003 to 2007-09-03, but only 569 bytes more compared to the MICARDO cards issued in 2002.
    - These cards have new cold and warm ATRs. The electrical communication parameters encoded in the ATR enable slightly faster data transmission between the card and terminal.

To verify whether the MICARDO-powered ID cards did indeed use three different chips as reported by the MICARDO operating system, we decapsulated the microcontroller from the chip and observed it using a high-magnification (1000x) digital microscope. After removing the chip module from the card we noticed that for the cards issued in 2002 and cards issued from 2003 to 2007-09-03, the microcontroller was covered by black opaque epoxy[8] while for the cards issued from 2007-09-03, the epoxy was transparent. In both cases the epoxy was successfully dissolved by dipping it into 96% sulfuric acid heated at 200 °C. Photos of the chip modules and microcontrollers are shown in Figure 6.

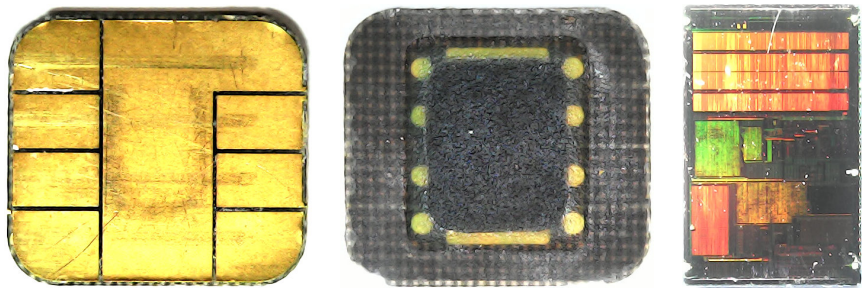
The cards issued in 2002 were likely embedded with a SLE66CX320P microcontroller as specified in the MICARDO User Manual. According to the CMB presentation (slide 3 in [187]) the cards issued from 2003 to 2007-09-03 may have used Infineon's 16KB microcontroller SLE66CX160P[9], while the cards issued after that used Infineon's 68KB microcontroller SLE66CX680PE.

[8] Hard opaque tamper-evident coating on a chip is required by FIPS 140-2 standard [189].

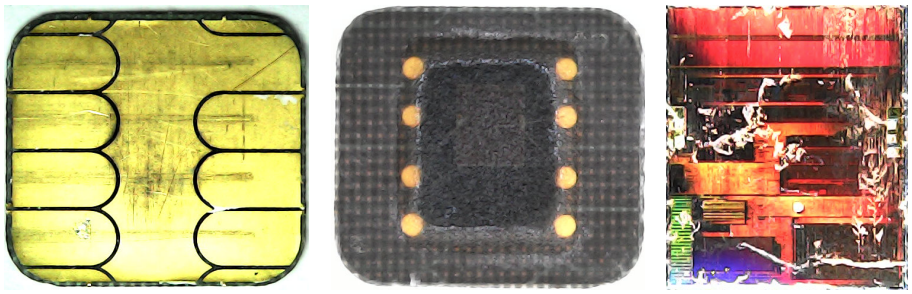
[9] This also corresponds to the Estonian EU notification on SSCDs [190], where for the ID cards issued until 2016-12-31, the certification reference is specified as TUVIT-DSZ-ITSEC-9121-2001, which corresponds to the certification of Infineon's SLE66CX160P microcontroller.



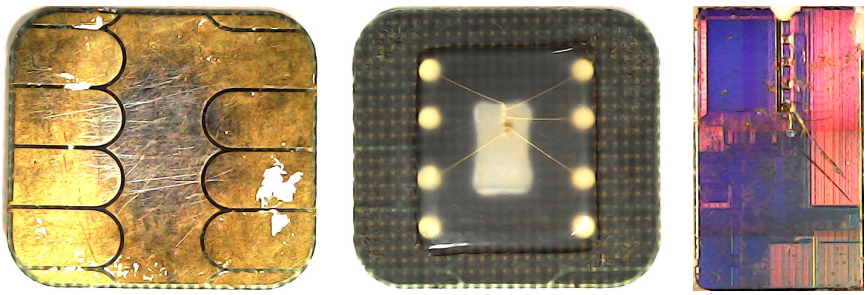
(a) Cards issued in 2002 (SLE66CX320P)



(b) Cards issued in 2002 with squared contact layout chip (SLE66CX320P)



(c) Cards issued from 2003 to 2007-09-03 (SLE66CX160P)



(d) Cards issued from 2007-09-03 (SLE66CX680PE)

Figure 6: The chip modules and microcontrollers used on MICARDO-powered ID cards

3.1.3. ITSEC certification

The MICARDO Public v2.1 chip card operating system had been certified according to ITSEC[10] (IT Security Evaluation Criteria) E4 evaluation level. The product was certified on 2001-08-28 by TÜV Informationstechnik GmbH (TÜViT) in Germany under certification reference TUVIT-DSZ-ITSEC-9126-2001 [191, 192]. However, based on the analysis below, we concluded that the MICARDO product used in the Estonian ID cards was not the exact product that had passed the ITSEC certification.

The certification report (Section 1.1 in [192]) states that the subject of the certification was smart card operating system MICARDO Public identifiable by value `0xD276000028FF051E000001` specified in the FMD of the MF. According to the MICARDO User Manual (Section 8.4.4. in [185]), the FMD data contains the ORGA registration identifier, chip identifier and operating system image identifier. We note that none of the MICARDO cards issued contain the same FMD data as specified in the certification report. The closest match was the ID cards issued in 2002, where the only difference was that the operation system image identifier was `0x000000` instead of `0x000001` as certified. We believe that these cards issued in 2002 were a pre-certified version of the product, as the MICARDO-powered EstEID specification [3] had already been drafted on 2001-06-07, before the product passed the certification on 2001-08-28.

The evaluated security functionality of the MICARDO operating system included physical protection provided by Infineon's SLE66CX320P microcontroller. The microcontroller was certified on 2000-08-04 by TÜViT under the reference TUVIT-DSZ-ITSEC-9115-2000 [193] and later recertified on 2002-09-12 by the German Federal Office for Information Security (BSI) under reference BSI-DSZ-ITSEC-0175-2002 [194]. Since the security of the MICARDO operating system embedded on a different chip had not been evaluated, the certification report (Section 1.4 in [192]) states that the use of a different microcontroller may lead to recertification. However, we see that the ID cards issued from 2003 and after 2007 contain Infineon chip identifiers `0x22` and `0x2D` (see Section 3.1.2.1 above), which are different from the certified chip identifier `0x1E` (SLE66CX320P). Furthermore, contrary to the MICARDO User Manual, these cards have a formatted `EF_ATR` file that contains a single record and the operation system image identifier on these cards are `0x020000` and `0x030000` which are different from the `0x000001` that has been subject to certification. Therefore, we conclude that at least formally the MICARDO cards used in Estonia were not the product that had been certified.

In the course of this research, we discovered a security flaw in the MICARDO mutual authentication protocol that was covered by the certification (see Section 5.1.4). This shows that the ITSEC certification process of the MICARDO platform at least to some extent failed to assure the security of the product.

[10] ITSEC is one of the predecessors of the Common Criteria certification standard.

We note that even if the security of the MICARDO operating system held, it could not guarantee the security of the final smart card solution, as the MICARDO operating system had to be configured using a secure configuration. This issue was well ilustrated by a misconfiguration we found in all MICARDO-powered ID cards, which allowed card management operations to be performed using PIN2 (see Section 6.2).

3.2. MULTOS platform

The MULTOS platform was introduced at the end of 2010 and was used until the end of 2014 exclusively for issuance of the newly introduced *digital identity cards*. The *digital identity card* and the MULTOS platform are described in the subsections below.

3.2.1. *Digital identity card*

In October 2010, a new type of identity document *digital identity card* (Digi-ID) was introduced. Since this document can only be used electronically, it can be personalized in PPA customer service points and issued instantly. The purpose of the Digi-ID is to provide a backup solution in the event the cardholder's *identity card* cannot be used. The Digi-IDs are distributed by the ID card manufacturer to PPA with the private keys pre-generated (Section 6.1.2.1 in [195]). The only electronic personalization that has to be done in the PPA service point is certificate loading in the card. Instead of high-security laser engraving, heat-transfer printing is used to print cardholder details on the Digi-ID blank [196]. Similarly as in the case of the *identity card*, a person can have only one valid Digi-ID. Usually persons apply for Digi-ID when applying for the *identity card*. The certificates for MULTOS-powered Digi-IDs were issued with the validity period of 3 years. According to PPA, the validity was limited to 3 years due to the durability of the plastic material used for the card. The Digi-ID certificates can be distinguished from the *identity card* certificates, since Digi-ID certificates have the Organization Name (O), in the Subject Distinguished Name field, set to "ESTEID (DIGI-ID)" (the *identity card* certificates have the value set to "ESTEID") [197]. The barcode on the back of the card encodes the document number.

Figure 7: A MULTOS-powered *digital identity card* issued from 2010-10-01 to 2014-11-30 [182]

3.2.2. MULTOS-powered ID cards

The MULTOS platform is used exclusively in Digi-ID cards issued from 2010-10-01 to 2014-11-30 (see Figure 7). The subsequent Digi-ID cards are implemented on the jTOP SLE78 platform (see Section 3.4.4) and have a slightly different coloring.

The card is built on the MULTOS I4E[11] platform [199] produced by KeyCorp and is masked on Infineon's SLE66CX???PE[12] chip (see Figure 8). We are not aware of any security certifications for this MULTOS platform.

The MULTOS card application was developed to mimic the MICARDO interface described in the EstEID specification. The EstEID application for the MULTOS-powered ID cards was coded in a legacy C programming language by a contractor from the Estonian company "ID Süsteemide AS" [186].

The development of the MULTOS-powered ID card platform was already completed in 2008. At that time it was believed that MULTOS would be a manufacturer-independent platform to which all applications would move in the future. Therefore, the MULTOS platform was devised as the future platform for the Estonian ID card, but its use was delayed due to the reorganization of CMB. Later, after Gemalto bought Keycorp, it became evident that MULTOS would not be a fully open platform and hence the MULTOS platform was abandoned, giving preference to the JavaCard platform instead. [184, 200]

The MULTOS-powered ID cards are limited to 1024-bit RSA keys and are only able to communicate over T=0 transmission protocol [9]. The cards can be identified by their cold[13] and warm[14] ATRs (Section 3.1 in [9]). The warm ATR is a copy of the cold ATR from the MICARDO-powered ID cards and hence the warm ATR of MULTOS cards falsely offers T=1 transmission protocol [201]. From our certificate dataset we see that in total, approximately 15 thousand MULTOS-powered Digi-ID cards have been issued. The last card expired on 2017-11-26.

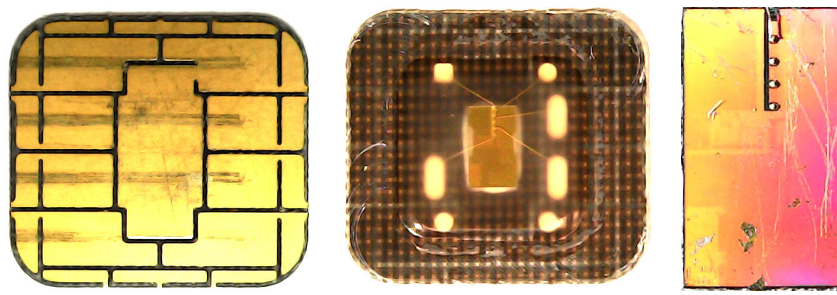


Figure 8: The chip module and microcontroller (SLE66CX???PE) used in MULTOS-powered ID cards

[11] The official documents [9, 198] have a typo as they refer to the implementation "IE4", which is non-existent.

[12] According to I4E specification the EEPROM size can either be 8 KB, 16 KB, 36 KB or 64 KB.

[13] 3B 6E 00 00 **45 73 74 45 49 44 20 76 65 72 20 31 2E 30**

[14] 3B FE 94 00 FF 80 B1 FA 45 1F 03 **45 73 74 45 49 44 20 76 65 72 20 31 2E 30 43**

3.3. jTOP SLE66 platform

Starting in 2011, a new platform based on Infineon's JCLX80JTOP20ID platform [198], further identified as the Estonian ID card platform jTOP SLE66, was used to manufacture the *identity card*. The same platform was also used for the new identity document type *residence permit card* introduced in 2011. The jTOP SLE66 platform and all further Estonian ID card platforms are based on JavaCard technology, therefore we will start this section by briefly introducing JavaCard technology and the related GlobalPlatform standard.

3.3.1. JavaCard and GlobalPlatform

JavaCard technology allows smart card application developers to write smart card applets using a subset of the Java programming language. The card operating system implements a JavaCard runtime environment and provides isolation between multiple JavaCard applets co-residing on a single smart card.

The functionality provided by JavaCard is identified by the JavaCard standard's API version. The operating system manufacturer can choose which subset of JavaCard API to implement, and can provide additional proprietary APIs, for example, to provide direct access to big number arithmetic. However, the use of proprietary APIs does not provide the cross-vendor applet interoperability that the JavaCard technology aims to achieve.

JavaCard platforms usually rely on the GlobalPlatform specification [202] for applet management. JavaCard applets can be installed and removed, but for security purposes the platform forbids the retrieval of installed applet instances and related data from the card.

GlobalPlatform card management operations with the card have to be performed over a secure channel based on symmetric or asymmetric keys. The GlobalPlatform standard defines several secure channel protocol versions and configurations the platform may implement. Several security domains can reside on a card and a separate set of card management keys can be used to manage each security domain and the applets associated with it.

3.3.2. jTOP SLE66-powered ID cards

The jTOP SLE66-powered ID cards were built on top of Infineon's product JCLX80jTOP20ID masked on Infineon's SLE66CX800PE chip [203] (see Figure 9). The cards run the jTOP (Java Trusted Open Platform) JavaCard operating system developed by Trusted Logic. The platform has a 75KB user-accessible EEPROM, and is compliant with JavaCard 2.2.1 API and GlobalPlatform 2.1.1 specification [203]. The platform also includes libraries available in JavaCard 2.2.2 API (page 8 in [204]). On this platform, the EstEID functionality was implemented in a JavaCard applet, which was the intellectual property of the ID card manufacturer (page 18 in [205]).

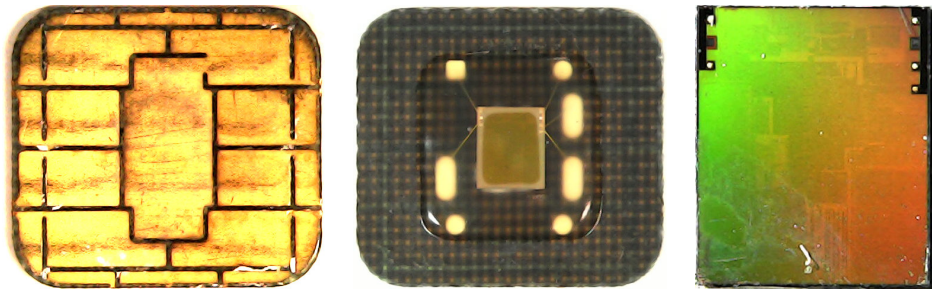


Figure 9: The chip module and microcontroller (SLE66CX800PE) used in jTOP SLE66-powered ID cards. The components are barely visible as the microcontroller is fully covered by protective mesh shielding (described by Infineon as "Active Shield").

The jTOP SLE66-powered ID cards support both T=0 and T=1 transmission protocols and can be identified by their cold[15] and warm[16] ATRs (Section 3.1 in [9]). Compared to the previous ID card platforms that used 1024-bit RSA keys, the jTOP SLE66 platform uses 2048-bit (and 2047-bit) RSA keys. The jTOP SLE66-powered ID cards were issued from January 2011 until the end of 2014. The last jTOP SLE66-powered ID card expired at the end of 2019.

The manufacturing of the first jTOP SLE66-powered ID cards faced issues. In January 2011, PPA recalled 46 ID cards that have been issued before 2011-01-18 for cardholders who had requested an urgent document issuance [206]. From the last byte of the warm ATR of the affected cards (see Section 3.1 in [9]), we see that the chip on these cards was not fully personalized, as the GlobalPlatfom card's life cycle state encoded in the last nibble corresponds to `OP_READY` (`0x01`) instead of the expected `SECURED` (`0x0F`) state. After the ID card manufacturer fixed the issue, the warm ATR of the card changed. Since the ID card software relied on the ATR to identify the Estonian ID card, the ID card software had to be updated [207].

At the end of 2011, a critical security flaw was found in the EstEID JavaCard applet v3.0 (see Section 6.4) and starting from 2012, a fixed EstEID applet v3.4 was installed on the jTOP SLE66-powered ID cards [208].

3.3.3. *Identity card*

With the switch to the jTOP SLE66 platform, the design of the *identity card* was updated (see Figure 10). The chip was moved to the back of the card to be in compliance with EU requirements for the *residence permit card* (see below). This caused increased technical support requests as cardholders' were accustomed to inserting the ID card in the reader with the cardholder's photo facing upward. Additionally, two barcodes, encoding the personal ID code of the cardholder and the document number, were added on the back of the card in the updated design.

[15] 3B FE 18 00 00 80 31 FE 45 **45 73 74 45 49 44 20 76 65 72 20 31 2E 30 A8**

[16] 3B FE 18 00 00 80 31 FE 45 **80 31 80 66 40 90 A4 16 2A 00 83 0F 90 00 EF**

Figure 10: A jTOP SLE66/SLE78-powered *identity card* issued from 2011-01-01 [182]

3.3.4. *Residence permit card*

With the introduction of the jTOP SLE66 platform, the *residence permit card* was launched (see Figure 11). The *residence permit card* is issued to non-EU third-country nationals residing in Estonia and was introduced to implement Regulation (EC) No 380/2008 [209], which established uniform format requirements for residence permits issued by EU member states to third-country nationals. The Estonian-specific design elements were the two barcodes on the back of the card that (the same as for the *identity card*) encoded the personal ID code of the cardholder and the document number.

Figure 11: A jTOP SLE66/SLE78-powered *residence permit card* issued from 2011-01-01 [182]. The 6-digit Card Access Number (CAN) was introduced for the SLE78-powered cards.

The *residence permit card* contains a separate contactless smart card chip (see Figure 12) that contains an ICAO-compliant electronic machine-readable travel document (eMRTD) ePassport application. The chip stores digitally signed cardholder data, including biometric data (a 480x640 pixel facial image and the cardholder's fingerprints). However, to read that information wirelessly, the terminal has to authenticate to the eMRTD applet and establish a secure channel using the Basic Access Control (BAC) mechanism [210]. To create the BAC key, the machine-readable zone (MRZ) of the *residence permit card* has to be optically read to extract the document number, expiration date and cardholder's date of birth. However, since the fields comprising the BAC key are also stored

on the EstEID applet in the personal data file, a contact reader can be used to construct the BAC key without needing to optically scan the MRZ.

To slow down BAC brute-force attacks, the BAC implementation of the eMRTD applet implements an incremental delay: after the first incorrect BAC try, the delay increases to 1 second; after the second try to 2 seconds; and after the third try it increases and stays at 10 seconds until a successful BAC is performed.

To provide integrity for the data stored on an eMRTD, the document issuer signs hashes of eMRTD data files and provides the signature together with the hash values in the `EF.SOD` file. However, the eMRTD chip on the jTOP SLE66-powered *residence permit card* has defects that prevent ICAO-compliant document inspection systems from verifying the signature. While the `EF.SOD` specifies SHA-256 as a hash function used to calculate hash values, the actual hash values contain SHA-1 hashes of the data files. Furthermore, the signature on the document signer certificate is not valid (signs a wrong hash), effectively preventing the verification of the authenticity of the data stored in the eMRTD.

To prevent eMRTD cloning attacks, the eMRTD applet supports the Chip Authentication mechanism (Section 6.2 in [210]) using the elliptic curve brainpoolP224r1.

The fingerprints of a cardholder are usually considered to be more sensitive biometric data than a facial image, therefore to access fingerprints, an additional Extended Access Control (EAC) mechanism with terminal authentication should be used. However, we found that contrary to the requirements of EU regulation [209], the fingerprints stored on the jTOP SLE66-powered *residence permit card* can be publicly read even without establishing a secure channel using the BAC mechanism.

The specification of the contactless eMRTD chip is not publicly available. The chip provides an ISO/IEC 14443 Type A contactless interface with a random 4-byte UID. The historical bytes of the ATS[17] (answer to select) encode ASCII string `MTCOSp`, which may suggest that the eMRTD functionality is implemented using the MTCOS Pro line of products.

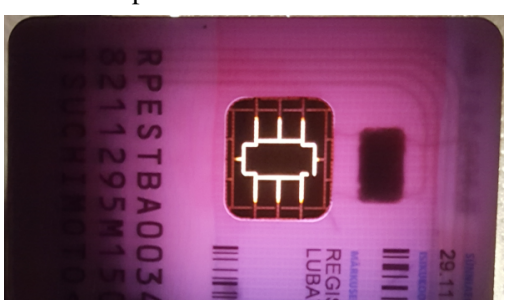


Figure 12: A contactless eMRTD chip and antenna in a *residence permit card*

[17] 3B 89 80 01 **4D 54 43 4F 53 70 02 01 05** 38

3.3.5. Common Criteria certification peculiarities

According to the information provided in the EU QSCD list [190] prepared by the Estonian authorities, the jTOP SLE66 platform used by the Estonian ID cards was Common Criteria certified on 2009-10-27 by the National Cybersecurity Agency of France (ANSSI) under the reference ANSSI-CC-2009/34 [211]. The certification process was based on the composite evaluation, where the object of evaluation was product JCLX80jTOP20ID, which was the result of embedding the jTOP JavaCard platform on top of a chip that had been independently certified by the chip manufacturer. The SLE66CX800PE chip was certified on 2008-05-27 by BSI under the reference BSI-DSZ-CC-0482-2008 [212].

The certification report for the JCLX80jTOP20ID platform (see Section 1.2.1 in [211]) describes how to correctly identify the unique product that had passed the certification. The historical bytes of the ATR among other things identify the software version of the product.

We found that contrary to the certification report, the expected patch version identifier 2.0 (`0x20`) was set to 0 (`0x00`) in the warm ATR of the actual jTOP SLE66-powered ID cards. This implies that the software version embedded on the chips used by the Estonian ID card were missing the version 2.0 patch, which was subject to certification.

Since the historical bytes of the ATR can be changed in the personalization phase through a JavaCard API call, the accuracy of the information contained in the ATR was confirmed by also retrieving the same software version identifier from the CPLC (Card Production Life Cycle) data using the GlobalPlatform [213] `GET DATA` command with a tag value of `0x9F7F`. This CPLC data cannot be changed by the ID card manufacturer in the card personalization phase.

On 2016-05-27, we informed the ID card manufacturer [214] about the discovered inconsistency. The ID card manufacturer explained that Infineon did not load 2.0 patch since this patch was only relevant for the ICAO LDS (eMRTD) application which was not used in Estonian ID cards.

As the certification report [211] does not state that the security objectives claimed in the security target are also met for a product that did not contain the patch, we have to accept this manufacturer's explanation at face value.

Another issue that questions whether the platform version delivered in the Estonian ID cards had been subject to the Common Criteria evaluation process is the fact that the RSA key generation algorithm implemented on the card, returns a 2047-bit RSA modulus 38% of the time when asked to generate a 2048-bit RSA key (see Section 4.3.1). While this is not a security issue, one would consider that such a basic functional bug would have been discovered in the extensive testing performed by the evaluation facility. According to the certification report [211], the evaluation of the JCLX80jTOP20ID platform was peformed by Serma Technologies in France.

3.4. jTOP SLE78 platform

Commencing at the end of 2014, the new jTOP SLE78 platform was used to produce the *identity card*, *residence permit card* and *digital identity card*. While all the previous Estonian ID card platforms used Infineon's SLE66 chip controller family, this platform used the improved Infineon's SLE78 security controller. The move away from the SLE66 microcontroller family was also motivated by the demonstration at the Black Hat 2010 conference, which showed that an expensive but practical attack against the SLE66 family of microcontrollers was possible [215].

The visual design of the *identity card* and *residence permit card* stayed the same as shown in Figure 10 and 11. The visual appearance of the *digital identity card* became more colorful (see Figure 14). The jTOP SLE78 platform was also used to power the newly introduced *e-resident's digital identity card* and *diplomatic identity card* described in the subsections below.

3.4.1. jTOP SLE78-powered ID cards

The jTOP SLE78 platform was implemented on top of Infineon's product "jTOP ID on SLE 78" [216] [18] using product configuration SLJ52GCA080CL [198]. The platform is masked on Infineon's SLE78CLX800P [217] chip (see Figure 13), runs jTOP ID JavaCard operating system developed by Trusted Logic, has an 80KB EEPROM, supports JavaCard 3.0.4 API and complies to GlobalPlatform 2.2.1 specification [216].

The chip supports both T=0 and T=1 transmission protocols and can be identified by their cold[19] and warm[20] ATRs. The warm ATR, however, is identical to the warm ATR of the jTOP SLE66-powered ID cards, therefore the updated EstEID specification recommends using the EstEID applet version identifier (Section 3.1 in [10]) to identify the card application.

Initially, the jTOP SLE78 platform used 2048-bit RSA keys, but due to Infineon's RSA key generation flaw (see Section 6.7), the switch to ECC keys using NIST curve P-384 was made at the end of 2017. The RSA key generation flaw was found in a component that was certified by the Common Criteria certification process. Therefore, the Common Criteria certification of the platform and the failure of this process are further analyzed in Section 6.7.5 under Infineon's RSA key generation flaw.

The jTOP SLE78-powered ID cards were issued until the end of 2018 and thus the last ID cards will expire at the end of 2023.

[18] In the Common Criteria certificate the product is identified as jTOP INFv#46 (SLJ 52G).

[19] 3B FA 18 00 00 80 31 FE 45 **FE 65 49 44 20 2F 20 50 4B 49 03**

[20] 3B FE 18 00 00 80 31 FE 45 **80 31 80 66 40 90 A4 16 2A 00 83 0F 90 00 EF**

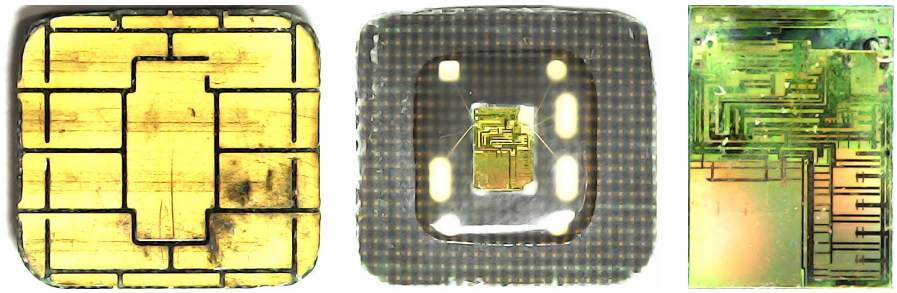


Figure 13: The chip module and microcontroller (SLE78CLX800P) used in jTOP SLE78-powered ID cards

3.4.1.1. EstEID applet versions

The EstEID functionality in the jTOP SLE78-powered ID cards was implemented as a version 3.5 EstEID JavaCard applet. Over the years, the EstEID applet received several updates. As the development of the EstEID applet was done by the ID card manufacturer and the authorities in secrecy, the exact changelog is not available. Below we list the most significant EstEID applet version 3.5 changes of which we are aware.

According to the EstEID specification (Section 1.3 in [10]) the jTOP SLE78-powered ID cards should return version identifier 3.5.1 or higher. However, the lowest version identifier we have observed on the jTOP SLE78-powered ID cards is version 3.5.2.

In fall 2015, EstEID applet version 3.5.3 was released to fix the public key encoding (see Section 6.6).

In order to replace the certificates with incorrectly encoded public keys, EstEID applet version 3.5.7 was released on 2016-06-20. This version was installed on new cards and also in the applet update process. Beginning with this version, RIA acquired the copyright ownership of the EstEID applet from the ID card manufacturer [218].

On 2017-10-25, EstEID applet version 3.5.8 was released as a response to the discovery of Infineon's RSA key generation flaw (see Section 6.7). This version replaced 2048-bit RSA keys with ECC keys using NIST curve P-384. The switch to ECC keys also resulted in an updated EstEID specification document [12] being released by RIA.

3.4.2. *Identity card*

Starting from 2014-10-17, the jTOP SLE78 platform was used to issue the *identity card*. As the chip contact layouts of jTOP SLE66 and jTOP SLE78 cards look the same, there is no visual feature that can be used to distinguish jTOP SLE66-powered *identity cards* from jTOP SLE78-powered *identity cards* (see Figure 10). However, jTOP SLE78-powered *identity cards* have the document numbers starting from AA0850000 and EA0040000 [219].

3.4.3. *Residence permit card*

Starting from 2014-12-19 [208], the jTOP SLE78 platform was used to issue *residence permit cards* with document numbers starting from BB0002000, FB0000100 and PB0010000 [219].

The jTOP SLE78-powered *residence permit card* had a new eMRTD contactless chip. The chip provided an ISO/IEC 14443 Type B contactless interface with a random 4-byte PUPI. The specification for the contactless eMRTD chip is not publicly available and the ATS[21] does not encode any known product identifier.

In addition to the BAC mechanism, the eMRTD implementation on the jTOP SLE78-powered *residence permit card* supports the Password Authenticated Connection Establishment (PACE) protocol that can be used to establish a secure channel with the applet. As a password, the PACE implementation allows the use of the key derived from the MRZ (the same that is used by BAC) or a 6-digit Card Access Number (CAN) that is printed on the front of the *residence permit card* (see Figure 11). The PACE implementation uses the elliptic curve brainpoolP256r1.

To slow down brute-force attacks, the eMRTD applet implements separate counters for BAC using MRZ, PACE using MRZ and PACE using CAN as a password. After each unsuccessful BAC/PACE establishment, the delay incrementally increases (0.2, 3.3, 6.4 and 10.6 seconds) and stays at 10.6 seconds until a secure channel is successfully established using that particular mechanism and password.

Similar to jTOP SLE66-powered *residence permit cards*, the signature on the document signer certificate is not valid (signs a wrong hash), effectively preventing the verification of the authenticity of the data stored in the eMRTD. Most likely because of this, on 2018-06-26, PPA issued a deviation list[22] [221] that contains document numbers of 169 422 jTOP SLE78-powered *residence permit cards* with the defect code `id-Deviation-LDSSODSignatureWrong`.

As with jTOP SLE66-powered *residence permit cards*, to prevent eMRTD cloning attacks, the Chip Authentication mechanism using the elliptic curve brainpoolP224r1 is supported.

Contrary to the eMRTD applet of the jTOP SLE66-powered cards, the fingerprints are not readable on jTOP SLE78-powered cards as they are likely protected by an EAC mechanism. European Commission decision C(2011) 5499 [222] required the PACE protocol to be implemented by 2014-12-31 at the latest. According to PPA [223], the new chip supporting the PACE protocol was implemented in *residence permit cards* starting from 2014-11-03. This means that there may also be jTOP SLE66-powered *residence permit cards* that have been shipped with this eMRTD chip and the CAN printed on the card.

[21] 3B 88 80 01 **00 00 00 00 77 81 91 00** 6E

[22] A deviation list is a machine-readable data file that is used by the document issuing state to notify relying parties of a non-conforming travel documents (Section 7 in [220]).

3.4.4. *Digital identity card*

The *digital identity card* issuance switched from MULTOS to jTOP SLE78 platform starting from 2014-12-01 [208]. These cards had document numbers starting from N0100000 [219]. Along with this switch, the visual appearance of the Digi-ID card became more colorful (see Figure 14).

Figure 14: A jTOP SLE78-powered *digital identity card* and *e-resident's digital identity card* issued from 2014-12-01 [182]

Starting from 2018-05-01, the validity period of *digital identity cards* was extended from 3 to 5 years. Starting from 2018-11-01, the cardholders of still valid *digital identity cards* that were valid for a 3-year period, were offered the opportunity to extend the validity of their certificates for an additional 2 years [224]. In total there were approximately 32 000 such cardholders, most of them e-residents (see the next section). The validity extension service was not provided in PPA service points but was instead only offered remotely through the use of the remote ID card applet replacement solution (see Section 5.4). The remote update possibility was discontinued on 2019-04-30 [225].

3.4.5. *E-resident's digital identity card*

On 2014-12-01, together with the switch from MULTOS to the jTOP SLE78 platform for Digi-ID issuance, the *e-resident's digital identity card* (e-resident's Digi-ID) was introduced. The e-resident's Digi-ID is issued to persons who are not residents of Estonia, but who have obtained an Estonian personal ID code through the e-Residency program [226]. In the context of the Identity Documents Act (IDA), the e-resident's Digi-ID is considered to be a subtype of Digi-ID, hence the term *digital identity card* in the law may also refer to the *e-resident's digital identity card*. In this work, however, we will use separate terms and abbreviations to refer to the particular document type.

The validity period of the e-resident's Digi-ID is inherited from Digi-ID and hence the validity period was also extended from 3 to 5 years for e-resident's Digi-ID on 2018-05-01. The visual design of the e-resident's Digi-ID is the same as for Digi-ID (see Figure 14). The document type can only be distinguished through the certificates. The certificates of the e-resident's Digi-ID have the Organization Name (O) set to "ESTEID (DIGI-ID E-RESIDENT)", while Digi-ID certificates have "ESTEID (DIGI-ID)" [197].

In the case of substantial public interest, the Minister of the Interior can decide to issue the e-resident's Digi-ID to a person without an application from that person. Over the years, several world-famous persons have received an e-resident's Digi-ID without them having asked for it [227]. This conflicts with the standard practice of CAs issuing the certificate based on a subject's application and acceptance of Terms and Conditions.

3.4.6. NFC-enabled *digital identity card*

In November 2014, the ID card manufacturer announced an NFC-enabled ID card pilot based on the jTOP SLE78-powered Digi-ID [200]. In March 2016, a video demonstrating the contactless Digi-ID prototype was published in the media [228].

The prototype was implemented on top of Infineon's "jTOP ID on SLE 78" product using the dual-interface product configuration SLJ52GCA080CL [216]. The certificates for the NFC-enabled Digi-IDs were issued with the validity period set to 2 years. In our certificate dataset we identified 50 such NFC-enabled Digi-IDs issued in 2014-12, 2015-03 and 2015-04, mostly to the employees of RIA, SK, PPA and the ID card manufacturer.

Even though qualified certificates for electronic signatures were issued for this NFC pilot platform, this NFC-enabled platform has been documented neither in law nor SK certificate policies. The conformance to SSCD requirements has not been assessed either, since the platform has not been included in the list of SSCDs used in Estonia [198].

3.4.7. *Diplomatic identity card*

On 2017-02-01, IDA introduced a *diplomatic identity card* (diplomatic ID card). Diplomatic ID cards are issued by the Ministry of Foreign Affairs (MFA) to replace the old diplomatic and service cards that did not provide the electronic functionality. The document numbers on the diplomatic ID cards start with A1, A2, A3, B1, B2, B3, C1, C2, C3, D1, D2, E1, F1, HC, G1, G2, G3 and G4, depending on the diplomatic status type (Section 2.2.1 in [229]). The design of the diplomatic ID card is shown in Figure 15.

Figure 15: A jTOP SLE78-powered *diplomatic identity card* issued from 2017-02-01 [230]

In the context of IDA, the diplomatic ID card is considered to be a subtype of the *identity card*, hence the term *identity card* in the law may also refer to the diplomatic ID card. SK, however, does not define the diplomatic ID card in their policies, but issues certificates for the diplomatic ID card under the Digi-ID certificate policy (i.e., the certificates of the diplomatic ID card have the Organization Name (O) set to "ESTEID (DIGI-ID)" in the Subject Distinguished Name field, the same as for Digi-IDs).

The validity period of the diplomatic ID card in IDA is regulated separately from the *identity card* and its validity period has been set to 5 years from the introduction of the diplomatic ID card.

3.5. IDEMIA platform

The latest generation Estonian ID card platform was introduced at the end of 2018. The cards are manufactured by IDEMIA (formerly Oberthur Technologies), but the personalization is performed by the Estonian company Hansab AS [131]. The IDEMIA-powered ID cards have a completely new design featuring a color photo of the cardholder, new security elements and a new smart card chip that has a new style contact layout and includes a contactless interface (see Figure 16, 17, 19 and 20). The added QR code at the back of the *identity card* encodes a link to the PPA website where the validity of the document can be verified[23]. The IDEMIA platform was used to issue every type of ID card starting from 2018-12-03. For cardholders who applied before 2018-12-03, the jTOP SLE78-powered ID cards were issued [134].

The new smart card platform is powered by the ID-One Cosmo v8.1 JavaCard open platform developed by Oberthur Technologies (now known as IDEMIA). The platform is embedded on the NXP P6022M VB microcontroller manufactured by NXP Semiconductors GmbH (see Figure 21). The platform has a 144KB EEPROM, supports JavaCard 3.0.4 API and complies to GlobalPlatform 2.2.1 (ID Configuration v1.0) specification.

The chip supports both T=0 and T=1 transmission protocols and returns the same cold and warm ATR[24] [13]. Digital signature and authentication keys are ECC keys using NIST P-384 curve.

In contrast to the previous generation *digital identity cards* (and *e-resident's digital identity cards*), the private keys on the IDEMIA-powered *digital identity cards* are not pre-generated, but are generated on the card by PPA in the personalization process (Section 3.2.1 in [70]). An additional change is that document numbers on IDEMIA-powered *digital identity cards* start with NA, while document numbers on IDEMIA-powered *e-resident's digital identity cards* start with UA.

[23] For example, `https://www2.politsei.ee/qr/?qr=AS0000302`.

[24] 3B DB 96 00 80 B1 FE 45 1F 83 **00 12 23 3F 53 65 49 44 0F 90 00 F1**

Figure 16: An IDEMIA-powered *identity card* [231]

Figure 17: The IDEMIA-powered *residence permit card* issued until 2020-09-30 [231]

Figure 18: The IDEMIA-powered *residence permit card* issued from 2020-10-01 [231]

Figure 19: An IDEMIA-powered *diplomatic identity card* [183]

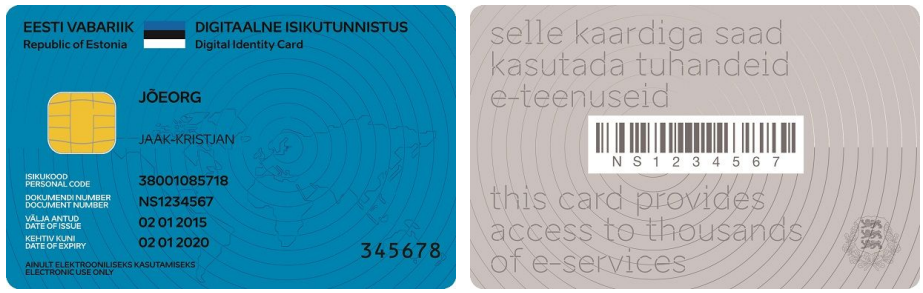


Figure 20: An IDEMIA-powered *digital identity card* and *e-resident's digital identity card* [231]

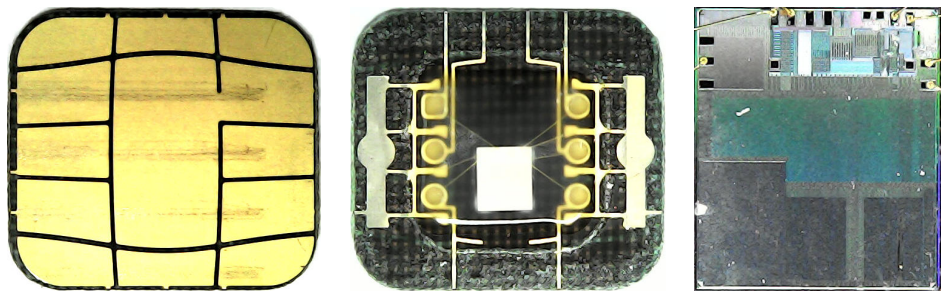


Figure 21: The chip module and microcontroller (NXP P6022M VB) used in IDEMIA-powered ID cards

3.5.1. IAS-ECC applet

On IDEMIA-powered ID cards the electronic functionality, which in previous ID card platforms was implemented by the EstEID applet, has been implemented using ID-One IAS-ECC V2 applet that was developed by IDEMIA. The applet is based on the Identification Authentication Signature – European Citizen Card (IAS-ECC) specification v1.01 [232] developed by the French smartcard industry association. The applet on top of the ID-One Cosmo v8.1 platform has been Common Criteria certified as a Secure Signature Creation Device (SSCD), thereby fulfilling the certification requirement of qualified electronic signature creation devices as required by eIDAS Article 30. The switch to the IAS-ECC applet was made mainly because the EstEID applet used in the previous generation Estonian ID card platforms had not been certified as is now required by eIDAS.

While most of the functionality provided by the EstEID applet has remained, the communication interface between the card and reader has changed. The main differences compared to the EstEID applet are: (1) the IAS-ECC applet is not the default selected applet[25] and hence it has to be explicitly selected by the reader

[25]The implicitly selected applet now is the GlobalPlatform Issuer Security Domain, except on *residence permit cards* where the eMRTD applet is the default selected applet.

after establishing a connection to the card; (2) certificates are stored in different locations; (3) each personal data file record is stored in a separate file; (4) key usage counters are not maintained; (5) to read the PIN retry counters an empty `VERIFY` command must be sent[26]. [233]

3.5.2. EE-GovCA2018

The certificates for IDEMIA-powered ID cards are issued by the intermediate CA ESTEID2018 under a new separate SK root CA EE-GovCA2018, which is used exclusively to issue certificates for Estonian state-issued identity documents. Several changes have been introduced in the ID card certificates: (1) CRLs are still issued, but the URL has been removed from the certificates; (2) the Organization Name (O) and Organizational Unit (OU) fields have been removed from the certificate's subject name; (3) the cardholder's personal ID code in the SerialNumber (SN) field of the subject name now contains the "PNOEE-" prefix; (4) the email address in the authentication certificate is now in the form `personal_ID_code@eesti.ee`. [233]

3.5.3. Contactless interface

The IDEMIA-powered ID cards are dual-interface cards (see Figure 22) that can be also used to communicate with the chip over the contactless interface. The contactless interface supports the ISO/IEC 14443 Type A standard and returns a random 4-byte UID and an ATS[27] containing the same historical bytes as the contact interface. To prevent unauthenticated contactless reading of the card, the card requires the commands sent over the contactless interface to be sent over a secure channel established with the PACE protocol using the elliptic curve NIST P-256. To establish the secure channel the 6-digit Card Access Number (CAN) printed on the ID card must be used as a password in PACE. The PACE protocol prevents offline password brute-force attacks, but to slow down online CAN brute-force attacks, the PACE implementation on the IAS-ECC applet introduces a 30 second delay after 10 consecutive tries to establish PACE using an incorrect password are made. [234]

The secure channel can also be optionally used over the contact interface to secure communication between the card and the application. Unfortunately, the privacy guarantees provided by the secure messaging are questionable, as the CAN is visible at the front of the card and the authorities have not informed the cardholders about the privacy sensitive nature of the CAN.

[26] This command is blocked when using smart card readers with a PIN pad and PIN firewall (see Section 2.14.1).

[27] 3B 8B 80 01 **00 12 23 3F 53 65 49 44 0F 90 00** A0

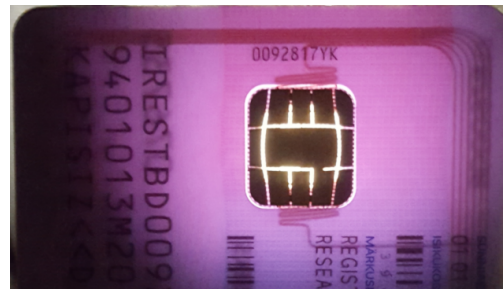


Figure 22: A dual-interface chip and antenna in an IDEMIA-powered ID card

3.5.4. *Residence permit card*

In the previous generation *residence permit cards* the eMRTD applet was implemented as a separate contactless chip. On the dual-interface IDEMIA-powered ID cards the eMRTD applet is loaded next to the IAS-ECC applet and is the default selected applet[28] as expected by eMRTD inspection systems.

To communicate with the eMRTD applet over both contact and contactless interfaces, a secure channel has to be established. To establish a secure channel either BAC or PACE can be used.

To slow down BAC brute-force attacks, after 15 consecutive incorrect passwords a 3 second delay is introduced in the BAC process. The PACE implementation on the eMRTD applet uses the elliptic curve BrainpoolP384r1 and allows the use of either CAN or MRZ as a password. To slow down CAN and MRZ brute-force attacks, the PACE implementation on the eMRTD applet maintains a separate incorrect password counter for CAN and MRZ. After 10 consecutive incorrect passwords of either type are used to establish PACE, a 15 second delay is introduced in the PACE process for that password type. It is interesting to note that the delay is not removed even after PACE is successfully established using a correct password, leaving the eMRTD applet in a permanently impaired state.

To prevent eMRTD cloning attacks, Chip Authentication using the elliptic curve brainpoolP256r1 is supported. The eMRTD applet on the IDEMIA-powered ID cards also supports the Active Authentication mechanism (Section 6.1 in [210]). To prove the authenticity of the eMRTD chip, in the Active Authentication process, the chip signs a random challenge (provided by the terminal) using ECDSA with the elliptic curve brainpoolP256r1.

With the introduction of IDEMIA-powered ID cards the personal ID code of the cardholder is now included in the MRZ and hence its authenticity is provided in the eMRTD of *residence permit cards*.

[28]The eMRTD applet is also the default selected applet on the contact interface. It can be explicitly selected using the AID `A00000024710FF`.

The visual design of the *residence permit card* changed on 2020-10-01 (see Figure 18). The EU member states were required to issue newly-designed residence permit cards on 2020-07-10 at the latest, however, they were allowed to use up the existing card stocks for an additional six months, i.e. until 2021-01-10 [235].

Starting from August 2021, the EU requires national identity cards to be equipped with an eMRTD applet [236]. Estonia plans to start shipping the *identity card* with an eMRTD applet starting in July 2021 (page 10 in [237]).

3.5.5. Common Criteria certification

The ID-One Cosmo v8.1 platform was Common Criteria EAL5+ certified on 2017-09-05 by ANSSI under the reference ANSSI-CC-2017/49 [238]. The platform is a composite product based on NXP P6022y VB microcontroller that was certified on 2016-10-11 by BSI under the reference BSI-DSZ-CC-0973-V2-2016 [239]. The ID-One IAS-ECC V2 applet was EAL5+ certified by ANSSI on 2018-05-04 under the reference ANSSI-CC-2018/17[29] [240].

The certification report of the IAS-ECC applet states that other JavaCard applets can be loaded next to the IAS-ECC applet, provided that the applet loading guidelines of the certified ID-One Cosmo v8.1 platform are followed (Section 1.2.2 in [241]). This means that additional non-certified applets can be loaded on the ID card without invalidating the certification of the IAS-ECC applet and hence the legal requirements for qualified electronic signature creation devices. Since the ID card chip has a significant amount of free space, RIA is considering opening the ID card platform to third-party JavaCard application developers [242].

3.5.5.1. Compliance issues

While studying the Common Criteria certification reports, we discovered that the software revision on the ID card chip of our IDEMIA-powered *digital identity card*, issued on 2019-01-02, did not correspond to the product identified in the Common Criteria certification report.

According to the certification report for the IAS-ECC applet (Section 1.2.2 in [241]), the certified product consists of three components: (1) the NXP microcontroller certified under the reference BSI-DSZ-CC-0973-V2-2016; (2) the ID-One Cosmo v8.1 platform certified under the reference ANSSI-CC-2017/49 and which has been the subject of maintenance ANSSI-CC-2017/49-M01; (3) and the ID-One IAS-ECC v2 applet in configuration #3.

[29] The ID-One IAS-ECC V2 applet was certified in 4 configurations. RIA and SK have confirmed that the Estonian ID card implements configuration #3.

After the initial certification of the ID-One Cosmo v8.1 platform (2017-09-05), three new revisions of the platform were the subject of certificate maintenance: M01 (2017-11-20), M02 (2019-01-24) and M03 (2020-01-21) [238]. To identify which revision of the platform had been installed on the ID card chip, we followed the instructions in the Security Target document (Section 2.4.3 in [243]) and used the `GET DATA` command with the tag value `0xDF52` to read the card identification data. This identified patch version 090871, which corresponds to the revision covered by the maintenance report for M02 (Section 2 in [244]).

From this we can see that while the ID-One IAS-ECC V2 applet had been certified to be used on top of revision M01 of the ID-One Cosmo v8.1 platform, the Estonian ID card chip contained revision M02 of the platform.

We informed RIA and SK about these findings on 2020-04-28. RIA's response was that according to IDEMIA the maintenance of the platform had no impact on the certification of the applications built on top of it. ANSSI confirmed this [245] by stating that they forgot to add such a statement in the maintenance reports for M01 and M02, but that it had been added to the maintenance report for M03. The question of whether the addition of such a statement to the certification maintenance report of the platform can retrospectively fix the applet's compliance to the certification requirement of eIDAS, we leave as an open issue.

Regardless of the issue discussed above, we note that the M02 revision of the ID-One Cosmo v8.1 platform only passed the maintenance assurance on 2019-01-24, meaning that the Estonian ID cards issued in the period from 2018-12-03 to 2019-01-24, strictly speaking, did not meet the eIDAS legal requirements for qualified electronic signature creation devices.

On 2020-07-21, the Director General of ANSSI wrote a clarification letter [246] stating that even though the maintenance report was only signed on 2019-01-24, the decision had been shared with IDEMIA on 2018-10-28 and hence the product issued by IDEMIA since 2018-12-03 is fully in accordance with ANSSI decisions.

3.6. ID card test cards

Over the years, SK has offered ID card test cards that can be used by developers and system integrators to test a system's compatibility with different ID card platforms [247]. The test cards fully replicate the visual appearance of the ID cards, including all the security features on it. The only difference from the real ID cards is that the test card has the word "SPECIMEN" placed on the front of the card and the identity information on the card is that of a fictitious cardholder.

The public-key certificates on these cards are issued under SK test CA hierarchy and SK provides test services where ID card test cards can be used. Before 2010, the test certificates were issued by intermediate CA TEST-SK,

chaining up to the SK production root CA. This was against Mozilla's CA root store policy as it requires each intermediate CA of a trusted root CA to be documented and audited. As a result, on 2009-12-01, SK revoked intermediate CA TEST-SK. [248]

In addition to standard operations, the MICARDO-powered ID card test cards could have also been used to test the card management operations as the card management keys on these test cards were set to the example values used in the specification (Section 14.3.2 in [5]).

The jTOP SLE66-powered ID card test cards were used by foreign researchers to test the susceptibility of the Estonian ID card's decryption functionality to the padding oracle attacks (see Section 6.3).

At the end of September, 2017, the ECC-enabled jTOP SLE78-powered ID card test cards were provided to service providers free of charge to facilitate the mitigation of the Estonian ID card crisis (Section 6.7).

3.7. SEB employee card

In September 2012, SEB introduced a corporate identity card (SEB employee card) built on top of Estonian ID card platform jTOP SLE66 and integrated with a separate contactless interface so that it could be used as a door card (see Figure 23) [249].

Figure 23: A jTOP SLE66-powered SEB employee card [249]

The SEB employee card has been included in this work as it uses (with minor differences) the same technical, legal and organizational solution as the Estonian ID card. The main difference is that the qualified certificates for the SEB employee card are not issued under the SK intermediate CAs "ESTEID-SK" which are used to issue qualified certificates for the state-issued identity documents. These certificates are instead issued under the SK intermediate CAs "EID-SK 2011" and "EID-SK 2016".

The SEB employee card can be used to create digital signatures with the same legal status as digital signatures created with the Estonian ID card. The only legal difference when compared to the Estonian ID card is that the authentication certificate of the SEB employee card does not have the state issued identity document status as provided by IDA. However, we have observed that in practice web servers configured for ID card authentication may allow (unless explicitly forbidden in the configuration) authentication with certificates issued by any SK intermediate CA (see Section III-B in [15]).

In the issuance process, SEB acts as a registration authority (RA) and has the right to suspend and revoke certificates. The certificates are issued with the validity period of 5 years. The main difference in the authentication certificate is that it contains the employee's SEB email address and includes the additional key usage "Microsoft Smart Card Logon" in the extended key usage extension [250]. The certificates of SEB employee cards are not published in the LDAP certificate repository. From 2015-03-30 to 2017-11-01, SEB employee card certificates were also issued to the employees of SEB sister companies in Latvia and Lithuania [251].

The switch to the jTOP SLE78 platform for the SEB employee card was done later than for the Estonian ID card – only in August 2015. The jTOP SLE78-powered SEB employee cards were also hit by Infineon's RSA key generation flaw (see Section 6.7) and the affected certificates were revoked in November 2017. The issuance of SEB employee cards (using ECC keys) later resumed, but only for SEB Estonia employees.

The issuance of SEB employee cards was terminated in April 2019 [251] as ID card manufacturer Gemalto ceased its operations in Estonia [55]. As of March 2020, SK has revoked all valid SEB employee card certificates and has ceased to provide certification services to SEB [251].

4. ASYMMETRIC CRYPTOGRAPHY PROVIDED BY ID CARD PLATFORMS

The main cryptographic functionality provided by the ID card is the on-card key generation of the cardholder's asymmetric authentication and digital signature key pairs and the execution of private key operations with them.

In this chapter we study the asymmetric cryptography algorithm implementations of each ID card platform. We measured the performance of key generation and private key operations on the platforms and attempted to reverse engineer the implementation of the RSA key generation algorithm by studying the properties of the keys generated by the platform. While two of the ID card platforms use Elliptic Curve Cryptography (ECC) for the cardholder's keys, we focus mainly on RSA, as the implementation of the RSA key generation algorithm can vary significantly. The properties of the generated RSA keys were analyzed based on the methods from the paper "The Million-Key Question – Investigating the Origins of RSA Public Keys" by Svenda et al. [252].

The properties of RSA implementations described in this chapter were used in the research outlined in Section 6.8 to verify whether the public keys contained in the ID card certificates had been generated by the on-card key generation algorithm on the respective ID card platform.

The timing for private key operations was measured by taking 100 000 measurements for each operation. The measurements were repeated at least twice to verify that the observed timing variance was stable. For RSA measurements, a 48-byte value was signed and a ciphertext containing encrypted 48-byte plaintext was decrypted. For ECC measurements, a 48-byte hash value was sent to the card for signing. The performance was only measured for the transmission of the command that sends the data and returns the result of the cryptographic operation, excluding any preprocessing such as setting security environments and performing PIN verification. Hence, the time was measured for sending a single smart card application protocol data unit (APDU) command and receiving its response, except for the 2048-bit RSA case where sending a 256-byte ciphertext required two APDUs. The APDU commands were sent over T=0 transmission protocol.

The sections below analyze each ID card platform separately with the final section providing a summary comparison of all the platforms.

4.1. MICARDO platform

The MICARDO-powered ID cards use 1024-bit RSA. In the subsections below we analyze 1024-bit RSA as implemented by the MICARDO operating system.

We note that unlike smart card microcontroller chips used for later ID card platforms, the SLE66 microcontrollers used by the MICARDO platform do not have an internal clock generator, which means that the performance of the chip is

proportional to the clock frequency supplied by the terminal. The chips used in later ID card platforms apparently have a built in clock as their performance does not depend on the clock frequency supplied by the terminal.

For the experiments, we used Gemalto IDBridge CT30 (formerly GemPC Twin/TR) smart card reader which supplies a 4.8 MHz clock frequency to the chip, while most of the smart card readers in the market supply a 4 MHz clock frequency [253].

4.1.1. RSA key generation

To study the RSA key generation algorithm implemented by the MICARDO platform, we analyzed more than 2 million 1024-bit RSA key pairs generated by 7 MICARDO-powered ID cards issued between 2002 and 2011. To generate and export the RSA keys, we exploited the flaw which allows the performance of card management operations using PIN2 (see Section 6.1). We note that the keys could have also been generated using the MICARDO-powered ID card test cards as the card management keys are known for these cards (see Section 3.6).

The MICARDO product supports RSA keys with a modulus length of up to 1024 bits. RSA key generation is performed using the `GENERATE PUBLIC KEY PAIR` command specifying the file identifier (FID) of the public key file in the APDU data field. The key generation routine reads the length of the expected RSA modulus N and the length of the public exponent e from the public key file (Section 4.7.3 in [185]). The maximum length of the exponent e can be set to either 2, 3 or 4 bytes. The value of the public exponent e cannot be set – it is generated randomly. The private key is stored in the Chinese remainder theorem (CRT) form (Section 4.7.2 in [185]). The FID of the file where the private key should be stored is specified in the `EF_KeyD` file. The key generation routine uses the key slot corresponding to the current security environment. The generated RSA modulus N and public exponent e are written to the public key file together with an RSA PKCS#1 signature over the values. The public key is either signed by the corresponding private key or by a key specified in the security environment.

The time distribution for 1024-bit RSA key generation on the tested cards is shown in Figure 24 (the outliers in the boxplot cover $< 5\%$ and $> 95\%$ percentiles). The cards generated 1024-bit RSA keys in 15.2 seconds on average (min – 4.7 seconds, max – 86.0 seconds).

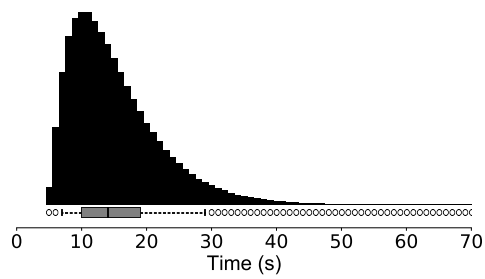


Figure 24: MICARDO: Time distribution for generating 1024-bit RSA keys

From the exported private keys we found that MICARDO implemented a non-standard RSA key generation algorithm as the generated primes were not fully balanced – the primes were never 512 bits long. The length of p varies from 513 to 520 bits (with equal probability) while the length of q varies respectively from 511 to 504 bits to produce a 1024-bit modulus N. Due to this imbalance, a 1024-bit RSA key generated by the MICARDO platform provides a lower security level than the standard 1024-bit RSA.

Figure 25 shows the distribution of the most significant byte (MSB) of p, q and N. Since the length of the primes is variable, the MSB is defined as the top 8 bits starting from the most significant bit that has been set. We note that the plots are the same also when looking separately at the MSB of primes with the same length. As we can see, the MSB of p and q has a bias that could be explained by some other nonstandard behavior of the algorithm, for instance, the primes being generated in a special form. The distribution of the most significant byte of N suggests that the rejection sampling method is used, meaning that the primes are regenerated if their product is not 1024 bits long (see Section 3.2 in [252]).

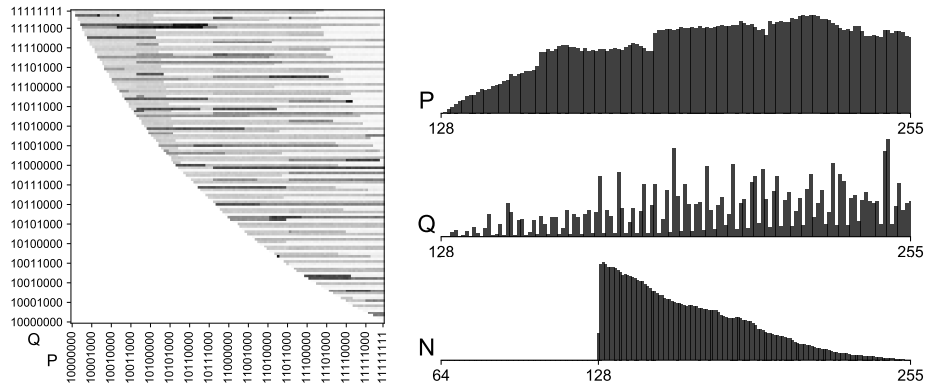


Figure 25: Distribution of most significant bytes of p, q and N from 1024-bit RSA keys generated by MICARDO platform. Lighter regions in the heatmap indicate where the MSB pair of p and q is less likely to appear and the darker regions – more likely.

The RSA algorithm requires public exponent e to be invertible, i.e., being co-prime to $\varphi(N) = (p-1)(q-1)$. A key generation algorithm will determine whether the selected primes are valid for the selected public exponent e when calculating the modular inverse to find the private exponent d. In the event that the values are not co-prime (i.e., the modular inverse cannot be found), the algorithm has to either select a different e or generate new primes until the private exponent d can be found. Today, it is a standard practice to use the public exponent value $e = 65537$. Since it is a prime number, the probability that it would not be co-prime to $\varphi(N)$ is insignificant (1/65537). As it has a special binary form, the exponentiation can be done very efficiently by implementing exponentiation by squaring. The random exponent e used by MICARDO has a

larger probability of having common factors (e.g., 3, 5, 7) with $p-1$ (or $q-1$) and therefore would not be valid for the chosen p and q primes. In the generated keys dataset, we see that the public exponents that have small prime factors are less represented. This means that when a randomly chosen e is not co-prime to $\varphi(N)$, another e is chosen so that the card does not need to regenerate the primes. For example, a randomly selected odd e should be divisible by 3 in 33.33% of the cases, while we find that only 8.37% of the chosen public exponents are divisible by 3. Furthermore, we find that among the generated exponents, the prime exponents preceded by prime exponents are less frequent when compared to prime exponents preceded by non-prime exponents (which are more likely to be rejected). This means that when e is not co-prime to $\varphi(N)$, the current e is incremented instead of choosing a new public exponent randomly.

Based on this information, we can define the possibly used MICARDO 1024-bit RSA key generation algorithm (see Algorithm 1).

Algorithm 1: The possibly used MICARDO 1024-bit RSA key generation algorithm

Input: *max_exponent_length*

1: $p\_len \leftarrow randomInt(513,520)$
2: $p \leftarrow getPrime(p\_len)$ {some bias in MSB is introduced}
3: $q \leftarrow getPrime(1024 - p\_len)$ {some bias in MSB is introduced}
4: $N \leftarrow p \cdot q$
5: **if** $bit\_length(N) < 1024$ **then**
6: **go to** 2
7: **end if**
8: $\varphi(N) \leftarrow (p-1)(q-1)$
9: $e \leftarrow randomBits(max\_exponent\_length)$
10: Set the least significant bit of e {make odd}
11: $d \leftarrow e^{-1} \bmod \varphi(N)$
12: **if** not d **then**
13: $e \leftarrow e+2$
14: **go to** 11
15: **end if**
16: **return** p, q, d, e, N

4.1.2. RSA private key operations

The MICARDO and later ID card platforms implement RSA signing using the RSA PKCS#1 v1.5 standard [254]. The value to be signed is received from the terminal, PKCS#1 v1.5 padding is applied to it, the signature value is calculated and then returned to the terminal.

Typically, for performance reasons, a hash value of data that needs to be signed is signed. The PKCS#1 and other digital signature standards expect the

ASN.1 DER-encoded `DigestInfo` structure to be signed, as, in addition to the hash value, it also encodes the algorithm identifier of the hash function used. Exceptions are older TLS protocol versions where a plain hash value of TLS protocol handshake has to be signed using PKCS#1 v1.5 in the client certificate authentication process. Starting with TLS v1.2, in the case of the RSA algorithm, the `DigestInfo` structure containing the hash value has to be signed.

The PKCS#1 v1.5 standard requires that the data that has to be signed is padded to the byte length of the modulus, where the padding is at least 3 bytes. Therefore the maximal length of data that can be signed is 3 bytes less than the RSA modulus size. The MICARDO platform, however, limits the value to be signed to 48 bytes (Section 5.1.3.7 in [185]). The result is that the MICARDO-powered ID cards can only be used to sign at a maximum SHA-224 hash values as the SHA-256 `DigestInfo` structure occupies 51 bytes. This caused issues when using the MICARDO-powered ID cards with the TLS v1.2 protocol [255].

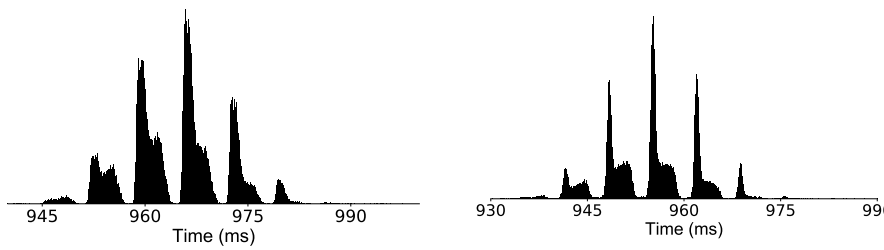
As an alternative, MICARDO supports the on-card hash calculation of data to be signed using the digital signature key (Section 15.2 [8]). In the on-card hash calculation process, the SHA-1 value of the data blocks received from the terminal is calculated, encapsulated in the `DigestInfo` structure and signed by applying PKCS#1 v1.5 padding to it. It is also possible to retrieve the calculated SHA-1 value without signing it. The on-card hash calculation has not been popular in practice, as it is only useful in cases where the terminal is unable to calculate the hash value itself due to limited resources.

The MICARDO and later ID card platforms support RSA decryption of ciphertexts padded according to the RSA PKCS#1 v1.5 standard [254]. The PKCS#1 v1.5 standard requires that the data that has to be encrypted is padded to the byte length of the modulus, where the padding is at least 11 bytes. Therefore the maximal length of data that can be encrypted is 11 bytes less than the RSA modulus size. The padding is in the format `0x00||0x02||PS||0x00||D`, where `PS` is a nonzero random padding at least 8 bytes in length and `D` is the plaintext data that needs to be encrypted. During decryption, the card removes the padding, returing decryption failure if the padding cannot be found.

The RSA PKCS#1 v1.5 encryption scheme is known to be vulnerable to a padding oracle attack [256] as it allows an attacker to forge a signature by learning whether the decryption of specially crafted ciphertexts contain valid PKCS#1 v1.5 padding. The complexity of the padding oracle attack depends on how permissive the padding check is – a more permissive check requires less oracle calls to forge a signature. The MICARDO platform verifies that the decrypted result starts with `0x00||0x02` and contains `0x00` somewhere after the first two bytes but not in the last byte (prohibiting empty plaintexts). We note that the decryption functionality of the Estonian ID card has been discussed in the context of padding oracle attacks, but the attacks have been found to lack practical exploitation scenarios (see Section 6.3).

It is interesting to note that contrary to other ID card platforms, the MICARDO platform does not require PIN2 verification after each decryption operation with the digital signature key.

On the MICARDO-powered ID cards, while using a 4.8 MHz smart card reader, RSA signing takes 0.964 seconds on average and RSA decryption 0.954 seconds on average. The timing distributions for the RSA signing and decryption operations for the MICARDO platform are shown in Figure 26. The signing and decryption operations experience a similar non-constant-time behavior as we can see several peaks following a symmetric timing distribution.



(a) signing (b) decryption

Figure 26: MICARDO: Timing distributions for RSA private key operations

4.2. MULTOS platform

The MULTOS-powered ID cards use 1024-bit RSA. We could not perform RSA key generation and export operations as we did not have access to a non-personalized card running the MULTOS platform. However, we used the public keys from our certificate dataset to study the potential RSA key generation algorithm implemented by the MULTOS platform. Our findings are outlined in the subsections below.

4.2.1. RSA key generation

According to the developer of the EstEID application for the MULTOS ID card platform, the RSA key generation algorithm was implemented using low-level code and the generation of a 1024-bit RSA key could take anywhere from 5 seconds to 2 minutes [186].

In our dataset we have 29 262 certificates issued for the MULTOS-powered ID cards. The public keys have random public exponents of up to 31 bits in length, mimicking the use of non-standard exponents as implemented by the MICARDO platform. All exponents, however, are prime numbers, the smallest exponent being the 24-bit prime 8565203. The use of prime exponents guarantees that the probability of e being co-prime to $\varphi(N)$ is significant.

We looked at the average distance between e and its previous prime to determine whether the algorithm selects candidate primes for public exponent e

using random sampling method or incremental search method (Section 3.1.1 in [252]). Primes which have a large gap between them and the previous prime are more likely to be selected in the incremental search. We calculated that for 31-bit primes, the incremental search resulted in an average distance of 36, while the random search had a distance of 20. Since we found that the distance for 31-bit public exponents from the MULTOS-powered ID card certificates is 36, we can conclude that the algorithm uses the incremental prime search method.

The distribution of the MSB values of the keys from the MULTOS-powered ID card certificates is shown in Figure 27. Such a distribution is the result of the RSA key generation algorithm setting the two most significant bits of p and q to 11_2 in order to guarantee that the generated modulus N is exactly 1024 bits long (see Section 3.2.2 in [252]).

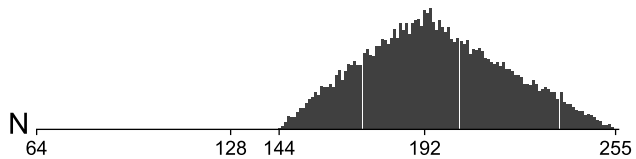


Figure 27: Distribution of the MSB of N from the MULTOS-powered ID card certificates

Based on the above, we can define the possibly used MULTOS 1024-bit RSA key generation algorithm (see Algorithm 2).

Algorithm 2: The possibly used MULTOS 1024-bit RSA key generation algorithm

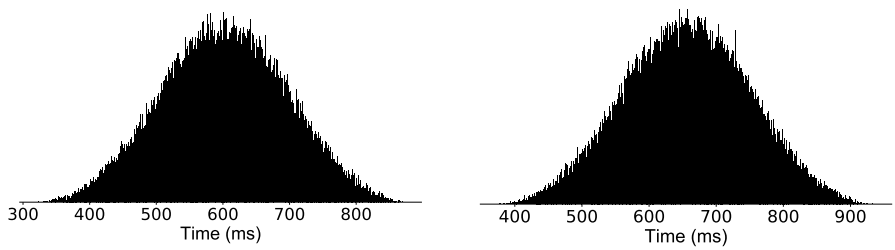
1: $p \leftarrow getPrime(512)$ {sets the two most significant bits}
2: $q \leftarrow getPrime(512)$ {sets the two most significant bits}
3: $N \leftarrow p \cdot q$
4: $\varphi(N) \leftarrow (p-1)(q-1)$
5: $e \leftarrow randomBits(31)$
6: $e \leftarrow nextPrime(e)$
7: $d \leftarrow e^{-1} \bmod \varphi(N)$
8: **if** not d {rare event} **then**
9: **go to** 1 or 6
10: **end if**
11: **return** p, q, d, e, N

4.2.2. RSA private key operations

The maximal size that can be signed using the MULTOS-powered ID card is 123 bytes, which is two bytes less than the maximal length allowed by the PKCS#1 v1.5 padding for a 1024-bit RSA modulus. The MULTOS-powered ID cards do not support on-card hash calculation of data to be signed.

The implemented PKCS#1 v1.5 padding check in the decryption process is very permissive as it only looks for a zero byte after the first byte in the decrypted result and returns everything that follows.

On the MULTOS-powered ID cards, RSA signing takes 0.603 seconds on average, while decryption takes 0.657 seconds on average. The timing distributions for RSA signing and decryption operations on the MULTOS platform are shown in Figure 28. Contrary to the other ID card platforms, the decryption and signing processes as implemented by the MULTOS platform has a high variance in time. This variance in the timing could be the result of applying some form of RSA blinding [257] in modular exponentiation to prevent side channel attacks.



(a) signing (b) decryption

Figure 28: MULTOS-powered ID card: Timing distributions for RSA private key operations

4.3. jTOP SLE66 platform

The jTOP SLE66-powered ID cards use 2048-bit RSA. Since we discovered that the ID card manufacturer had imported 2048-bit RSA keys generated outside the card on some of the jTOP SLE66-powered ID cards (see Section 6.8), we also describe the key import functionality supported by the platform in the subsection below.

4.3.1. RSA key generation

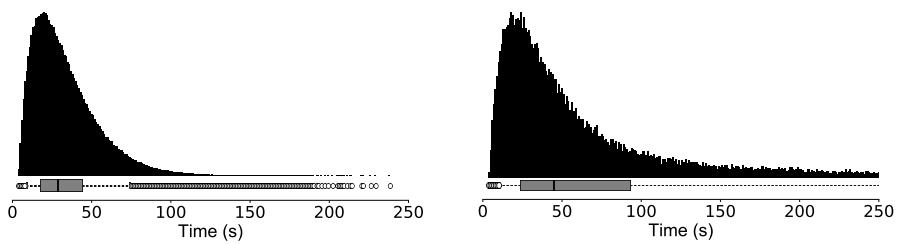
To study the RSA key generation algorithm and the properties of the generated keys, we exported 2048-bit RSA keys generated using blank jTOP SLE66 JavaCards. One million keys were generated using the default public exponent $e = 65537$ and an additional 63 000 keys were generated using a random public exponent. These additional keys were generated as the EstEID applet for jTOP SLE66-powered ID cards mimicked the behavior of the MICARDO platform and generated the keys using a random public exponent.

Since key generation is implemented in low-level native code on JavaCard platform, access to the manufacturer's proprietary EstEID JavaCard applet was not required. The JavaCard API provides the method `KeyPair.genKeyPair()` that can be used by a JavaCard applet to initiate key generation. The algorithm identifier and keylength is specified when initiating the `KeyPair` object.

The jTOP SLE66 platform supports 2048-bit RSA key generation using the `KeyPair.ALG_RSA_CRT` algorithm, i.e., the key pair is stored in a `KeyPair` object with the private key being stored in CRT format. Key generation in a non-CRT format using the `KeyPair.ALG_RSA` algorithm is only supported by the platform for up to 1024-bit RSA.

The JavaCard specification requires implementations to support arbitrary public exponent values for at least up to 4 bytes in length. The exponent can be set using the `RSAPublicKey.setExponent()` method. If no public exponent value is set, the key generation process will use the default value of $e = 65537$. The jTOP SLE66 platform accepts and is able to generate RSA key pairs with any odd value e of up to 4 bytes in length. From the jTOP SLE66-powered ID card certificates, we saw that the EstEID applet on the jTOP SLE66-powered ID cards initiated key generation using a random exponent of up to 31 bits in length.

The time distribution for 2048-bit RSA key generation on the jTOP SLE66 platform is shown in Figure 29. Key generation using the default public exponent $e = 65537$ takes 33 seconds on average (min – 4 seconds, max – 238 seconds). However, key generation using a random public exponent takes 87 seconds on average (min – 4 seconds, max – 1824 seconds) with 5% of the keys taking longer than 318 seconds (5 minutes and 18 seconds) to generate. In this case key generation is significantly slower, because a random exponent e has a larger probability of having common factors with $\varphi(N)$. In contrast to the MICARDO platform, the RSA key generation algorithm provided by the JavaCard API has to generate the key pair using the specified e, and therefore it has to search for new primes until $p-1$ and $q-1$ are co-prime to e.



(a) $e = 65537$ (b) Random e from interval $[1, 2^{31} - 1]$

Figure 29: jTOP SLE66: Time distribution for generating 2048-bit RSA keys

We note that the additional time introduced by the generation of keys with a random exponent and the rejection of 2047-bit RSA keys (see below) significantly decreased the throughput of the jTOP SLE66 ID card production line. This created a strong incentive to seek time-saving shortcuts and most likely was the reason why the ID card manufacturer decided to ignore the requirements and, during the ID card update process, imported keys generated outside the ID card (see Section 6.8).

The key generation algorithm can test whether the selected prime is valid for the specified e by checking whether $p-1$ is co-prime to e (i.e., by calculating the greatest common divisor (GCD) between $p-1$ and e). If $p-1$ is not co-prime to e (i.e., $GCD(p-1,e) > 0$), the algorithm can regenerate the prime until it is compatible. The alternative is to not test the compatibility of the primes separately, but regenerate both primes in the event that the private exponent d cannot be found. The regeneration of both primes if $\varphi(N)$ is not co-prime to e will take more time than only regenerating the incompatible prime, but will save two GCD computations if $\varphi(N)$ turns out to be co-prime to e.

To deduce which approach of prime compatibility checking is used by the jTOP SLE66 key generation algorithm, we measured the average key generation time when the exponent e values 3, 5, 7, 9, 11 and 13 are used (using 10 000 samples for each e). We then compared these values with the average key generation time when the public exponent value $e = 65537$ is used, as the probability that the chosen prime will not be compatible is insignificant ($1/65537$). The results from these measurements along with the expected values for both approaches are given in Table 2.

e	$GCD(p-1,e)$	$e^{-1} \bmod \varphi(N)$	jTOP SLE66
3	2.00	4.00	1.93
5	1.33	1.78	1.78
7	1.20	1.44	1.44
9	2.00	4.00	3.96
11	1.11	1.23	1.24
13	1.09	1.19	1.20

Table 2: The key generation time increase rate in respect to the public exponent e and the key generation algorithm used. The values given are the factors by which the time rate increases when compared to the key generation time when $e = 65537$.

We see that the time increase for the jTOP SLE66 key generation algorithm closely matches the expected increase when both primes are regenerated if the private exponent d cannot be found. The exception is $e = 3$, in which case an optimized key generation algorithm is likely used, as in this case the probability that at least one prime will not be co-prime to 3 is high.

The key generation algorithm has an uncommon property, because approximately 38% of all generated moduli are 2047 bits long. This ratio is close to the theoretical ratio of 38.6294% when p and q are chosen uniformly from the distribution of 1024-bit primes. Usually RSA key generation algorithms use either the rejection sampling method (regenerating primes until their product is of the required length) or sample the primes ensuring that a k-bit prime is larger than $\sqrt{2} \cdot 2^{k-1}$ (Section 3.2 in [252]). However, all primes generated by the jTOP SLE66 platform are 1024 bits in length with no apparent constraints put on the range from where these 1024-bit numbers are sampled (see Figure 30).

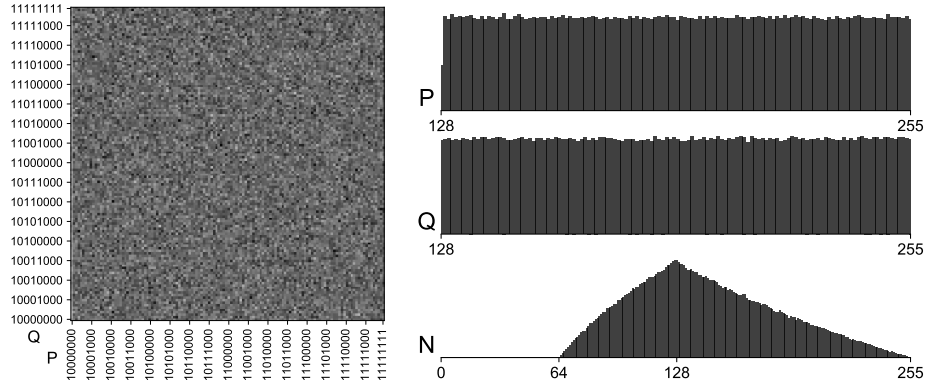


Figure 30: Distribution of the most significant bytes of p, q and N from the 2048-bit RSA keys generated by the jTOP SLE66 platform ($e = 65537$)

The bit distribution of primes is uniform, except that the most significant and the least significant bit are always set. The most significant byte of p (but not q) has a small bias – the probability of observing value 128 (10000000_2) is half that of observing a value greater than 128. We believe that this may be caused by a random number generator (RNG) "quality check" built into the RSA key generation algorithm which rejects RNG outputs that contain zero in the MSB. We found that this check only affects the initial selection of p, because the bias in p is barely observable in keys generated for public exponent $e = 9$, as in this case the probability that the primes will be regenerated is significant.

To determine whether the algorithm selects candidate primes using the random sampling method or the incremental search method, we analyzed the average distance between p and its previous prime. We calculated that for 1024-bit primes, the incremental search results in an average distance of 1410, while the random search has a distance of 710. These average distances will be slightly larger when the square region sampling method or the rejection sampling method is used, as the resulting primes will also be slightly larger. Since we found that the distance for 1024-bit primes generated by the jTOP SLE66 platform is 1410, we can conclude that the algorithm uses the incremental prime search method.

Based on the analysis given above, we can define the possibly used jTOP SLE66 2048-bit RSA key generation algorithm (see Algorithm 3).

4.3.2. RSA key import

The jTOP SLE66 platform supports 2048-bit RSA key import using the CRT format by initiating the `RSAPrivateCrtKey` object using the algorithm `KeyBuilder.TYPE_RSA_CRT_PRIVATE`. The CRT private key components p, q, $d \bmod (p-1)$, $d \bmod (q-1)$ and $q^{-1} \bmod p$ can be set with the methods `setP()`, `setQ()`, `setDP1()`, `setDQ1()` and `setPQ()`, respectively. Only odd p and q values can be set. Setting inconsistent CRT component values causes the card to crash when a private key operation is performed.

Algorithm 3: The possibly used jTOP SLE66 2048-bit RSA key generation algorithm (in case e is not 3)

Input: e (default: 65537)

1: While MSB of RNG buffer is 0, obtain a new RNG bufffer
2: $p \leftarrow randomBits(1024)$
3: $q \leftarrow randomBits(1024)$
4: Set the most significant bit of p and q
5: $p \leftarrow nextPrime(p)$
6: $q \leftarrow nextPrime(q)$
7: $\varphi(N) \leftarrow (p-1)(q-1)$
8: $d \leftarrow e^{-1} \bmod \varphi(N)$
9: **if** not d **then**
10: **go to** 5
11: **end if**
12: $N \leftarrow p \cdot q$
13: **return** p, q, d, N

The platform also supports 2048-bit RSA key import in the non-CRT format. This can be done by creating an empty `RSAPrivateKey` object using the `KeyBuilder.buildKey()` method, specifying the algorithm type `KeyBuilder.TYPE_RSA_PRIVATE` and the `KeyBuilder.LENGTH_RSA_2048` key length. The modulus and private exponent can be set using the `setModulus()` and `setExponent()` methods. The jTOP SLE66 platform accepts any odd private exponent d from 1 to 2048 bits in length and any odd modulus N from 1 to 2048 bits in length. However, private key operations with a modulus smaller than 10 bits in length produce an incorrect result and operations with a modulus smaller than 9 bits causes the card to enter an endless loop.

The keys imported using the non-CRT format are not usable in practice, as private key operations using a non-CRT key usually take around 30 seconds, instead of 1.4 seconds as is the case of RSA keys imported using the CRT format. Such a delay is not observed when importing 1024-bit RSA keys. It is somewhat unexpected to see that 2048-bit RSA import in the non-CRT format is even partially supported by the platform, as on-card 2048-bit RSA key generation in the non-CRT format is not supported at all.

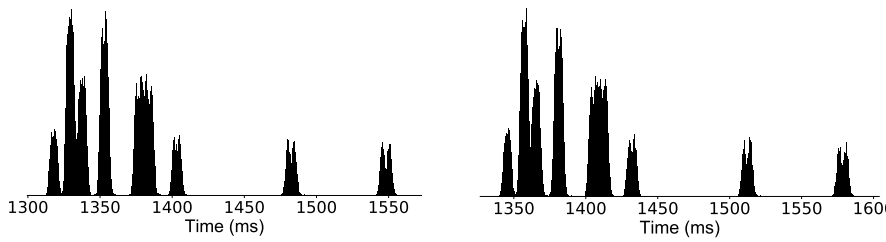
4.3.3. RSA private key operations

The maximal size that can be signed using the JavaCard API call provided by the jTOP SLE66 platform is 245 bytes, which is 8 bytes less than the maximal length allowed by the PKCS#1 v1.5 padding for a 2048-bit RSA modulus. The jTOP SLE66-powered ID cards support the on-card hash calculation of data to be signed, but do not provide an option to retrieve the calculated SHA-1 value.

The PKCS#1 v1.5 decryption padding check implemented by the JavaCard

platform is very strict as it checks that the first two bytes in the decrypted result are `0x00||0x02` followed by at least 8 nonzero bytes and then a zero byte somewhere after the nonzero padding.

On the jTOP SLE66-powered ID cards (EstEID v3.4 applet), RSA signing takes 1.413 seconds on average using the authentication key and 1.416 seconds on average using the digital signature key. RSA decryption takes 1.440 seconds on average using the authentication key and 1.443 seconds on average using the digital signature key. We noticed that the EstEID applet added some overhead, as our JavaCard applet on the jTOP SLE66 platform achieved an average of 1.375 for RSA signing and 1.403 for decryption. This overhead may have been caused by the additional state checks required (e.g., check if PIN has been entered). The timing distribution for RSA signing and decryption operations on the jTOP SLE66 platform are shown in Figure 31. The signing and decryption operations experience identical non-constant-time behavior as eight peaks with different distances from each other are visible in the distribution.



(a) signing (b) decryption

Figure 31: jTOP SLE66: Timing distributions for RSA private key operations

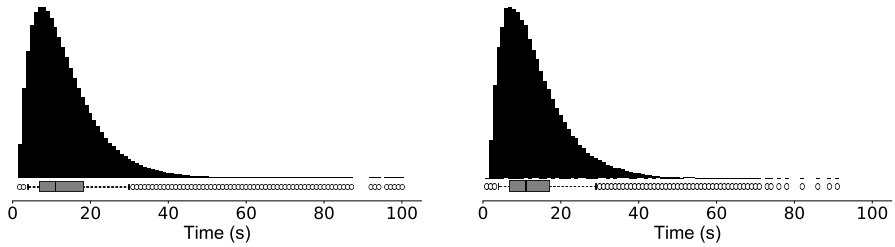
4.4. jTOP SLE78 platform

The jTOP SLE78-powered ID cards issued before 2017-10-25 use 2048-bit RSA. However, due to the discovery of Infineon's RSA key generation flaw (Section 6.7), the ID cards issued and renewed starting from 2017-10-25 use ECC with NIST curve P-384.

4.4.1. RSA key generation

To study the RSA key generation algorithm and the properties of the generated keys, we exported 2048-bit RSA keys generated using blank jTOP SLE78 JavaCards. Since the jTOP SLE78 platform supports 2048-bit RSA key generation using the CRT and non-CRT format, we generated 1 336 000 keys using the CRT format and 135 800 keys using the non-CRT format. We used the default public exponent $e = 65537$ as the ineffectual practice of generating keys with a random public exponent was discontinued for the EstEID v3.5 applet deployed on the jTOP SLE78 platform.

The time distribution for 2048-bit RSA key generation on the jTOP SLE78 platform is shown in Figure 32. Key generation in the CRT format takes 13.430 seconds on average (min – 2 seconds, max – 100 seconds), while key generation in the non-CRT format is slightly faster – 13.220 seconds on average (min – 1 second, max – 91 seconds).



(a) CRT form (b) non-CRT form

Figure 32: jTOP SLE78: Time distribution for generating 2048-bit RSA keys ($e = 65537$)

Private key operations using the private key in the CRT format only take around 0.3 seconds on average, while using the private key in non-CRT format takes around 0.8 seconds on average. Since the private key operations on the jTOP SLE78-powered ID cards take around 0.4 seconds, we can conclude that the EstEID applet uses RSA keys generated in the CRT format.

The distribution of the MSB of p, q and N is shown in Figure 33. As we can see, the distribution matches that of Infineon's faulty RSA key generation algorithm as shown in [252] (Card: Infineon JTOP 80K in Figures 10 and 11).

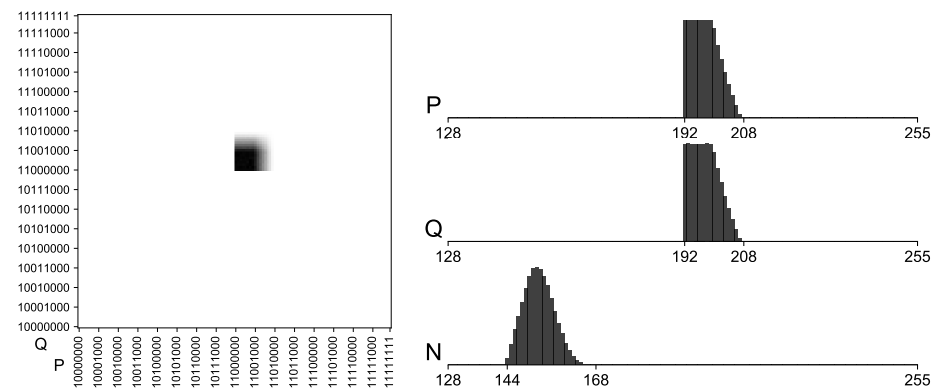


Figure 33: Distribution of the most significant bytes of p, q and N from the 2048-bit RSA keys generated by the jTOP SLE78 platform ($e = 65537$)

From the paper "The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli" by Nemec et al. [258], we already know that the faulty Infineon RSA key generation algorithm generates candidate primes for 2048-bit RSA in the form $p = k \cdot M + (65537^a \bmod M)$, where M is a 971-bit constant (the product of the first 126 primes $M = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 701$), k is a 53-bit

random number and a is a 255-bit random number. Such prime candidate generation method guarantees that the candidate will not have prime divisors up to 701, increasing the probability that the candidate is a prime from 1/355 (in the case of naive sampling of odd integers) to 1/60 (in the case of Infineon's approach above) on average.

Produit in [259] analyzed the keys generated by the jTOP SLE78 platform and observed that in practice k and a are not truly random as some of the most significant bits are fixed and have other biases. This observation allowed the original attack described in [258] to be significantly optimized.

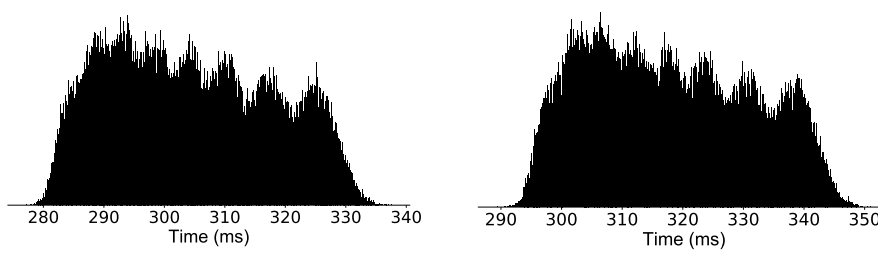
4.4.2. RSA private key operations

The JavaCard implementation of PKCS#1 v1.5 signing and decryption matches that of the jTOP SLE66 platform – the maximal size that can be signed is 245 bytes and the decryption padding check is strict. However, the EstEID applet of the jTOP SLE78 platform has a bug – the maximal size that can be signed using the digital signature key is only 127 bytes. It is possible that the bug in the EstEID applet is interpreting the APDU data length byte as a signed integer. This would result in byte values larger than 127 being interpreted as negative numbers, hence leading to an error condition when used.

The jTOP SLE78-powered ID cards support the on-card hash calculation of data to be signed and provide an option to retrieve the calculated SHA-1 value. However, support for on-card hash calculations has been removed from EstEID v3.5.7 and higher versions.

In contrast with the previous ID card platforms, the EstEID applet of the jTOP SLE78 platform does not support decryption with the digital signature key.

On the jTOP SLE78-powered ID cards RSA signing takes 0.391 seconds on average using the authentication key and 0.411 seconds on average using the digital signature key, while decryption using the authentication key takes 0.503 seconds on average. We noticed that the EstEID applet adds significant overhead, as our JavaCard applet on the jTOP SLE78 platform achieved an average of 0.304 for RSA signing and 0.318 for decryption. The timing distributions for RSA signing and decryption operations on the jTOP SLE78 platform are shown in Figure 34. The signing and decryption operations experience similar non-constant-time behavior as several peaks are visible in the distribution.



(a) signing (b) decryption

Figure 34: jTOP SLE78: Timing distributions for RSA private key operations

4.4.3. ECC key generation

Compared to the RSA key generation process where the algorithm has to find two large prime numbers, the generation of an Elliptic Curve (EC) key pair is trivial. In Elliptic Curve Cryptography (ECC) the private key is simply a random number that is smaller than the size of the EC group and the public key is calculated by multiplying the base point of the curve with the private key.

The JavaCard API provides the method `KeyPair.genKeyPair()` that can be used by a JavaCard applet to initiate key generation. To generate a NIST P-384 EC key the `KeyPair` object has to be initiated by specifying the algorithm identifier `KeyPair.ALG_EC_FP` and the keylength 384. The JavaCard API does not implement the standard named curves. The applet has to manually set the domain parameters of the curve on the `ECPrivateKey` and `ECPublicKey` interface objects.

We generated 2 million EC keys using the NIST P-384 curve on the jTOP SLE78 platform. The card generates EC keys in 0.365 seconds on average (min – 0.336 seconds, max – 0.414 seconds). The timing distribution for ECC key generation on the jTOP SLE78 platform is shown in Figure 35. The key generation process experiences non-constant-time behavior as there are several peaks visible in the distribution.

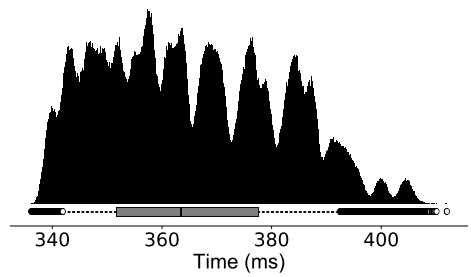


Figure 35: jTOP SLE78: Timing distribution for generating NIST P-384 EC keys

4.4.4. ECC private key operations

The signing operation using EC keys is implemented using the Elliptic Curve Digital Signature Algorithm (ECDSA). In ECDSA the `DigestInfo` structure is not used, but a raw hash value of the data to be signed is signed. The hash values that are longer that the size of the curve are truncated by ECDSA to the bit-length of the curve. Therefore, for the NIST P-384 curve used by the platform, it is optimal to use hash function SHA-384 as it provides the same security level of 192 bits as the 384-bit curve.

ECC support was introduced in the EstEID applet v3.5.8 released on 2017-10-25. The EstEID applet accepts 20, 28, 32, 48 or 64-byte hash value to be signed using ECDSA. These lengths correspond to the output of the hash functions SHA1, SHA-224, SHA-256, SHA-384 and SHA-512. As a response,

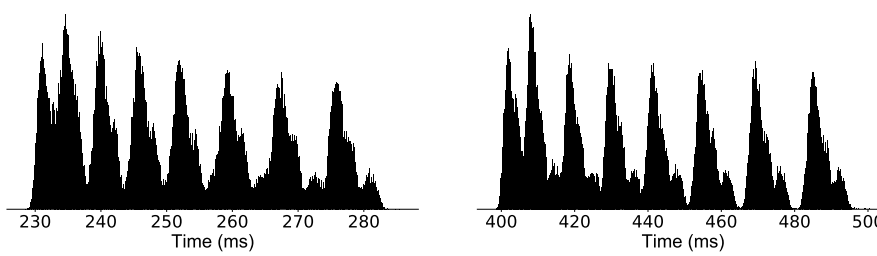
the card returns 96-byte signature value $r||s$, which is a concatenation of the 48-byte ECDSA signature components r and s. The JavaCard API method `signPreComputedHash()` returns ECDSA signature components r and s in a DER-encoded ASN.1 structure consisting of two `INTEGER` values in a `SEQUENCE`, therefore the EstEID applet has to parse the DER structure and extract the components.

In the ECC version of the EstEID applet, the decryption functionality is provided by the Elliptic Curve Diffie-Hellman (ECDH) key agreement using the cardholder's authentication key (see Section 2.9.1.1).

For ECDH calculations, the card expects a 97-byte value which is a public key point on the curve in the uncompressed format – the byte `0x04` followed by the 48-byte x and y coordinates of the point. The JavaCard runtime performs the check to verify that the received public key point is on the curve. As a response, the card returns a 48-byte shared secret which is the x-coordinate from the result of the received public key point multiplied by the cardholder's authentication private key. The shared secret can then be used by the decryption software to derive the symmetric encryption key.

On the jTOP SLE78-powered ID cards, the ECDSA operation takes 0.308 seconds on average using the authentication key and 0.314 seconds on average using the digital signature key, while the ECDH operation using the authentication key takes 0.481 seconds on average. The EstEID applet adds some overhead, as our JavaCard applet on the jTOP SLE78 platform achieved an average of 0.252 seconds for ECDSA and 0.440 seconds for ECDH. The timing distributions for ECDSA and ECDH operations on the jTOP SLE78 platform are shown in Figure 36. We see that both operations experience non-constant-time behavior as eight peaks of similar size are clearly visible.

We note that the authors of the paper "Minerva: The curse of ECDSA nonces; Systematic analysis of lattice attacks on noisy leakage of bit-length of ECDSA nonces" [260] have analyzed the jTOP SLE78 platform (Table 1 in [260]) but have not found that the non-constant-time implementation can be exploited to recover the bit-length of the scalar used in the scalar multiplication of ECDSA.



(a) ECDSA (b) ECDH

Figure 36: jTOP SLE78: Timing distributions for ECC private key operations using NIST P-384 curve

4.4.4.1. Invalid ECDSA signatures

The ECC-enabled EstEID applet has a bug that results in an invalid ECDSA signature being returned once per around every 60 000 signatures on average. The returned invalid signature values contain a pattern. The component r for the invalid signatures starts with two zero bytes and the component s starts with byte `0x02`. Additionally, in half of the cases the `0x02` byte is followed by `0x30` or `0x31`, which is a valid DER header for encoding a 48-byte `INTEGER` value. The observed rate of invalid signatures closely matches the probability of 1/65536 for observing r with two zero bytes in the leftmost position. Since we found no valid signatures where the r component would have two zero bytes in its leftmost position among the produced signatures, we concluded that the faulty signature is caused by the ASN.1 DER parsing as implemented by the EstEID applet in cases when the component r contains two zero bytes in its leftmost position.

We had already noticed the bug in the test cards running the EstEID applet v3.5.8 (v0.5 2017-09-21) and informed RIA about it on 2017-10-05. The bug, however, was not fixed and is also contained in the production version of the EstEID applet v3.5.8 shipped as of 2017-10-25.

4.4.4.2. Randomness in ECDSA signing process

From the security perspective, the move from RSA to ECC increases the reliance on the unpredictability of the RNG built into the chip. In the case of RSA PKCS#1 v1.5 signature scheme the use of good randomness is only required in the key generation process, whereas in the case of ECC, the ECDSA signing process requires unpredictable randomness for every signature. The quality of randomness in ECDSA is crucial, since the recovery of the private key from signatures becomes possible if only a few bits of the secret nonce k used in the ECDSA signing process are biased [261–263].

RFC 6979 standard [264] defines the deterministic generation of secret nonce k based on hash h to be signed and the private key d. The JavaCard API, however, does not support deterministic ECDSA and the ECDSA API in the JavaCard specification does not allow the nonce to be specified.

This issue is especially concerning due to the observation that the random number generator on the jTOP SLE78 platform produces numbers that are missing some properties of a random sequence (see Section 6.1 in [252]).

To test the randomness of EC private keys and ECDSA nonces, we generated and exported 2 million P-384 private key values. For each generated key we used on-card ECDSA to sign a 20-byte hash h with the JavaCard API method `signPreComputedHash()`. Using the private key d and hash h, we recovered the 48-byte secret nonce k from the signature (r,s) by computing $k = s^{-1}(h + rd) \bmod n$. In total, we obtained 91 MB of randomness from the private keys and 91 MB from the ECDSA nonces. The collected random data was tested using the Dieharder random number test suite [265], which did not detect any statistical issues in the data.

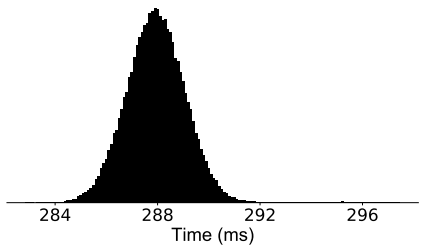
4.5. IDEMIA platform

The IDEMIA-powered ID cards are currently using ECC with the same NIST P-384 curve as used in the jTOP SLE78 platform. As we did not have access to a blank JavaCard powered by the IDEMIA platform, we were unable to perform key generation operations and related measurements. The observations described below have been made using a personalized IDEMIA-powered ID card.

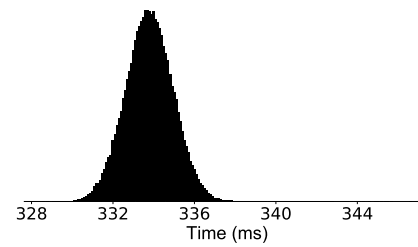
4.5.1. ECC private key operations

The IAS-ECC applet on IDEMIA-powered ID cards provides ECDSA and ECDH functionality similar to that of the EstEID v3.5.8 applet on the jTOP SLE78 platform. The ECDSA operation using the authentication key requires hash values of up to 48 bytes, while the ECDSA operation with the digital signature key requires values of exactly 48 bytes.

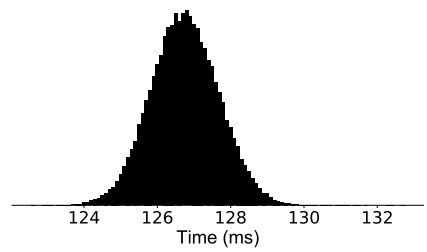
The timing distributions for ECDSA and ECDH operations on the IDEMIA-powered ID card are shown in Figure 37. The ECDSA operation using the digital signature key takes 0.334 seconds on average, while using the authentication key takes only 0.288 seconds on average. The ECDH operation using the authentication key takes 0.127 seconds on average. Contrary to the jTOP SLE78 platform, ECDSA and ECDH implementations on the IDEMIA platform exhibit constant-time behavior.



(a) ECDSA (authentication key)



(b) ECDSA (digital signature key)



(c) ECDH (authentication key)

Figure 37: IDEMIA-powered ID card: Timing distributions for ECC private key operations using NIST P-384 curve

4.6. Summary comparison

The summary for the asymmetric cryptography performance of the ID card platforms is shown in Table 3 and 4. The value specified in parenthesis is the lower bound that we obtained with our implementation on the chip used by the ID card platform.

Operation	1024-bit RSA		2048-bit RSA	
	MICARDO	MULTOS	jTOP SLE66	jTOP SLE78
Key generation	15.2	?	87 (33)	13.4
RSA signing (auth)	0.964	0.603	1.413 (1.375)	0.391 (0.304)
RSA signing (sign)	0.964	0.603	1.416 (1.375)	0.411 (0.304)
RSA decryption (auth)	0.954	0.657	1.440 (1.403)	0.503 (0.318)
RSA decryption (sign)	0.954	0.657	1.443 (1.403)	–

Table 3: RSA performance on the ID card platforms (average times in seconds)

Operation	NIST P-384	
	jTOP SLE78	IDEMIA
Key generation	0.365	?
ECDSA signing (auth)	0.308 (0.252)	0.288
ECDSA signing (sign)	0.314 (0.252)	0.334
ECDH key agreement (auth)	0.481 (0.440)	0.127

Table 4: ECC performance on the ID card platforms (average times in seconds)

As expected, the newer platforms provide better performance and asymmetric cryptography with a higher security level. ECC is significantly faster than RSA, performing key generation and cryptographic operations in less than half a second. While ECC was already supported by the jTOP SLE66 ID card platform introduced in 2011, the switch to ECC was only triggered in fall 2017 by the discovery of the flaw in the RSA implementation on the jTOP SLE78 platform. The migration from RSA to ECC, however, could not have happened any earlier, as ECC support for TLS client certificate authentication was only introduced in TLS v1.2[1] and the move to the BDOC digital signature file format supporting ECC only started in 2016.

It is interesting to find that the RSA algorithm on each ID card platform is implemented with slight differences. It is interesting to note that only on the IDEMIA platform the cryptographic operations exhibit constant-time behavior. The differences in the timing behavior also show that the math operations on each platform are implemented differently. It would be useful to further research the exact reasons behind these differences.

[1]The widely used OpenSSL library only introduced support for TLS v1.2 in OpenSSL v1.0.1 released on 2012-03-14.

5. ID CARD REMOTE UPDATE SOLUTIONS

The ability to remotely update some components of the ID card after it has already been issued has been an important feature, that has helped the card issuer to sustain the validity of certificates and even fix critical security flaws. While some of the ID card remote update scenarios are limited to simple certificate overwriting, some scenarios go as far as replacing the whole EstEID applet on the card.

This chapter describes the ID card remote update functionality used throughout the years for different ID card platforms and analyzes the security of the implementations. We start by first introducing the EstEID secure messaging protocol which is used to perform card management operations on the ID card platforms that implement the EstEID specification.

5.1. EstEID secure messaging

The Estonian ID card supports several card management operations that can be performed by the card issuer after the ID card has been personalized and delivered to the cardholder. To authenticate the card issuer, the card management commands sent to the card have to be sent over a secure messaging protocol. The secure messaging protocol is applied to the smart card APDUs exchanged between the card and the card issuer's backend.

In the card personalization phase several card-specific symmetric 3DES keys are loaded on the card, each used to authenticate a particular card management operation (see Section 19 in [9]). For PIN reset operations and the generation of new RSA key pairs, a secure messaging session must be negotiated using the card management keys CMK1 and CMK2a, respectively. Card management keys are used to derive unique session keys, which are then used to secure the actual smart card command (C-APDU) and response (R-APDU) pairs. The same secure messaging protocol is also used for passphrase authentication using 3DESKey1 and 3DESKey2 (see Section 6.1.1).

For card management operations such as certificate overwriting, changing active key references and creating additional application structures on the card, a secure messaging session is not established. However, to provide integrity for the APDUs exchanged, a message authentication code (MAC) is applied to the APDUs using the card management keys CMK2b and CMK3 directly.

The secure messaging protocol is defined in Sections 5.5.2, 5.5.3 and 5.6 of the MICARDO User Manual [185]. The same protocol has been reimplemented in EstEID implementations on subsequent ID card platforms. In this section we provide the description of the EstEID secure messaging protocol and describe the security flaws found in it.

5.1.1. EstEID secure messaging protocol

The EstEID secure messaging protocol consists of a session key negotiation phase and a secure messaging phase in which cryptographically protected APDUs are exchanged. The EstEID secure messaging protocol is depicted in Figure 38. The figure shows the content of the APDU data field for C-APDU and R-APDU pairs. The C-APDU header and R-APDU status word (SW) fields are sent in clear text. These values are not depicted in the figure. The number in square brackets represents the byte length of the value.

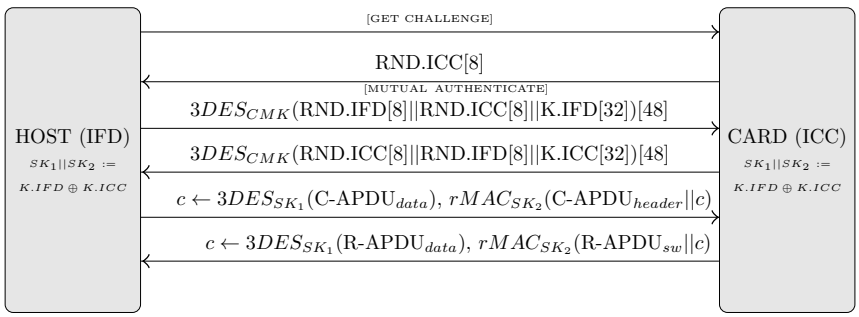


Figure 38: EstEID secure messaging protocol

5.1.2. Session key negotiation phase

The goal of the key negotiation protocol is to negotiate two 16-byte session keys SK1 (used for encryption) and SK2 (used for MAC), and the 8-byte send sequence counter SSC which serves as an IV (initialization vector) for encryption and MAC calculations. The values are negotiated based on a specific 3DES card management key (CMK) which is shared between the card and the host (terminal). The session key negotiation process involves mutual authentication, guaranteeing that the session key can only be derived by the parties who share the CMK. The security of the protocol relies on the randomness provided by the card and the terminal.

The protocol starts with the card returning 8-byte randomness RND.ICC as a response to the host's `GET CHALLENGE` command. The host then generates its 8-byte randomness RND.IFD and a random 32-byte session key share K.IFD. The host concatenates values RND.IFD, RND.ICC and K.IFD, and encrypts them with the CMK using 3DES in CBC mode without padding and an IV set to zeros. The ciphertext is sent to the card with the `MUTUAL AUTHENTICATE` command. By finding the card's randomness RND.ICC in the decrypted plaintext, the card is convinced that the host knows the CMK. As a response, the card returns the encryption of RND.ICC, RND.IFD and its randomly generated 32-byte session key share K.ICC (encrypted using an IV set to zeros). By decrypting the ciphertext and finding its randomness RND.IFD in the plaintext, the host is convinced that the card knows CMK.

The session keys are derived by XORing the 32-byte values K.IFD and K.ICC. The first 16 bytes of the result are used as SK1 and the remaining 16 bytes as SK2. The 8-byte SSC is constructed by concatenating the last four bytes of RND.IFD and RND.ICC.

5.1.3. Secure messaging phase

The APDU commands and responses sent over a secure messaging session have integrity protection and are optionally encrypted. The secure messaging format follows the so-called Encrypt-then-MAC approach. The original APDU data in the command and response APDUs is encrypted with SK1 by applying 3DES in CBC mode with SSC as the IV. It is important to note that the secure messaging protocol is only able to encrypt APDU data, therefore the C-APDU header which includes the command identifier and parameters P1 and P2 cannot be concealed. Similarly, the status word (SW) of the R-APDUs is sent in clear text.

The MAC protects the C-APDU header (and R-APDU SW) and encrypted (if encryption is applied) APDU data. The MAC is calculated using SK2 by applying the retail MAC (ISO 9797-1 MAC algorithm 3) in CFB mode with SSC as the IV. The SSC is incremented before each command and response APDU is produced. This prevents message reordering and replay attacks.

5.1.4. Card impersonation attack

The MICARDO User Manual [185] in Section 5.5.2 claims that "the negotiation of session keys involves mutual authentication of the chip card and the external world". However, we found a flaw in the MICARDO session key negotiation protocol that allows an attacker to impersonate the card without knowing the CMK.

5.1.4.1. Attack

The attack works by answering the host's `MUTUAL AUTHENTICATE` command with a modification of the same ciphertext that was sent by the host itself (so-called reflection attack). An important observation is that the host will accept the response of the `MUTUAL AUTHENTICATE` command as valid if the host's RND.IFD value is located in the second plaintext position of the card's encrypted response (for this particular attack we assume that the host will ignore the verification of RND.ICC in the first plaintext position). Due to the fact that the plaintext data fields (RND.IFD, RND.ICC and K.IFD) align to the block boundaries of the 3DES cipher (8-byte block size), it is possible to forge a valid response without knowing the CMK key. In order to do so, the attacker's response must be the host's own ciphertext with the first 2 blocks modified. The first ciphertext block must be moved to the second position and the new first ciphertext block must be set to zeros, as the first block of the original ciphertext was encrypted using an IV of zeros.

Even though the host's key share K.IFD (and hence the forged K.ICC) is not known to the attacker, the session keys derived by the host are, however, known due to the properties of the XOR operation. All the bytes of K.IFD $\oplus$ K.ICC will be zeros, except for the first 8 bytes, because during the decryption of the forged ciphertext, a wrong IV will be used by the host to decrypt the first block of K.ICC. We know, however, that these first 8 bytes is the value of the second ciphertext block of the host's ciphertext XORed with the second ciphertext block of the forged ciphertext.

Since the host's RND.IFD is not known to the attacker, the 4 bytes of the SSC value are unknown. The whole SSC, however, can be trivially recovered from the first C-APDU received in the negotiated secure messaging session. To recover the SSC, the MAC process has to be reversed – the MAC has to be decrypted using SK2 and the MACed data.

We implemented a proof-of-concept card impersonation attack on a JavaCard and tested it against our own protocol implementation on the host's side. The pseudocode of the attack is shown in Algorithm 4. The square brackets represent the byte slicing operator.

Algorithm 4: Card impersonation attack (card's side)

1: $RND.ICC \leftarrow random(8)$
2: Return $RND.ICC$ as a response to host's `GET CHALLENGE`
3: $req \leftarrow$ C-APDU data from host's `MUTUAL AUTHENTICATE`
4: $resp \leftarrow 0000000000000000_{hex} \,||\, req[0:8] \,||\, req[16:24]$
5: Return $resp$ as a response to host's `MUTUAL AUTHENTICATE`
6: $SK_1 \leftarrow req[8:16] \oplus resp[8:16] \,||\, 0000000000000000_{hex}$
7: $SK_2 \leftarrow 0000000000000000_{hex} \,||\, 0000000000000000_{hex}$
8: $req\_MACed \leftarrow$ C-APDU data of first MAC-protected C-APDU
9: $SSC \leftarrow retail\_MAC\_reverse_{SK2}(req\_MACed)$

There seem to be two scenarios where this card impersonation attack could be exploited in practice. The first exploitation scenario is the remote ID card update process that involves the generation of new key pairs. The attack would allow qualified certificates for keys generated outside the ID card to be obtained. Another exploitation scenario is the new PIN envelope issuance process, which would allow the attacker to simulate a successful change of PIN codes, without the operations being actually performed on the ID card (at first glance, however, it seems that this does not give any benefit to the attacker).

5.1.4.2. Mitigation

The correct protocol-level fix would be to protect the ciphertexts exchanged in the `MUTUAL AUTHENTICATE` command against modification using a MAC. This, however, would require the EstEID implementation on the card's side to be updated, which is not feasible.

To mitigate this particular attack, the host could reject the APDU responses of the `MUTUAL AUTHENTICATE` command, when the decryption of the first block does not contain the RND.ICC that was obtained using the `GET CHALLENGE` command. This, however, is vulnerable to an impersonation attack in a man-in-the-middle scenario, where the response from the real card is modified by replacing K.ICC with a copy of K.IFD from the host's ciphertext. To prevent both attacks, the host must reject `MUTUAL AUTHENTICATE` responses where a ciphertext block in any position of the card's ciphertext matches a ciphertext block in any position of the host's ciphertext.

5.1.4.3. Disclosure

We informed RIA of the flaw on 2016-02-03. As a response, the ID card remote update implementation for the jTOP SLE78-powered ID cards (Section 5.4) was modified to introduce the mitigation measure described above. We are not aware, however, if the mitigation measure was also applied for the implementation used to issue new PIN envelopes.

5.1.4.4. Failure of ITSEC certification process

After we realized that the EstEID secure messaging protocol was a reimplementation of the MICARDO secure messaging protocol, which was ITSEC certified by certification body TÜViT (Section 3.1.3), we decided to disclose the flaw to TÜViT. On the grounds that the certificate was no longer valid and that the documentation had been deleted 10 years after the certification, TÜViT expressed no interest in the flaw [266].

On 2018-10-25 and 2018-11-13, we submitted a message using the contact form available on `morpho.com` and `idemia.com` asking for security contacts to report the security vulnerability in the MICARDO product. In both cases we received a confirmation from IDEMIA that our message had been received, but no one ever contacted us back on the matter.

We observed that the newer versions[1] of the MICARDO card operating system have been Common Criteria certified by BSI, with the latest certificate issued in 2014 under certification reference BSI-DSZ-CC-0861-2014 for the product "MICARDO V4.0 R1.0 eHC v1.2". On 2019-01-23, we contacted BSI asking them to verify if these related products certified by BSI were also affected and to use their authority to ensure that the information about the flaw reached the affected parties. On 2019-02-20, BSI responded with IDEMIA's project manager specified in the email's CC field, asking for more details on the flaw, which we provided on 2019-02-25. On 2019-03-29, BSI responded with the findings that these other certified MICARDO products were not affected, since

[1] MICARDO certifications in question: BSI-DSZ-CC-0358-2006, BSI-DSZ-CC-0390-2007, BSI-DSZ-CC-0392-2007, BSI-DSZ-CC-0602-2009, BSI-DSZ-CC-0391-2009, BSI-DSZ-CC-0603-2010, BSI-DSZ-CC-0604-2010, BSI-DSZ-CC-0673-2010, BSI-DSZ-CC-0661-2011 and BSI-DSZ-CC-0861-2014.

they do not use the unauthenticated DES-based symmetric authentication protocol. To cite BSI: "The ITSEC certified MICARDO product was implemented according to a very old HPC specification in the framework of the German Health Care System. This specification shows the security problem of an unauthenticated DES-based symmetric authentication protocol, but this was detected several years ago and appropriately fixed for following specification versions as those are used e.g. for the MICARDO products listed above." [267]

This case (and the case described in Section 6.7) shows that the security certification process does not guarantee an absence of security flaws. Even though the flaw was later found in the specification, this information did not reach the affected parties, which in this particular case led to the flawed protocol being reimplemented on the later Estonian ID card platforms.

5.1.5. MAC protection using the CMK directly

For the card management operations secured with the CMK2b and CMK3 keys, the APDUs are MAC-protected without establishing a secure messaging session. The MAC is calculated using the respective card management keys directly by applying the retail MAC in CBC mode with the IV set to zeros.

It is important to note that this type of MAC protection provides very weak integrity properties. The MAC is not bound to any session-specific information, which means that the C-APDUs observed can be successfully replayed against the card at a later time. Furthermore, while the card's R-APDUs are MAC protected, the MAC is not bound to the C-APDU for which the R-APDU has been produced. This means that by observing one MAC-protected R-APDU with the status word `0x9000` (success), the same R-APDU can be replayed to the backend as a response to another command, convincing the backend that the command was executed successfully without actually passing the C-APDU to the card.

We informed RIA on 2016-02-12 of this weakness and its effect on the remote update process.

5.2. MICARDO-powered ID cards

The MICARDO-powered ID cards issued from 2002 were valid for 10 years, while the validity period of certificates was set to 3 years. On 2004-12-03, SK and the Ministry of the Interior agreed that the state would pay SK for the renewal of certificates whose 3-year validity period will be soon ending. From January 2005, cardholders were provided with a certificate update possibility either in customer service points of CMB and SK or remotely over the internet. The certificates could be updated free of charge if the remaining validity of the certificate was less than 105 days. While suspended and revoked certificates could not be updated, it was possible to update (including remotely) expired certificates, as long as the ID card itself was valid. The customer service points of SK (branches of Swedbank and SEB) provided a paid service to update certificates that qualified for the free of charge update described above. [74, 268].

On the client side, the remote updating was implemented using an ActiveX component which served as a communication relay between the SK backend and the smart card connected to the user's computer. The solution, however, only worked with Microsoft Windows and Internet Explorer. Later, when the new generation ID card desktop software launched at the end of 2010, the remote update functionality was also implemented in the ID-card utility and hence was also available for other operating systems [269].

For the ID cards produced since 2007, the validity term of the document and certificates were aligned to 5 years, therefore for these cards the updating of certificates was not used. The last MICARDO-powered ID card with a validity of 10 years expired on 2016-12-31. The remote update service, however, was already closed down on 2016-01-01, due to SK's decision to deprecate the certification of 1024-bit RSA keys.

5.2.1. Remote update protocol

The EstEID implementation on MICARDO-powered cards was preconfigured to provide a list of card management operations, which could be performed remotely after card issuance (see Section 17 in [5]). The operations were authenticated using symmetric 3DES card management keys loaded by the card manufacturer in the personalization phase. To provide the RSA key regeneration option, the cards had two pairs of RSA key slots. The key reference could be updated to specify which pair of authentication and digital signature key was currently active. In general, the certificate update process consisted of generating new RSA keys in the inactive slots, reading out the public keys, overwriting the certificate files and updating the key references to point to the newly generated keys.

Below we provide the description of the remote update protocol steps. The description is based on a non-public specification [270] shared by SK.

1. The personal ID code and the document number is read from the personal data file. The personal ID code is required to derive the correct card management keys and the card's eligibility for certificate update is checked using the document number.
2. The active key references are read from the first record of the `EEEE/EF_SE` file (FID: `MF/EEEE/0033`) to find which RSA key slots are inactive and hence can be used to store the new keys.
3. The inactive key slots are flagged for key generation by updating the appropriate records in the `EEEE/EF_KeyD` file (FID: `MF/EEEE/0013`). The `UPDATE RECORD` command is MAC-protected using the CMK2b key. The prerequisite is a successful cardholder verification using PIN1.
4. Cardholder verification using PIN1 is performed (required to execute the commands in the next step).

5. A secure messaging channel is established using the CMK2a key, and over the MAC-protected (unencrypted) secure messaging session:
   (a) The authentication key pair is generated using the `GENERATE PUBLIC KEY PAIR` command.
   (b) The public key is read from the `MF/EEEE/1000` file.
   (c) The digital signature key pair is generated using the `GENERATE PUBLIC KEY PAIR` command.
   (d) The public key is read from the `MF/EEEE/1000` file.
6. The files storing authentication (FID: `MF/EEEE/AACE`) and digital signature (FID: `MF/EEEE/DDCE`) certificates are overwritten. The `UPDATE BINARY` commands are MAC-protected using the CMK2b key. The prerequisite is a successful cardholder verification using PIN1.
7. The active key references are changed in the first record of the `EEEE/EF_SE` file. The `UPDATE RECORD` command is MAC-protected using the CMK2b key. The prerequisite is a successful cardholder verification using PIN1.

The generation of the authentication key pair is done first, because both public keys are signed using the authentication key. An interrupted certificate update process can be restarted by executing the process from the beginning. If, however, the key generation (step 5) is completed successfully, the protocol resumes with the certificate overwrite (step 6).

In the event a card generated an RSA key with a 32-bit public exponent, the key was regenerated until the randomly chosen exponent was smaller than 32 bits. This was done because Microsoft Windows 98 SE only accepted 32-bit exponents if they were encoded in 4 bytes, ignoring the ASN.1 DER encoding rules, which specify that the most significant bit is the sign bit (correct certificate encoding is further discussed in Section 6.6). This only affected the MICARDO-powered ID cards issued in 2002. For the later cards, the maximum exponent size was decreased to 3 bytes (see Section 3.1.2.1).

The same protocol was also used for updating certificates in CMB and SK customer service points. In the process the cardholder was asked to enter PIN1. In case PIN1 was unknown, the cardholder was issued a new PIN envelope [271].

5.2.2. Security analysis

In this section we provide a brief security analysis of the remote update protocol. We note that the remote update protocol described above has been built utilizing the most secure functionality provided by the card's configuration. Therefore, the security risks described below could not have been mitigated by utilizing a different functionality provided by the card.

5.2.2.1. Bringing a card to an inconsistent state

As described in Section 5.1.5, the MAC protection applied for the commands using the CMK2b key provides very weak integrity properties. In the remote update context it means that an attacker can, for example, prevent certificate overwrite or prevent the change of active key references without the backend being able to detect it. While not a security threat by itself, the flaw allows an attacker to bring the card to an inconsistent state.

5.2.2.2. Exporting the generated key

The file identifier, where the `GENERATE PUBLIC KEY PAIR` writes the private key of the generated RSA key pair, is stored in the `EEEE/EF_KeyD` file. By exploiting the flaw which allows card management operations to be performed using PIN2 (see Section 6.2), the attacker can modifying the private key file identifier in the `EEEE/EF_KeyD` file to point to a publicly readable file. After the remote update process, the attacker could export the generated RSA keys from the card, thereby breaching the legal requirement that the private key must reside on a secure signature creation device. In practice, this attack would have required intercepting the APDU commands sent to the smart card – sending a few extra APDUs before the APDUs from step 5 of the protocol have been transmitted.

5.2.2.3. Obtaining a certificate for a key generated outside the card

The flaw in the MICARDO key negotiation phase of the secure messaging protocol (see Section 5.1.4) allows the attacker to impersonate the card in step 5 of the remote update protocol. This would allow the attacker to trick the backend into believing that the public key returned has been generated by the card and hence the corresponding private key resides on the card, while the public key provided in the impersonated session actually corresponds to a private key generated by the attacker. As in the case described above, the legal requirement that requires the private key to reside on a secure signature creation device would be breached.

5.2.2.4. Disclosure

The security issues described above were discovered after the remote update functionality of the MICARDO-powered ID cards was already discontinued, therefore no action was required to handle these risks.

5.3. jTOP SLE66-powered ID cards

For the ID cards based on the jTOP SLE66 platform, the ID card remote update capability was used starting from 2016-03-18 to update certificates containing an invalid ASN.1 DER encoding (see Section 6.6 for more details). The updating of jTOP SLE66-powered ID cards was terminated on 2017-07-01, due to the eIDAS technical requirements (see Section 6.6.2.2).

The remote update process used the card management key CMK2b to overwrite the authentication and digital signature certificates. The RSA keys in the process were not regenerated, therefore SK had to update their CP (Section 4.2.5 in [272]) which previously required certificate replacement based on a newly generated key pair.

Due to the weak security properties of the MAC-protection using the CMK2b key (Section 5.1.5), the attacker could interfere with the process, e.g., by preventing some parts of the certificate from being overwritten or writing the authentication certificate in the digital signature certificate file and vice versa. This, however, does not give any advantage to the attacker, other than turning certificates on the card into an inconsistent state without the backend being able to detect it.

Figure 39 shows the sequence diagram of the remote certificate update process. The diagram has been drawn from the APDU traces collected in the remote update process performed on 2016-06-29. The activities colored in blue are user GUI activities.

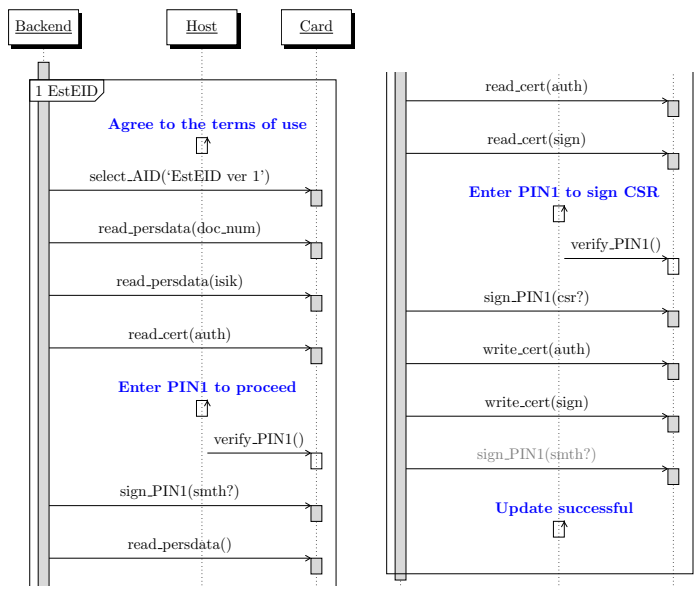


Figure 39: jTOP SLE66: Sequence diagram of the remote certificate update process

5.4. jTOP SLE78-powered ID cards

The remote update solution for the jTOP SLE78-powered ID cards was introduced in spring 2016. It was first used starting from 2016-06-22 [273] to update certificates containing an invalid ASN.1 DER encoding (see Section 6.6). For an unknown reason, the EstEID v3.5 applet deployed on the jTOP SLE78-powered ID cards did not support certificate updating using the card

management operations. Therefore, a simple certificate overwrite (as it was done for the jTOP SLE66-powered ID cards) was not possible and hence the whole EstEID applet had to be reinstalled. Later, in fall 2017, the readiness of such a remote applet updating solution turned out to play a crucial role in solving Infineon's RSA key generation flaw (see Section 6.7), enabling the cardholders to remotely update their cards from 2017-10-25 to 2018-03-31. The same solution was used, starting from 2018-11-01 to 2019-04-30, to extend the validity period for certificates on Digi-IDs and e-resident's Digi-IDs from 3 to 5 years (see Section 3.4.4).

In 2016, the author of this work participated in the security review of the remote update protocol. Therefore some protocol peculiarities documented below may have been introduced as a result of the feedback provided in the protocol design process.

5.4.1. Remote applet update protocol

The sequence diagram of the remote applet update process is shown in Figure 40. The diagram has been drawn from the APDU traces of the remote update processes performed on 2016-06-29 and 2017-10-26. Since the APDU header is not encrypted, the protocol can be reverse-engineered to a great extent. The protocol steps are split into 14 protocol sessions based on the smart card applet in which the activities take place. The activities starting with the "secure_" prefix are performed over an EstEID secure messaging session. The activities colored in blue are user GUI activities. The activities colored in brown are extra APDUs introduced in the protocol in 2017. The activities colored in gray are smart card requests for which we were unable to determine a purpose.

5.4.1.1. Applet management

To provide JavaCard applet management functionality, the jTOP SLE78 platform implements GlobalPlatform (GP) Card Specification 2.2.1 [202]. The applet management operations such as applet installation and deletion are authenticated using the GP Issuer Security Domain (ISD) keys which are loaded on the card by the card platform manufacturer and handed over to the ID card manufacturer together with the cards. The card management APDUs are sent over a secure messaging session established using the GP Secure Channel Protocol (SCP).

5.4.1.2. Handling of new PIN codes

The PIN codes are stored and handled by the EstEID applet, therefore the PIN codes of the old EstEID applet can not be retained. The new PIN codes were encrypted using the authentication key of the old EstEID applet, and after PIN1 verification they were decrypted and shown to the user on the screen (session 6). To verify whether the user had written down the new codes, the entry of the new PIN1 was requested in the next step (session 7).

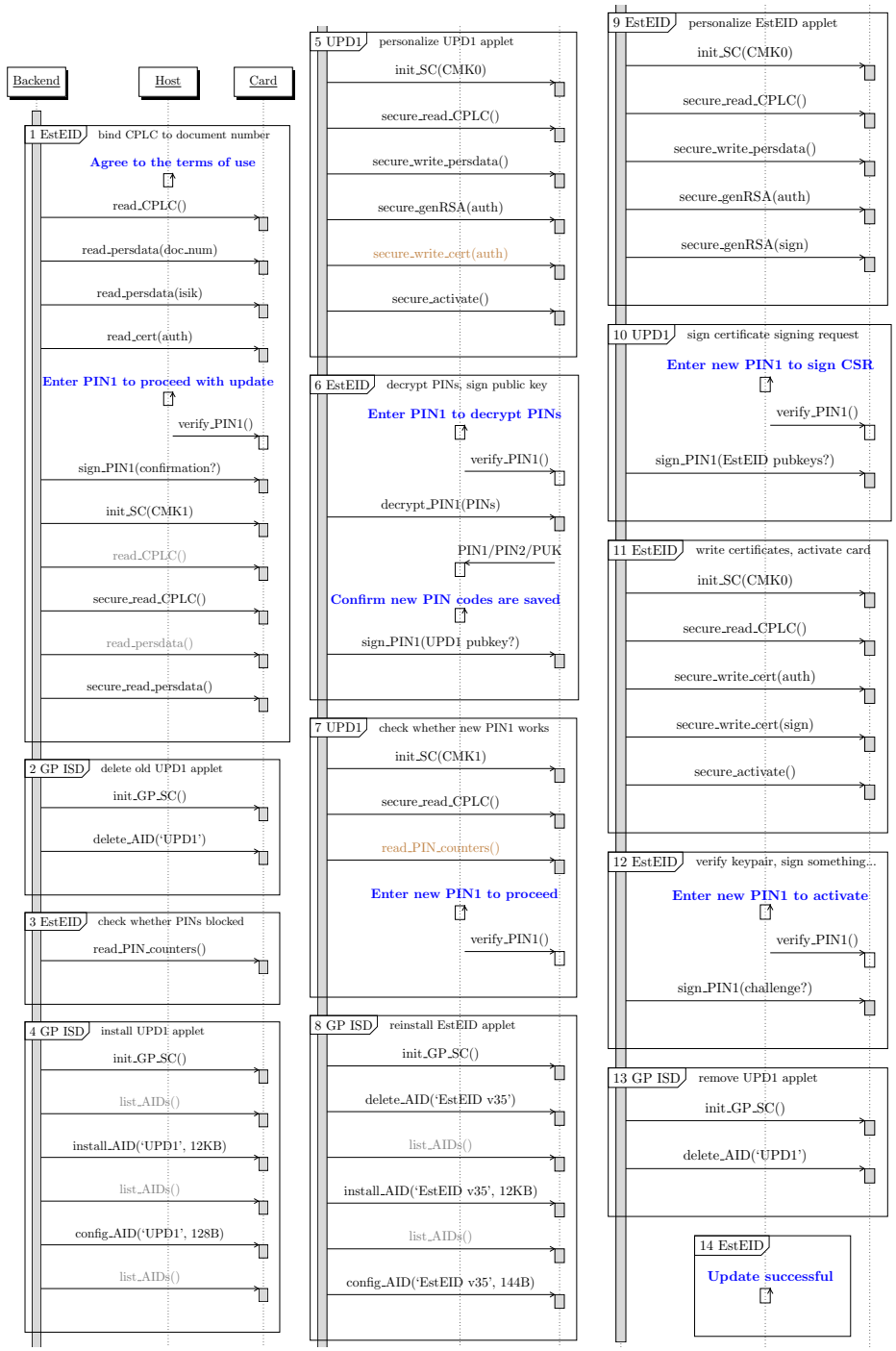


Figure 40: jTOP SLE78: Sequence diagram of the remote applet update process

5.4.1.3. The purpose of the temporary applet

Since several applets can reside on a single smart card, only one applet can be automatically selected by the card without the need for the terminal to send an explicit `SELECT` command. At that time, the systems communicating with the ID card were built under the assumption that the EstEID applet is the implicitly selected applet. To not break this compatibility, the implicit selection of the EstEID applet had to be maintained. The Default Selected privilege can only be set at the applet installation phase and cannot be set if some other applet already has the privilege. Therefore, before installing the new EstEID applet, the old applet had to be removed (session 8). To avoid the situation where the ID card ended up without any applet residing on the card, temporary update applet UPD1 was installed on the card. The applet was configured with the PIN codes of the new EstEID applet (session 4). A temporary authentication key was generated for the applet and (only in 2017) an authentication certificate issued by "Trueb Baltic" that was valid for 10 days was loaded on the card (session 5). The applet was removed at the end of a successful update process (session 13).

The use of the temporary applet ensures that even if the update process is interrupted, the ID card always contains an applet-level RSA key bound to the cardholder's identity. The temporary RSA public key of the UPD1 applet is signed by the cardholder's old EstEID applet's RSA key, and the new EstEID applet's public RSA key is signed by the UPD1 applet's temporary RSA key. This guarantees that the attacker, who has obtained the GP ISD key but does not have access to the cardholder's authentication key, cannot impersonate the cardholder in the remote applet update process.

5.4.2. Security of GlobalPlatform SCP02 protocol

The jTOP SLE78 platform implements the GlobalPlatform Secure Channel Protocol SCP02 implementation option '55' (SCP02_55) (see Section E in [202]). The purpose of the protocol is to provide encryption and authentication for the C-APDUs sent to the card (the R-APDUs are not protected). In every SCP session, the APDU is encrypted and MACed using unique session keys *enc* and *cmac*. The session keys are not negotiated by the card and host, but are derived from the sequence number *seq* and a 16-byte symmetric GP ISD key shared between the card and host. The value of *seq* is maintained by the card and incremented by 1 after every successfully completed `EXTERNAL AUTHENTICATE` command.

The protocol is depicted in Figure 41. The number in square brackets represents the byte length of the value. The square brackets with a colon represent the byte slicing operator.

The protocol works as follows:

1. Using the `INITIALIZE UPDATE` command the host sends an 8-byte random challenge r_h to the card.

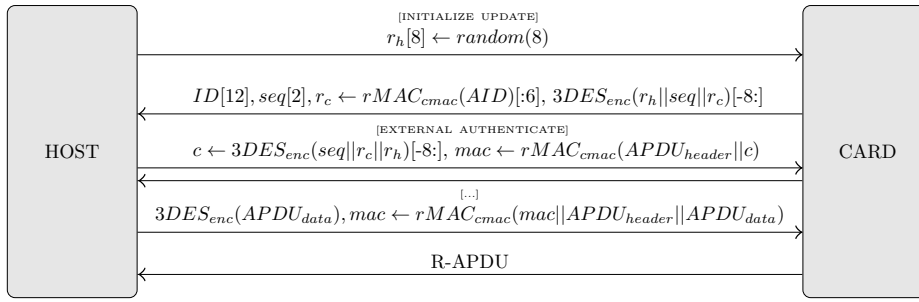


Figure 41: GlobalPlatform SCP02 implementation option '55'

2. The card responds with a 12-byte card identifier, a 2-byte sequence number and a 6-byte challenge r_c (note that the card's challenge is not random, but depends on the *cmac* of the particular session). In addition, the card returns the last ciphertext block of the sequence number *seq* and the challenges encrypted using padded 3DES in CBC mode with session key *enc*. This proves to the host that the card knows the session key *enc* as the card could not have otherwise provided the encryption of r_h.
3. Using padded 3DES in CBC mode with session key *enc*, the host encrypts the sequence number *seq* and the challenges that have been reordered. Using the `EXTERNAL AUTHENTICATE` command, the host sends the last ciphertext block to the card. The C-APDU is MAC-protected using the retail MAC in CBC mode with session key *cmac*. This proves to the card that the host knows the session keys *enc* and *cmac* as the host could not have otherwise provided a modified encryption of the challenges with a correct MAC.
4. All further C-APDUs sent to the card are encrypted using padded 3DES in CBC mode with an IV set to zeros, and are MACed using the retail MAC in CBC mode over the plaintext version of the APDU. To prevent replay attacks, the previous MAC value is encrypted and used as an IV for the next MAC calculation.

5.4.2.1. Untrustworthy R-APDUs

The failure of the GP SCP02_55 protocol to at least provide MAC protection for R-APDUs returned by the card means that the backend cannot trust the data returned by the card and it is not even able to verify whether the C-APDUs were at all executed by the card.

The C-APDU protection provided by the GP SCP02_55 protocol, however, can be used by the backend to securely install the EstEID applet and set the applet-specific card management keys (session 4 and 8). The actual success of the installation can then be verified by establishing an EstEID-level secure messaging session using the EstEID card management keys (session 5 and 9). The remote applet update process, however, cannot verify whether, for example, the deletion of the temporary update applet (session 13) was actually performed.

5.4.2.2. The same GP ISD key shared among ID cards

It turned out that the GP ISD keys set by the manufacturer are not card-specific, but are shared among different batches of ID cards. We collected the responses to the `INITIALIZE UPDATE` command from *identity cards* issued on 2015-02-20, 2015-07-02 and 2015-08-31. All three cards responded with the *seq* number 1 and the same r_c value `0xDC8A9A723FE9` showing that the same GP ISD key was loaded on these cards.

To check if the ID card manufacturer had also failed to diversify GP ISD keys for the jTOP SLE66-powered ID cards, we collected the responses to the `INITIALIZE UPDATE` command from a jTOP SLE66-powered *identity card* issued on 2013-01-08 and jTOP SLE66-powered *residence permit card* issued on 2013-09-16. Both cards responded with the *seq* number 2, the same r_c value `0x87C4530FD38B` and the card's cryptogram `0xBFD3C912D020DC07`, showing that the same practice was also employed for the jTOP SLE66-powered ID cards.

The use of the same GP ISD key on several cards is a bad practice in general. If the key is compromised, all the cards sharing the same GP ISD key can be attacked. The knowledge of the GP ISD key does not allow an attacker to directly obtain secrets stored by the EstEID applet, because the GP card specification forbids the retrieval of an applet and its associated data. The loading of additional (potentially malicious) applets, however, opens the card to attacks against the on-card bytecode verifier. If the attack is successful, an attacker would be allowed to bypass the applet isolation enforced by the JavaCard virtual machine [274]. The knowledge of the GP ISD key would also allow the attacker who has captured C-APDUs exchanged between the backend and the card to decrypt the corresponding PIN codes and CMK keys configured for the applet.

In the context of the remote applet update, the use of non-unique GP ISD keys means that the update process cannot ensure that the applet, PIN codes and CMK keys are installed in the correct ID card. As a counter-measure, the first step after a secure messaging session with the applet has been established is to verify (via `secure_read_CPLC()` which returns the card's serial number) whether the backend is communicating with the EstEID applet installed on the correct ID card.

Furthermore, since the GP SCP02 secure messaging protocol does not involve any randomness from the card, the C-APDUs captured can be replayed against any card sharing the same GP ISD key as long as the *seq* number of the target card matches the *seq* of the C-APDUs to be replayed. This, for example, allows the attacker who has captured C-APDUs from some remote update session to delete the EstEID applet or install the EstEID applet with the corresponding PIN codes and CMK keys on any ID card sharing the same GP ISD key.

The response to the `INITIALIZE UPDATE` command obtained from a jTOP SLE78-powered *identity card* issued on 2018-09-04 shows that the ID card manufacturer has not changed the practice of using the same GP ISD key even long after being informed of the negative security implications it introduced.

5.4.2.3. Encryption with IV set to zeros

The APDU data in the GP SCP02 protocol C-APDUs is encrypted using an IV of all zeros. This means that the encrypted APDU data sent over the same SCP session which starts with a common plaintext prefix will share the same ciphertext prefix. Sabt et al. in [275] point out this GP SCP02 protocol deficiency.

Due to the ID card manufacturer's practice of using the same GP ISD key among different ID cards, an attacker sniffing communication between the backend and the ID cards will be able to cross reference ciphertexts sent to different ID cards. Whether the attacker can deduce something useful from the similarity of ciphertext blocks depends on how the data in the APDU is arranged.

We informed RIA of this weakness on 2016-02-12, recommending that they ensured that the plaintext data encrypted for the same GP session starts with a unique 8-byte block of data.

5.4.2.4. Padding oracle attack

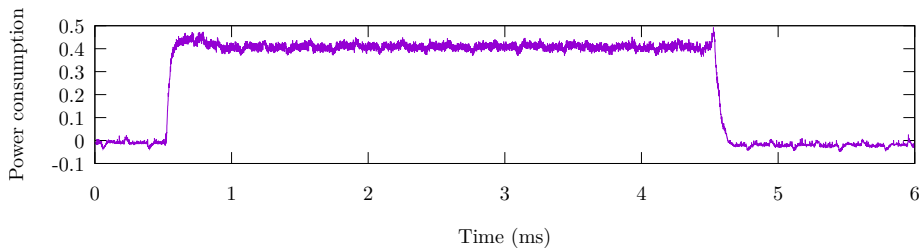
The GP SCP02 protocol follows the so-called Encrypt-and-MAC approach, which means that the card first has to decrypt potentially modified ciphertext, before its integrity using the MAC can be verified. This opens the card to the padding oracle attack [276], allowing an attacker to decrypt ciphertexts, unless special implementation-level counter-measures are implemented by the platform.

To exploit the flaw, the attacker has to determine the reason why the card rejects a modified C-APDU (incorrect decryption padding vs correct padding but incorrect MAC). Since the smart card returns the same status word `0x6982` (security status not satisfied) in both cases, the attacker has to deduce whether the decryption was successful, and hence the MAC verification was performed, using some side-channel information, such as execution time or power consumption of the card.

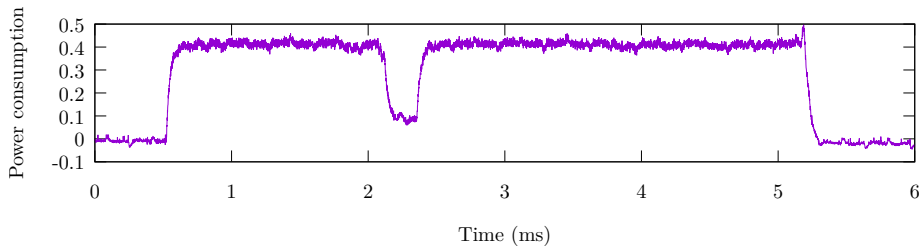
To measure the power consumption of the card, we built a power consumption measurement setup (see Figure 42) by adding a 50 ohm resistor to the terminal's ground wire and measuring the voltage drop across the resistor using the Hantek DSO-5200A USB oscilloscope.

Figure 42: Smart card power consumption analysis setup

We collected power traces of the jTOP SLE78 card processing malformed C-APDUs. From the traces (see Figure 43) we see that the card's power consumption allows us to clearly distinguish between the cases, therefore the GP SCP02 implementation on the jTOP SLE78 platform is vulnerable to the padding oracle attack. From the traces we also see that the timing side channel could be used to distinguish between the cases. Avoine et al. in [277] have shown that padding oracle attacks using timing as a side-channel is a common problem among GP SCP02 protocol implementations on smart cards produced by different card manufacturers.



(a) Status word `0x6982` caused by decryption failure



(b) Status word `0x6982` caused by MAC verificaton failure

Figure 43: Power trace of a jTOP SLE78 card processing malformed GP SCP C-APDUs

In the context of the remote applet update process, the exploitation of the padding oracle attack would allow the attacker to decrypt C-APDUs, thereby gaining access to sensitive data such as the JavaCard bytecode of the EstEID applet, PIN codes and CMK keys.

The exploitation of the padding oracle attack requires the remote update backend to send the encryption of the same plaintext repeatedly over many GP SCP02 sessions, with the attacker modifying the ciphertext and observing the card's behavior in handling the malformed ciphertext. In each GP SCP02 session the attacker can verify only one guess of the plaintext byte. The attacker cannot replay a single session against a single ID card over and over, because the card increments the session sequence number *seq* after every successfully established session. Establishing thousands of unsuccessful remote update sessions would most likely be noticed by the backend. However, due to the manufacturer's

practice of using the same GP ISD key on several ID cards and the GP SCP02 protocol's failure to involve randomness from the card, the attacker can replay the captured C-APDUs against ID cards sharing the same GP ISD key, without the need to involve the backend. This would, however, require access to a large number of different ID cards, unless the attacker had collected a large number of sessions that could be replayed and needed to decrypt only a few bytes of the ciphertext.

We informed RIA of the flaw on 2016-10-12, recommending that they increase the cost of the attack by limiting the number of ciphertexts produced using different *seq* values.

5.4.3. Obtaining new PIN codes using PIN1

The decision to require only PIN1 to authenticate the remote update process led to controversy. Contrary to other ID card remote update scenarios, in this case at the end of the remote applet update process a new PIN envelope is shown to the user. This means that an attacker who has access to a cardholder's ID card and 4-digit PIN1 could perform the remote applet update, thereby obtaining the 5-digit PIN2 and would be able to use the PIN2 functionality of the card (i.e., create digital signatures).

As a type of counter-measure, RIA introduced an SMS notification for the cardholders who had registered their phone number in the state portal `eesti.ee`. The SMS text (in Estonian) was as follows: "Dear [cardholder's name]! Your certificates are updated. If they were not updated by you, call number 1777 or (+372) 677337."

This case is another example showing that the ID card ecosystem in practice fails to isolate PIN1 from PIN2, in some cases allowing the abuse of the PIN1 privilege to escalate to PIN2. In our opinion, adding an extra GUI activity for PIN2 entry would not have considerably affected the user experience, since the update process already demands 5 entries of PIN1.

5.5. IDEMIA-powered ID cards

For the IDEMIA-powered ID cards, the remote ID card update functionality has not yet been used. However, in their 2020 yearbook, RIA announced plans to develop a solution to remotely update the software and certificates on the chip of the IDEMIA-powered ID cards as one day, it may become necessary (page 10 in [237]).

6. SECURITY INCIDENTS AND OTHER ISSUES

In this chapter we analyze a list of security incidents and similar issues that the Estonian ID card has experienced over the years. We note that some security issues related to the Estonian ID card were already mentioned in the previous chapters of this work when discussing the particular topic. Most of the issues covered in this chapter have been previously publicly disclosed, however, several issues were found by us and reported during the course of this research.

While the occurrence of an incident cannot be completely avoided, an important aspect is how the involved parties respond and resolve the incidents once they occur. Therefore, we have made a significant effort to try to document the events that took place, and, when possible, provide an analysis describing the root cause and impact of the incident, the applied mitigation measures and how effective and timely they were, how the risk was communicated to the affected parties and the public, and what lessons were learned from the incident and what improvements were made to the ID card ecosystem.

6.1. Authentication key operations using PIN2

The ID cards powered by the MICARDO, MULTOS and jTOP SLE66 platforms support the not so well-known passphrase authentication feature, which can be used to perform cardholder verification using a passphrase instead of a PIN code. The passphrase authentication is configured in such a way that it creates a design flaw, as it allows the usage of PIN2 to perform private key operations with the authentication key (rather than only with the digital signature key). This functionality has been documented in the EstEID specification, but has been overlooked by the community and hence has never been called into question. In the subsection below, we describe the passphrase authentication process and the related security issues in detail.

6.1.1. Passphrase authentication

In addition to cardholder verification using PIN1 and PIN2 codes, the EstEID specification provides cardholder verification using so-called passphrases (see Section 17 in [9]). The passphrase authentication feature is implemented as follows. From the user's submitted passphrase, a symmetric 3DES key is derived, which is then used to establish a mutually-authenticated secure channel with the card (see Section 5.1.1 for the description of the secure messaging protocol). Since the operations with the cardholder's asymmetric keys are performed over a mutually-authenticated channel, additional cardholder verification using a PIN code is not required.

Passphrase authentication introduces two 3DES keys: 3DESKey1 to perform operations with the authentication key and 3DESKey2 to perform operations with

the digital signature key. Each key is assigned a retry counter with an initial value of 255, which is decreased whenever an incorrect 3DES key is used to establish a secure channel. After a successful passphrase authentication the retry counter is not reset. Once the counter reaches zero, the respective 3DES key cannot be used and there are no means to reset the counter.

The initial value for the 3DES passphrase keys is set to all zeros and therefore (because of the 3DES key parity error) cannot be used to initiate a secure channel. The cardholder can set both keys by updating the `EF_Key_DES` (FID: `MF/0010`) file on the smart card file system. The prerequisite for updating the `EF_Key_DES` file is a successful cardholder verification using PIN2. Hence, the knowledge of PIN2 also allows a cardholder to set the 3DESKey1 which in turn can be used to perform cryptographic operations with the PIN1-protected authentication key.

It is interesting to note that the preliminary version of the EstEID specification drafted in June 2001 required PUK verification to set both 3DES keys (see Section 6.2.3.3 in [3]). It is not clear why the PUK code was replaced with PIN2 in the final specification (see Section 4.2.2.3 in [4]). Requiring the PUK code to set the 3DES keys would not have introduced the security issue, because the PUK code is intended as a super-PIN that can be used to reset both PIN codes.

Since the 3DES keys are not reset after the PIN codes are changed, the passphrase authentication feature can be used as a type of backdoor to maintain persistent access to the private key functionality even after the cardholder has changed the PINs. The passphrase authentication feature can also be abused to bypass the security restrictions enforced by the smart card readers with a PIN pad (Section 2.14.1). Once the cardholder has entered PIN2 on the PIN pad, malware can set the 3DES keys, thereby having full access to the cryptographic functionality of the ID card whenever the card is in the reader.

The only security advantage provided by the passphrase authentication is in the case where an attacker is sniffing the communication channel between the terminal and the card – the attacker will not be able to see the plaintext and modify the data. Offline brute-force attacks to recover the passphrase, however, will still be possible.

The passphrase authentication feature that was first introduced in the MICARDO platform has also been reimplemented in the MULTOS and jTOP SLE66 ID card platforms, with the only difference being that the initial value of the retry counter is 3 and it is reset to the initial value after each successful passphrase authentication or successful change of the passphrase. The passphrase authentication feature has never been implemented in the client-side software and has not been implemented in the EstEID implementation on the jTOP SLE78 ID card platform. We note that SK's CPS [129] and CP [130] documentation (which is subject to audits) have never mentioned this passphrase authentication feature as a method of private key activation.

6.1.2. Disclosure

We informed RIA of this design flaw on 2019-02-27. As far as we know, this has not resulted in any type of action. We note that while all MICARDO and MULTOS-powered ID cards have already expired, the last jTOP SLE66-powered ID card only expired at the end of 2019.

6.2. Card management operations using PIN2

While studying the configuration of MICARDO-powered ID cards, we found that all MICARDO-powered ID cards issued from 2002 to 2011[1] have a configuration flaw which can be used to perform all card management operations if the PIN2 code of the ID card is known.

6.2.1. Cause of the flaw

The EstEID specification uses several symmetric 3DES keys. In the MICARDO operating system the secret keys are stored in the file `EF_Key_DES` (FID: `MF/0010`). The first two records store 3DESKey1 and 3DESKey2, which are used to enable cardholder verification using the passphrase authentication feature (see Section 6.1.1). The other records store card management keys (CMK1, CMK2a, CMK2b and CMK3) loaded by the card manufacturer in the ID card personalization process (see Section 19 in [9]).

The descriptions of the keys are stored in the read-only file `EF_KeyD` (FID: `MF/0013`). Each key in the MICARDO operating system is identified by a 2-byte key reference. In order to map the keys defined in `EF_KeyD` to the secret key values stored in `EF_Key_DES`, each secret key in the `EF_Key_DES` file has the corresponding key reference value prepended. The access rules of the `EF_Key_DES` file allow the cardholder to update the first two records so that the the passphrases can be set. The prerequisite for updating the `EF_Key_DES` file is a successful cardholder verification using PIN2.

The flaw lies in the fact that when updating the records the cardholder can specify an arbitrary key reference in the first two bytes of the record. When the cardholder specifies the reference of a card management key for the cardholder's 3DES key, it will be used by the MICARDO OS to authenticate the corresponding card management operation. This is because the MICARDO OS uses the first record with the matching key reference when looking for a secret key in the `EF_Key_DES` file. Section 4.7.1 of the MICARDO User Manual [185] states that the MICARDO OS does not detect the existence of entries with duplicate key references, the requirement for uniqueness being left up to the entity who writes the data.

[1] We verified that the flaw was present on a card issued on 2002-01-17 as well as a card issued on 2010-09-01.

The flaw could have been avoided if the cardholder's 3DES keys were stored in the last two records of the EF_Key_DES file – in the order they are defined in the EF_KeyD file. This way, the card management keys set by the manufacturer would always take precedence over potentially colliding key references set by the cardholder. Taking into account the above, we classify this flaw as a misconfiguration of the MICARDO OS.

The EstEID applets of the MULTOS and jTOP SLE66-powered ID cards are not affected, because their applets ignore the key reference specified when updating the EF_Key_DES file, instead using the record in which the key was written, to look up the corresponding CMK key.

6.2.2. Impact

The ability to perform all card management operations can be used to reset PIN and PUK codes, generate new RSA key pairs, overwrite certificates and create additional application structures on the card. A less obvious abuse scenario that can be done in the context of the EstEID applet is the ability to update the EEEE/EF_KeyD file (FID: MF/EEEE/0013), changing the RSA private key FID reference to point to a readable and writable file, hence allowing the generation of exportable keys and the use of imported keys. The access rules set in the personalization process, however, prevent the RSA private keys that have been generated in the ID card personalization process from being exported. The flaw could have been exploited in the ID card remote update process, for example, by forcing the generated private key to be saved in a publicly readable file, thereby allowing private key export (see Section 5.2.2 for more details).

6.2.3. Disclosure

We informed RIA of the flaw on 2019-02-27. Since, at that time, all MICARDO-powered ID cards had already been expired, RIA decided that there was no need to take action.

6.3. Padding oracle attack in the decryption functionality

In April 2012, Bardou et al. published the research paper "Efficient Padding Oracle Attacks on Cryptographic Hardware" [256]. In this paper the researchers presented an optimized version of the Bleichenbacher's padding oracle attack against the RSA PKCS#1 v1.5 encryption scheme [278] and tested the attack against commonly available cryptographic hardware that impemented the PKCS#1 v1.5 standard. Also among the devices tested was the Estonian ID card (jTOP SLE66-powered ID card test card), which naturally was found to be vulnerable, as it supports the decryption functionality and implemented the RSA PKCS#1 v1.5 standard.

Essentially, the researchers found that by sending tens of thousands of specially crafted ciphertexts to the card (it takes tens of hours due to the performance limitations of a smart card) and by observing the smart card responses (whether the decryption was successful, i.e., the decrypted plaintext contained a valid PKCS#1 v1.5 padding), it is possible to calculate the decryption of some ciphertext or forge a signature on arbitrary data.

A potential exploitation scenario could be the case where a machine in which the ID card is used accepts RSA ciphertexts from some untrusted source, automatically sends the ciphertexts to the ID card for decryption and allows an attacker to learn (by error message or some side channel) whether the decryption was successful or not. Since in practice the decryption function of the ID card is not used in this manner, the attacker is unable to gain access to such an oracle and hence this PKCS#1 v1.5 vulnerability in the context of the standard use of the Estonian ID card is not exploitable.

6.3.1. Incident response

On 2011-11-11, the researchers informed SK about their findings, rating the severity of the vulnerability as "moderate", but added that exploitation in practice may be difficult [279].

SK reported the flaw to CERT-EE and later gave the researchers the response that a signature forged using such an attack would not be usable in practice, as the authentication key is used mainly to authenticate to TLS servers. However, the particular attack was too slow to forge a signature before the attacker's TLS handshaking process with the TLS server timed out (see Appendix C in [256]).

We note that the argument used in the CERT-EE response is not sound for two reasons. Firstly, we found that in practice, the TLS server implementations of 63% of the Estonian service providers allow the TLS handshaking process to be kept open for days (see Section F in [15]). Secondly, the padding oracle attack also concerns the forging of digital signatures, because the MICARDO, MULTOS and jTOP SLE66-powered ID cards also enabled decryption function with the digital signature key (see Section 2.9).

6.3.2. Reflections in the media

On 2012-06-08, the news portal Delfi.ee published an article about the findings of the researchers together with a comment from Cybernetica AS and PPA [280]. Cybernetica's specialist noted that the findings had been analyzed and there was no reason to panic. PPA stated that the attack was theoretical and could not be used to commit a crime, providing an analogy that this was an attack where in order to break the door, access to the key is first needed.

On 2012-06-25, Ars Technica published an article titled "Scientists penetrate hardened security devices in under 15 minutes" [281], mentioning that the Estonian ID card was among the affected devices. As a response to this article,

RIA on 2012-06-27 published PPA's press release [282] (English version on 2012-07-03 [283]) refuting the claims made in the article.

On 2012-09-13, two of the authors of the research paper visited Estonia and gave a guest lecture on the topic at the University of Tartu [279]. The researchers refused to give their assessment on the practical exploitability of this attack in the context of the Estonian ID card. A government official later published a newspaper article accusing the researchers of presenting sensational half-truths for cheap publicity [284]. The media coverage directed at this theoretical issue actually might have been an attempt to divert attention from the very practical security risk affecting the Estonian ID card of which the officials at that time were well aware (see the following section).

6.4. Security flaw in the ID cards issued in 2011

In September 2012, PPA announced an invitation for cardholders whose ID cards were issued in 2011 to visit PPA customer service points to renew the electronic component of their ID cards [285–287].

According to RIA, "During a routine ID card analysis process we discovered that one of the electronic security measures of the ID card needs to be renewed. ID card users have no reason to be concerned. The card is secure and all transactions made with the card are fully reliable. The assessment points to only one of many security measures – all other security components are still of high quality." [285]

The wording of the press release suggested that some purely theoretical risk was being mitigated and the renewal was just a formality. As a result, this ID card recall received very little public interest [288]. Only later, in 2017, the authorities partly acknowledged (see Section 6.4.2) that this was how one of the most serious security flaws in the Estonian ID card history was to be mitigated.

6.4.1. Incident response

According to two sources, the flaw was discovered by Finnish penetration testers who were contracted by RIA to perform a code audit of the EstEID v3.0 JavaCard applet shipped on the jTOP SLE66-powered ID cards. Most likely the flaw was discovered at the end of 2011, as the ID cards issued from 2012 were not vulnerable. According to PPA, in total 120 000 cards issued in 2011 were affected [285]. This covered all jTOP SLE66-powered *identity cards* and *residence permit cards*. The *digital identity cards* issued in 2011 were not affected, because they were powered by the MULTOS platform.

In the first half of 2012, the ID card manufacturer developed a solution that allowed the vulnerable ID cards to be patched in PPA customer service points. In the renewal process, the old EstEID JavaCard applet was removed and a new applet with new RSA keys and PIN codes was installed on the card [289].

The renewal process was opened from 2012-09-10, the same day the PPA press release [285] was issued. From the certificate data we learned that the first cards (issued to SK and PPA employees) were already renewed on 2012-07-12, apparently to test the ID card renewal functionality.

The entire procedure was reported to take between 5–10 minutes [289]. The customer service points had a separate priority queue for cardholders coming to renew their 2011 ID cards [290].

The invitations to renew the card were sent to the affected cardholders via email and regular mail. The invitation stated that the update was needed to ensure higher security and reliability of the document [290]. The first invitations were sent to 40 000 cardholders who had used the card electronically at least once [285]. For the persons registered abroad a new ID card was issued [289].

In the initial press release [285] the cardholders were informed that the certificates of the non-renewed ID cards would be revoked in March 2013. For unknown reasons the revocation date was later moved to July 2013 [289]. From the CRL data we found that the actual revocation took place from 2013-07-24 until 2013-07-28. In total, certificates of 78 760 ID cards were revoked. According to SK [291], the certificates were revoked based on an official letter from state authorities and the legal basis for revocation was the Digital Signatures Act [63], clause 14 (1) 2): "an opportunity for using the private key corresponding to the public key set out in the certificate without the consent of the certificate holder".

The possibility to renew the affected ID cards was also provided to cardholders after the certificates were revoked. The timeline of ID card updating activity is shown in Figure 44.

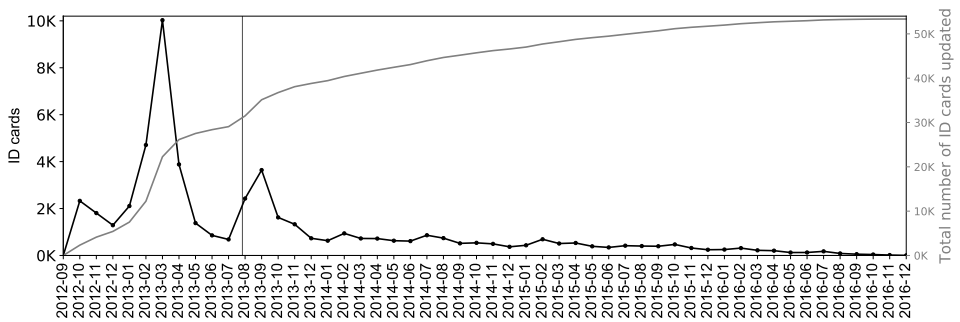


Figure 44: Updating activity of vulnerable 2011 ID cards (source: certificate data)

We noticed that the certificates issued in the PPA renewal process can be distinguished from the certificates on initially issued ID cards by the value of RSA public exponent e. For initially issued ID cards, the 2048-bit RSA key was generated with a 4-byte random public exponent. However, the RSA keys generated in the PPA renewal process were generated with the public exponent set to the value 65537. As it later turned out (see Section 6.8), the difference was due to these keys issued in the PPA renewal process being generated outside the ID card.

6.4.2. Decision to hide the nature and the true risk of the flaw

It was already clear in 2012 that the assuring statements of the authorities contradicted with the actions taken (see slide 17 in [292]). The costly ID card recall and the revocation of the affected certificates sharply differed from the usual response of the authorities, where the severity of even practically exploitable security flaws is downplayed. It was clear that the flaw was actually very serious and trivially exploitable.

Only in 2017, after the ROCA flaw broke out (Section 6.7), was it disclosed in the media that the flaw in the 2011 ID cards was exploitable by having access to the card [288]. Based on this fact, we are quite certain that the flaw manifested in the possibility of using the ID card without knowledge of the PIN codes. The only alternative is that the flaw could have been used to extract the private key. This, however, would require the applet to contain JavaCard private key export code, but there is no legitimate reason for the applet to contain such code.

It is not hard to see why the authorities in 2011 and even now[2] lack the courage to disclose the details about the flaw. If the nature and the true risk of the flaw became public, some political parties would immediately question the trust of i-voting and the results of parliamentary elections that were held in the beginning of 2011[3]. It would also be evident that the affected ID cards did not satisfy the legal requirements for secure-signature-creation devices (see Section 2.10.1), and would also raise a fair but embarrassing question of why the EstEID JavaCard applet was put into production without it being audited.

The authorities could argue that the nature of the flaw was not disclosed to prevent the reverse-engineering of the flaw, thereby limiting the number of people who knew how to exploit it. The result, however, was that the affected cardholders could not evaluate the risk and implement local mitigation measures. Furthermore, since the authorities were not honest about the risk, the affected cardholders did not rush to renew their ID cards.

It is not known on which level the decision to hide the nature and the true risk of the flaw was made. According to [288], at least to some extent the prime minister was informed of the flaw. According to the Minister of Finance at that time, the topic had been discussed among the members of the government, and it was concluded that the risk was unlikely to occur and there was no need to send confusing messages [294].

The opportunity for the state and the public to learn from this case was missed. The manner in which this flaw was mitigated created distrust, which fueled our research, in hopes of finding the security issues the authorities and the ID card manufacturer were likely trying to hide.

[2] According to PPA, information about the flaw is protected under the termless confidentiality clauses of the ID card manufacturing contract [293].

[3] The i-voting took place from February 24 to March 2, 2011. At least 12 000 vulnerable ID cards were issued by that time.

6.5. Certificates with incorrect `@eesti.ee` email addresses

In September 2015, SK and PPA announced that in June and July 2015, due to a software error, the authentication certificates for 4 120 ID cards had been issued with the wrong `@eesti.ee` email addresses [295]. In a nutshell, the process of assigning email addresses lost track of already occupied addresses. Therefore, certificates were issued with an email address that matched the email address of the namesake. This led to a situation where two (or more) namesakes had the same email address specified in their certificate and emails sent to that address were forwarded to both namesakes.

6.5.1. Mitigation

After the flaw was discovered, the colliding `@eesti.ee` addresses were deassigned from the persons to whom they had been wrongly assigned. The authorities, however, decided not to revoke the certificates containing the wrong email addresses, stating that "The security or use of the document is not affected in any way. To solve the problem, the renewal of the certificate in the service is sufficient." [295]. Hence, the security risk that emails sent to the address specified in a valid certificate would reach the wrong person, was not mitigated.

Starting from 2015-09-14, the affected cardholders were invited to PPA customer service points to renew the certificates of their jTOP SLE78-powered ID cards. In the certificate update process the entire EstEID applet was reinstalled and a new PIN envelope was issued to the cardholder. The updating service was offered for the next 6 months. [295]

The flaw not only affected the initially issued jTOP SLE78-powered ID cards, but also the Mobile-ID certificates and certificate renewals for the MICARDO and jTOP SLE66-powered ID cards. We were unable to obtain any information from SK on how the problem was resolved for these cases and we were also unbable to obtain details of the issue, as according to SK, the details were confidential and they did not have the right to share all the information.

In the certificate dataset we observed that the email address collisions had occurred before the 2015 incident and had also occurred relatively recently. The latest collision we observed was in a ertificate issued on 2018-04-20, where the email address `siim.jarve.2@eesti.ee` matched the address specified in another person's (namesake's) certificate issued on 2018-03-24.

6.5.2. Other issues

In the certificate dataset we noticed that throughout the years there had been several issues with the correct handling of `@eesti.ee` email addresses. As far as we know, none of the issues described below have resulted in a decision to revoke or replace the affected certificates.

6.5.2.1. Certificates lacking an email address

In the certificate data we found that from 2010-10-04 to 2011-07-09 a total of 4 878 *digital identity card* and Mobile-ID authentication certificates were issued with the email address field set to text value "`none`". Most likely this was caused due to an error in the certificate profile used to issue *digital identity card* and Mobile-ID authentication certificates.

6.5.2.2. The email address in digital signature certificates

For a year, from 2011-07-10 to 2012-07-05, contrary to the SK certificate profile specification [197], the cardholder's `@eesti.ee` email address was also included in the digital signature certificates of all types of identity documents.

6.5.2.3. Email address change without reason

We found 512 persons whose number extension for their `@eesti.ee` email had changed over the years without apparent reason (e.g., `name.surname@` turning into `name.surname.1@` or vice versa). These 512 persons do not include the persons who have namesakes and hence whose address extension may have changed due to the abovementioned collisions.

For some persons the allocation of a new extension occurred because the person changed their name temporarily and then reverted to their original name, but the original email address was not reassigned. We selected one person from the set that we know has no namesakes and their name had never changed. We found that while the new email address was specified in the certificate, the `@eesti.ee` server actually only accepts email forwarding for the old address. According to SK, this situation was caused by an error.

6.5.2.4. Email addresses not corresponding to person's name

We found 52 certificates issued in the period from 2016-07 to 2018-03, where the surname in the email address did not match the surname specified in the certificate's subject name. All of the certificates belonged to persons who had recently changed their names. The affected certificates were issued in the ID card certificate renewal process and the email address corresponded to the person's previous email, before the surname of the person had changed[4]. Apparently, the renewed certificate was filled with the person's name from the population register, while the email address was copied from the old certificate. This also resulted in an inconsistency, where the person's name printed on the ID card was different from the name in the renewed certificate. To prevent such a situation, the business rules of SK should disallow the renewal of an ID card when the person's name has changed, as a CA must not issue certificates with incorrect data (person's previous name).

[4]We found that at the time of the renewal, the new ID card with the new name was already manufactured, but not yet issued to the person.

6.5.2.5. Truncated email address

We found an ID card authentication certificate issued on 2010-05-07, containing a truncated email address "`sergo.lanno@ees`" which belonged to another person, whose authentication certificate with full email address was issued on the same day. Most likely the certificates were issued at the same time and the corruption was caused by some software error.

6.5.2.6. Invalid email addresses

We found 354 email addresses where the local-part of the `@eesti.ee` address was not valid according to RFC 3696 [296]. The local-part contained space characters or parenthesis, started or ended with a dot character or had dot characters appearing consecutively. This problem most likely has been fixed permanently, because the last certificate where such a problem is present was issued on 2013-03-25.

6.5.2.7. Email addresses not accepted by `@eesti.ee` server

It is interesting to note that we found 22 persons who have no given name (given name is set to "–") and hence their `@eesti.ee` email address starts with a hyphen. RFC 3696 [296] allows the address to start with a hyphen, however, for security reasons [297] the `@eesti.ee` Postfix server does not accept email to addresses starting with a hyphen. An error "`501 5.1.3 Bad recipient address syntax`" is returned when trying to send an email to these recipients.

6.6. Certificates with incorrectly encoded public keys

In September 2015, while testing the beta version of Google Chrome version 46, it was discovered that the certificates of recent ID cards could not be used with Google Chrome to perform TLS client certificate authentication [298]. The problem manifested because Google implemented stricter encoding checks in their OpenSSL fork BoringSSL, enforcing integer values in the certificate to be correctly encoded according to ASN.1 DER encoding rules.

According to ASN.1 DER encoding, integers are encoded using two's complement representation interpreting the most significant bit as a sign bit. This requires positive integer values that have the most significant bit of the most significant byte set to be zero-padded. The ID card certificates were encoded ignoring this requirement. This resulted in the RSA public key values being interpreted as negative values and led to an error condition in the development version of Chrome. Some of the certificates had the integer value padded even though the most significant bit was not set and hence the padding was not necesary. This resulted in another decoding error.

6.6.1. Affected ID cards

While at first it was announced that around 250 000 ID cards issued in the one-year period after September 2014 were affected [298], later in February 2016 it was announced that 420 000 ID cards needed to be updated to be compatible with Chrome [299]. We counted 434 567 ID cards (MICARDO – 131, jTOP SLE66 – 175 994, jTOP SLE78 – 258 442) that were valid on 2015-10-01 that had certificates with incorrectly encoded RSA public key.

In the certificate data we found that there were already certificates issued from 2002 for the MICARDO-powered ID cards that had incorrect ASN.1 DER encoding for the public key exponent value. The value of the random public exponent was always padded to 3 bytes hence introducing excess padding in the case where the exponent was smaller than 16 bits. The problem was fixed in 2014-09, because since then the certificates for MICARDO cards (issued in the certificate update process) encoded the exponent correctly.

The certificates for jTOP SLE66-powered ID cards contained negative moduli starting from their issuance in 2011. Starting from 2011-07-19, the modulus was correctly padded. However, for a year (from 2011-08 to 2012-09), every day there were still several certificates issued with the negative encoding of the modulus. Most likely the ID card manufacturer failed to implement the fix on one of their ID card personalization lines. From 2012-09 the modulus was correctly encoded until 2014-01, when the ID card manufacturer modified the personalization process to also include 2047-bit RSA keys generated by the jTOP SLE66 platform in the certificates (see Section 4.3.1). All the certificates containing 2047-bit keys (until the end of the jTOP SLE66-powered ID card issuance in 2014-12) had extra padding for the modulus encoding. The encoding of the public exponent also had problems. The public exponent was always padded to 4 bytes, which led to unnecessary padding if the randomly generated exponent was smaller than 24 bits. The 32-bit exponent values were not generated to avoid the compatibility problem discovered for the MICARDO cards (see Section 5.2.1). The jTOP SLE66-powered ID card certificates issued in the renewal process in PPA service points (from 2012-09 until 2017-07) had a correctly encoded 2048-bit modulus and the fixed public exponent 65537.

The problem also affected jTOP SLE78-powered ID cards from the beginning of their issuance in 2014-10. All the moduli were negative, but the fixed exponent 65537 was correctly encoded. All the certificates for Mobile-ID and MULTOS-powered cards were issued with correctly encoded public key values. Mobile-ID used a fixed exponent, while the MULTOS-powered cards had a random exponent.

6.6.2. Mitigation

The certificate issuance process was largely fixed on 2015-09-22. However, up until 2015-10-21 some certificates for the jTOP SLE78-powered ID cards were still issued with negative modulus. These certificates were issued in the PPA renewal process to fix the colliding `@eesti.ee` email addresses (Section 6.5).

After it became evident that the affected ID card holders would not be able to use Google Chrome version 46 (planned to be released on 2015-10-13), two bug reports [300, 301] were opened in Chrome's issue tracker. The involved parties asked Google to implement a workaround for the next 5 years, until the affected ID cards expired. At the end, it was agreed that Google would give the authorities 6 months to replace the malformed certificates. The Estonian state decided to develop a solution for remote certificate update, since the service points of PPA would be overwhelmed otherwise and the update process would be especially complicated for all of the roughly 5 000 Estonian e-residents at that time [302].

6.6.2.1. ID card update process

The remote update solution had to be ready by March 2016 at the latest, when the strict certificate validation was expected to be enabled in Chrome. While for the jTOP SLE66-powered ID cards the remote update only involved the replacement of the certificate (see Section 5.3) the jTOP SLE78-powered ID cards required the entire EstEID applet to be reinstalled to update the certificate (see Section 5.4). In the case where the cardholders of the jTOP SLE66-powered ID cards renewed the certificates in PPA service points, the entire applet was reinstalled and new PIN codes issued instead of only replacing the certificate. It was done this way because the applet replacement did not require the knowledge of PIN1, and the applet replacement solution and the workflow was already in place to fix the security flaw discovered in 2011 (see Section 6.4). For the jTOP SLE78-powered ID card renewal in PPA the software solution and workflow was already in place to fix the certificates with wrong email addresses (see Section 6.5).

The certificate renewal (both remotely and in PPA service points) was open from 2016-03-18 [303]. At first the remote update option was offered to around 155 000 cardholders of affected jTOP SLE66-powered ID cards [273]. Starting from 2016-06-22, the remote update was also offered to around 264 000 cardholders of the affected jTOP SLE78-powered ID cards [273]. In January 2017, the remote update solution was also offered to those cardholders whose certificates were signed with the SHA-1 hash algorithm [304]. This allowed cardholders whose certificates were issued before January 2015 [125] to obtain certificates signed using a stronger hash algorithm SHA-256. Since most of the MICARDO-powered ID cards expired at the end of 2016, for the cardholders of MICARDO-powered ID cards the solution to update incorrectly encoded certificates was not provided [273]. Around 23 000 ID cards could only be renewed in PPA service points [273], most likely because the EstEID card management features for some reason could not be used for these cards.

Even though the initially agreed deadline was 6 months, Google only implemented strict certificate validation for Chrome version 61 which was released on 2017-09-05 [305]. At that time there were still 250 000 valid ID cards with incorrectly encoded certificates, 116 000 of them were jTOP

SLE66-powered ID cards which after 2017-07-01 could not be updated anymore (see below). The updating of the jTOP SLE78-powered ID cards was still possible until 2017-10-25, when all the jTOP SLE78-powered ID cards had to be renewed to fix the ROCA flaw (see Section 6.7).

6.6.2.2. Introduction of eIDAS-compatible certificates

In 2016-11, SK issued a new certificate profile [306] which brought the ID card certificate profile in compliance with eIDAS technical requirements. This required additional fields to be placed in the certificate, in particular, new `QCStatements` attributes and the Authority Information Access (AIA) extension specifying the OCSP address where the validity of the certificate could be verified free of charge.

Due to the additional fields added and the 4096-bit RSA signature used by the intermediate CA "ESTEID-SK 2015", the size of the certificate become larger than 1536 (`0x600`) bytes which is the certificate size limit hardcoded in the EstEID applet. For the EstEID v3.5 applet on the jTOP SLE78-powered ID cards the certificate size was increased to 2048 bytes (`0x800`) [12]. Updating the EstEID v3.4 applet on the jTOP SLE66-powered ID cards was considered to be too expensive and therefore was not pursued. It was decided that the eIDAS-compatible profile would be only used for the certificates on newly issued ID cards and for the ID card renewals certificates would be issued using the old certificate profile. However, the auditors required the eIDAS-compatible certificates to be issued by 2017-07-01 at the latest, which meant that the renewals of the jTOP SLE66-powered ID cards had to be ended on 2017-06-30.

In the certificate data we found that by some error on 2016-11-07 eight jTOP SLE66-powered cards were already (unsuccessfully) renewed in PPA by issuing the eIDAS-compatible oversized certificates for these cards. For some cardholders the card was later renewed successfully using the old certificate profile, while for others a jTOP SLE78-powered warranty replacement was issued.

We see that contrary to the public announcement about the end of jTOP SLE66 renewals, four jTOP SLE66-powered ID cards were also renewed in PPA on the first working day of July 2017 (2017-07-03). The renewals were successful as the certificates were issued using the eIDAS non-compatible certificate profile. Also on the same day four jTOP SLE78-powered ID cards were renewed with eIDAS non-compatible certificates. The next day (2017-07-04) one jTOP SLE66-powered ID card was renewed (unsuccessfully) using an eIDAS-compatible certificate. For this card a jTOP SLE78-powered warranty replacement was issued on 2017-07-20.

The eIDAS requirement that was not implemented until the IDEMIA-powered ID cards were introduced was the use of the country code PNOEE for encoding the personal ID code in the `SerialNumber` field of the certificate's subject name. It was explainted to the auditors that the migration to the PNO prefix needed some more time, as many legacy services needed to be updated to use the new form of the `SerialNumber` field.

6.6.3. Discussion

The problem of incorrect integer encoding was ignored for so long because several software products created certificates with incorrect encoding and most of the crypto software tolerated such incorrect encodings. For instance, the most popular crypto library OpenSSL, since v0.9.3 released on 1999-05-24, had a compilation option `NEG_PUBKEY_BUG` which enabled the handling of broken certificates that encoded public key elements as negative integers [307]. This option was enabled by default in OpenSSL v0.9.6 released on 2000-09-24 [307]. It has been reported that at least until 1998, Microsoft software encoded integer values ignoring the sign bit [308]. The GnuTLS `certtool` up until 2010-03-15 (when the GnuTLS v2.8.6 was relased) also ignored the sign bit when encoding the certificates [309].

According to SK CEO, the reason why this error went through and was permanent was that no browser had discovered it before [310]. However, in 2012 it was already known that some software failed to handle ID card certificates (slide 6/1 in [311]). Even the EstEID v3.4 specification produced in 2012 states that the public keys of v3.0 cards do not comply with the ASN.1 standard and therefore may not be accepted by some information systems (page 113 in [9]). Only the actions of the software giant Google were able to force the ID card manufacturer and SK to finally fix the problem.

The incorrect encoding of the public key is not a security issue on its own. However, it showed the non-conformance to the standards against which SK was audited. The certificates were not issued according to CA's certificate policy referenced in the certificate and hence could have been revoked on the basis that incorrect data had been entered in the certificate.

This case once more gave the state an opportunity to practice fixing large scale ID card production errors. Later, the readiness of the remote update solution played an important role in fixing the ROCA flaw (see Section 6.7).

6.7. Infineon's RSA key generation flaw

In October 2017, researchers from Masaryk University in Czech Republic published a paper "The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli" [258] describing the vulnerability CVE-2017-15361 (hereinafter – the ROCA flaw) in secure hardware chips manufactured by Infineon. The vulnerability in Infineon's proprietary RSA key generation algorithm allowed the factorization of 2048-bit RSA keys in only 140.8 CPU-years. This vulnerable RSA key generation function was present in the jTOP SLE78 platform, powering more than 750 000 Estonian ID cards. The discovery of the flaw started the so-called Estonian ID card crisis described in this section.

6.7.1. Timeline of developments

The first suspicion of an anomaly in Infineon's RSA key generation algorithm was already raised by the same Czech researchers in their "The Million-Key Question – Investigating the Origins of RSA Public Keys" paper presented in USENIX Security Symposium in August 2016 [252]. At that time it was evident that the distribution of RSA primes generated by the Infineon jTOP 80K smart card platform did not match the distribution of randomly generated numbers. At the end of 2016, the researchers were able to reverse-engineer the proprietary RSA key generation algorithm used in the card and found a method to practically factor a faulty 2048-bit RSA key in only 140.8 CPU-years, requiring an estimated $40 000 in computational power [258].

On 2017-02-01, the researchers disclosed the vulnerability together with proof-of-concept code and CPU-year figures to Infineon. Infineon confirmed the severity of the flaw and started to inform their customers. From the security advisory [312] of Yubico (one of Infineon's customers), we learnt that Yubico was informed in May 2017, under strict coordinated vulnerability disclosure restrictions. Yubico was allowed to implement mitigation measures, but was forbidden from disclosing the flaw until October 2017, when vendors of the Trusted Platform Module (TPM) chip would release a fix for their firmware and the researchers would publish initial information about the flaw.

In August 2017, a reviewer at the ACM CCS conference, where the ROCA paper [258] was submitted, suggested testing the Estonian public certificate repository to see whether the keys on Estonian ID cards were also affected. The test confirmed that the Estonian ID cards were affected and were still being produced with the vulnerable keys. On 2017-08-30, the researchers sent formal notification to CERT-EE, which initiated the incident handling process. [313, 314]

As the flaw affected ID cards issued to more than half of the Estonian population, the incident escalated into a crisis. As the crisis management process involved more than 200 people from a wide range of areas, a journalist soon learned of the flaw [315]. The government then decided to make a proactive press conference on 2017-09-05 with the participation of the Estonian prime minister [315]. The public narrative was that the risk was "great enough to take it seriously, but not enough to cancel the cards" [316]. The state knew the CPU-year figures, but only provided an obscure statement that 60 billion euros would be required to compromise all of the affected cards [317].

Immediately, the security of i-voting, planned from October 5th to October 11th, for electing local government councils was called into question [318]. On 2017-09-06, the National Electoral Committee unanimously decided to allow i-voting with the affected ID cards as doing otherwise would be peculiar in a situation where the government had publicly stated that the Estonian ID card was still secure [319]. The decision was appealed [320], but the Supreme Court rejected the appeal, stating that based on current information the resources

needed were too great to exploit the flaw to an extent that could affect the election results [321].

On 2017-10-10, Microsoft issued a security advisory informing the public about the vulnerability in TPM chips [322]. On 2017-10-16, other affected vendors issued software updates and guidelines for mitigation, and Infineon sent vulnerability notifications [323] to those customers who were not informed before. On the same date, the researchers published initial information about the flaw and released tools to test whether a particular RSA key was affected [324]. RIA announced that the published information did not contain anything new for the Estonian ID card, but mainly provided more details about the content and the impact of the vulnerability [325].

On 2017-10-25, the production of new ID cards was switched from the flawed RSA implementation to the ECC algorithm which was also supported by the chip, but was not affected by the flaw. On the same date, the affected cardholders were provided with an option to update their ID cards either at PPA customer service points or remotely over the internet.

On 2017-10-30, the researchers published the full paper describing the flaw in detail and presented their findings at the ACM CCS conference on 2017-11-02 [324].

On 2017-11-02, the Estonian government decided to suspend the affected certificates on the midnight of 2017-11-03 [326]. The decision was justified by the increased risk of exploitation due to the full research paper being made available and a malware allegedly being available on the black market[5] that could exploit the flaw [326]. For the next five months, the holders of the suspended certificates were still able to update their ID cards at PPA customer service points and remotely over the internet.

On 2017-11-05, Daniel J. Bernstein and Tanja Lange published an improved attack code which according to their claims could provide 5-25% faster factorization for 2048-bit keys when compared to the Czech researchers' initial estimates [327].

The possibility to renew the affected ID cards was closed on 2018-03-31. The certificates of the non-renewed ID cards were revoked on 2018-04-01.

On 2018-04-19, a news article was published [328] informing the public that at the beginning of April, as a proof-of-concept, RIA was able to factor the vulnerable ID card authentication key belonging to RIA's eID Domain Manager Margus Arm. The attack tool was developed by the Estonian research and development company Cybernetica AS. RIA used their own internal resources for the computation. They did not disclose the computation effort spent on the attack, only the electricity cost of a few thousand euros was provided.

In spring 2018, RIA commissioned Tallinn University of Technology to perform a study on the lessons that could be learned from this ID card crisis. The report [329], based on expert interviews, was presented together with other

[5]The malware information was not confirmed by any source.

presentations on the subject in an international conference "The Lessons We Learned" held on 2018-05-09 in Tallinn [330]. RIA released a booklet about the case and the lessons learned [76]. Since then, several research papers have been published about the case [18, 331, 332].

In June 2019, a master thesis (supervised by the author of this work) was defended in the University of Tartu, showing that based on the properties observed from the keys generated by the affected jTOP SLE78 platform, the original attack can be optimized to 35.2 CPU-years for 90% of the keys and 70.4 CPU-years for the remaining 10% of the keys [259].

6.7.2. Mitigation

As the first mitigation step, before the public press conference on 2017-09-05, the authorities restricted access to the public LDAP repository holding public-key certificates of the issued ID cards (see Section 2.17). The restriction was made on the grounds that without access to the public keys, the security flaw could not be exploited [333].

As another form of mitigation, officials recommended the use of Mobile-ID whose private keys were not affected by the flaw [316]. For such a measure to have practical value, the ID card holders would not only have to use Mobile-ID, but would also have to revoke the vulnerable certificates of their ID cards. This, however, was not emphasized in public communication [334].

Since the production of ID cards, using a brand new ID card platform (supplied by the winner of the new ID card procurement – IDEMIA), could not start for at least 3 months [315], a solution was needed to issue secure ID cards using the current jTOP SLE78 platform.

In the course of finding a solution, three technical options were considered: (1) to generate 2048-bit RSA keys externally and import them inside the card; (2) to generate 3072-bit RSA keys using the vulnerable algorithm, which would still provide at least 100 bits of security [335]; (3) to switch to the elliptic curve cryptography (ECC) algorithm supported by the platform. The option to generate keys outside the chip was discarded as it would violate the security principles, increasing the risk of someone having a copy of the private key. The jTOP SLE78 platform was discovered not to support RSA keys above 2048 bits. Thus, it was decided to modify the EstEID JavaCard applet to move from 2048-bit RSA to ECC keys using NIST curve P-384. The NIST standard curve was chosen as it is well supported in cryptographic libraries. [96, 336]

The date 2017-10-25 can be considered the date when the technical solution was implemented and the patch was available to cardholders. The production of ID cards with RSA keys ended on the evening of 2017-10-24. Starting from 2017-10-25: the ID cards were produced with ECC keys; the cardholders could renew their ID cards in PPA service points and remotely over the internet; and PPA employees were instructed to renew the already produced vulnerable ID cards before handing them out to the cardholders. [337]

From an operational perspective, the switch from RSA to the ECC algorithm required service providers to adjust their systems to support the ECC algorithm for ID card authentication. To facilitate this, ECC-enabled ID card test cards were provided to service providers free of charge at the end of September 2017. The DigiDoc Client application for digital signatures and the digital signature file format (BDOC) already had support for ECC, since Mobile-ID moved from 1024-bit RSA to the ECC algorithm using NIST curve P-256 on 2015-01-01 [338]. It was later discovered that the ECC-enabled ID cards did not work when used to log into Windows workstations, but soon after, the ID card client software was updated to include a new Windows minidriver [339]. The encryption function also had to be updated, as the CDOC encryption scheme [95], used for encrypting data, only supported the RSA key exchange mechanism.

6.7.2.1. ID card remote renewal

A crucial factor in the success of solving the crisis was the possibility to remotely renew the ID card applet (see Section 5.4). The technical capability for the remote ID card update was already designed and implemented in 2016, due to the need to update incorrectly encoded certificates (see Section 6.6). The remote update process only took a few minutes, required knowledge of PIN1 and in the process new PIN and PUK codes were displayed to the user [340].

The remote renewal of the vulnerable ID cards did not come without risks, as the remote update solution was not designed to securely update vulnerable ID cards with invalid certificates.

For instance, successful exploitation of the ROCA flaw could have allowed an entity who had access to the vulnerable ID card but not the PIN codes to successfully complete the renewal process without entering PIN1, instead signing the renewal request using the private key factored from the public key. This was possible because the remote applet update protocol (Section 5.4) was not able to provide proof to the backend that a successful PIN1 verification was done by the card.

Another security risk was related to the confidentiality of the new PIN codes issued in the remote update process, as the new PIN codes were end-to-end encrypted using the vulnerable authentication key of the cardholder.

6.7.2.2. ID card update process

In the first days the remote update process faced serious difficulties, as the system could only handle 1 000 parallel update processes. Since the process was dependent on stable and timely interractions between the servers of four parties (RIA, PPA, SK and Gemalto) the system experienced temporary downtimes. Many cardholders experienced delays caused by overloads and had to attempt the update at a later time. [341–343]

After the affected certificates were suspended on 2017-11-03, the remote update service was provided exclusively to a list of the 35 000 most active

ID card users, betwen noon on Friday and the end of Sunday. The list consisted primarily of medical workers and employees of state institutions. Two thirds of those on the list updated their certificates over the weekend. [344, 345]

It was discovered that around 18 000 ID cards could not be remotely renewed because the card manufacturer for an unknown reason was unable to use the symmetric applet-level card management keys loaded onto the card in the personalization process [346].

The PPA service points in larger cities were also kept open on weekends and the original opening hours were extended by one hour [343]. PPA opened service points in five hospitals across Estonia to enable medical workers to update their ID cards[6] [347]. On the weekends of November and December, PPA opened temporary service points in shopping centers across all of Estonia [348, 349]. The state made billboard ads and radio clips, and later, a TV campaign asking cardholders to update their ID cards was also made [350].

The ID card holders abroad were offered the possibility to apply for a free ID card replacement over email. The new ID card, however, had to be collected by visiting the closest Estonian embassy in person [351].

The update period ended on 2018-03-31, when the certificates on non-updated ID cards were revoked. From the affected 760 000 ID cards, over 494 000 of the cards were updated (see Figure 45 for the timeline). The number of updated cards accounted for 95% of the ID cards used electronically at least once. The majority of the updated cards (356 000 or 72%) were updated remotely. [352]

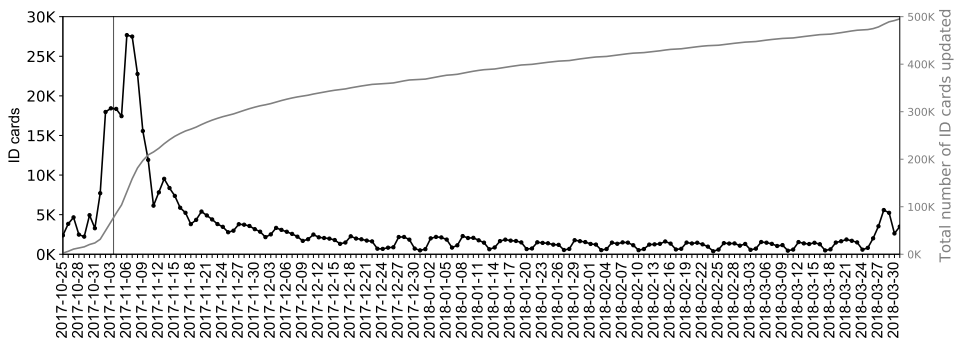


Figure 45: Updating activity of ROCA vulnerable ID cards (source: certificate data)

6.7.2.3. Encryption support

During the crisis, the government's priority was authentication and digital signature support for ECC-enabled ID cards [353]. The encryption function for ECC-enabled cards was not available until 2017-12-21 when the CDOC format was updated [95] to use the ECIES encryption scheme (ECDH algorithm for

[6]Estonia's healthcare system is organized through the e-health information system where the ID card is used as the authentication means.

symmetric key derivation) and an updated ID card client-side software was released. The unavailability of an encryption function brought difficulties, for some hospitals, in exchanging sensitive documents with various partner institutions [354].

As the digital signature scheme used in Estonia requires a timestamp to prove that the certificate was valid at the time of signing and since the majority of service providers perform certificate revocation checks in the process of ID card authentication (Section 2.8.1), the certificate suspension on 2017-11-03 largely mitigated the security risk introduced by the ROCA flaw. The revocation of the certificates, however, does not prevent an attacker who has factorized a vulnerable key from decrypting data encrypted for that vulnerable key. Therefore, sensitive information exchanged using the above mentioned CDOC files is still under the risk of compromise. The ROCA case shows that the current CDOC encryption scheme, that lacks the forward secrecy feature, should only be used to encrypt data for which confidentiality must only be protected for a very short time.

6.7.3. Legal issues solving the crisis

In our paper "Solving the Estonian ID Card Crisis: the Legal Issues" [18], we analyzed the extent by which the involved parties were able to precisely follow the applicable laws and regulations in the field when solving the crisis. We found a number of cases where the requirements were not fully followed, either due to the lack of technical preparedness, suboptimal decisions made under high time pressure, or the critical nature of the situation.

Due to the lack of technical preparedness, SK was not able to quickly invalidate a large number of certificates and correctly indicate their status in the certificate validity services. For similar reasons, SK and PPA failed to notify the affected cardholders as required by the law and SK certificate policies.

We questioned the effectiveness and the legal basis for the decision to restrict access to the public LDAP certificate repository and the decision to suspend (and not revoke) the affected certificates.

While convenient for the cardholders and significantly cost-saving for the state, remotely updating the certificates that had been suspended was not in compliance with the legal requirements. Similarly, the changes to the EstEID JavaCard applet were done without recertification of the ID card platform as required by eIDAS. The crisis also highlighted the problems in the current regulation of certificate validity suspension, in particular, when the suspension is requested by a party other than the certificate holder. It also showed that it is not clear who (if anyone) is legally liable for the ID card private key security.

Due to the critical nature of the situation, the state was forced to compel SK not to invalidate the certificates as soon as it became clear that the affected ID card platform did not satisfy the legal requirements of QSCDs. In this case, SK and their auditor TÜViT agreed to follow the plan of the authorities, but, in our opinion, this should not be a norm.

We came to the conclusion that the root cause of the crisis was the fact that the failure in a single, monocultural ID card platform put the reputation and operation of the entire digital state at risk. To reduce the risk of a future incident of a similar nature escalating into a crisis, the state should seek to equip its residents with a secondary electronic identity tool that would rely on a different technological platform (preferably a different public-key cryptography algorithm also) and a different CA. This, in case of a similar event, would allow residents to access e-services in a legally compliant manner, and potentially to also safely renew the affected electronic identity tool remotely.

6.7.4. Gemalto's failure to inform the state

From the Yubico security advisory [312] we know that Infineon had already disclosed the vulnerability to their direct customers in May 2017. This information, however, did not reach the Estonian authorities until the end of August, when the Czech researchers directly contacted RIA. Since the Estonian state is not a direct customer of Infineon, Infineon was expected to inform Gemalto, who in turn was obligated to inform PPA. Gemalto did not explain the cause of the delay, stating only that they were analyzing how the information was spread about the flaw [355].

In November 2017, it became known that the state had filed a claim against Gemalto regarding the damage caused by the ROCA flaw [356]. Before the ROCA flaw incident, Gemalto and the state had an already ongoing legal dispute with Gemalto challenging PPA's decision regarding the winner of the new ID card manufacturing procurement [357].

On 2017-11-22, Gemalto's local representative Andreas Lehmann posted a comment in LinkedIn stating that he had already informed the responsible state authorities of the flaw on 2017-06-15, but the authorities kept it quiet possibly due to the upcoming summer holidays [358]. Soon after, Lehmann deleted his comment stating that this was his personal statement.

RIA acknowledged that on 2017-06-15, in the regular meeting between Gemalto and PPA, Lehmann had hinted about an ID card flaw, but the hints were too vague to act upon. The Gemalto representative promised to provide more information when it became available, but never did so. [359]

In response to Lehmann's claims, Gemalto, in a private communication with PPA, apologized for the claims stating that Lehmann's comment was a personal statement and cannot be considered as an official statement by Gemalto [360]. Starting from December 2017, Gemalto replaced Lehmann as its representative in communications with PPA [361].

On 2017-12-01, Postimees published an article [362] arguing that the state was aware of the ID card flaw on 2017-06-20 at the latest, when the ENISA incident report [363] about the flaw in signature creation devices used in Austria was sent to all EU member states. According to state officials, the ENISA report was indeed received, but as it referred to the smart card platform CardOS

developed by Atos, which is not used by Estonian ID cards, the state was unable to associate the Estonian ID card with the Austrian incident. Furthermore, on 2017-06-22 RIA received a statement from Lehmann that the problem mentioned by him at the meeting on 2017-06-15 was not related to the Austrian report [364].

On 2017-12-06, Postimees published [365] an excerpt from our (at that time) non-public incident report made in January 2017, which informed RIA about several key management flaws found in the ID card personalization process (Section 6.8). This excerpt published by Postimees was the excerpt that was shared with Gemalto under the confidentiality clauses of the ID card manufacturing contract. The Postimees article did indeed contain RIA's comment that the findings mentioned in this incident report were not related to the ROCA case, however, the article also referred to the ENISA Austrian incident report and the statements by Gemalto representative Lehmann claiming that he had already informed the state about the ROCA flaw on 2017-06-15.

At the end of 2017, PPA sent a letter to Gemalto announcing that PPA had withdrawn their letters of recommendation which Gemalto used when applying for procurements. PPA stated that the behavior and attitude of Gemalto did not give the opportunity to continue the situation where the Estonian letters of recommendation would still be true and credible [366].

In the last days of December 2017 the Tallinn Administrative Court dismissed Gemalto's appeal about the results of the ID card procurement. Gemalto, however, later appealed this decision in the second instance court and the case is still pending [53].

On 2018-08-30, PPA announced [367] that they had made some steps in reaching a compromise, where Gemalto would be required to only reimburse the direct costs spent on addressing the ROCA flaw. The direct costs of PPA, RIA and SMIT combined were 2 million euros [368], of which EUR 1 115 616 was the cost incurred by RIA [369].

It later became evident that Gemalto had several counterclaims against PPA for violating the confidentiality clauses of the agreement. Allegedly, Gemalto submitted a new claim whenever a PPA representative publicly mentioned Gemalto or an agreement between them [367].

A few days after PPA's statement, presentation slides (apparently made by Gemalto representative Lehmann) containing the timeline of the ROCA vulnerability warnings communicated to Estonia were leaked to journalists [370] along with information regarding the possible conditions of the compromise [371].

The slides did not provide any new factual evidence that would substantiate Lehmann's claims about informing the state in an appropriate manner. Postimees, however, used the leaked information as new evidence to prove that the state officials were "lollygagging" [371].

As a response, PPA announced that Gemalto had failed to demonstrate willingness to reach a compromise and PPA would take Gemalto to court in the

following weeks [372]. In a public statement Gemalto denied leaking the information and expressed their interest in still reaching a compromise [372].

On 2018-09-26, PPA submitted a claim to the Harju County Court demanding a contractual penalty of 152 million EUR from Gemalto regarding a separate issue of private key generation outside the ID card (see Section 6.8). The claim was submitted first and separately from the others due to each violation being legally and technically very complicated. [373]

On 2018-10-25, it was made public that Gemalto had filed a breach of contract action against PPA for severely interrupting the compromise negotiations in September [374].

On 2018-11-05, PPA submitted a new claim to the court demanding a contractual penalty of 300 000 EUR from Gemalto for violating the contractual responsibility of immediately forwarding significant information, since the existence of the ROCA security risk was only confirmed by Gemalto on 2017-09-05 in response to a PPA inquiry made on 2017-09-04 [375].

A news article published on 2020-08-28 [376] stated that the judge had decided to merge both of PPA's claims into one proceeding but a new hearing date had not been set as yet. In August 2019, a preliminary hearing had been held where the possibility of finding a compromise was discussed. This, however, was not successful as of 2020-08-28 no compromise had been reached with both parties having submitted a number of different requests that the court had to resolve.

At the end of 2020, the court reversed their earlier decision and decided to split the claims and hold the next court hearings in February 2021 [377]. However, after working hours on Friday, 2021-02-05, PPA issued a press release [378] to inform the media that a compromise had been reached, with Gemalto agreeing to pay the state 2.2 million EUR in compensation. The press release stated that the agreement was signed to "close the claims on the potential vulnerability to the Estonian ID card which occurred in 2017". However, PPA confirmed to us personally that the compromise covered all the claims from both sides, including the claim regarding the generation of private keys outside the ID card.

6.7.5. Failure of Common Criteria certification

According to Infineon, the flawed RSA key generation function was certified by BSI, but no mathematical weaknesses were known, or had been discovered during the certification processes [379]. This section aims to investigate the Common Criteria (CC) certification process according to which the security of the vulnerable smart card chip (used in the Estonian ID card) was certified.

The affected Infineon jTOP SLE78 (jTOP INFv#46) platform was certified on 2013-08-07 by ANSSI under the reference ANSSI-CC-2013/55 [380, 381]. The certification process was based on a "compositional approach", where the evaluation of the product to some extent relies on the evaluation results of the microcontroller chip independently certified by the chip manufacturer. The

composite certification process evaluated the product which is the result of embedding an operating system (jTOP JavaCard platform) on top of an already certified chip.

For a secure implementation of RSA on-card key generation, according to ANSSI, the certification relied on the evaluation results of Infineon's M7820 A11 family of microcontrollers certified on 2012-09-05 by BSI [382]. The same microcontroller with some minor modifications was recertified by BSI on 2015-08-03 [382].

The Security Target of the evaluated microcontroller lists RSA key generation (implemented by the flawed cryptographic library RSA2048/4096 v1.02.013) as a Security Functional Requirement "FCS_CKM.1/RSA" (see Section 7.1.4.5 in [383]). The Security Target requires RSA keys to be generated in accordance with the key generation algorithm specified in the PKCS#1 v2.1 standard [384]. This standard, however, only specifies the format of RSA keys and does not describe the key generation algorithm. The referencing of this irrelevant standard allowed the parties involved in the CC certification process to hide the fact that a proprietary algorithm was actually used for RSA key generation. It is hard to estimate how many more CC-certified products are using proprietary shortcuts, that have not been analyzed by the cryptanalysis community.

The evaluation of the microcontroller was conducted by the evaluation facility TÜV Informationstechnik GmbH (TÜViT) in Germany. According to TÜViT [385], "although the know how of the Coppersmith method as well as the implementation [of the algorithm] was both available to the involved parties, we have to conclude that the threat was simply overlooked.". TÜViT did not answer the question as to what lessons they had learned from this ROCA case.

In response to the ROCA flaw, BSI now plans to improve transparency by requiring that the certification report at least specifies if the implemented proprietary cryptography is not exactly conformant to a recommended standard. BSI does not plan on requiring the proprietary algorithm to be published in any way. [386]

Even though the certification bodies are now aware that the security claims specified in the CC certificates do not hold anymore, neither ANSSI nor BSI have revoked the corresponding certificates. According to BSI [386], a certificate can only be withdrawn when it was issued under misconception, e.g., when it turns out that wrong evidence was submitted. After a CC certificate is issued, it must be presumed that the validity of the certificate decreases over time by improved and new attacks being discovered. In the case of the ROCA flaw, the users of the certified end products should have been informed of the flaw by the vendors. Certification bodies can issue maintenance reports and even perform a re-certification of the product. These activities, however, have to be initiated and sponsored by the vendor. Hence, it can be concluded that the responsibility of the certification body for the claims laid in the certification report ends with the issuance of the certificate.

On 2017-10-24, apparently in response to the ROCA flaw, BSI issued the maintenance report MA-02 [382] for Infineon's M7820 A11 family of microcontrollers. A careful reader can find that the updated Security Target in Section "8.5 SF_CS: Cryptographic Support" now includes an extra statement: "The RSA cryptographic key generation is not supported for this Target of Evaluation (TOE)." This statement, however, contradicts other parts of the Security Target, where "FCS_CKM.1/RSA" is still listed as a Security Functional Requirement of the TOE. The BSI maintenance report does not mention that the change to the TOE affects certified cryptographic security functionality, but concludes that the change is at the level of guidance documentation and has no effect on assurance. This BSI statement seems to be, at the very least, misleading.

While several CC certified products have been affected by the ROCA flaw, vendors' responses in the context of certification have been different. For instance, for the BSI certified CardOS platform [387] referenced in the ENISA Austrian incident report [363], on 2017-07-07 a maintenance report [335] was issued, which states that only RSA keys with a length of 3072 and 3584 bits have a security level of at least 100 bits.

6.8. Security flaws in key management

The initial draft version of the Digital Signatures Act [388] in 1999 envisioned that the digital signature key pair would be generated by the signatories themselves to prevent the possible misuse of the private keys. The final version of the act, however, no longer foresaw the possibility for persons to generate their keys themselves. The result is that in practice the cardholder's private keys are generated by the ID card manufacturer in the ID card personalization process. This situation introduces the risk of the ID card manufacturer collecting the private keys stored on the ID card.

The involved parties have maintained [389] that the manufacturing contract requires that the private key be generated inside the chip, such that it never leaves the card and no copies of the private key exist (only the public key should be exported from the card). Concerns of how this requirement would be enforced in practice were met with the standard response that everything is being regularly audited and, as the ID card manufacturer is in the business of trust, it would never risk its reputation, for example, by copying private keys.

Unfortunately, the findings of our research described in the paper "Estonian Electronic Identity Card: Security Flaws in Key Management" [19], among other things, show that contrary to the requirements, the ID card manufacturer Gemalto had generated keys outside the card and these previously mentioned security audits were not able to discover the practice over the 5 year period while it happened. We provide a brief summary of our findings in the sections below.

6.8.1. Certificates with duplicate RSA public keys

Our findings started with the discovery of 10 certificate pairs that contained duplicate RSA public keys. In most of the cases, the same public key was shared between the authentication and digital signature certificate of the same ID card. However, in two cases the same public key was shared among certificates of different cardholders. The certificates that shared a public key were issued at roughly the same time, with only a few seconds difference. In most of the cases, the involved ID cards were renewed in PPA to fix the security flaw in the 2011-issued ID cards (Section 6.4). If the cards did indeed contain duplicate private keys, then the only plausible explanation is that their keys had been generated outside the card and due to some error the private key was imported twice.

We decided to investigate the pair where the public key was shared between two cardholders – Toivo and Ülle, in more detail. In that pair, the authentication certificate of Toivo and digital signature certificate of Ülle shared the same public key, meaning that if the ID cards did indeed contain the corresponding private key, Toivo could forge digital signatures in the name of Ülle and Ülle could impersonate Toivo. We contacted Toivo and he confirmed that his ID card did indeed contain the corresponding private key. Later, we obtained convincing evidence that Ülle's ID card also contained the corresponding private key.

In the meantime the ID card manufacturer had discovered the problem and had issued new ID cards for Toivo and Ülle. However, it was still not clear to us whether the authorities were fully aware of the true reasons behind these faults, therefore we decided to inform the authorities about the case of Toivo and Ülle, and our suspicion that the keys were generated outside of the ID card. According to the authorities, the ID card manufacturer denied that the ID cards contained duplicate private keys, leading to a deadlock in solving this issue.

6.8.2. RSA private keys generated outside the ID card

Fortunately, in the meantime, there was a breakthrough in the research world due to the paper "The Million-Key Question - Investigating the Origins of RSA Public Keys" by Svenda et al. [252]. The researchers found that the RSA public key modulus N carries a fingerprint that can be used to distinguish between key generation algorithms. In particular, the range from where primes p and q are selected to obtain a modulus of the required length. This fingerprint can be observed from the probability distribution of the most-significant byte of a public key modulus.

We generated and exported millions of reference keys from each ID card platform and analyzed whether the properties in these keys (see Chapter 4) matched the properties of the public keys contained in the ID card certificates.

As a result, we found that the jTOP SLE66-powered ID card keys renewed in PPA (the renewal was offered from July 2012 to July 2017) had been generated

outside the card. We came to this conclusion because the public keys in the certificates of these ID cards were generated by setting the two most significant bits of p and q, while the key generation algorithm implemented by the platform did not set these bits.

It is important to note that this could not have been done by accident as the key import feature had to be programmed into the applet. The original EstEID v3.4 applet that was subject to a security review did not have this functionality. We hope, however, that the intent was just to speed up the ID card renewal process, because on-card key generation is quite slow and would have added an extra 5 minutes on average to the ID card renewal process. However, this case clearly demonstrates what a malicious ID card manufacturer could have done without it being discovered. It is important to note that this is not only limited to key import, as the keys could have also been exported after generating them inside the card.

At that time, from more than 74 000 jTOP SLE66-powered ID cards renewed in PPA, only 12 500 were still valid. After receiving our findings, in May 2018 PPA announced the replacement of the affected ID cards. Gemalto in their public statement denied the findings, saying that they had fulfilled the ID card agreement [390]. Later, in September 2018, PPA brought Gemalto to court demanding a contractual penalty of 152 million euros [373] (see Section 6.7.4).

Today, Gemalto has left Estonia and the latest ID cards are manufactured by a different company IDEMIA (formerly Oberthur). It is hard to say, however, if any lessons have been learned from this case, because nothing has fundamentally changed in the ID card production process to prevent similar incidents from happening again. Preferably, we would like to have a technological solution that is secure even if the ID card manufacturer is malicious. Schemes based on threshold cryptography [391–393] could help here, but introducing such changes, of course, would require a strong political will.

6.8.3. Certificates with corrupted RSA public keys

By analyzing the certificates, we found a separate key management flaw. A set of certificates from the jTOP SLE78-powered ID cards contained a corrupted RSA public key modulus, meaning that the modulus contained small factors (e.g., 3, 5, 7). The inclusion of corrupted RSA key moduli led to a security issue, because the corrupted modulus (which essentially is a random integer) if fully factored, could result in to the corresponding private exponent being calculated. We succeeded in fully factoring one of the corrupted modulus, demonstrating the potential risk [394].

Eventually, we came to the conclusion that the corruption happened on the ID card personalization line during the transmission of the modulus of the generated key from the card to the terminal. More details on this and the previously mentioned findings are available in our paper [19].

6.9. The ID card's built-in security measure causing it to lock itself

In October 2017, RIA announced that the use of an ID card as a loyalty card in point of sale (POS) terminals could lead to an ID card being locked. According to RIA, the lock is triggered by the security mechanism that is built into the card as it perceives too excessive ID card usage as an attack and locks the card as a security measure. The cardholders who experienced such a lock were invited to turn to PPA. [395]

Later, it turned out that the locking was not related to the use of the ID card in POS terminals, but was triggered by the Pkcs11Interop software component that used the OpenSC driver (`opensc-pkcs11.dll`) to read the ID card certificates [396]. From the related OpenSC bug report [397], we can see that the EstEID v3.5 applet had a bug in the code that handles the `READ BINARY` command which was used to read the certificates from the card. The bug was triggered when the `READ BINARY` command was sent over a T=0 connection with a positive offset specified in `P1` and `P2` and the `Le` byte of the APDU set to `0x00`. The EstEID v3.5 cards, in this case, responded with the status word `0x6F00` (No precise diagnosis is given).

We replayed the faulty `READ BINARY` command against jTOP SLE78-powered ID cards running EstEID applet v3.5.2, v3.5.3, v3.5.7 and v3.5.8, and were able to trigger the card lock in EstEID applet v3.5.7 and v3.5.8 after sending 60 faulty `READ BINARY` commands to the card. It seems that these versions of the EstEID applet were counting the number of unexpected exceptions raised by the applet and after the counter reached 60, the EstEID applet invoked JavaCard's `GPSystem.lockCard()` call, which resulted in the GlobalPlatfom card's life cycle state being set to `CARD_LOCKED`. Once the card was locked, only the GlobalPlatform ISD applet could be selected. The card could only be unlocked using the GlobalPlatform ISD security keys. Effectively, this makes the ID card vulnerable to logical denial-of-service attacks even if a smart card reader with a PIN pad and a PIN firewall is used.

It is common for smart card platform manufacturers to provide a confidential operational guide document which contains a list of applet-level security recommendations that should be implemented by applet developers to protect the applet against fault injection and other attacks (page 21 in [398]). Since unexpected software exceptions could be caused by an unsuccessful fault injection attack, the locking of the card limits the number of attack attempts the attacker can perform against the card. However, we found out experimentally, that the locking mechanism implemented by the EstEID applet may not be effective in practice, as the card lock is only enforced after the EstEID applet is re-selected. This means that as long as the EstEID applet is not deselected (the card is not powered off or reset), the attacker could perform an unlimited number of attack attempts long after the counter has been exceeded and the card's life cycle state has been set to the state `CARD_LOCKED`.

In November 2017, PPA received a few hundred [399] and in 2018 more than 150 [396] applications for the replacement of locked cards under warranty. In 2019, it became known that IDEMIA-powered ID cards were also affected by a similar problem [400]. The use of some "incorrect" card readers has been mentioned as the cause, but the exact reason why the IDEMIA-powered ID cards are being locked is still unknown [401].

In 2020, the Chancellor of Justice raised an issue about PPA's procedure for the replacement of locked ID cards [402]. To replace a locked ID card under warranty without paying the state fee, the ID card must be examined by IDEMIA experts in France. Since the examination can take a considerable amount of time, understandably the affected cardholders will want to apply for a new identity document before the results of the examination are known. However, if an application for a new ID card is submitted, the state fee that was paid is not reimbursed even if the examination later reveals that the ID card lock was not the fault of the cardholder. We note that the replacement ID cards issued under warranty terms are issued with the expiration date of the original, therefore PPA's reluctance to reimburse the state fee for a new ID card with a fresh expiration date is understandable.

The Chancellor of Justice found that the state must be responsible for cases where the electronic part of the ID card becomes unusable due to errors caused by the peculiarities of the ID card software solution. Although the state has been aware of this technical problem at least since spring of 2019, the state has not informed cardholders that such a risk may exist and how to avoid it. The Chancellor of Justice recommended that PPA should consider changing the practice when replacing faulty ID cards such as introducing an initial examination where the probable cause of the card's failure could be received within 5 working days.

Initially, PPA announced that they did not intend to follow the recommendations [403], but later rescinded [404] promising to introduce the opportunity to establish the cause of ID card malfunction in PPA customer service points. Furthermore, PPA promised to return the non-functioning ID card while the replacement ID card is being produced so that the person could at least use it as a physical identity document.

6.10. Failure to revoke ID card certificates of deceased cardholders

On 2019-06-26, a news article [405] was published informing the public about an incident where, due to a technical failure, the certificates for more than 15 000 automatically revoked ID cards were not revoked.

When a person dies or a resident obtains citizenship, the ID card is automatically revoked and hence also the certificates therein (see Section 2.15). Once a day, an information system run by the IT and Development Centre at the Estonian Ministry of the Interior (SMIT) sends requests to SK to revoke the

affected certificates. According to PPA, from 2014 to mid-2015 the data was submitted to the SK information system using incorrect request parameters and this resulted in an error. As the data exchange was not monitored on either side, the error was only discovered in the middle of 2015. However, even then, the past faults were not investigated and the failed requests were not resent. [405]

In spring 2019, the issue was finally investigated as some errors began to appear when attempting to revoke the certificates of certain ID cards. It was discovered that the certificates for more than 15 000 ID cards had not been revoked in the period from 2014 to mid-2015. Furthermore, 353 of these ID cards had been used electronically[7] after their certificates should have been revoked. In 285 of the cases the ID cards belonged to deceased cardholders. [405]

6.10.1. Police investigation and liability

The police (PPA) initiated a criminal proceeding to investigate the transactions that were made with the ID cards of deceased cardholders. In most cases the scenario was the same: the deceased person's bank account was accessed, bills were paid, contracts were terminated and money was transferred to the first-degree heir. PPA ruled that such conduct was not criminal as it is common for heirs to pay for funeral expenses as well as current bills using the deceased's funds. In the course of the inheritance proceedings the notary will make a set-off if necessary. [406]

In 2020, the Chancellor of Justice was approached [406] by a person whose grandmother's ID card was one of the documents that was electronically used after the revocation. Some other relative had made bank transfers with the deceased person's ID card and as a result the amount of inheritance decreased. The person asked whether the state could be jointly held liable for the damage caused by the use of this card.

In their reply to the Chancellor of Justice, PPA stated that no damage could have occurred in this situation because such conduct does not affect the right to inherit as the notary can make a set-off if necessary during the inheritance proceedings. However, the Chancellor of Justice found that since PPA had not contacted persons (potential heirs) who may have been affected by this issue, the heirs would not have been able to raise this issue during the inheritance proceedings. We note that, while PPA did not contact potential heirs directly, the news article [405] published on 2019-06-26 contained a link to a web service [407] where by entering the personal ID code of a deceased person it was possible to determine whether their ID card was electronically used after the document had been revoked.

[7] The database of OCSP certificate validity responses maintained by SK can be used to infer whether relying parties had requested validity confirmation of the involved certificates and hence whether the ID card had been used electronically. Cardholders can see a log of OCSP requests with the requester's IP address for their certificates at `https://minutoimingud.sk.ee/`.

In addition, PPA noted that they could not be held liable for the damage suffered as there was no causal link between the actions and/or omissions of PPA and the use of the deceased person's ID card. The existence of a technical opportunity to use the card did not give the right to use it. PPA noted that although the existence of access indirectly allowed the continued use of the ID card, they, however, did not cause the damage and there was no right to claim compensation from them for the damage done.

The Chancellor of Justice did not agree with PPA's position, as the activities of PPA created the opportunity to use an ID card after a person's death. The obligation that a document must be declared invalid after a person's death serves the purpose of preventing third parties from using the deceased person's identity document and making transactions on behalf of that person. Thus, there is a causal link between the activities of PPA and the continued use of the card. The Chancellor of Justice concluded that if a person had suffered damage, it is possible to apply for compensation from PPA under the State Liability Act.

6.10.2. Analysis of certificate revocation data

In their 2020-03-10 reply to the Chancellor of Justice [406], PPA confirmed that at present it was no longer possible to continue using revoked ID cards electronically and that SMIT had significantly improved data exchange and monitoring to prevent the same error from happening again.

However, our analysis of certificate revocation data shows that the failure to revoke certificates of deceased persons was not an isolated case as was reported in the media. Instead it has been present on a smaller scale throughout the years and even today there are currently several deceased persons' ID cards whose certificates have not been revoked.

We obtained the date of death for deceased persons using the e-service provided by the Population Register [408] and correlated this data with the ID card certificate revocation data from CRLs.

First of all, we found that the date of death registered in the Population Register for 138 persons was set to a date in the future – the dates 2020-12-29, 2020-12-30 and 2020-12-31. For 92 of these persons their ID cards were revoked on 2019-12-30, 2019-12-31 and 2020-01-01, hence the actual date of their death had been most likely a year earlier than the date registered in the Population Register. On 2020-10-07, we informed SMIT of this issue. They informed us that this was a data display error that should be fixed at the end of 2020-10, and that the correct date of death is indeed a year earlier.

Using the revocation data, we calculated the average time in which the ID card certificates of a deceased person are revoked or expire after their death. The average by month is shown in Figure 46.

For the year 2011 our dataset did not have enough deceased persons' certificates, therefore for that period we cannot provide a comprehensive overview. The average time in 2011 was determined by the outliers – a few certificates that had not been revoked and thus have appeared in our dataset.

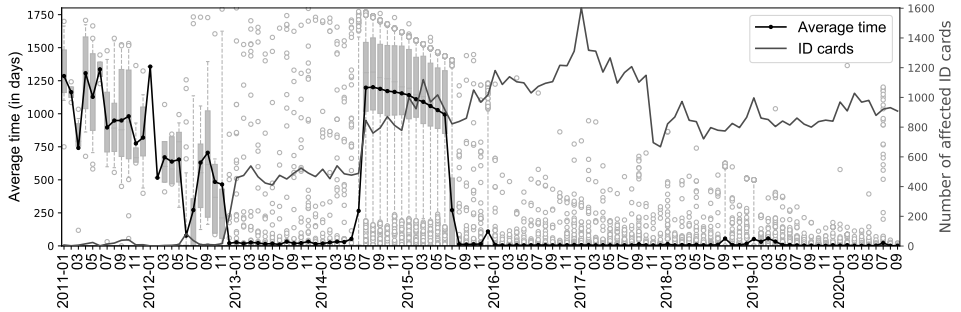


Figure 46: Time in which a deceased person's certificates are revoked or expire after their death (source: our dataset)

From the data we found that the large scale data exchange error that was reported in the media was present from 2014-07-01 to 2015-07-10 and the affected certificates were revoked on 2019-05-09. We also found that data exchange errors appeared on a smaller scale during the last days of 2015, in September 2018 and in the first months of 2019. Most of the certificates affected in these cases were revoked on 2019-05-09, 2019-05-10 and 2019-05-16. We found that the incident was not fully investigated and solved in 2019 as the revocation of missed certificates once again occurred on 2019-09-30, after PPA's media announcement.

As we can see in Figure 46, even in the periods where no significant data exchange errors occurred, there were quite a few certificates that had not been timely revoked.

Furthermore, we identified 27 deceased persons whose ID card certificates were still valid on 2020-10-08. The date of death for these persons were in the period from 2019-04-08 to 2020-09-28, with the date of death for the majority of the cases (17) being 2020-07-28. On 2020-10-08, we informed PPA, SK and RIA of the case. We received no response, but observed that the certificates of most of the affected ID cards were revoked on 2020-10-14. However, the certificates of 4 of the affected ID cards were still valid on 2020-10-21, of which we once again informed PPA, SK and RIA. It turned out that these four ID cards were part of another type of failure, as even the identity documents of the corresponding deceased persons were not revoked. We observed that the certificates of these ID cards were finally revoked on 2020-10-30. Later, on 2021-02-03, a news article was published about these findings [409].

It is important to note that the risk of a deceased person's ID card being abused cannot be completely avoided as there is a window of time available (might be up to several days) until the doctor prepares a death certificate and the data is registered in the Population Register. Furthermore, for certain types of residents (especially e-residents[8]), if the person dies outside Estonia, the Estonian state may not learn about the person's death for years.

[8]From our dataset of 49 360 persons who have had the status of e-resident, the Population Register has only recorded the death of 12 of them.

6.11. Transparent PIN envelopes

On 2002-05-03, just a few months after the first ID card was issued, it was discovered that the security envelope holding the PIN codes did not meet the security requirements – the PIN codes were clearly visible through the enclosed envelope by holding it under a lamp (see Figure 47) [410].

The ID card manufacturer Trüb responded by blaming the German company producing the envelopes [410]. However, as we can see from the fix (Figure 47), the envelopes were not changed, but the printing was adjusted to replace the black digits with white digits on a light gray background.

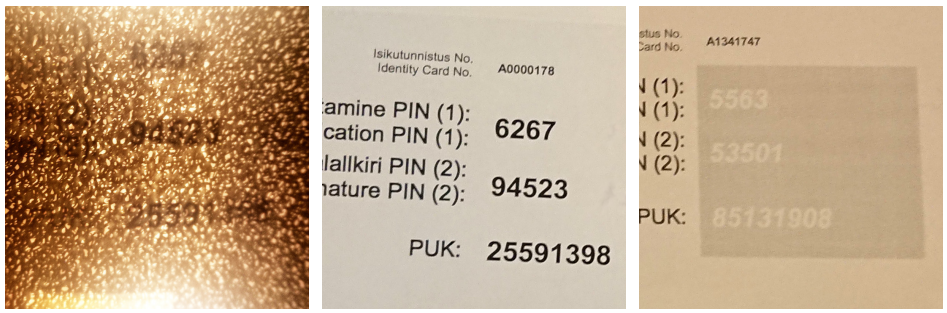


Figure 47: Trüb-issued flawed (left and center) and fixed (right) PIN envelopes [411]

It can be estimated that at least 15 000 ID cards had been produced with the flawed PIN envelopes [41]. CMB did not recall the already manufactured ID cards residing in banks that had not yet been delivered to the cardholders, but made a recommendation for cardholders to change the PINs either by themselves or by visiting a CMB customer service point in Tallinn [412]. CMB considered claiming damages from Trüb for the poor quality work [412], but it is unknown whether the claims were made in practice.

The same flaw was reintroduced by the new ID card manufacturer IDEMIA in 2018. On 2018-12-20, a few weeks after the issuance of the new generation ID cards, it become known that the PIN codes were clearly visible through the envelope using an ordinary pocket lamp (see Figure 48) [413].

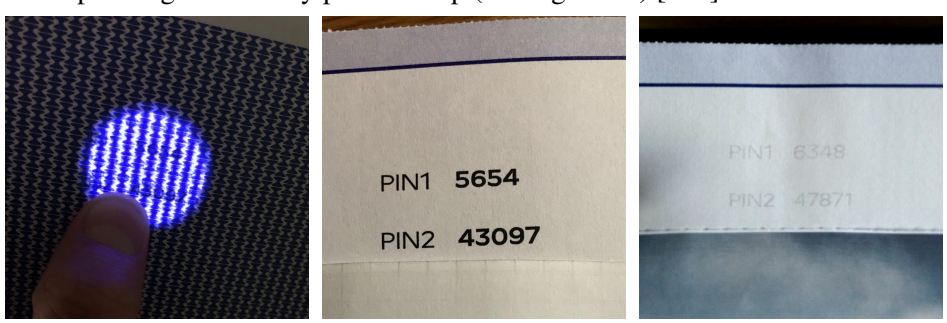


Figure 48: IDEMIA-issued flawed (left and center) and fixed (right) PIN envelopes [414]

In total, around 5 000 cards with the flawed envelopes were issued. According to PPA, the flaw did not present a direct security risk because internal security procedures for transportation and storage exclude the possibility that the ID card along with the PIN envelope would reach some third persons and the risk that PPA employees would look through the envelopes using a pocket lamp did not exist[9]. Furthermore, the card could only be used electronically from the moment the card is issued to the cardholder. [417, 418]

We note that in practice, the non-transparent PIN envelope is the only physical measure in place that prevents a PPA employee from using the electronic functionality of the cardholder's ID card without it being detected. As an additional measure, the corresponding certificates are not valid until a PPA employee activates the card, marking it in the system as handed out to the cardholder (Section 2.15). Since the cardholder receives an email notification from SK about the certificate activation, it would be risky for a PPA employee to activate the certificates before the cardholder has arrived for the card. However, during the PIN replacement procedure (Section 2.11.4) the PPA employee has access to an already active card which can be abused to its full extent using the PIN codes visible through the flawed PIN replacement envelope. It is not uncommon for the PPA employee to perform the PIN replacement procedure in a separate room which makes the execution of the attack even easier.

We note that the security measure of certificate activation does not prevent the creation of a valid digital signature in the period before the ID card is activated (see Section 2.10.3). At its best, it prevents an attacker from using not yet valid certificates to authenticate in e-services that check certificate revocation status. For Gemalto-manufactured ID cards, SK validity services acted as if not yet valid certificates had not been issued, therefore they were not listed in CRLs and the OCSP returned the status "unknown" (see clause 4.3.1.1 in [172]). This resulted in a security risk in the case of CRLs, since certificates not listed in the CRL are presumed to be valid. Only after the introduction of the IDEMIA-powered ID cards and the new root CA in fall 2018, the SK validity services started to correctly list not yet valid certificates as being technically suspended.

On 2018-12-28, a news article was published with PPA stating that on 2018-12-20 the ID card manufacturer had fixed the envelopes and it was no longer possible to see the PIN codes through them [419]. On 2019-01-16, we observed that in the PPA customer service point Tammsaare in Tallinn the flawed PIN envelopes were still being used for the issuance of *digital identity cards* and PIN replacement envelopes. We informed the representatives of PPA, SK and RIA about our observations, but received no promise on when the flawed PIN envelope supplies were to be replaced.

[9]The fact that insider threat should not be underestimated is well illustrated by the large scale document forgery case exposed in 2015 involving four PPA employees [415, 416].

7. DISCUSSION AND RECOMMENDATIONS

In this work we have enumerated a list of challenges that the Estonian ID card and its ecosystem have experienced over the years. In this chapter we discuss broader reasons behind why they happened and provide some specific and less specific recommendations that, in our opinion, would improve the security management of the ID card and its ecosystem. In the last section of this chapter we list some open issues that would benefit from a solution.

7.1. Audits and security certifications

Currently, the ID card security management mainly relies on the mandatory CA audits and the security certifications of ID card platforms. However, this work has highlighted several shortcomings with these security mechanisms. We discuss these shortcomings below.

In a traditional PKI, the role of the CA is to issue certificates, keeping an audit record which proves that the identity of the subject, to whom a certificate is issued, has been verified, and that the public key included in the certificate is the public key that the subject has asked to bind to their identity. These audit records help to solve disputes, keeping CA accountable and deterring CA from abusing their position of trust. However, in the case of the ID card, the accountability of a CA is impaired as it is the CA itself that generates keys[1] for the cardholders and chooses which public key will be bound to a cardholder's identity. As a result, a very high level of trust is granted to the CA, without the ability to verify that the CA behaved properly using the traditional audit measures.

We find that the CA compliance audits that are required by eIDAS are superficial in their nature and only provide a very limited level of assurance. It is best illustrated by the fact that the manufacturer's malpractice of generating private keys outside the ID card was discovered neither in the CA's internal nor the external audits over the 5-year period while it happened. Furthermore, since the audits are episodic, they cannot provide any guarantees for what occurs between the audits. There is also an inherent confict of interest, as the auditor is chosen by the CA itself. The CA, of course, would be interested in purchasing auditing services from an auditor that would provide a positive audit report with the least effort required. We see that a number of intentional non-compliances made by the CA were made with the auditor's approval (Section 6.6.2.2 and 6.7.3).

According to article 20(2) of eIDAS, the supervisory body may at any time audit or request a conformity assessment at the expense of the CA. The state, however, takes a passive role, generally only relying on the reports of the mandatory biennial conformity assessment audits.

[1] While in practice the keys are generated by the ID card manufacturer, in the context of eIDAS, key management is the responsibility of the CA.

Recommendation 1: In case of non-compliance the state should actively exercise their supervisory function by performing or requiring ad hoc CA conformity assessment audits.

The security certification is commonly used in the smart card industry to provide assurance that the product meets certain security requirements. However, the ITSEC certification of the MICARDO secure channel and the CC certification of Infineon's flawed RSA key generation algorithm show that the certification process does not provide a complete guarantee against security flaws. We see that the certification bodies are willing to certify products that use proprietary cryptographic algorithms and in the event that flaws are found, information about them fails to reach the end-users. The certification bodies are not allowed to revoke security certificates of known flawed products, but the issuance of a maintenance report requires the vendor's initiative. As we saw from the cases with the MICARDO, jTOP SLE66 and IDEMIA ID card platforms, in practice it is also a challenge to ensure that the certified version of the product is the one that ends up in the ID card.

While the security evaluation performed in the certification process increases the product's security, it limits the technical flexibility. The eIDAS requirement that only certified products can be used will likely limit the use of innovative technical solutions such as the remote applet updating solution that has been successfully used in Estonia.

7.2. Supervision and legal compliance

The state currently does not supervise the security of ID card manufacturing process. The manufacturing contract requires the delivery of a secure product, leaving the security aspects to the ID card manufacturer. The state assumes that the fines imposed by the manufacturing contract for delivering a defective product will motivate the ID card manufacturer to apply the best security measures.

We see that this assumption does not hold in practice. The ID card manufacturer did not apply the basic security practice of diversifying the GlobalPlatform ISD keys (Section 5.4.2.2) and in 2011 shipped a flawed JavaCard applet whose security was not reviewed. Furthermore, imposing fines in the event of non-compliance is not an effective measure, as a security breach endangering national security, such as the copying of a cardholder's private keys, may leave no visible trace. We only discovered it by accident due to the manufacturer's failure to deliver unique private keys.

Recommendation 2: The state should take an active role in supervising the security aspects of the ID card manufacturing process by requiring the implementation of security measures that are transparent and publicly verifiable.

Preferably, the ID card technological solution should be redesigned to involve threshold cryptography [391–393]. This would decrease the risk of accidental

failures and ensure that intentional malice would require higher conspiracy, hence increasing the risk of detection and attribution.

Recommendation 3: The state should look for an ID card technological solution that is more security fault tolerant (e.g., involves threshold cryptography).

It is necessary to make the ID card personalization protocols and security procedures public. While this alone would not provide a way of verifying whether the manufacturer followed them, this would allow the public to assess whether the applied security measures are adequate.

It is common to hear that information about security measures has to be kept secret due to security concerns. We note that if a security measure can be broken by just knowing what the security measure is, such a measure cannot provide the high level of security which is needed here. Such an approach of "security through obscurity" contrasts with the widely accepted Kerckhoffs's assumption and Shannon's maxim that "the enemy knows the system" and hence the security of a system should not rely on the secrecy of its design. Usually, the actual reason for not disclosing the measures is that there are no measures or they are not documented, but if they are, then the measures are so poor that the responsible party is too ashamed to disclose them to the public.

Another benefit of having the measures being made public is that it forces the involved parties to commit to these procedures, making them aware of their illegitimate actions in case the procedures are not followed. The trustworthiness of the parties can then be already called into question when insignificant deviations from the protocol are found without the need to wait until non-compliances resulting in a tangible security impact occur.

Recommendation 4: The state should require the ID card manufacturer and other involved parties to publish a detailed description of security procedures and measures.

In the course of this research, we have observed numerous legal non-compliances on different levels.

One could argue that the ability to bend the rules provides some advantages. For example, the use of ID card platforms whose SSCD compliance had not been assessed have allowed the state to experiment with new ID card platforms (the NFC-enabled Digi-ID), and to quickly mitigate ID card security flaws (the ROCA flaw). By not revoking the ID card platforms that did not satisfy SSCD requirements (the 2011 ID cards and the cards affected by the ROCA flaw), the state was able to make a reasonable decision to sustain the digital society at the expense of increased security risks.

On the other hand, this loose attitude towards the compliance requirements has created an atmosphere where the requirements (including security) are arbitrary and left up to an individual. The decision by the authorities to approach their duty of SSCD conformance assessments as only a formality, resulted in the ID card manufacturer shipping flawed 2011 ID cards with an applet whose security had

not been reviewed. Since no serious consequences followed, this created a feeling of impunity, which reached its extreme when the ID card manufacturer made an intentional decision to not follow the security requirements and generated ID card private keys outside the card.

We have observed a number of cases where the ID card certificates were not compliant with the certificate policies. Compliance violations are also frequent issues among web browser CAs [420]. Browser vendors enforce strict policy compliance by requiring CAs to revoke affected certificates even when seemingly unimportant non-compliances are found. Unfortunately, applying this model in the context of the Estonian ID card is problematic, as cardholders will blame the state who provided a faulty ID card to them instead of the CA that provided a defective service. We see that there is a conflict of interest as the state is expected to supervise a CA scheme where the state itself is a stakeholder. This would explain why the authorities do not even require the revocation of certificates whose non-compliance result in a security risk (e.g., the case of certificates with an incorrect `@eesti.ee` email address). The problem, however, is that this approach does not motivate the parties to avoid non-compliances. As we saw from the incident where certificates had incorrectly encoded public keys (Section 6.6), a seemingly unimportant non-compliance can one day became a serious issue.

Throughout this work we have observed a list of quality issues related to the ID card manufacturing process. In several cases, the involved parties failed to find and fix the root cause of the problem, which led to the same or a similar issue to occur again. For instance, over the years there have been recurring issues with the encoding of RSA public keys, handling of @eesti.ee email addresses in the certificates and revocation of the certificates of deceased cardholders. The case of the duplicate RSA public keys showed that the same issue had to be dealt with repeatedly. The case of the corrupted RSA public keys showed that the warranty ID cards were issued to cardholders without finding and fixing the cause of the flaw and without realizing the security impact of the flaw.

This shows a systematic failure to sufficiently investigate incidents when they occur. To ensure that every incident and quality issue related to the manufacturing of ID cards is sufficiently investigated, the authorities should require detailed incident reporting. For example, browser vendors require CAs to publish detailed reports, describing how the incident was detected, the security impact, root causes of the problem and what has been done to make sure that it does not happen again [421, 422]. It is also important to ensure that the quality assurance procedures developed in this process are transferred to successors, because otherwise it may occur as in the case of transparent PIN envelopes, where the exact same flaw was reintroduced 16 years later by another ID card manufacturer.

Recommendation 5: The state should require the involved parties to publish detailed incident reports, describing the root cause analysis and the measures taken to prevent it from happening again.

To be able to exercise its supervisory function, the state should avoid practices that increase their dependence on a particular ID card manufacturer or CA. For instance, the state increased its dependence on the ID card manufacturer when it granted the ID card manufacturer the manufacturing contract for two terms without competition. The authorities' decision to cover up the manufacturing flaw in the 2011 ID cards increased this interdependence even more.

The latest ID card manufacturing procurement suggests that the authorities may have learned a lesson. We hope that the state's decision to only provide ID card-related services in PPA customer service points was made to decrease the dependence on SK, in the event the certification services would one day have to be provided by a different CA. An idea worth considering, is to give the smart card chip manufacturing to a manufacturer other than the one manufacturing and personalizing the plastic document.

The ID card crisis in 2017 also showed that the reliance on a single, monocultural eID solution backed by a single CA creates a risk that endangers the sustainability of the Estonian digital society.

Recommendation 6: The state should strive to decrease the dependence on the ID card manufacturer and CA by encouraging an open competition and by diversifying the technical solutions and their suppliers.

So far we have not seen fundamental changes in the organization and execution of the ID card manufacturing process, therefore incidents like the ones described in this work, in one form or another, are likely to occur again. We hope, however, that the public knowledge of these incidents have changed the perception of the ID card as being infallible. This should now allow the construction of better security systems and legal rules which are able to deal with potential security failures of the ID card. A good example here is Swedbank's internetbank, where an additional authentication factor (a relatively unpredictable user ID) is required for ID card authentication.

Recommendation 7: Security-critical systems that rely on the ID card should be built with the assumption that someone may have a database containing the private keys of all Estonian residents.

7.3. Research and expert opinion

It is positive to see that the authorities are seeking expert opinions by procuring regular Cryptographic Algorithms Lifecycle reports [96, 121, 423–425]. However, there have been ID card-related developments where the expert opinion and analysis have been missing. For instance, the ID card's 3DES passphrase authentication feature was enabled and reimplemented in later ID card platforms without analyzing the security implications and its usefulness.

The decision to generate ID card keys with a random RSA public exponent was made without consulting cryptographers. This caused serious ID card key generation performance issues and resulted in the ID card manufacturer's decision to breach the security requirements by generating keys outside the ID card.

The decision to introduce the timestamping application TeRa was made without understanding the implications of the weaknesses of the SHA-1 algorithm. As a result, significant resources were spent by the public and private sector trying to solve a problem that did not exist (the risk of a second-preimage attack against SHA-1). At the same time, the authorities still have not completely deprecated SHA-1 as the DDOC signatures created using SHA-1 are still accepted by the state-supplied digital signature validation software.

Recommendation 8: The authorities should gather and consider a wide range of expert opinions before deciding on eID-related developments.

We appreciate the state's decision to develop the ID card software using the open-source development model, as it provides transparency and supports contributions from the wider public. However, the development of the EstEID chip application has always been a non-transparent activity even after RIA became the holder of its intellectual property rights. We invite the authorities to promote the use of open eID technological solutions, at a minimum, providing open access to the ID card JavaCard platform such that at least black-box testing of the platform could be performed. We note that the discovery of Infineon's RSA key generation flaw was only possible because the Czech researchers had somehow obtained a sample of the affected JavaCard platform and could experiment with it.

The fact that during this research we had to go to such great lengths to establish basic facts (e.g., which chips were used and their capabilities), shows that more information should be placed in the public domain to support research in this field.

Recommendation 9: In the eID-related procurements, the state should give preference to open and transparent eID solutions and should strive to publish as much technical documentation as possible.

7.4. Transparency about security issues

Today, the private sector relies on the ID card to secure transactions that are worth millions, the ID card authentication is used to protect loads of sensitive personal data, and the ID card is used by citizens to exercise their constitutional right to elect the parliament. In this situation, the security of the ID card and the eID in general is not the authorities' internal issue, but the personal concern of the entire digital society.

We have observed a number of cases where the state has not been honest with the public. The most serious deception was in the communication of the 2011 ID card incident. The situation improved greatly in 2017 with the disclosure of the ROCA flaw, but even there the authorities hid the CPU-year figures of the attack and the details about the successful factorization attack performed later in 2018.

We note that the public would not have known about the incidents of certificates with duplicate RSA public keys and certificates with corrupted RSA

public keys had we not ourselves discovered this in the certificate data. Hence, the list of ID card-related security incidents covered in this work may just be the tip of the iceberg, with the other incidents being kept hidden from the public.

Complete transparency about the security issues concerning the ID card is needed, as it is the only way the involved parties can learn from the incidents and the public and private sector can conduct appropriate risk assessments. This would also support research as it would allow researchers to focus on real world issues making the research more beneficial for society.

There is a natural desire for the authorities to sweep the eID-related security issues under the carpet as there is a fear that this will damage the reputation of the state. However, it is a trap, as it only serves as a short-term reputation management strategy. We hope that the Estonian society has reached the maturity level where the digital society-related security issues can be discussed in the open.

Recommendation 10: The state should release detailed information about the flaw in the 2011 ID cards and all other security incidents that have not been made public. The state should define a clear transparency policy for security incidents with the nation's eID system.

There are worries in the Estonian society that certain government agencies may hold a copy of the ID card private keys [426]. As our findings have shown, this can be technically done with only a handful of people being aware of it.

There is also a concern that the state may compel the CA to issue false certificates [427] that can be used by the intelligence agencies to covertly intercept encrypted communications and impersonate cardholders. The Estonian law does not explicitly forbid this.

The law does not foresee imposing criminal liability for knowingly mishandling the security of the ID card. As we saw from the aftermath of the incident where key were generated outside the ID card, the liability for the ID card manufacturer and its contractors is rather limited – in the worst case only the breach of contract action can be brought against the ID card manufacturer.

Recommendation 11: To increase the trustworthiness of the state-issued eID solutions, the state should declare a transparent, backdoor-free policy for its technologies. Furthermore, the law should foresee introducing criminal liability for knowingly issuing false certificates and knowingly mishandling the security of cardholders' private keys (e.g., holding copies of the private key or generating weak keys). The ideas from Certificate Transparency[2] should be implemented to make it very difficult for a CA to issue a certificate without it being visible to the certificate holder.

[2] `https://www.certificate-transparency.org/`

7.5. Other open issues

In this work we have touched upon a list of open issues related to the ID card ecosystem that would benefit from a solution. We summarize them below.

1. The current ID card-based encryption solution (CDOC) does not provide a forward secrecy feature, meaning that confidentiality for the encrypted data is only provided as long as the cardholder's private key is not compromised (Section 2.9).
   Recommendation 12: The ID card encryption solution should be redesigned to provide confidentiality guarantees even after the cardholder's private key gets compromised.
2. The current use of the ID card for authentication to a machine provides very little security as the authenticity of the data in the personal data file is not cryptographically protected (Section 2.12).
   Recommendation 13: The state should consider equipping the ID card with a cryptographically secure method for using it as a physical authentication token.
3. The ID card is not protected against malware attacks even if a smart card reader with a PIN pad and PIN firewall is used (Section 2.14.1). Currently, we have not seen any malware exploiting this weakness, but this may change any day.
   Recommendation 14: The next generation eID solutions should work towards solving this weakness (e.g., by introducing a next generation ID card that has a built-in trusted display).
4. The ID card crisis in 2017 highlighted the fact that it is not clear who bears liability for the security of the ID cards private keys.
   Recommendation 15: The legal framework should be updated to provide clear answers to the liability questions of ID card security.
5. The current certificate suspension mechanism creates a security issue when suspension is requested by a party other than the certificate holder (Section 6.7.3). Furthermore, the possibility of certificate suspension allows the validity of any digital signature created with the Estonian ID card to be challenged (Section 2.10.3).
   Recommendation 16: The certificate suspension mechanism should be deprecated and the revocation mechanism should be adjusted to accomplish the same purpose.
6. Signing using the ID card browser extension provides a very poor evidentiary value as the signatory is not able to see what is being signed (Section 2.10).
   Recommendation 17: The ID card browser signing extension should be redesigned to show the content that a service provider is requesting to be digitally signed.

7. A complete deprecation of SHA-1 in the context of the digital signature format DDOC has still not occurred.
   Recommendation 18: The digital signature validation software should refuse to validate recently created digital signatures that use the DDOC format.
8. Currently there is no legally binding technical standard that would precisely describe the validation rules of a digital signature.
   Recommendation 19: To provide legal certainty for digital signature validity the authorities should actively engage in the creation of a digital signature validation standard that complies with the digital signature legal requirements set by eIDAS.

8. CONCLUSIONS

In this work, we have shown that over the two decades of the existence of the Estonian ID card, the ID card and its ecosystem have experienced various faults and security issues in different parts of the ecosystem.

We put forward three concluding statements that should summarize the answer to our work's second research question on how sufficiently the technical, organizational and managerial mechanisms address the ID card related security challenges:

1. *Security audits and security certifications of the ID card components are not able to guarantee security and enforce compliance.* The findings of this work have shown that the audit mechanisms used have catastrophically failed, as they could not discover serious non-compliances in the ID card personalization process over the 5 years while it happened. Similarly, we have seen that the security certifications of the ID card components are not enough to guarantee that security flaws are not present in these components. We propose that instead of relying on organizational security measures that are difficult to audit and enforce, the security of the ID card should rely on a more fault-tolerant design. Such designs can be invented and implemented but require further research and interest from the involved parties.
2. *There is a systematic failure to sufficiently investigate incidents and learn the lessons.* The findings of this work show that the involved parties have repeatedly failed to sufficiently investigate incidents to understand their security impact, scale, root causes, and implement measures that would prevent similar incidents in the future. The incident reporting (if it is present at all) is not transparent. Furthermore, the impartiality of the state's supervisory function is greatly encumbered as the state is also directly responsible for the development of the solutions under its supervision.
3. *Greater transparency, openness and expert involvement is needed.* Throughout this research, we have encountered difficulties obtaining some rather basic technical information, especially on issues concerning the card. Similarly, the state has not released information regarding the ID card incident in 2011 and other security issues that have not emerged to the public. We have seen that in some cases, the involved parties rely on the flawed "security by obscurity" approach, which eventually makes life more difficult for the defenders rather than the attackers. We have also found that some ID card related developments have missed sufficient expert opinion and security analysis, which has introduced security issues. We call for greater transparency and openness, as it can increase security by encouraging research in this field and allowing the participants of the ecosystem to learn from the incidents and thereby conduct more appropriate risk assessments.

BIBLIOGRAPHY

[1] Estonian Information System Authority. ID card usage statistics inferred from queries to the OCSP service, 2019.

[2] Thomas W. Edgar and David O. Manz. Part II. Observational Research Methods. In Thomas W. Edgar and David O. Manz, editors, *Research Methods for Cyber Security*, page 93. Syngress, 2017.

[3] ID Süsteemide AS. EstEID Secure token (smartcard) application and interface v1.0, June 7, 2001. http://cybersec.ee/storage/20010607_esteid_specification_v100.rtf.

[4] ID Süsteemide AS. EstEID card specification v2.01 (in Estonian), November 20, 2002. http://www.id.ee/public/EstEID_Spetsifikatsioon_v2.01.pdf.

[5] ID Süsteemide AS. EstEID card application manual (in Estonian), November 1, 2003. https://www.id.ee/public/EstEID_kaardi_kasutusjuhend.pdf.

[6] Estonian Centre for Standardisation. EVS 827:2004 – Security chip – Application and interface, 2009. https://www.evs.ee/products/evs-827-2004.

[7] Trüb Baltic AS. EstEID card application notes – part 1, November 29, 2010. http://www.id.ee/public/EstEID_card_application_notes_part_1-2011.pdf.

[8] Trüb Baltic AS. EstEID v3.4 card specification v1.0, June 11, 2012. https://cybersec.ee/storage/20120611_TB-SPEC-EstEID-Chip-App-v3.4.pdf.

[9] Trüb Baltic AS. EstEID v3.4 card specification v1.1, September 12, 2013. http://www.id.ee/public/TB-SPEC-EstEID-Chip-App-v3.4.pdf.

[10] Trüb Baltic AS. EstEID v3.5 card specification v1.20, March 27, 2014. https://www.id.ee/public/TB-SPEC-EstEID-Chip-App-v3_5-20140327.pdf.

[11] Trüb Baltic AS. EstEID v3.5 card specification v1.30, March 14, 2017. http://www.id.ee/public/TB-SPEC-EstEID-Chip-App-v3.5-20170314.pdf.

[12] Estonian Information System Authority. EstEID v3.5.8 card specification v1.40, October 18, 2017. https://www.id.ee/public/RIA-EstEID-Chip-App-v3.5.8_fix_form.pdf.

[13] Estonian Information System Authority. Estonia ID1 Chip/App 2018 – Technical Description, V0.9, July 1, 2020. https://installer.id.ee/media/id2019/TD-ID1-Chip-App.pdf.

[14] SK ID Solutions AS. CA documentation repository, December 2020. https://www.skidsolutions.eu/en/repository/.

[15] Arnis Parsovs. Practical Issues with TLS Client Certificate Authentication. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2014. http://dx.doi.org/10.14722/ndss.2014.23036.

[16] Danielle Morgan and Arnis Parsovs. Using the Estonian Electronic Identity Card for Authentication to a Machine. In Helger Lipmaa, Aikaterini Mitrokotsa, and Raimundas Matulevičius, editors, *Secure IT Systems*, pages 175–191, Cham, 2017. Springer International Publishing.

[17] Tõnu Mets and Arnis Parsovs. Time of signing in the Estonian digital signature scheme. *Digital Evidence and Electronic Signature Law Review*, 16:40–50, 2019. https://journals.sas.ac.uk/deeslr/article/view/5076.

[18] Arnis Parsovs. Solving the Estonian ID Card Crisis: the Legal Issues. In Amanda L. Hughes, Fiona McNeill, and Christopher Zobel, editors, *Proceedings of the 17th International Conference on Information Systems for Crisis Response and Management ISCRAM 2020*, pages 459–471, Blacksburg, VA, May 2020.

[19] Arnis Parsovs. Estonian Electronic Identity Card: Security Flaws in Key Management. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1785–1802. USENIX Association, August 2020.

[20] Tarvi Martens. Electronic identity management in Estonia between market and state governance. *Identity in the Information Society*, 3(1):213–233, July 2010. https://doi.org/10.1007/s12394-010-0044-0.

[21] Tarvi Martens. The Story of the Estonian ID Card. In *Baltic IT&T Review #24*, January 2002. http://www.ebaltics.com/doc_upl/Martens(2).pdf.

[22] Olev Sepp. ID card program development (in Estonian), October 1, 2002. https://web.archive.org/web/20021001102839/http://id.ee/programmiareng.html.

[23] AS Aprote. ID card preliminary survey report (in Estonian), 1998. https://www.id.ee/public/aptaruanne.pdf.

[24] Küberneetika AS. Preliminary study of Estonian ID card technology (in Estonian), 1998. https://www.id.ee/public/cyberaruanne_20070322030353.pdf.

[25] Küberneetika AS. Application instructions for carrying out ID-card pilot projects (in Estonian), 1999. https://www.id.ee/public/rakendusjuhised.pdf.

[26] Küberneetika AS. Estonian draft standards in the field of EID cards (in Estonian), 1999. https://www.id.ee/public/idkaardistandardid.pdf.

[27] Küberneetika AS. Overview of international standards for identification cards (in Estonian), 1999. https://www.id.ee/public/rahv_std_aruanne.pdf.

[28] Arnis Parsovs. *Security Analysis of Internet Bank Authentication Protocols and their Implementations*. MSc thesis, Tallinn University of Technology, 2012. https://kodu.ut.ee/~arnis/bankauth/thesis.pdf.

[29] Estonian Information System Authority. As of 1 March, it will no longer be possible to access certain public e-services via a bank link, February 10, 2021. https://www.ria.ee/en/news/1-march-it-will-no-longer-be-possible-access-certain-public-e-services-bank-link.html.

[30] Ohtuleht.ee. ID card half-mandatory (in Estonian), November 9, 2001. https://epl.delfi.ee/artikkel/50811395/id-kaart-poolkohustuslikuks.

[31] Riigi Teataja. Identity Documents Act – RT I, 21.04.2018, 5. English translation, 1999. https://www.riigiteataja.ee/en/eli/521062017003.

[32] Riigi Teataja. Form and technical description of the identity card and list of data to be entered on the identity card and period of validity of digital data – RT I, 09.11.2018, 5 (in Estonian), 2018. https://www.riigiteataja.ee/akt/109112018005.

[33] Riigi Teataja. Form and technical description of the residence permit card and list of data to be entered on the residence permit card and period of validity of digital data – RT I, 09.11.2018, 4 (in Estonian), 2018. https://www.riigiteataja.ee/akt/109112018004.

[34] Riigi Teataja. Form and technical description of the digital identity card and list of data to be entered on the digital identity card – RT I, 09.11.2018, 3 (in Estonian), 2018. https://www.riigiteataja.ee/akt/109112018003.

[35] Riigi Teataja. Procedure for Issuance and Revocation of a Diplomatic Identity Card, Form, Technical Description and List of Data to be Entered in a Diplomatic Identity Card and Procedure for Registration of Non-Residents Exempt from Income Tax – RT I, 24.11.2018, 6 (in Estonian), 2018. https://www.riigiteataja.ee/akt/124112018006.

[36] Police and Border Guard Board. Electronic machine readable travel document (eMRTD) Public Key Infrastructure (PKI) of the Republic of Estonia, November 3, 2020. https://www.politsei.ee/en/electronic-machine-readable-travel-document-emrtd-public-key-infrastructure-pki-of-the-republic-of-estonia.

[37] Delfi.ee. The Estonian company fell out of the competition to manufacture ID cards (in Estonian), September 21, 2000. https://epl.delfi.ee/artikkel/50841294/eesti-firma-langes-id-kaartide-tootja-konkursil-valja.

[38] Delfi.ee. ID card base price: production line (in Estonian), October 22, 2001. http://www.delfi.ee/news/paevauudised/eesti/id-kaardi-pohjamineku-hind-trukiliin?id=2348259.

[39] Delfi.ee. ID cards ready for printing (in Estonian), October 24, 2001. https://arileht.delfi.ee/artikkel/50810719/id-kaardid-trukivalmis.

[40] Delfi.ee. Yesterday the ID card opened the era of new documents (in Estonian), January 29, 2002. https://epl.delfi.ee/artikkel/50907758/id-kaart-avas-eile-uute-dokumentide-ajastu.

[41] Delfi.ee. By the end of the year there will be 100 000 ID card users in Estonia (in Estonian), June 10, 2002. https://arileht.delfi.ee/artikkel/50926152/aasta-lopuks-on-eestis-100-000-id-kaardi-kasutajat.

[42] Delfi.ee. Overpaid by 100 million (in Estonian), June 30, 2004. https://arileht.delfi.ee/artikkel/8116658/100-miljoniga-ule-makstud.

[43] Delfi.ee. ID card may start serving the banks (in Estonian), November 22, 2001.

[44] Delfi.ee. Hansapank: our cards do not come from the same production line as the ID cards (in Estonian), November 23, 2001. https://epl.delfi.ee/artikkel/50812039/hansapank-meie-kaardid-ei-tule-id-kaartidega-uhelt-liinilt.

[45] Delfi.ee. The ID-card manufacturer received a new contract without tender (in Estonian), October 30, 2006. https://epl.delfi.ee/artikkel/51062104/id-kaartide-tootja-sai-uue-lepingu-konkursita.

[46] Postimees. Will it be different this time? The state has so far procured ID cards from only one company (in Estonian), February 10, 2016. https://tehnika.postimees.ee/3577961/kas-sel-korral-laheb-teisiti-riik-on-seni-tellinud-id-kaarte-vaid-uhelt-ettevottelt.

[47] Ohtuleht.ee. New ID cards have a shorter validity period and higher price (in Estonian), March 20, 2011. https://www.ohtuleht.ee/419123/uutel-id-kaartidel-luhem-kehtivusaeg-ja-kallim-hind.

[48] Delfi.ee. Competitors suspect unfair game in the 40 million euro PPA procurement (in Estonian), May 30, 2016. http://arileht.delfi.ee/news/uudised/konkurendid-kahtlustavad-40-miljoni-eurose-ppa-hanke-juures-valemangu?id=74676029.

[49] Postimees. Large companies sued the Estonian police in court (in Estonian), June 8, 2017. https://tehnika.postimees.ee/4140063/laane-suurettevotted-kaebasid-eesti-politsei-kohtusse.

[50] ERR News. PPA signs deal with France's Oberthur to produce IDs beginning 2019, April 28, 2017. http://news.err.ee/592722/

ppa-signs-deal-with-france-s-oberthur-to-produce-ids-beginning-2019.

[51] ERR News. Court rejects current ID card manufacturer's appeal against Estonia, December 29, 2017. https://news.err.ee/651254/court-rejects-current-id-card-manufacturer-s-appeal-against-estonia.

[52] Postimees. The World Bank imposed a boycott on the next Estonian ID card manufacturer (in Estonian), December 27, 2017. https://tehnika.postimees.ee/4356255/maailmapank-kehtestas-boikoti-eesti-id-kaardi-jargmisele-tootjale.

[53] Postimees. Gemalto and PPA are carrying tens of millions after the conflict (in Estonian), March 8, 2018. https://tehnika.postimees.ee/4432811/gemalto-ja-eesti-politsei-veavad-kumnete-miljonite-parast-vagikaigast.

[54] Postimees. Gemalto appeals to the Supreme Court to cancel the ID card procurement (in Estonian), April 30, 2019. https://majandus24.postimees.ee/6648601/gemalto-nouab-riigikohtus-edasi-id-kaardi-hanke-voidu-tuhistamist.

[55] Aripeav. ID card manufacturer closes doors in Estonia (in Estonian), March 6, 2019. https://www.aripaev.ee/uudised/2019/03/06/id-kaartide-tootja-paneb-eestis-uksed-kinni.

[56] ABI research. Smart card & secure ICs, October 2018.

[57] Delfi.ee. Eesti Telekom creates public key infrastructure (in Estonian), March 22, 2000. http://arileht.delfi.ee/news/uudised/eesti-telekom-loob-avaliku-votme-infrastruktuuri?id=50823824.

[58] Geenius. Lithuanian media: The state simply gave the mobile signature market to Estonians (in Estonian), November 15, 2018. https://digi.geenius.ee/rubriik/uudis/leedu-meedia-riik-kinkis-mobiiliallkirja-turu-eestlastele/.

[59] Estonian Banking Association. Press release: Internet bank password card payment limits will be reduced today (in Estonian), May 18, 2009. https://www.pangaliit.ee/uudised-ja-teated/aktuaalset-2009-uudis3.

[60] Mai-Liis Palginõmm. *Diffusion of the Estonian ID-Card and Its Electronic Usage: Explaining the Success Story*. MSc thesis, Tallinn University of Technology, 2016. https://digi.lib.ttu.ee/i/?5674.

[61] The European Parliament and the Council of the European Union. Regulation 910/2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC, 2014. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.257.01.0073.01.ENG.

[62] Riigi Teataja. Electronic Identification and Trust Services for Electronic Transactions Act – RT I, 25.10.2016, 1. English translation, 2016. https://www.riigiteataja.ee/en/eli/527102016001.

[63] Riigi Teataja. Digital Signatures Act – RT I 2000, 26, 150. Repealed 26.10.2016. English translation of last wording in force before being repealed., 2000. https://www.riigiteataja.ee/en/eli/508072014007.

[64] The European Parliament and the Council of the European Union. Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures, 2000. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:31999L0093.

[65] SK ID Solutions AS. Terms and Conditions for Use of Certificates of Personal Identification Documents of the Republic of Estonia, July 1, 2018. https://sk.ee/upload/files/SK-TCU-ESTEID-EN-20180701.pdf.

[66] ID Help Centre. Compete information about the terms of use of certificates, January 21, 2021. https://www.id.ee/en/article/compete-information-about-the-terms-of-use-of-certificates/.

[67] Estonian Cyber Security News Aggregator. SK ID Solutions declared provider of vital services, July 3, 2017. https://cybersec.ee/2017/07/03/sk-id-solutions-declared-provider-of-vital-services/.

[68] Riigi Teataja. The description and requirements for ensuring the continuity of digital identification and digital signing as a vital service – RT I, 15.01.2019, 11. English translation, 2019. https://www.riigiteataja.ee/en/eli/510102019001.

[69] SK ID Solutions AS. SK ID Solutions AS – Certification Practice Statement for EE-GovCA2018, October 1, 2018. https://www.sk.ee/upload/files/SK-CPS-EE-GovCA2018-EN-v1_0-20181001.pdf.

[70] PPA. Certificate Policy for identity card, digital identity card, residence permit card and diplomatic identity card, Version 1.0, October 1, 2018. https://www.id.ee/public/CP_ESTEID_01.10.2018_version1.0.pdf.

[71] AS Sertifitseerimiskeskus. SK additional services portfolio updates (in Estonian), November 3, 2016. https://www.sk.ee/upload/files/2_Lisateenuste%20portfelli%20uuendused_Liisa%20Lukin_AK2016.pdf.

[72] ID Help Centre. For organisations that sign large quantities of documents using DigiDoc4 Client and RIA DigiDoc mobile application, January 21, 2021. https://www.id.ee/en/article/for-organisations-that-sign-large-quantities-of-documents-using-digidoc4-client/.

[73] ERR News. Nearly half a million people using Smart-ID in Estonia, December 30, 2019. https://news.err.ee/1018838/nearly-half-million-people-using-smart-id-in-estonia.

[74] AS Sertifitseerimiskeskus. SK customer service points (in Estonian), March 21, 2008. http://web.archive.org/web/20080321033024/http://www.sk.ee/pages.php/0202010302.

[75] ERR News. Banks to cease issuing ID Card PINs, February 1, 2019. https://news.err.ee/906671/banks-to-cease-issuing-id-card-pins-as-well-as-pass-code-cards.

[76] Estonian Information System Authority. ROCA Vulnerability and eID: Lessons Learned, May 2018. https://www.ria.ee/sites/default/files/content-editors/kuberturve/roca-vulnerability-and-eid-lessons-learned.pdf.

[77] Estonian Information System Authority. Electronic Identity Application Guide: Securing e-mail, VPN-s and logging into the desktop, May 2014. https://eid.eesti.ee/index.php/EID_email_etc.

[78] Riigi Teataja. Riigikogu Election Act – RT I, 25.10.2016, 21. English translation. https://www.riigiteataja.ee/en/eli/527102016003.

[79] Official Journal of the European Union. Electronic identification schemes notified pursuant to Article 9(1) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market (2018/C 401/08), November 7, 2018. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.C_.2018.401.01.0007.01.ENG.

[80] Applied Cyber Security Group, University of Tartu. List of Estonian e-services using eID, November 30, 2020. https://acs.cs.ut.ee/services_using_eid/.

[81] Timo Toots. Memopol-1 installation, 2010. https://www.timo.ee/memopol1/.

[82] Arnis Parsovs. SEB Estonia Internet bank ID card authentication bypass, August 25, 2015. https://www.youtube.com/watch?v=rRB8jZnS5nY.

[83] Arnis Parsovs. Swedbank Estonia Internet bank ID card authentication bypass, August 3, 2016. https://www.youtube.com/watch?v=5mWMu0KinFg.

[84] Semjon Kravtšenko. Coop Pank Internet bank ID card authentication bypass, February 16, 2021. https://www.youtube.com/watch?v=cObPmkK7zaY.

[85] Semjon Kravtšenko. elisa.ee, printincity.ee, arved.ee: ID card authentication bypass, February 18, 2021. https://www.youtube.com/watch?v=IQ5UK2VwN4w.

[86] Mart Sõmermaa. Web eID: electronic identity cards on the Web, June 3, 2019. https://github.com/web-eid/web-eid-system-architecture-doc.

[87] ID Help Centre. Encryption and decryption of documents, August 12, 2020. https://www.id.ee/en/article/encryption-and-decryption-of-documents/.

[88] Riigi Teataja. Procedure for formalizing, forwarding and preservation of digitally signed and other digital documents in offense proceedings – RT I, 11.02.2016, 8 (in Estonian), 2016. https://www.riigiteataja.ee/akt/111022016008.

[89] Estonian Police and Border Guard Board. Warning fine 223012001507 sent from domain: politsei.ee (in Estonian), January 19, 2012. http://www.kalale.ee/foorum/arutelu/6D5A?page=3800.

[90] Toomas Lepik, Jaan Priisalu, Anna-Maria Osula, and Sten Mäses. Preliminary analysis to find the optimal solution for the secure e-mail exchange system for Estonian public authorities (in Estonian), 2016. https://www.ria.ee/sites/default/files/content-editors/publikatsioonid/eelanaluus-riigiasutuste-e-kirjavahetussusteem.pdf.

[91] Mart Oruaas and Jan Willemson. Developing Requirements for the New Encryption Mechanisms in the Estonian eID Infrastructure. In Tarmo Robal, Hele-Mai Haav, Jaan Penjam, and Raimundas Matulevičius, editors, *Databases and Information Systems*, pages 13–20, Cham, 2020. Springer International Publishing.

[92] AS Sertifitseerimiskeskus. Encrypted DigiDoc Format Specification, Document Version 1.1, June 25, 2012. https://www.id.ee/public/SK-CDOC-1.0-20120625_EN.pdf.

[93] D. Eastlake and J. Reagle. W3C Recommendation 10 December 2002: XML Encryption Syntax and Processing, 2002. https://www.w3.org/TR/xmlenc-core/.

[94] Daniel J. Bernstein. Break a dozen secret keys, get a million more for free, November 20, 2015. https://blog.cr.yp.to/20151120-batchattacks.html.

[95] Cybernetica AS. Required modifications to CDOC for elliptic curve support, September 27, 2017. https://www.ria.ee/sites/default/files/content-editors/EID/cdoc.pdf.

[96] Cybernetica AS. Cryptographic Algorithms Lifecycle Report 2017 (in Estonian), February 9, 2018. https://www.ria.ee/sites/default/files/content-editors/publikatsioonid/kruptograafiliste_algoritmide_elutsukli_uuring_2017.pdf.

[97] Ingrid Pappel, Ingmar Pappel, Jaak Tepandi, and Dirk Draheim. *"Systematic Digital Signing in Estonian e-Government Processes"*, pages 31–51. Springer Berlin Heidelberg, Berlin, Heidelberg, 2017.

[98] Delfi.ee. It is safer and cheaper to register a vehicle in the e-service (in Estonian), March 26, 2015. https://maaleht.delfi.ee/tasubteada/e-teeninduses-saab-soiduki-turvalisemalt-ja-soodsamalt-registrisse?id=71103263.

[99] State Electoral Office of Estonia. General Framework of Electronic Voting and Implementation thereof at National Elections in Estonia, June 20, 2017. https://www.valimised.ee/sites/default/files/uploads/eng/IVXV-UK-1.0-eng.pdf.

[100] Riigi Teataja. Code of Civil Procedure – RT I, 19.03.2019, 22. English translation, 2019. https://www.riigiteataja.ee/en/eli/512042019002.

[101] John Allberg. Bug 328346: Certificates with keyusage nonRepudiation should not be used as SSL client certificates. https://bugzilla.mozilla.org/show_bug.cgi?id=328346.

[102] Raul Metsma. hwcrypto.js: Browser JavaScript library for working with hardware tokens, August 30, 2017. https://github.com/hwcrypto/hwcrypto.js/.

[103] European Commission. 2000/709/EC: Commission Decision of 6 November 2000 on the minimum criteria to be taken into account by Member States when designating bodies in accordance with Article 3(4) of Directive 1999/93/EC of the European Parliament and of the Council on a Community framework for electronic signatures, 2000. https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex:32000D0709.

[104] European Commission. 2003/511/EC: Commission Decision of 14 July 2003 on the publication of reference numbers of generally recognised standards for electronic signature products in accordance with Directive 1999/93/EC of the European Parliament and of the Council, 2003. https://eur-lex.europa.eu/legal-content/GA/ALL/?uri=celex:32003D0511.

[105] Ministry of Economic Affairs and Communications. Conformity of secure signature creation device to the requirements of Digital Signatures Act (in Estonian), March 4, 2016. https://cybersec.ee/storage/20160304_MKM_SSCD_conformance_assessment/.

[106] Kristiina Laanest and Laura Kask. Legal Issues of ID Card Security Risk (unofficial translation), October 2017. https://cybersec.ee/storage/RIA_MKM_idcard_risklegal.pdf.

[107] European Commission. Comission Implementing Decision (EU) 2016/650, 2016. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2016.109.01.0040.01.ENG.

[108] European Commission. Compilation of Member States notification on SSCDs and QSCDs, June 3, 2019. https://cybersec.ee/storage/eidas-art.31-list-2019-06-03.pdf.

[109] SK ID Solutions AS. DigiDoc Format Specification. Version 1.3.0, May 12, 2004. http://www.id.ee/public/DigiDoc_format_1.3.pdf.

[110] Tarvi Martens. *Real-life Digital Signatures with Long-Term Validity*, pages 164–168. Vieweg+Teubner Verlag, Wiesbaden, 2004.

[111] SK ID Solutions AS. BDOC – format for digital signatures. Version 2.1.2, 2014. http://www.id.ee/public/bdoc-spec212-eng.pdf.

[112] Official Journal of the European Union. Commission Implementing Decision (EU) 2015/1506, September 8, 2015. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:JOL_2015_235_R_0006.

[113] ID Help Centre. ID-software version information (release notes), December 10, 2019. https://www.id.ee/en/article/id-tarkvara-versioonide-info-release-notes/.

[114] Estonian Centre for Standardisation. EVS 821:2009 – BDOC – Format for Digital Signatures, 2009. https://www.evs.ee/products/evs-821-2009.

[115] Estonian Centre for Standardisation. EVS 821:2014 – BDOC – Format for Digital Signatures, 2014. https://www.evs.ee/products/evs-821-2014.

[116] European Telecommunications Standards Institute. ETSI EN 319 102-1 V1.1.1 (2016-05) Electronic Signatures and Infrastructures (ESI); Procedures for Creation and Validation of AdES Digital Signatures; Part 1: Creation and Validation, May 2016. https://www.etsi.org/deliver/etsi_en/319100_319199/31910201/01.01.01_60/en_31910201v010101p.pdf.

[117] European Telecommunications Standards Institute. Draft ETSI TS 119 172-4 V0.0.7 (2019-08) Electronic Signatures and Infrastructures (ESI); Signature policies; Part 4: Signature validation policy for European qualified electronic signatures/seals using trusted lists, August 2019. https://docbox.etsi.org/esi/Open/Latest_Drafts/draft-ts_119172-4v007-public.pdf.

[118] ID Help Centre. Handling signature validation warnings in DigiDoc libraries, March 20, 2019. https://www.id.ee/en/article/handling-signature-validation-warnings-in-digidoc-libraries-2/.

[119] Arnis Parsovs. Estonian digital signature: modifying the time of signing, November 4, 2019. https://www.youtube.com/watch?v=ysYouhl1Yx4.

[120] Bruce Schneier. SHA-1 Broken, February 2005. https://www.schneier.com/blog/archives/2005/02/sha1_broken.html.

[121] Cybernetica AS. Study on the lifecycle and areas of use of cryptographic algorithms v1.1 (in Estonian), July 15, 2011. https://www.id.ee/wp-content/uploads/2020/01/kryptoalgoritmide_elutsykli_uuring_15-07-2011.pdf.

[122] Eduard Kovacs. New Collision Attack Lowers Cost of Breaking SHA1, October 2015. http://www.securityweek.com/new-collision-attack-lowers-cost-breaking-sha1.

[123] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The first collision for full SHA-1. Cryptology ePrint Archive, Report 2017/190, 2017. http://eprint.iacr.org/2017/190.

[124] Erling Ellingsen. SHA1 collider, February 2017. https://alf.nu/SHA1.

[125] SK ID Solutions AS. SHA-256 hash algorithm was adopted in the certificates of Estonian personal identity documents, February 4, 2015. https://www.sk.ee/en/News/sha-256-hash-algorithm-was-adopted-in-the-certificates-of-estonian-personal-identity-documents.

[126] Estonian Information System Authority. An ID card software update brings several significant changes, June 27, 2017. https://www.ria.ee/en/news/id-card-software-update-brings-several-significant-changes.html.

[127] BalTstamp UAB. BalTstamp time-stamping practice statement. Version 2.1, March 2017. https://www.baltstamp.lt/files/BalTstamp_TSPS_2017-03-31.pdf.

[128] BalTstamp UAB. BalTstamp TSA Disclosure Statement, July 2016. https://www.baltstamp.lt/files/BalTstamp_TSA_disclosure_statement_2016-07-01.pdf.

[129] AS Sertifitseerimiskeskus. ESTEID-SK Certification Practice Statement, Version 1.0, November 1, 2016. https://sk.ee/upload/files/SK-CPS-ESTEID-EN-v1_0_20161101.pdf.

[130] AS Sertifitseerimiskeskus. Certificate Policy for ID card, Version 6.0, November 1, 2016. https://sk.ee/upload/files/SK-CP-ID%20CARD-EN-v6_0_20161101.pdf.

[131] SK ID Solutions AS. ESTEID2018 Certification Practice Statement, Version 1.0, November 1, 2018. https://sk.ee/upload/files/SK-CPS-ESTEID2018-EN-v1_0-20181101.pdf.

[132] Federal Law Gazette I, p. 2959. Act on Identity Cards of 18 June 2009, amended by Article 4 of the Act of 22 December 2011. https://www.gesetze-im-internet.de/englisch_pauswg/englisch_pauswg.html.

[133] DigiDoc4-Client pull requests. #214: New PIN code validation rules, March 31, 2018. https://github.com/open-eid/DigiDoc4-Client/pull/214.

[134] ERR News. Estonia's first new ID cards to be issued this week, December 11, 2018. https://news.err.ee/883962/estonia-s-first-new-id-cards-to-be-issued-this-week.

[135] Geenius. You can order an ID-card and a passport more cheaply from the New Year (in Estonian), December 17, 2019. https://raha.geenius.ee/rubriik/uudis/uuest-aastast-saab-id-kaardi-ja-passi-soodsamalt-tellida/.

[136] SK ID Solutions AS. Personal communication (support ticket #INC255135), July 30, 2018.

[137] Estonian Information System Authority. The ID card software is updated (in Estonian), July 22, 2008. https://www.ria.ee/et/uudised/id-kaardi-tarkvara-uueneb.html.

[138] Postimees. ID card software will be developed by AS Sertifitseerimiskeskus (in Estonian), July 27, 2010. https://majandus24.postimees.ee/292458/id-kaardi-tarkvara-hakkab-arendama-sertifitseerimiskeskus.

[139] AS Sertifitseerimiskeskus. Since October 2014, ID software development has been migrated to GitHub (in Estonian), December 9, 2014. https://www.id.ee/artikkel/alates-oktoobrist-2014-on-id-tarkvara-arendus-ule-viidud-github-keskkonda-mis-on-leitav-https-github-com-open-eid/.

[140] ID Help Centre. Operating systems supported by ID-software, October 17, 2020. https://www.id.ee/en/article/operating-systems-supported-by-id-software/.

[141] Estonian Information System Authority. RIA: update your ID-card software today!, January 4, 2019. https://www.ria.ee/en/news/ria-update-your-id-card-software-today.html.

[142] ID Help Centre. RIA DigiDoc application version information (release notes), January 15, 2020. https://www.id.ee/en/article/ria-digidoc-application-version-information-release-notes-2/.

[143] Open Electronic Identity: linux-installer issues. Horribly broken awp package, December 9, 2018. https://github.com/open-eid/linux-installer/issues/37.

[144] Antti Andreimann. Mr. Certification center, what are you leaking from there? (in Estonian), November 1, 2010. http://id.anttix.org/leak/leak.html.

[145] Jaanus Kase. The ID card may transfer your name and personal identification code without your permission (in Estonian), November 4,

2010. https://e.jaanus.com/id-kaart-vib-luba-ksimata-nime-ja-isikukoodi-edastada/.

[146] Estonian Public Broadcasting. Pealtnägija: 408 (in Estonian), ETV, November 17, 2010. http://arhiiv.err.ee/vaata/pealtnagija-408.

[147] ERR News. RIA intends to correct the ID card security error (in Estonian), November 17, 2010. https://www.err.ee/412790/ria-kavatseb-id-kaardi-turvavea-parandada.

[148] Delfi.ee. Concern: ID card data can be collected secretly? (in Estonian), November 8, 2010. https://epl.delfi.ee/eesti/kahtlus-id-kaardi-andmeid-saab-salaja-koguda?id=51286140.

[149] AS Sertifitseerimiskeskus. New ID card software for Windows ready (in Estonian), January 23, 2011. https://www.id.ee/artikkel/valmis-uus-id-kaardi-tarkvara-windowsile/.

[150] Ohtuleht.ee. E-election software verifiers: there are errors, but the results can't be falsified (in Estonian), August 17, 2013. https://www.ohtuleht.ee/537447/e-valimiste-tarkvara-kontrollijad-vigu-on-aga-tulemusi-voltsida-ei-saa.

[151] ERR News. After Bug Fix, Some Fear Thousands May Not Update Digital Signing Software, August 28, 2013. https://news.err.ee/108093/after-bug-fix-some-fear-thousands-may-not-update-digital-signing-software.

[152] AS Sertifitseerimiskeskus. Install new ID software version 3.7.2! (in Estonian), August 23, 2013. https://www.id.ee/artikkel/paigaldage-uus-id-tarkvara-versioon-3-7-2/.

[153] Estonian Information System Authority. Install the new version of the ID-card software 3.7.2 (in Estonian), August 23, 2013. https://www.ria.ee/et/uudised/paigaldage-id-kaardi-baastarkvara-uus-versioon-372.html.

[154] Delfi.ee. Toomas Lepa's public question to the Estonian state: Who is responsible that digital signatures are not secure for us? (in Estonian), August 27, 2013. https://www.delfi.ee/news/paevauudised/eesti/toomas-lepa-avalik-kusimus-eesti-riigile-kes-vastutab-et-digiallkiri-pole-meil-turvaline?id=66642425.

[155] Delfi.ee. 30,000 people have not yet updated their DigiDoc software (in Estonian), August 27, 2013. https://epl.delfi.ee/eesti/30-000-inimest-pole-digidoci-tarkvara-veel-uuendanud?id=66642309.

[156] Chrome-token-signing pull requests. #51: Authentication support, July 6, 2017. https://github.com/open-eid/chrome-token-signing/pull/51.

[157] Arnis Parsovs. ID card (browser signing extension) authentication man-in-the-middle attack, February 3, 2021. https://www.youtube.com/watch?v=Qr638sbaZ_M.

[158] Geenius. Swedbank used the flawed ID card extension for two years: the bank considered it more reliable (in Estonian), February 5, 2021. https://digi.geenius.ee/eksklusiiv/swedbank-kasutas-turvanorkusega-id-kaardi-laiendust-kaks-aastat-pank-pidas-seda-tookindlamaks/.

[159] Estonian Information System Authority. The Information System Authority (RIA) and its partners fixed a critical bug in the ID-card browser extension, February 3, 2021. https://www.ria.ee/en/news/information-system-authority-ria-and-its-partners-fixed-critical-bug-id-card-browser-extension.html.

[160] Delfi.ee. Another discovery by the University of Tartu security specialist: there was a critical hole in the ID card software for years (in Estonian), February 28, 2020. https://www.delfi.ee/news/paevauudised/eesti/tartu-ulikooli-turvaspetsi-jarjekordne-avastus-id-kaardi-tarkvaras-oli-aastaid-kriitiline-auk?id=89080065.

[161] Arnis Parsovs. Creating a technically valid but legally invalid EU Qualified Electronic Signature, February 26, 2020. https://www.youtube.com/watch?v=eYG17IGOCi0.

[162] AS Sertifitseerimiskeskus. The Look@World Foundation brings "everyone's ID card reader" to Estonia (in Estonian), December 19, 2006. https://www.sk.ee/uudised/vaata-maailma-toob-eestisse-igauhe-id-kaardi-lugeja.

[163] Ronald Liive. Estonian startup launches a compact and stylish Smart Card reader, 2014. http://www.kahvel.ee/28882/estonian-startup-launches-compact-stylish-smart-card-reader/.

[164] David Harley. Dr. Zeus: the Bot in the Hat, November 5, 2010. https://www.welivesecurity.com/2010/11/05/dr-zeus-the-bot-in-the-hat/.

[165] Postimees. Malware creators target chip cards similar to the ID card (in Estonian), November 29, 2010. https://tarbija24.postimees.ee/349057/arvutiviiruste-loojad-sihivad-id-kaardi-sarnaseid-kiipkaarte.

[166] Estonian Information System Authority. RIA recommends that authorities use ID card readers with a PIN pad on high-risk computers (in Estonian), December 11, 2012. https://www.ria.ee/et/uudised/ria-soovitab-ametiasutuste-korgema-riskastmega-arvutites-kasutada-koodisormistikuga-id.html.

[167] Martin Paljak. Insecure HP USB Smart Card Keyboard, March 19, 2011. https://web.archive.org/web/20110802021123/http:

//martinpaljak.net/2011/03/19/insecure-hp-usb-smart-card-keyboard/.

[168] Gemalto. IDBridge CT700 & CT710 brochure. https://cybersec.ee/storage/Gemalto_IDBridgeCT700_CT710_brochure.pdf.

[169] Liisa Lukin, SK ID Solutions AS. Personal communication, September 27, 2017.

[170] Postimees. RIA: ID card readers should be replaced with safer readers (in Estonian), December 10, 2012. https://www.postimees.ee/1070224/ria-id-kaardi-lugejad-tuleks-turvalisemate-vastu-vahetada.

[171] Gemalto. IDBridge CT710 ESTONIA PinPad product specification. https://cybersec.ee/storage/IDBridgeCT710Estonia_specs.docx.

[172] SK ID Solutions AS. ESTEID-SK Certification Practice Statement, Version 4.0, April 1, 2018. https://sk.ee/upload/files/SK-CPS-ESTEID-EN-v4_0_20180401.pdf.

[173] SK ID Solutions AS. LDAP directory service, May 2019. https://www.sk.ee/en/repository/ldap/.

[174] AS Sertifitseerimiskeskus. The Estonian ID Card and Digital Signature Concept: Principles and Solutions, March 7, 2003. https://www.id.ee/public/The_Estonian_ID_Card_and_Digital_Signature_Concept.pdf.

[175] Estonian Public Broadcasting. Pealtnägija: 202 (in Estonian), ETV, January 19, 2005. https://arhiiv.err.ee/vaata/pealtnagija-202.

[176] Chancellor of Justice. Chancellor of Justice Activity Review 2006. Page 269: ID card holder personal ID code publication on the Internet (in Estonian), 2007. https://www.oiguskantsler.ee/sites/default/files/6iguskantsleri_2006._aasta_tegevuse_ylevaade.pdf.

[177] AS Sertifitseerimiskeskus. Change in LDAP directory lookup (in Estonian), December 1, 2006. https://sk.ee/uudised/ldap-kataloogi-otsing-muutus.

[178] Mihai Barbulescu, Adrian Stratulat, Vlad Traista-Popescu, and Emil Simion. RSA Weak Public Keys available on the Internet. Cryptology ePrint Archive, Report 2016/515, 2016. https://eprint.iacr.org/2016/515.

[179] Märt Laur. eID application guide for software developers (in Estonian). Presentation at SK annual conference 2011, November 15, 2012. https://www.sk.ee/upload/files/Mart%20Laur_eid-rakendusjuhend-sk-laur.pdf.

[180] Geenius. Did you know that in this portal everyone can see what kind of real estate you own free of charge? (in Estonian), February 2, 2019.

https://digi.geenius.ee/rubriik/uudis/kas-teadsid-sellest-portaalist-saab-igauks-tasuta-vaadata-millist-kinnisvara-sa-omad/.

[181] Estonian Information System Authority. Changes in eesti.ee mailbox notifications (in Estonian), February 19, 2019. https://www.ria.ee/et/uudised/muutus-eestiee-postkasti-teavitus.html.

[182] Estonian Police and Border Guard Board. Document descriptions issued by Police and Border Guard Board, November 24, 2017. https://www2.politsei.ee/en/nouanded/dokumentide-naidised/.

[183] Council of the European Union. PRADO - Public Register of Authentic travel and identity Documents Online: EST - Estonia, 2020. https://www.consilium.europa.eu/prado/en/prado-documents/EST/index.html.

[184] European Communities. eID Interoperability for PEGS: Update of Country Profiles study. Estonian country profile, July 2009. http://ec.europa.eu/idabc/servlets/Doc7398.pdf?id=32304.

[185] ORGA Kartensysteme GmbH. MICARDO Public Chip Card Operating System Version 2.1 User Manual, September 2001. https://cybersec.ee/storage/mic21_druck.pdf.

[186] Andres Overst. Personal communication, February 14, 2019.

[187] Ivar Jung. Latest developments in Estonian eID, October 2007. https://slideplayer.com/slide/11130818/.

[188] ISO/IEC. ISO/IEC 7816-3 – Identification cards – Integrated circuit cards – Part 3: Cards with contacts – Electrical interface and transmission protocols, 2006.

[189] NIST. FIPS PUB 140-2: Security Requirements for Cryptographic Modules, 2002.

[190] European Commission. Compilation of Member States notification on SSCDs and QSCDs, June 14, 2018. https://ec.europa.eu/futurium/en/content/compilation-member-states-notification-sscds-and-qscds.

[191] TÜV Informationstechnik GmbH. TUVIT-DSZ-ITSEC-9126-2001: Certification / MICARDO Public Version 2.1 64/32 R1.0, August 28, 2001. https://cybersec.ee/storage/TUVIT-DSZ-ITSEC-9126-2001_micardo_certificate.pdf.

[192] TÜV Informationstechnik GmbH. TUVIT-DSZ-ITSEC-9126-2001: Certification Report / Smart Card Operating System MICARDO Public Version 2.1 64/32 R1.0/, August 28, 2001. https://cybersec.ee/storage/TUVIT-DSZ-ITSEC-9126-2001_micardo_report.pdf.

[193] TÜV Informationstechnik GmbH. TUVIT-DSZ-ITSEC-9115-2000: Certification of chipcard security controller SLE66CX320P from Infineon

Technologies AG, August 4, 2000. https://cybersec.ee/storage/TUVIT-DSZ-ITSEC-9115-2000_SLE66CX320P_certificate.pdf.

[194] German Federal Office for Information Security (BSI). Certification Report BSI-DSZ-ITSEC-0175-2002 for Smart Card IC (Security Controller) SLE66CX320P / m1421b25 from Infineon Technologies AG, September 12, 2002. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Reporte/Reporte01/0175a_pdf.pdf.

[195] AS Sertifitseerimiskeskus. ESTEID Card Certification Policy, Version 3.3, September 1, 2012. https://sk.ee/upload/files/SK-CP-ESTEID-20120901v3_3_en.pdf.

[196] Helar Laasik. Digi-ID is a big help. In *Estonian Information Society Yearbook 2011/2012*, 2012. http://www.digar.ee/arhiiv/et/perioodika/51716.

[197] SK ID Solutions AS. Certificate, CRL and OCSP Profile for Personal Identification Documents of the Republic of Estonia. Version 8.1, November 4, 2017. https://sk.ee/upload/files/SK-CPR-ESTEID-EN_v8_1_20171104.pdf.

[198] Ministry of Economic Affairs and Communications. Secure Signature Creation Devices (SSCD) used in Estonian electronic signature ecosystem, June 20, 2016. https://cybersec.ee/storage/20160620_table_of_sscd_in_estonia.pdf.

[199] MULTOS. MULTOS Implementation Reports: Multos International I4E, December 18, 2017. https://www.multos.com/products/approved_platforms/MIR/multos_international/i4e.

[200] Andreas Lehmann. New Generation of eID Smartcard, November 6, 2014. https://sk.ee/upload/files/AK2014_New%20Generation%20of%20eID%20Smartcard_Andreas%20Lehmann.pdf.

[201] Martin Paljak. Cold/warm ATR and protocol (re-)selection in pcsc-lite, November 4, 2010. http://muscle.musclecard.narkive.com/DsjHXzip/cold-warm-atr-and-protocol-re-selection-in-pcsc-lite.

[202] GlobalPlatform Inc. GlobalPlatform Card Specification, Version 2.2.1, January 2011. https://github.com/martinpaljak/GlobalPlatformPro/blob/master/docs/pdfs/GPC_Specification-2.2.1.pdf.

[203] Infineon. Product Brief: JCLX80JTOP20ID: Java Card™ Open Platform for Identification, 2008. https://cybersec.ee/storage/infineon_JCLX80JTOP20ID_product_brief.pdf.

[204] Trusted Logic S.A. Java Card Open Platform (jTOP v42) Security Target Lite. Version: PU-2006-RT-389-v42-1.2-LITE, September 15,

2009. https://www.commoncriteriaportal.org/files/epfiles/anssi-cible_2009-34en.pdf.

[205] e-Governance Academy. Study on the functionality of documents in ID-1 format (in Estonian), December 2013. https://www.siseministeerium.ee/sites/default/files/dokumendid/Uuringud/Isikut_toendavad_dokumendid/2013_id-1_formaadis_dokumentide_funktsionaalsuse_uuring.pdf.

[206] Postimees. ID cards issued this year are faulty (in Estonian), January 11, 2011. http://www.postimees.ee/373211/tanavu-valjastatud-id-kaardid-on-vigased?id=373211.

[207] Postimees. The manufacturer fixed an error on this year's ID cards (in Estonian), February 10, 2011. https://www.postimees.ee/386132/tootja-korvaldas-tanavustel-id-kaartidel-ilmnenud-vea.

[208] ID Help Centre. ID-card generations supported in different ID-software versions, October 26, 2017. https://www.id.ee/en/article/id-card-generations-supported-in-different-versions-of-the-id-software/.

[209] The Council of the European Union. Council Regulation (EC) No 380/2008 of 18 April 2008 amending Regulation (EC) No 1030/2002 laying down a uniform format for residence permits for third-country nationals, 2008. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32008R0380.

[210] International Civil Aviation Organization. DOC 9303. Machine Readable Travel Documents. Part 11: Security Mechanisms for MRTDs, 2015. https://www.icao.int/publications/Documents/9303_p11_cons_en.pdf.

[211] National Cybersecurity Agency of France (ANSSI). ANSSI-CC-2009/34: CC Certified Product: JCLX80jTOP20ID : Java Trusted Open Platform IFX#v42, with patch version 2.0, emedded on SLE66CLX800PE or SLE66CLX360PE (in French), October 27, 2009. https://cybersec.ee/storage/carte-a-puce-jclx80jtop20id-java-trusted-open-platform-ifxv42-avec-patch-en-version-2-0-masquee-sur-composants-sle66clx800pe-et-sle66clx360pe.html.

[212] German Federal Office for Information Security (BSI). Archive Certification Reports, October 2018. https://www.bsi.bund.de/EN/Topics/Certification/certified_products/Archiv_reports.html.

[213] GlobalPlatform Inc. GlobalPlatform Card Specification, Version 2.1.1, March 2013. http://www.win.tue.nl/pinpasjc/docs/Card%20Spec%20v2.1.1%20v0303.pdf.

[214] Andreas Lehmann, General Manager, Trüb Baltic AS. Personal communication, May 27, 2016.

[215] David Everett. What the silicon manufacturer has put together let no man put asunder, March 2010. https://www.smartcard.co.uk/articles/Whatthesiliconmanufacturerhasputtogetherletnomanputasunder.

[216] Infineon. jTOP ID on SLE 78: Java Card™ platform for government ID projects, April 2017. https://www.infineon.com/dgdl/Infineon-jTOP_ID_on_SLE78-PB-v04_17-EN.pdf?fileId=5546d4624cb7f111014d4d1cfb004279.

[217] Infineon. SLE 78CLX800P: Dual-interface and contactless security cryptocontroller, July 2012. https://cybersec.ee/storage/SPO_SLE%2078CLX800P_2012-07.pdf.

[218] Estonian Information System Authority. Personal communication, January 2, 2018.

[219] Director General of the Police and Border Guard Board. Decision No 15.2-9/277-1 (in Estonian), November 2, 2017. https://cybersec.ee/storage/20171102_PPA_decision_ROCA_suspension.bdoc.

[220] International Civil Aviation Organization. DOC 9303. Machine Readable Travel Documents. Part 3: Specifications Common to all MRTDs, 2015. https://www.icao.int/publications/Documents/9303_p3_cons_en.pdf.

[221] Police and Border Guard Board. Deviation list (issued on 2018-06-26) listing residence permit cards with a wrong signature contained in the EF.SOD file, June 26, 2018. https://www.politsei.ee/files/eMRTD/erpdeviationlist.bin.

[222] European Commission. Commission Decision C(2011) 5499 of 4.8.2011 amending Commission Decision C(2006) 2909 final laying down the technical specifications on the standards for security features and biometrics in passports and travel documents issued by Member States, 2011. https://ec.europa.eu/home-affairs/sites/homeaffairs/files/e-library/docs/comm_native_c_2011_5499_f_en.pdf.

[223] Estonian Police and Border Guard Board. Personal communication, July 31, 2017.

[224] Police and Border Guard Board. Estonia is extending the validity period of 32,000 digital ID cards, March 20, 2020. https://www.id.ee/en/article/estonia-is-extending-the-validity-period-of-32000-digital-id-cards/.

[225] Police and Border Guard Board. Police recommends extending Digi-IDs with a 3-year validity period before April 30 (in Estonian), April 18, 2019. https://www.ria.ee/et/uudised/politsei-soovitab-pikendada-kolmeaastase-kehtivusajaga-digi-id-enne-30-aprilli.html.

[226] Republic of Estonia. e-Residency, May 14, 2019. https://e-resident.gov.ee/.

[227] Postimees. Bill Gates made into e-resident without asking, October 23, 2018. https://news.postimees.ee/6436641/bill-gates-made-into-e-resident-without-asking.

[228] Postimees. A contactless Estonian ID-card has been built (in Estonian), March 5, 2016. http://tehnika.postimees.ee/3607697/video-valminud-on-kontaktivaba-eesti-id-kaart.

[229] Ministry of Foreign Affairs of the Republic of Estonia. Handbook: Diplomatic Immunities and Privileges in Estonia, April 2017. http://vm.ee/sites/default/files/content-editors/state-protocol/kasiraamat_2017_3.pdf.

[230] Official Journal of the European Union. Update of model cards issued by the Ministries of Foreign Affairs of Member States to accredited members of diplomatic missions and consular representations and members of their families (2017/C 279/04), August 23, 2017. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.C_.2017.279.01.0005.01.ENG.

[231] Estonian Police and Border Guard Board. ID card samples (in Estonian), August 28, 2020. https://www.politsei.ee/et/juhend/id-kaardi-naeidised-1.

[232] GIXEL Digital Identity. Identification Authentication Signature – European Citizen Card. Technical Specifications, Revision: 1.0.1, 2008. http://www.unsads.com/specs/IASECC/IAS_ECC_v1.0.1_UK.pdf.

[233] ID Help Centre. New ID-card and its changes, April 2, 2019. https://www.id.ee/en/article/new-id-card-and-its-changes/.

[234] Sander-Karl Kivivare. *Secure Channel Establishment for the NFC Interface of the New Generation Estonian ID Cards*. BSc thesis, University of Tartu, 2020. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=70557&year=2020.

[235] Uwe Seidel. EU initiatives in the ID document domain. Part 1: Answering the counterfeit pressure on EU visa and residence permits, October 4, 2019. https://platform.keesingtechnologies.com/eu-id-document/.

[236] The European Parliament and the Council of the European Union. Regulation (EU) 2019/1157 of the European Parliament and of the Council of 20 June 2019 on strengthening the security of identity cards of Union citizens and of residence documents issued to Union citizens and their family members exercising their right of free movement, 2019. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32019R1157.

[237] Estonian Information System Authority. The 2020 yearbook of the Information System Authority, 2020. https://www.ria.ee/sites/default/files/content-editors/ria_aastaraamat_2020_48lk_eng.pdf.

[238] National Cybersecurity Agency of France (ANSSI). ANSSI-CC-2017/49: CC Certified Product: ID-One Cosmo v8.1-N platform - Large, embedded on the NXP P6022M VB (in French), September 5, 2017. https://www.ssi.gouv.fr/entreprise/certification_cc/plateforme-id-one-cosmo-v8-1-n-large-masquee-sur-le-composant-nxp-p6022m-vb/.

[239] German Federal Office for Information Security (BSI). BSI-DSZ-CC-0973-V2-2016: NXP Secure Smart Card Controller P6022y VB including IC Dedicated Software, October 11, 2016. https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/CC/SmartCards_IC_Cryptolib/0973_0973V2.html.

[240] National Cybersecurity Agency of France (ANSSI). ANSSI-CC-2018/17: CC Certified Product: IAS ECC V2 in configuration #3 in composition on the ID-One Cosmo open platform v8.1-N - Large R2, embedded on the NXP P6022M VB (in French), May 4, 2018. https://www.ssi.gouv.fr/certification_cc/ias-ecc-v2-en-configuration-3-en-composition-sur-la-plateforme-ouverte-id-one-cosmo-v8-1-n-large-r2-masquee-sur-le-composant-nxp-p6022m-vb/.

[241] National Cybersecurity Agency of France (ANSSI). Certification Report ANSSI-CC-2018/17: IAS ECC V2 in configuration #3 in composition on the ID-One Cosmo open platform v8.1-N - Large R2, embedded on the NXP P6022M VB (in French), May 4, 2018. https://www.ssi.gouv.fr/uploads/2018/05/anssi-cc-2018_17fr.pdf.

[242] Gregor Johannson. *Technical Prerequisites for Enabling Third-Party Applications on the New Estonian ID-card*. MSc thesis, Tallinn University of Technology, 2019. https://digikogu.taltech.ee/et/Item/64c83d8f-8f2d-4311-b548-b07c9b58a6cb.

[243] Oberthur Technologies. ID-One Cosmo V8.1 – Security Target Lite, August 2017. https://www.ssi.gouv.fr/uploads/2017/09/anssi-cible-cc-2017_49en.pdf.

[244] National Cybersecurity Agency of France (ANSSI). Maintenance Report ANSSI-CC-2017/49-M02: ID-One Cosmo v8.1-N platform - Large, embedded on the NXP P6022M VB (in French), January 24, 2019. https://www.ssi.gouv.fr/uploads/2017/09/anssi-cc-2017_49_m02.pdf.

[245] Julie Chuzel, Head of Certification Body, ANSSI. Personal communication, May 20, 2020.

[246] Guillaume Poupard, Director General, ANSSI. Clarification letter addressed to Caroline Jardon (IDEMIA), July 21, 2020. https://cybersec.ee/storage/20200721_NP_ANSSI_SDE_PSS_CCN_1658-Lettre_IDEMIA.pdf.

[247] SK ID Solutions AS. Cards for testing, July 01, 2017. https://sk.ee/en/services/testcard/.

[248] Lembitu Ling. Bug 414520: Add Sertifitseerimiskeskus AS root CA certificate. https://bugzilla.mozilla.org/show_bug.cgi?id=414520.

[249] Kaido Raiend. SEB Employee Card (in Estonian), November 7, 2013. https://sk.ee/upload/files/AK2013_Kaido%20Raiend_SEB%20tootoendist.pdf.

[250] SK ID Solutions AS. SK ID Solutions AS – Certificate and OCSP Profile for SEB-cards, October 24, 2018. https://www.sk.ee/upload/files/SK-CPR-SEB%20CARD-EN-v6_1-20181024.pdf.

[251] SK ID Solutions AS. SK ID Solutions AS – Certificate Policy for the SEB card, April 10, 2020. https://www.skidsolutions.eu/upload/files/SK-CP-SEB%20CARD-EN-v6_0-20200410.pdf.

[252] Petr Svenda, Matus Nemec, Peter Sekan, Rudolf Kvasnovsky, David Formanek, David Komarek, and Vashek Matyas. The Million-Key Question – Investigating the Origins of RSA Public Keys. In *FI MU Report Series, FIMU-RS-2016-03*, pages 1–83. Masaryk University, 2016. https://crocs.fi.muni.cz/_media/public/papers/usenixsec16_1mrsakeys_trfimu_201603.pdf.

[253] Ludovic Rousseau. CCID descriptor statistics: dwDefaultClock, July 2, 2014. https://ludovicrousseau.blogspot.com/2014/07/ccid-descriptor-statistics.html.

[254] B. Kaliski. PKCS #1: RSA Encryption Version 1.5. RFC 2313 (Proposed Standard), March 1998. http://www.ietf.org/rfc/rfc2313.txt.

[255] Bug 915408: [Client side] TLS 1.2 client authentication needs to support SHA-1 signatures. https://bugzilla.mozilla.org/show_bug.cgi?id=915408.

[256] Romain Bardou, Riccardo Focardi, Yusuke Kawamoto, Lorenzo Simionato, Graham Steel, and Joe-Kai Tsay. Efficient Padding Oracle Attacks on Cryptographic Hardware. Rapport de recherche RR-7944, INRIA, April 2012. http://hal.inria.fr/hal-00691958/PDF/RR-7944.pdf.

[257] Paul C Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Annual International Cryptology Conference*, pages 104–113. Springer, 1996.

[258] Matus Nemec, Marek Sys, Petr Svenda, Dusan Klinec, and Vashek Matyas. The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 1631–1648, New York, NY, USA, 2017. ACM.

[259] Bruno Produit. *Optimization of the ROCA (CVE-2017-15361) Attack*. MSc thesis, University of Tartu, 2019. `https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=66866&year=2019`.

[260] Jan Jancar, Vladimir Sedlacek, Petr Svenda, and Marek Sys. Minerva: The curse of ECDSA nonces; Systematic analysis of lattice attacks on noisy leakage of bit-length of ECDSA nonces. In *Conference on Cryptographic Hardware and Embedded Systems (CHES) 2020*. Ruhr-University of Bochum, Transactions on Cryptographic Hardware and Embedded Systems, 2020.

[261] N. A. Howgrave-Graham and N. P. Smart. Lattice attacks on digital signature schemes. *Designs, Codes and Cryptography*, 23(3):283–290, August 2001.

[262] Elke De Mulder, Michael Hutter, Mark E. Marson, and Peter Pearson. Using Bleichenbacher's Solution to the Hidden Number Problem to Attack Nonce Leaks in 384-Bit ECDSA. In Guido Bertoni and Jean-Sébastien Coron, editors, *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, volume 8086 of *Lecture Notes in Computer Science*, pages 435–452. Springer, 2013.

[263] Diego F. Aranha, Felipe Rodrigues Novaes, Akira Takahashi, Mehdi Tibouchi, and Yuval Yarom. LadderLeak: Breaking ECDSA With Less Than One Bit Of Nonce Leakage. *IACR Cryptol. ePrint Arch.*, 2020:615, 2020.

[264] T. Pornin. Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA), August 2013. `http://www.ietf.org/rfc/rfc6979.txt`.

[265] Robert G. Brown. Dieharder: A Random Number Test Suite, Version 3.31.1, 2014. `https://webhome.phy.duke.edu/~rgb/General/dieharder.php`.

[266] Christoph Sutter, IT Infrastructure Division Manager, TÜV Informationstechnik GmbH. Personal communication, October 24, 2018.

[267] Gereon Killian, Head of Certification Section D22 (Certification of hardware related products), German Federal Office for Information Security (BSI). Personal communication, March 29, 2019.

[268] AS Sertifitseerimiskeskus. Certificate renewal (in Estonian), February 7, 2006. https://web.archive.org/web/20060207071228/http://www.sk.ee:80/pages.php/0202040102,674.

[269] AS Sertifitseerimiskeskus. Renewal of ID-card certificates, September 4, 2014. https://web.archive.org/web/20140904165340/http://id.ee/index.php?id=34416.

[270] SK ID Solutions AS. Remote certificate update for MICARDO cards (in Estonian), October 20, 2018.

[271] Postimees. Many ID-card certificates need to be updated (in Estonian), October 5, 2005. https://sakala.postimees.ee/2166139/paljude-id-kaartide-sertifikaarte-tuleb-uuendada.

[272] AS Sertifitseerimiskeskus. ESTEID Card Certification Policy, Version 5.0, January 25, 2016. https://sk.ee/upload/files/SK-CP-ESTEID-20160125v5_0_en.pdf.

[273] Anto Veldre. The second wave of card renewals, June 22, 2016. https://blog.ria.ee/kaardiuuenduse-teine-laine/.

[274] Wojciech Mostowski and Erik Poll. Malicious Code on Java Card Smartcards: Attacks and Countermeasures. In Gilles Grimaud and François-Xavier Standaert, editors, *Smart Card Research and Advanced Applications*, pages 1–16, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[275] Cryptanalysis of GlobalPlatform Secure Channel Protocols. Mohamed Sabt and Jacques Traore. Cryptology ePrint Archive, Report 2017/032, 2017. https://eprint.iacr.org/2017/032.pdf.

[276] Serge Vaudenay. Security Flaws Induced by CBC Padding — Applications to SSL, IPSEC, WTLS... In Lars R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, pages 534–545, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[277] Gildas Avoine and Loïc Ferreira. Attacking GlobalPlatform SCP02-compliant Smart Cards Using a Padding Oracle Attack. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):149–170, 2018.

[278] Daniel Bleichenbacher. Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1. In *Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '98, pages 1–12, London, UK, UK, 1998. Springer-Verlag.

[279] Yusuke Kawamoto and Joe-Kai Tsay. Guest lecture: Efficient Padding Oracle Attacks on Estonian Electronic ID Card, Security Tokens and Smart Cards, September 13, 2012. https://www.uttv.ee/naita?id=13146.

[280] Delfi.ee. Foreign cryptologists: The Estonian ID card is not secure (in Estonian), June 8, 2012. http://epl.delfi.ee/news/

eesti/valismaa-kruptoloogid-eesti-id-kaardid-pole-turvalised?id=64512104.

[281] Ars Technica. Scientists crack RSA SecurID 800 tokens, steal cryptographic keys, June 25, 2012. https://arstechnica.com/information-technology/2012/06/securid-crypto-attack-steals-keys/.

[282] Estonian Information System Authority. Police and Border Guard Board: International Research Work Addresses Security of Estonian ID Card (in Estonian), June 27, 2012. https://www.ria.ee/et/uudised/politsei-ja-piirivalveameti-teade-rahvusvahelises-teadustoos-puudutati-eesti-id-kaardi.html.

[283] Estonian Information System Authority. Police and Border Guard Board: International Research Work Addresses Security of Estonian ID Card, July 3, 2012. https://www.ria.ee/en/news/police-and-border-guard-board-international-research-work-addresses-security-estonian-id-card.html.

[284] Agu Kivimägi. Padding oracle and the responsibility of the researcher (in Estonian), September 27, 2013. https://www.sirp.ee/s1-artiklid/c9-sotsiaalia/polsterdusoraakel-ja-teadlase-vastutus/.

[285] Police and Border Guard Board. The Police and Border Guard Board is renewing ID-Cards issued in 2011, September 11, 2012. https://www.id.ee/en/article/the-police-and-border-guard-board-is-renewing-id-cards-issued-in-2011/.

[286] ERR News. Police Begin Renewing Faulty ID Cards, September 26, 2012. https://news.err.ee/105405/police-begin-renewing-faulty-id-cards.

[287] Postimees. ID cards issued last year will be updated (in Estonian), September 7, 2012. https://www.postimees.ee/965078/eelmisel-aastal-valja-antud-id-kaarte-uuendatakse.

[288] Delfi.ee. Estonia's largest PR operation: Saving the ID card (in Estonian), September 13, 2017. https://ekspress.delfi.ee/kuum/eesti-suurim-pr-operatsioon-id-kaardi-paastmine?id=79478038.

[289] Police and Border Guard Board. FAQ: Renewing ID-Cards issued in 2011, May 17, 2013. https://www.id.ee/index.php?id=36348.

[290] Police and Border Guard Board. Email notification to the cardholders of ID cards issued in 2011, 2012. https://cybersec.ee/storage/PPA_2012_notice_to_2011_ID-card_holders.txt.

[291] Urmo Keskel, SK ID Solutions AS. Personal communication, April 19, 2019.

[292] Arnis Parsovs. Practical Issues with ID Card Authentication. Presentation at CERT-EE Symposium 2013, June 14, 2013. https://cybersec.ee/storage/CERT-EE_0ct0b3rf3st_2013_slides.pdf.

[293] Police and Border Guard Board. Response to request for explanation 08.04.2019 no 15.1-2/49-2, April 8, 2019. https://cybersec.ee/storage/PPA_2011_response_20190408.pdf.

[294] ERR News. Ex-head of RIA about disclosing the security risk: the government did the right thing (in Estonian), September 8, 2017. https://www.err.ee/617401/ria-eksjuht-turvariski-avalikustamisest-valitsus-tegi-oigesti.

[295] Police and Border Guard Board. The eesti.ee email addresses of four thousand documents must be renewed (in Estonian), September 1, 2015. https://sk.ee/uudised/neljal-tuhandel-dokumendil-tuleb-uuendada-eestiee-meiliaadressi.

[296] J. Klensin. Application Techniques for Checking and Transformation of Names. RFC 3696, February 2004. http://www.ietf.org/rfc/rfc3696.txt.

[297] Wietse Venema. Postfix Configuration Parameters: allow_min_user, November 6, 2018. http://www.postfix.org/postconf.5.html#allow_min_user.

[298] ERR News. 250,000 Estonian ID cards could be faulty, September 29, 2015. https://news.err.ee/116849/250-000-estonian-id-cards-could-be-faulty.

[299] Postimees. All Estonians need to update ID-card software online (in Estonian), February 21, 2016. https://tehnika.postimees.ee/3593619/koigil-eesti-elanikel-tuleb-id-kaardi-tarkvara-veebis-uuendada.

[300] Chromium. Issue 532048: Unable to use client certificate with negative modulus, September 15, 2015. https://bugs.chromium.org/p/chromium/issues/detail?id=532048.

[301] Chromium. Issue 534766: Unable to use client certificate with MSB=0 bad encoding, September 22, 2015. https://bugs.chromium.org/p/chromium/issues/detail?id=534766.

[302] Postimees. All e-residents got faulty cards, October 2, 2015. https://news.postimees.ee/3348383/all-e-residents-got-faulty-cards.

[303] Estonian Information System Authority. Remote updating of Estonian ID card certificates begins today, March 18, 2016. https://www.ria.ee/en/news/remote-updating-estonian-id-card-certificates-begins-today.html.

[304] Anto Veldre. The third wave of card renewals, December 21, 2016. https://blog.ria.ee/kauguuendamise-kolmas-laine/.

[305] Estonian Information System Authority. From next week non-updated ID cards will not work in Google Chrome browser (in Estonian), August 31, 2017. https://www.ria.ee/et/uudised/jargmisest-nadalast-ei-saa-uuendamata-id-kaarte-google-chromei-veebibrauseriga-kasutada.html.

[306] SK ID Solutions AS. Certificate, CRL and OCSP Profile for Personal Identification Documents of the Republic of Estonia. Version 7.0, November 1, 2016. https://sk.ee/upload/files/SK-CPR-ESTEID-EN-v7_0_20161101.pdf.

[307] The OpenSSL Project. OpenSSL CHANGES, November 10, 2018. https://github.com/openssl/openssl/blob/65042182fcafbd4c0dd8fdabaefdf1fd38dc6287/CHANGES.

[308] Peter Gutmann. X.509 Style Guide, October 2000. https://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt.

[309] The Go programming language. Issue 6837: crypto/tls: rejects CA cert from certtool, November 27, 2013. https://github.com/golang/go/issues/6837.

[310] Postimees. In Estonia there are hundreds of thousands of faulty ID cards in circulation (in Estonian), January 11, 2011. https://tehnika.postimees.ee/3342861/eestis-on-kaibel-sadu-tuhandeid-tarkvaraveaga-id-kaarte.

[311] Raul Kaidro, Martin Paljak, and Hillar Aarelaid. Modulus is negative, May 26, 2016. http://slides.com/hillar/modulus-is-negative#/.

[312] Yubico Inc. Security Advisory 2017-10-16 – Infineon Weak RSA Key Generation, October 16, 2017. https://www.yubico.com/support/security-advisories/ysa-2017-01/.

[313] Petr Svenda. The goal of the research (ROCA), May 8, 2018. https://lessonslearned.publicon.ee/userfiles/RIA/lessonslearned2018/1_Petr%20Svenda%20-%20CROCS_ROCA_Tallin_20180508_finalCut.pdf.

[314] Postimees. Beer saved the Estonian ID card (in Estonian), May 9, 2018. https://tehnika.postimees.ee/4485598/olu-paastis-eesti-id-kaardi.

[315] Ilmar Raag. States are held by people's will. 3 briefs & 2 lessons from the ID card saga (in Estonian), May 9, 2018. https://edasi.org/23317/ilmar-raag-riike-hoiab-pusti-inimeste-tahe-3-briifi-2-oppetundi-id-kaardi-saagast/.

[316] ERR News. Potential security risk could affect 750,000 Estonian ID cards, September 5, 2017. https://news.err.ee/616732/potential-security-risk-could-affect-750-000-estonian-id-cards.

[317] ERR News. The agencies are hoping to eliminate the security risk of ID cards in two months (in Estonian), September 5, 2017. https://www.err.ee/616731/ametid-loodavad-id-kaardi-turvariski-likvideerida-kahe-kuuga.

[318] Postimees. Estonian ID card security risk calls into question security of e-elections, September 5, 2017. https://news.postimees.ee/4233385/estonian-id-card-security-risk-calls-into-question-security-of-e-elections.

[319] Postimees. ID-card tip from Czech scientists, September 7, 2017. https://news.postimees.ee/4236857/id-card-tip-from-czech-scientists.

[320] ERR News. EKRE challenges electoral committee's decision to allow e-voting, September 11, 2017. https://news.err.ee/617916/ekre-challenges-electoral-committee-s-decision-to-allow-e-voting.

[321] ERR News. Supreme Court rejects EKRE's appeal of e-vote in upcoming elections, September 21, 2017. https://news.err.ee/631752/supreme-court-rejects-ekre-s-appeal-of-e-vote-in-upcoming-elections.

[322] Microsoft. Security Advisory ADV170012: Vulnerability in TPM could allow Security Feature Bypass, October 10, 2017. https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/ADV170012.

[323] Carsten Loschinsky, Vice President Chip Card & Security Sales and Marketing, Infineon Technologies AG. Customer information, October 20, 2017. https://cybersec.ee/storage/20171020_Infineon_Customer_Info.pdf.

[324] Petr Svenda. ROCA: Vulnerable RSA generation (CVE-2017-15361), October 16, 2017. https://crocs.fi.muni.cz/public/papers/rsa_ccs17.

[325] ERR News. Scientists published a technical description of the Estonian ID card security risk (in Estonian), October 16, 2017. https://www.err.ee/636860/teadlased-avalikustasid-eesti-id-kaardi-turvariski-tehnilise-kirjelduse.

[326] ERR News. Government to suspend ID card certificates with security risk at midnight, November 3, 2017. https://news.err.ee/640385/government-to-suspend-id-card-certificates-with-security-risk-at-midnight.

[327] Daniel J. Bernstein and Tanja Lange. Reconstructing ROCA, November 11, 2017. https://blog.cr.yp.to/20171105-infineon.html.

[328] Delfi.ee. ID card broken. RIA proves that fuss about ID card flaw was not in vain (in Estonian), April 19, 2018. http://epl.delfi.ee/news/eesti/id-kaart-murti-lahti-ria-toestas-et-kara-id-kaardi-turvanorkuse-parast-polnud-asjata?id=81807683.

[329] Ahto Buldas, Martha Jung, Kaja Kuivjõgi, Anna-Maria Osula, Rain Ottis, Jaan Priisalu, Liisa Tallinn, and Toomas Vaks. ID card case lessons (in Estonian), 2018. https://www.ria.ee/public/PKI/ID-kaardi_oppetunnid.pdf.

[330] Estonian Information System Authority. The Lessons We Learned, May 9, 2018. https://lessonslearned.publicon.ee/.

[331] Silvia Lips, Ingrid Pappel, Valentyna Tsap, and Dirk Draheim. Key Factors in Coping with Large-Scale Security Vulnerabilities in the eID Field. In *Electronic Government and the Information Systems Perspective - 7th International Conference, EGOVIS 2018, Regensburg, Germany, September 3-5, 2018, Proceedings*, pages 60–70, 2018.

[332] Andreas Ventsel and Mari-Liis Madisson. Semiotics of threats: Discourse on the vulnerability of the Estonian identity card, 2019. https://doi.org/10.12697/SSS.2019.47.1-2.05.

[333] Estonian Information System Authority. Security Vulnerability Detected in the Estonian ID Card Chip (in Estonian), September 5, 2017. https://www.ria.ee/ee/id-kaardi-kiibis-avastati-turvarisk.html.

[334] ERR News. RIA recommends state officials use Mobile-ID to minimize security risks, September 20, 2017. https://news.err.ee/619703/ria-recommends-state-officials-use-mobile-id-to-minimize-security-risks.

[335] German Federal Office for Information Security (BSI). BSI-DSZ-CC-0833-2013-MA-01: Maintenance Report for the product CardOS V5.0 with Application for QES, V1.0, July 7, 2017. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Reporte/Reporte08/0833ma1a_pdf.pdf.

[336] Geenius. Claims about ID card backdoor spreads across the internet, the state rejects it (in Estonian), November 13, 2017. https://geenius.ee/uudis/internetis-levib-vaide-id-kaarti-tekkinud-uuest-tagauksest-riik-lukkab-selle-umber/.

[337] Tanel Kuusk, Security Officer, SK ID Solutions AS. Personal communication, June 14, 2018.

[338] Police and Border Guard Board. Important information for Mobile-ID users, January 19, 2015. https://sk.ee/en/News/important-information-for-mobile-id-users.

[339] Postimees. The ID cards closing time undecided, PPA is going to shopping malls (in Estonian), November 2, 2017. https://tehnika.postimees.ee/4297103/id-kaartide-sulgemise-aeg-otsustamata-politsei-laheb-kaubanduskeskustesse.

[340] Geenius. Video: see how ID card is updated (in Estonian), March 22, 2018. https://geenius.ee/uudis/juhend-kui-sa-selle-kuu-jooksul-id-kaarti-ei-uuenda-siis-seda-enam-teha-ei-saagi/.

[341] ERR News. Information System Authority advises not to rush with ID card update, October 26, 2017. http://news.err.ee/638865/information-system-authority-advises-not-to-rush-with-id-card-update.

[342] ERR News. Technical errors making ID card certificate renewal difficult, November 1, 2017. https://news.err.ee/639902/technical-errors-making-id-card-certificate-renewal-difficult.

[343] ERR News. RIA: ID card patch ready, expect errors as thousands update, October 31, 2017. http://news.err.ee/639702/ria-id-card-patch-ready-expect-errors-as-thousands-update.

[344] ERR News. Government to suspend ID card certificates with security risk at midnight (in Estonian), November 2, 2017. https://www.err.ee/640301/valitsus-peatab-turvariskiga-id-kaartide-sertifikaadid-alates-reede-ohtust.

[345] ERR News. Updating of ID card certificates has not affected medical facilities' work, November 6, 2017. http://news.err.ee/640962/updating-of-id-card-certificates-has-not-affected-medical-facilities-work.

[346] ERR News. 18,000 people unable to update their ID cards remotely, November 1, 2017. https://news.err.ee/639948/18-000-people-unable-to-update-their-id-cards-remotely.

[347] ERR News. ID card updates: PPA offices open on Saturday and Sunday, November 4, 2017. http://news.err.ee/640617/id-card-updates-ppa-offices-open-on-saturday-and-sunday.

[348] Postimees. PPA will bring ID card renewal to shopping centers across Estonia (in Estonian), November 16, 2017. https://tehnika.postimees.ee/4313017/ppa-viib-id-kaardi-uuendamise-kaubanduskeskustes-ule-eestiliseks.

[349] ERR News. Police to open ID card service points in malls on December weekends, November 30, 2017. http://news.err.ee/645870/police-to-open-id-card-service-points-in-malls-on-december-weekends.

[350] ERR News. 272,000 renew ID card certificates, authorities say busiest time over, November 21, 2017. http://news.err.ee/643906/272-

000-renew-id-card-certificates-authorities-say-busiest-time-over.

[351] ERR News. Estonians abroad can apply for a new ID card via email, November 3, 2017. https://news.err.ee/640426/estonians-abroad-can-apply-for-new-id-card-via-email.

[352] ERR News. Nearly 300,000 ID card certificates not renewed by March 31 deadline, April 3, 2018. https://news.err.ee/693660/nearly-300-000-id-card-certificates-not-renewed-by-march-31-deadline.

[353] ERR News. Updated ID cards cannot yet be used for encryption, November 8, 2017. https://news.err.ee/641417/updated-id-cards-cannot-yet-be-used-for-encryption.

[354] ERR News. Stopping encryption created additional problems for hospitals (in Estonian), November 9, 2017. https://www.err.ee/641657/krupteerimise-peatamine-tekitab-haiglatele-lisaprobleeme.

[355] ERR News. Paper: ID card producers won't explain delay in informing of security flaw, November 14, 2017. https://news.err.ee/642562/paper-id-card-producers-won-t-explain-delay-in-informing-of-security-flaw.

[356] Postimees. State files claim against ID-card manufacturer, November 15, 2017. https://news.postimees.ee/4311613/state-files-claim-against-id-card-manufacturer.

[357] ERR News. Card manufacturer Gemalto helps to solve ID-card security risk (in Estonian), September 14, 2017. https://www.err.ee/618524/kaarditootja-gemalto-aitab-lahendada-id-kaardi-turvariski.

[358] ERR News. Gemalto rep: Estonian authorities notified of ID card flaw in June, November 22, 2017. https://news.err.ee/644250/gemalto-rep-estonian-authorities-notified-of-id-card-flaw-in-june.

[359] ERR News. RIA representative: Lehmann hinted at ID card flaw, but hints were vague, November 24, 2017. https://news.err.ee/644679/ria-representative-lehmann-hinted-at-id-card-flaw-but-hints-were-vague.

[360] ERR News. PPA still awaiting public apology from Gemalto, December 1, 2017. http://news.err.ee/646126/ppa-still-awaiting-public-apology-from-gemalto.

[361] Postimees. Gemalto to replace representative in Estonia, November 30, 2017. https://news.postimees.ee/4329113/gemalto-to-replace-representative-in-estonia.

[362] Postimees. Estonia had already received a notification about the potential ID card risk in June (in Estonian), December 1, 2017. https://tehnika.postimees.ee/4330169/eesti-sai-juba-juunis-teate-id-kaardi-voimaliku-riski-kohta.

[363] European Union Agency for Network and Information Security (ENISA). Incident report ID 163484 Austria, June 20, 2017. https://cybersec.ee/storage/Incident-report-ID-163484-Austria.pdf.

[364] Delfi.ee. Palo dismissed the claims of the ID card security risk from an earlier notification: the hint from the Gemalto Estonia representative was vague and the Austrian report was incomplete (in Estonian), December 11, 2017. http://arileht.delfi.ee/news/uudised/palo-lukkas-umber-vaited-id-kaardi-turvariski-varasemast-teavitusest-gemalto-eesti-esindaja-vihje-oli-ebamaarane-ja-austria-raport-oli-puudulik?id=80450470.

[365] Postimees. The agencies had already closed ID cards with a security flaw in the spring, as Tartu scientist had faked digital signature (in Estonian), December 6, 2017. https://tehnika.postimees.ee/4334991/ametid-sulgesid-turvaveaga-id-kaarte-juba-kevadel-kuna-tartu-teadlane-oli-voltsinud-digiallkirja.

[366] Geenius. PPA no longer allows Gemalto to advertise itself with the e-state of Estonia (in Estonian), January 19, 2018. https://geenius.ee/uudis/politsei-ei-luba-gemaltol-end-enam-eesti-e-riigiga-reklaamida/.

[367] Delfi.ee. The country is looking for a compromise with the ID card manufacturer (in Estonian), August 30, 2018. http://www.delfi.ee/news/paevauudised/eesti/video-riik-otsib-id-kaardi-tootjaga-voimalust-kompromissiks?id=83516467.

[368] Postimees. PPA vs. Gemalto: New Developments (in Estonian), September 7, 2018. https://tehnika.postimees.ee/6399519/politsei-vs-gemalto-uued-arengud.

[369] ERR News. ID card crisis cost Information System Authority 1 million euros, March 9, 2018. https://news.err.ee/688602/id-card-crisis-cost-information-system-authority-1-million.

[370] Geenius. Hans Lõugas: how the documents about the ID card crisis were leaked to us and why we do not believe them (in Estonian), September 6, 2018. https://geenius.ee/uudis/hans-lougas-kuidas-meile-id-kaardi-kriisi-kohta-dokumendid-lekitati-ja-miks-me-neid-ei-usu/.

[371] Postimees. Cyber-lollygagging cost the state millions, September 6, 2018. https://news.postimees.ee/6383968/cyber-lollygagging-cost-the-state-millions.

[372] ERR News. PPA say no to ID card compromise, Gemalto still hope for accord with state, September 7, 2018. https://news.err.ee/859618/ppa-say-no-to-id-card-compromise-gemalto-still-hope-for-accord-with-state.

[373] ERR News. Police claim 152 million from ID card producer Gemalto, September 27, 2018. https://news.err.ee/864523/police-claim-152-million-from-id-card-producer-gemalto.

[374] ERR News. Former ID card manufacturer Gemalto files against PPA, October 25, 2018. https://news.err.ee/871871/former-id-card-manufacturer-gemalto-files-against-ppa.

[375] ERR News. PPA seeking EUR 300,000 from Gemalto, November 6, 2018. https://news.err.ee/874973/ppa-seeking-300-000-from-gemalto.

[376] Delfi.ee. Police vs Gemalto: two years in court and no outcome (in Estonian), August 28, 2020. https://forte.delfi.ee/news/tehnika/politsei-vs-gemalto-kaks-aastat-kohtuveskeid-ja-ei-tuhjagi?id=90871257.

[377] Delfi.ee. It's not even funny anymore: two years have passed and the court has gone nowhere in the police and Gemalto case (in Estonian), January 7, 2021. https://forte.delfi.ee/artikkel/92191151/see-pole-enam-isegi-naljakas-moodunud-on-kaks-aastat-ja-kohus-pole-joudnud-politsei-ja-gemalto-kohtuasjas-mitte-kuskile.

[378] Police and Border Guard Board. A settlement agreement has been signed between the Police and Border Guard Board and Gemalto AG Tallinn, February 5, 2021. https://www.politsei.ee/en/news/a-settlement-agreement-has-been-signed-between-the-police-and-border-guard-board-and-gemalto-ag-tallinn-2021.

[379] Infineon Technologies AG. Background Information on software update of RSA key generation function, October 16, 2017. https://www.infineon.com/cms/en/product/promopages/rsa-update/rsa-background.

[380] National Cybersecurity Agency of France (ANSSI). ANSSI-CC-2013/15: CC Certified Product: Platform jTOP INFv#46 embedded on Infineon components SLE78CLX1600PM, SLE78CLX800P and SLE78CLX360PM (in French), August 7, 2013. https://cybersec.ee/storage/plateforme-jtop-infv46-masquee-sur-composants-infineon-sle78clx1600pm-sle78clx800p-et-sle78clx360pm.html.

[381] National Cybersecurity Agency of France (ANSSI). Certificate ANSSI-CC-2013/15 (in French), August 7, 2013. https:

//cybersec.ee/storage/jTOP_INFv46_CERTIFICATE-PLATFORM_ANSSI_CC-2013-55.pdf.

[382] German Federal Office for Information Security (BSI). BSI-DSZ-CC-0829: Infineon smart card IC (Security Controller) M7820 A11 and M11 with optional RSA2048/4096 v1.02.013, EC v1.02.013, SHA-2 v1.01 and Toolbox v1.02.013 libraries and with specific IC dedicated software, September 5, 2012. https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/CC/SmartCards_IC_Cryptolib/0829_V2.html.

[383] Infineon Technologies AG. Security Target M7820 A11 and M11 including optional Software Libraries RSA – EC – SHA-2 – Toolbox, Version 1.6, August 28, 2012. https://www.commoncriteriaportal.org/files/epfiles/0829b_pdf.pdf.

[384] J. Jonsson and B. Kaliski. Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1. RFC 3447 (Proposed Standard), February 2003. http://www.ietf.org/rfc/rfc3447.txt.

[385] Patrick Boedeker, IT Security Head of Department Hardware Evaluation, TÜV Informationstechnik GmbH. Personal communication, September 10, 2018.

[386] Gereon Killian, Head of Certification Section D22 (Certification of hardware related products), German Federal Office for Information Security (BSI). Personal communication, August 6, 2018.

[387] German Federal Office for Information Security (BSI). BSI-DSZ-CC-0833-2013: CardOS V5.0 with Application for QES, V1.0, July 26, 2013. https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/CC/Digitale_Signatur-Sichere_Signaturerstellungseinheiten/0833.html.

[388] Explanatory Memorandum for Digital Signatures Act (in Estonian). Digitaalallkirja seadus (151 SE) Tallinn, 1999. https://www.riigikogu.ee/download/f2f6398a-30ab-337e-b6d4-40d30b64186a.

[389] Agu Kivimägi. Rebuttal: Estonian ID Card Secure, Says Rep, November 1, 2013. https://news.err.ee/108797/rebuttal-estonian-id-card-secure-says-rep.

[390] Postimees. Gemalto: Statements made by the Estonian state are a surprise (in Estonian), May 18, 2018. https://tehnika.postimees.ee/4490671/gemalto-eesti-riigi-tehtud-avaldused-on-ullatus.

[391] Vasilios Mavroudis, Andrea Cerulli, Petr Svenda, Dan Cvrcek, Dusan Klinec, and George Danezis. A Touch of Evil: High-Assurance Cryptographic Hardware from Untrusted Components. In *24th ACM Conference on Computer and Communications Security (CCS'2017)*, pages 1583–1600. ACM, 2017.

[392] Ahto Buldas, Aivo Kalu, Peeter Laud, and Mart Oruaas. Server-Supported RSA Signatures for Mobile Devices. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *Computer Security – ESORICS 2017*, pages 315–333, Cham, 2017. Springer International Publishing.

[393] Arnis Parsovs. Identity Card Key Generation in the Malicious Card Issuer Model, 2014. https://courses.cs.ut.ee/MTAT.07.022/2014_spring/uploads/Main/arnis-report-s14.pdf.

[394] Arnis Parsovs. Factorization of a corrupted RSA public key from an Estonian ID card, November 6, 2020. https://www.youtube.com/watch?v=tfttN6YUGrs.

[395] Postimees. An ID card used as a loyalty card can be unexpectedly locked (in Estonian), October 6, 2017. https://www.postimees.ee/4268277/kliendikaardina-kasutatud-id-kaart-voib-ootamatult-lukustuda.

[396] Geenius. The non-updated software has already locked the ID cards of 150 people: what do you need to do? (in Estonian), October 16, 2018. https://digi.geenius.ee/rubriik/uudis/uuendamata-tarkvara-on-juba-150-inimese-id-kaardi-lukustanud-mida-pead-ette-votma/.

[397] OpenSC Issues. OpenSC fails to read file with T=0 recursive APDUs, November 8, 2017. https://github.com/OpenSC/OpenSC/issues/1190.

[398] Trusted Logic S.A. Java Card Open Platform Security Target-LITE. Version: PU-2012-RT-751-v46-1.0-LITE, May 18, 2013. https://www.ssi.gouv.fr/uploads/IMG/certificat/ANSSI-CC-cible_2013-55en.pdf.

[399] ERR News. The Estonian state filed a claim against Gemalto for damages due to ID card problems (in Estonian), November 14, 2017. http://www.err.ee/642521/eesti-riik-esitas-id-kaardi-probleemide-tottu-gemaltole-kahjunoude.

[400] Delfi.ee. ID card-based loyalty systems have errors, more than a hundred cards have become unusable (in Estonian), August 9, 2019. https://forte.delfi.ee/news/digi/id-kaardil-pohinevates-pusikliendisusteemides-on-torkeid-ule-saja-kaardi-on-muutunud-kasutuskolbmatuks?id=87087775.

[401] Geenius. Using the wrong ID card reader can lock the card, the state has been aware of the problem since last spring (in Estonian), July 13, 2020. https://digi.geenius.ee/rubriik/uudis/vale-id-kaardi-lugeja-kasutamine-voib-kaardi-lukustada-riik-on-probleemist-teadlik-olnud-mullu-kevadest/.

[402] Chancellor of Justice. Letter No 7-4/191757/2003053: Replacement of a non-functioning ID card (in Estonian), May 26, 2020. https://

www.oiguskantsler.ee/sites/default/files/field_document2/Mittetoimiva%20ID-kaardi%20%C3%BCmbervahetamine.pdf.

[403] Delfi.ee. Police rejects recommendations of the Chancellor of Justice (in Estonian), July 13, 2020. https://forte.delfi.ee/news/tarkvara/politsei-saatis-oiguskantsleri-soovitused-kuu-peale?id=90436319.

[404] Chancellor of Justice. Letter No 7-4/191757/2004840: Examination of a non-functioning ID card (in Estonian), September 8, 2020. https://cybersec.ee/storage/20200908_Mittetoimiva%20ID-kaardi%20ekspertiisi%20tegemine.pdf.

[405] Geenius. The police discovered 15,000 faulty ID cards, over 300 have been used (in Estonian), June 26, 2019. https://digi.geenius.ee/rubriik/uudis/politsei-avastas-15-000-veaga-id-kaarti-ule-300-on-kasutatud/.

[406] Chancellor of Justice. Letter No 7-4/200479/2002668: ID card's use after person's death (in Estonian), May 7, 2020. https://www.oiguskantsler.ee/sites/default/files/field_document2/ID-kaardi%20kasutamine%20p%C3%A4rast%20inimese%20surma.pdf.

[407] State Portal eesti.ee. Services for a citizen: Document check, October 8, 2020. https://www.eesti.ee/portaal/!portaal.document_check.

[408] IT and Development Centre of the Ministry of the Interior. E-service: requesting death information, October 4, 2020. https://www.rahvastikuregister.ee/records/death.

[409] Geenius. For the second time, a number of deceased people's ID card certificates were not revoked (in Estonian), February 3, 2021. https://digi.geenius.ee/eksklusiiv/teist-korda-jaid-hulga-surnud-inimeste-id-kaartide-sertifikaadid-kehtetuks-tunnistamata/.

[410] Delfi.ee. Security hole found in ID-card (in Estonian), May 3, 2002. http://epl.delfi.ee/news/eesti/id-kaardis-leiti-turvaauk?id=50922213.

[411] Peeter Marvet. Personal communication, December 24, 2018.

[412] Delfi.ee. Recommendation: ID card codes must be changed immediately (in Estonian), May 4, 2002. http://epl.delfi.ee/news/eesti/soovitus-id-kaardi-koodid-tuleb-kohe-muuta?id=50922339.

[413] Postimees. Expert discovers security issue with new Estonian ID cards (in Estonian), December 20, 2018. https://tehnika.postimees.ee/6481827/ekspert-avastas-eesti-uue-id-kaardiga-seotud-turvaprohmaka.

[414] Martin Paljak. TL;DR: always, ALWAYS change your PIN codes!, December 20, 2018. https://martinpaljak.net/wellthisisodd/.

[415] ERR News. Enormous "document factory" scam exposed, October 28, 2015. https://news.err.ee/117074/enormous-document-factory-scam-exposed.

[416] Postimees. "Passport mafia" led by babushka, October 29, 2015. https://news.postimees.ee/3379293/passport-mafia-led-by-babushka.

[417] Delfi.ee. PPA expert: The manufacturer implements a fix for a new ID card PIN envelopes (in Estonian), December 20, 2018. http://www.delfi.ee/news/paevauudised/eesti/video-ppa-ekspert-tootja-muudab-uute-id-kaardi-turvaumbrikute-lahendust?id=84815297.

[418] TV3. Expert discovers security risk with the new ID cards (in Estonian), December 20, 2018. https://uudised.tv3.ee/eesti/uudis/2018/12/20/ekspert-avastas-uue-id-kaardiga-seotud-turvariski/.

[419] Postimees. ID card security envelope is not transparent anymore (in Estonian), December 28, 2018. https://tehnika.postimees.ee/6486878/id-kaardi-turvaumbrik-ei-paista-enam-labi.

[420] Nicolas Serrano, Hilda Hadan, and L. Jean Camp. A Complete Study of P.K.I. (PKI's Known Incidents), July 23, 2019. http://dx.doi.org/10.2139/ssrn.3425554.

[421] Mozilla. CA/Responding To An Incident, July 24, 2019. https://wiki.mozilla.org/CA/Responding_To_An_Incident.

[422] Chromium. Root Certificate Policy, September 26, 2019. https://www.chromium.org/Home/chromium-security/root-ca-policy.

[423] Cybernetica AS. Study on the lifecycle and use of cryptographic algorithms v3.1 (in Estonian), December 31, 2013. https://www.ria.ee/public/PKI/kruptograafiliste_algoritmide_elutsukli_uuring_II.pdf.

[424] Cybernetica AS. Study on the lifecycle of cryptographic algorithms v4.0 (in Estonian), June 3, 2015. https://www.ria.ee/public/RIA/Kruptograafiliste_algoritmide_uuring_2015.pdf.

[425] Cybernetica AS. Cryptographic Algorithms Lifecycle Report 2016, June 22, 2016. https://www.ria.ee/public/RIA/Cryptographic_Algorithms_Lifecycle_Report_2016.pdf.

[426] Otto de Voogd. The Flaw in the Estonian ID Card, October 29, 2013. https://news.err.ee/108556/the-flaw-in-the-estonian-id-card.

[427] Christopher Soghoian and Sid Stamm. Certified Lies: Detecting and Defeating Government Interception Attacks against SSL (Short Paper). In George Danezis, editor, *Financial Cryptography and Data Security*, pages 250–259, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

ACKNOWLEDGEMENTS

This research has been supported by the European Regional Development Fund through the Estonian Centre of Excellence in ICT Research (EXCITE) under grant number EU48684 and by the European Social Fund through the IT Academy programme.

We would like to thank our supervisors: Prof. Dominique Unruh from University of Tartu for supporting this research direction; and the industry supervisor Jan Willemson (and also Sven Heiberg) from Cybernetica AS for, among other things, supporting this research by allocating industrial funding for the respective STACC work project. We thank the pre-reviewers Peeter Laud, George Danezis and Petr Svenda for their feedback.

We would like to thank the following persons for donating their ID cards, participating in the experiments, providing feedback or otherwise supporting this research: Cesar Pereida Garcia, Evely Kukk, Imre Hohensee, Ivar Smolin, Ivo Kubjas, Janar Haidak, Juhan Aasaru, Juri Hudolejev, Kaire Valge, Kaur Virunurm, Kazuaki Tsuchimoto, Lennart Pungas, Liisa Lukin, Marek Pagel, Peeter Marvet, Rando Kulla, Robert Tiismus, Shahla Atapoor, Taimo Peelo, Tiina Viirelaid, Tiit Pikma, Toivo Reitalu and Tõnu Samuel.

We thank Arne Ansper for giving us the idea of using ROCA vulnerable moduli detection tests to recover the corrupted public keys, Alex Halderman for the initial ID card LDAP certificate dataset (December 2012), Olivier Nonga for the chemical removal of the chip epoxy coating and Danielle Morgan for language improvements. The access to the computing resources provided by UT HPC is also greatly appreciated.

We thank the persons who provided insightful information about the Estonian eID ecosystem and its related matters: Andres Overst, Anto Veldre, Inguss Treiguts, Karina Egipt, Margus Freudenthal, Martin Paljak, Tanel Kuusk and Tarvi Martens.

We thank the employees of state institutions RIA (Andrei Kargin, Margus Arm, Mark Erlich, Tõnis Reimo) and PPA (Eliisa Sau, Kaija Kirch) for their cooperation. An outstanding acknowledgement goes to SK ID Solutions AS and their employees who over the years have provided instant and insightful high-quality answers about the functionality of the Estonian eID ecosystem.

LIST OF ABBREVIATIONS

3DES	triple DES (data encryption standard)
AES	Advanced Encryption Standard
AID	application identifier
ANSSI	National Cybersecurity Agency of France
APDU	application protocol data unit
API	application programming interface
ASCII	American Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation One
ATR	answer to reset
ATS	answer to select
BAC	Basic Access Control
BSI	German Federal Office for Information Security
CA	Certificate Authority
CAN	Card Access Number
C-APDU	command APDU
CBC	cipher block chaining
CCA	client certificate authentication
CC	Common Criteria
CMB	Citizenship and Migration Board
CMK	card management key
CP	Certificate Policy
CPLC	Card Production Life Cycle
CPS	Certification Practice Statement
CPU	central processing unit
CRL	Certificate Revocation List
CRT	Chinese remainder theorem
CSCA	Country Signing Certification Authority
DER	Distinguished Encoding Rules
DF	dedicated file
DSA	Digital Signatures Act
EAC	Extended Access Control
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EC	Elliptic Curve
EEPROM	electrically erasable programmable read-only memory
EF	elementary file
eIDAS	EU regulation on electronic identification and trust services
EITSETA	Electronic Identification and Trust Services for Electronic Transactions Act

eMRTD	electronic machine-readable travel document
ENISA	European Union Agency for Cybersecurity
FCP	file control parameters
FID	file identifier
FMD	file management data
GCM	Galois/Counter Mode
GP	GlobalPlatform
GUI	graphical user interface
IAS-ECC	Identification Authentication Signature – European Citizen Card
IDA	Identity Documents Act
IEC	International Electrotechnical Commission
ISD	Issuer Security Domain
ISO	International Organization for Standardization
ITSEC	IT Security Evaluation Criteria
IV	initialization vector
jTOP	Java Trusted Open Platform
LDAP	Lightweight Directory Access Protocol
MAC	message authentication code
MB	megabyte
MFA	Ministry of Foreign Affairs
MF	master file
MIME	Multipurpose Internet Mail Extensions
MKM	Ministry of Economic Affairs and Communications
MRZ	machine-readable zone
MSB	most significant byte
NFC	near field communication
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
OS	operating system
PACE	Password Authenticated Connection Establishment
PIN	personal identification number
PKCS	Public Key Cryptography Standards
PKI	public key infrastructure
POS	point of sale
PPA	Police and Border Guard Board (Politsei- ja Piirivalveamet)
PUK	personal unblocking key
PUPI	pseudo-unique proximity integrated circuit card identifier
QES	qualified electronic signature
QSCD	qualified electronic signature creation device
QTSP	qualified trust service provider
R-APDU	response APDU
RFC	Request for Comments

RIA	Estonian Information System Authority (Riigi Infosüsteemi Amet)
RNG	random number generator
ROCA	Return of Coppersmith's Attack (vulnerability CVE-2017-1536)
RSA	Rivest, Shamir and Adleman Algorithm
SCP	Secure Channel Protocol
SHA	Secure Hash Algorithm
SK	SK ID Solutions AS (formerly AS Sertifitseerimiskeskus)
S/MIME	Secure/Multipurpose Internet Mail Extensions
SMIT	IT and Development Centre at the Estonian Ministry of the Interior
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SSCD	secure signature creation device
SSC	send sequence counter
SSH	Secure Shell
SW	status word
TJA	Technical Regulatory Authority (formerly Technical Surveillance Authority)
TLS	Transport Layer Security
TOE	target of evaluation
TPM	Trusted Platform Module
TSA	timestamping authority
TÜViT	TÜV Informationstechnik GmbH
UID	unique identifier
USB	Universal Serial Bus
VPN	virtual private network
XML	Extensible Markup Language
XOR	logical exclusive OR operation

SISUKOKKUVÕTE

Eesti elektrooniline ID-kaart ja selle turvaväljakutsed

Eesti elektrooniline isikutunnistus (ID-kaart) on Eesti kodanikele pakkunud turvalist elektroonilist identiteeti üle 18 aasta. See on olnud turvalise elektroonilise autentimise ja õiguslikult siduvate digitaalallkirjade tehnoloogiline nurgakivi. ID-kaart on võimaldanud turvalist identifitseerimist erinevates e-teenustes, kõige ambitsioonikam neist on valimised interneti teel. Eesti ID-kaardi kasutuselevõtt algas 2002. aastal ja seda peetakse levitamise ja aktiivse kasutamise aspektist üheks edukamaks kiipkaardipõhiste riiklike ID-kaardisüsteemide juurutamiseks maailmas.

Käesolevas töös uuritakse põhjalikult Eesti ID-kaarti ning sellega seotud turvaväljakutseid. Käsitleme teemat ulatuslikult, kuna ID-kaardi turvalisus sõltub lisaks kiibis olevast krüptograafilisest funktsionaalsusest ka selle kasutamisviisist ja kogu selle ümber ehitatud ökosüsteemi turvalisusest. See töö põhineb avalikult kättesaadaval dokumentatsioonil, meedias kajastatud teabel, kaasatud osapoolte teabel ning meie enda analüüsil ja selles valdkonnas tehtud katsetel. Oluline andmekogum, mida kasutati meie enda andmete valideerimiseks ja analüüsimiseks, oli aastate jooksul kogunenud ID-kaardi avaliku võtme sertifikaatide kogu ID-kaardi avaliku sertifikaatide hoidlast. Lõputöö on kirjutatud monograafia stiilis ning võtab lühidalt kokku ja viitab meie lõputöö lõpus loetletud originaalväljaannetele.

Selle töö esimene panus on anda põhjalik ülevaade Eesti ID-kaardist ja sellega seotud aspektidest. Töös käsitletakse praktilisi aspekte, teadmisi ja nüansse, mida pole eelnevalt mujal käsitletud. Oluline panus on ka teemaga seotud õiguslik ja ajalooline kontekst. Avastasime selle töö kirjutamise käigus mitmeid õiguslikke rikkumisi ja tõime välja Common Criteria sertifitseerimissmeemi puudused. See töö võib olla lähtepunkt teadlastele, kes on huvitatud Eesti ID-kaardist ja selle ökosüsteemist.

Selle töö teine panus on detailne ülevaade Eesti ID-kaardi tehnoloogilistest platvormidest (kiipkaardi kiibiversioonidest) ja vastavatest isikut tõendavate dokumentide tüüpidest. Iga platvormi jaoks esitatakse ID-kaardi elektroonilise funktsionaalsuse rakendamiseks kasutatava tehnoloogilise lahenduse kirjeldus ja kiipkaardi mikrokontrolleri fotod ning vastavate isikut tõendavate dokumentide näidiste fotod. Oleme analüüsinud iga platvormi pakutavat krüptograafilist funktsionaalsust, selle toimivusomadusi ja vastavust standarditele. Rakendades erinevaid musta kasti analüüsimeetodeid, oleme osaliselt taastanud platvormide rakendatud RSA võtmete genereerimise algoritmid. Leidsime, et mitme ID-kaardi platvormi puhul ei vasta ID-kaardi tootjate tarnitud kiibid ametlikus spetsifikatsioonis määratletud sertifitseeritud versioonile.

Eesti ID-kaardi puhul on kasutatud ainulaadseid tehnilisi lahendusi, mis on võimaldanud ID-kaardi omanikel oma kaartidel olevaid andmeid ja tarkvara

Interneti kaudu kauguuenduse kaudu ajakohastada. Oleme selles töös dokumenteerinud ja analüüsinud selle funktsionaalsuse rakendamiseks kasutatud protokolle. Selle tulemusena oleme leidnud, et ITSEC-i sertifitseeritud turvalise sõnumivahetuse protokollil on krüptograafiline puudus, mis võimaldas läbi viia teesklusrünnakuid ja GlobalPlatformi turvalise sõnumivahetuse protokolli juurutamisel jäi märkamata oraakli täidise haavatavus, mis võimaldas teabevahetuse dekrüpteerida. Need leiud on aidanud kauguuenduse lahenduse turvalisust parandada ja loodame, et see töö on kasulik viide tulevaste ID-kaardi kaugvärskenduslahenduste väljatöötamiseks.

Selle töö veel üks panus on üksikasjalik ülevaade minevikus toimunud Eesti ID-kaardi turvaintsidentidest. Toimunud sündmused on rekonstrueeritud avalikest allikatest leitud teabekildude põhjal. Kui see oli võimalik, kogusime täiendavat teavet, võttes ühendust asjaomaste osapooltega ja analüüsides meie sertifikaatide andmekogumit ning tehes ise analüüse ja katseid. Oleme pingutanud intsidentide põhjuste ja tagajärgede kindlakstegemiseks ning kaasatud poolte reageeringute ja avaliku suhtluse kajastamiseks. Töös leidub mitmeid uudseid järeldusi ja see võib olla aluseks juhtumitest täiendavate õppetundide saamiseks. Eesti ID-kaardi kogemus annab teistele üleriigilist avaliku võtme taristut rakendavatele riikidele põhjaliku ülevaate tekkida võivatest probleemidest.

Avastasime selle töö käigus erineva raskusastmega turvaprobleeme. Kõige olulisemad avastused, mis on seotud võtmehalduse turvavigadega, on kaasa toonud reaalse mõju, näiteks ID-kaartide tagasikutsumise ja kohtuprotsessi ID-kaartide tootja Gemalto vastu.

Ehkki see pole selle töö peamine eesmärk, oleme teinud laiahaardelisemaid järeldusi ja pakkunud välja soovitusi, mida saab kasutada sisendina ID-kaardi turvalisuse korraldamisega seotud poliitikamuudatuste tegemiseks.

Selle töö sisu on jagatud seitsmeks peatükiks:

- 1. peatükis tutvustatakse teemat lühidalt, tuuakse välja selle töö panused ja lõputöö ülesehitus.
- 2. peatükis tutvustatakse muuhulgas Eesti ID-kaardi ökosüsteemi, andes ülevaate: isikutest, kes on seotud ID-kaardi tootmise, väljastamise ja järelevalvega; ID-kaardi põhilisest elektroonilisest funktsionaalsusest ja sellega seotud juriidilistest mõistetest; autentimisest, dekrüpteerimisest ja digitaalse allkirjastamise kasutamisest ning sellega seotud probleemidest; seotud tugikomponentidest, näiteks kiipkaardilugejad ja ID-kaardi tarkvara; sertifikaatidest, avalikust sertifikaatide hoidlast ja `@eesti.ee` e-posti aadressist; ning ID-kaardi ja selle sertifikaatide elutsüklist.
- 3. peatükk tutvustab aastate jooksul kasutusel olnud ID-kaardi platvorme (kiipkaardi kiibiversioone) kronoloogiliselt ja isikut tõendavate dokumentide tüüpe, mis nendele platvormidele väljastatud on.
- 4. peatükis kirjeldatakse üksikasjalikult iga ID-kaardi platvormi pakutavat

asümmeetrilise krüptograafia funktsionaalsust. Analüüsides võtmete omadusi ja krüptograafiliste toimingute ajastusteavet, püüdsime taastada igal platvormil kasutatavate asümmeetriliste krüptograafia algoritmide rakendusdetailid. Leidsime, et igal platvormil rakendati krüptograafilisi algoritme väikeste erinevustega.

- 5. peatükis kirjeldatakse ID-kaardi kauguuenduse lahendusi ja sellega seotud turvaanalüüsi. Toome välja turvalised sõnumivahetuse protokollid, mida kaardihaldustoimingute tarbeks kasutatakse ja kirjeldame kauguuendamise lahendusi, mida on kasutatud iga ID-kaardi platvormi jaoks.
- 6. peatükis on välja toodud turvaintsidentide ja teiste sarnaste probleemide loetelu, millega Eesti ID-kaart aastate jooksul kokku puutunud on. Mõnest selles peatükis loetletud ja meie poolt töö kirjutamise käigus leitud probleemist on ka asjaomastele osapooltele teada antud.
- 7. peatükis kirjeldatakse käesoleva töö tulemusi laiemalt ning esitatakse loetelu soovitustest, mis meie arvates võiksid tugevdada Eesti ID-kaardi ja selle ökosüsteemi turvalisust.
- 8. peatükis esitatakse doktoritöö kokkuvõtvad seisukohad.

CURRICULUM VITAE

Personal data

Name:	Arnis Paršovs
Date of birth:	1986-08-05
Citizenship:	Latvian
Contact:	arnis.parsovs@eesti.ee

Education

2012–...	PhD studies in Computer Science, University of Tartu
2010–2012	Master of Science in Engineering (Cyber Security), Tallinn University of Technology and University of Tartu
2009–2010	Professional Bachelor Degree in Computer Systems, Riga Technical University
2006–2009	First Level of Higher Professional Education, Qualification: Computer Systems and Networks Administrator, Riga Technical College
2002–2006	Vocational Secondary Education, Qualification: Programming Technician, Riga Technical College Vocational Secondary School

Employment

2019–...	Information Security Team Lead at the Institute of Computer Science of University of Tartu
2012–2019	Researcher at Software Technology and Applications Competence Centre OÜ
2003–2011	Information Systems Security Administrator at State Land Service of Latvia

ELULOOKIRJELDUS

Isikuandmed

Nimi:	Arnis Paršovs
Sünniaeg:	1986-08-05
Kodakondsus:	Läti
Kontakt:	arnis.parsovs@eesti.ee

Haridus

2012–...	Informaatika doktoriõpe, Tartu Ülikool
2010–2012	Tehnikateaduse (küberkaitse) magister, Tallinna Tehnikaülikool ja Tartu Ülikool
2009–2010	Professionaalne bakalaureusekraad arvutisüsteemides, Riia Tehnikaülikool
2006–2009	Rakenduskõrghariduse esimene tase, kvalifikatsioon: arvutisüsteemide ja võrkude administraator, Riia Tehnikakolledž
2002–2006	Kutsekeskharidus, kvalifikatsioon: programmeerimistehnik, Riia Tehnikakolledž kutsekeskkool

Teenistuskäik

2019–...	Infoturbe töörühma juht, arvutiteaduse instituut, Tartu Ülikool
2012–2019	Teadur, Tarkvara Tehnoloogia Arenduskeskus OÜ
2003–2011	Infosüsteemide turbe administraator, Läti Maateenistus

LIST OF ORIGINAL PUBLICATIONS

I. Arnis Parsovs. Practical Issues with TLS Client Certificate Authentication. In Proceedings of the Network and Distributed System Security Symposium (NDSS), The Internet Society, San Diego, CA, February 2014.

II. Danielle Morgan and Arnis Parsovs. Using the Estonian Electronic Identity Card for Authentication to a Machine. In Secure IT Systems: 22nd Nordic Conference, NordSec 2017, Tartu, Estonia, November 2017.

III. Tõnu Mets and Arnis Parsovs. Time of signing in the Estonian digital signature scheme. In Digital Evidence and Electronic Signature Law Review, vol 16, 2019, pp. 40-50.

IV. Arnis Parsovs. Solving the Estonian ID Card Crisis: the Legal Issues. In Proceedings of the 17th International Conference on Information Systems for Crisis Response and Management ISCRAM 2020; Blacksburg, VA, May 2020.

V. Arnis Parsovs. Estonian Electronic Identity Card: Security Flaws in Key Management. In 29th USENIX Security Symposium (USENIX Security 20), August 2020.

DISSERTATIONES INFORMATICAE PREVIOUSLY PUBLISHED IN DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS

19. **Helger Lipmaa.** Secure and efficient time-stamping systems. Tartu, 1999, 56 p.
22. **Kaili Müürisep.** Eesti keele arvutigrammatika: süntaks. Tartu, 2000, 107 lk.
23. **Varmo Vene.** Categorical programming with inductive and coinductive types. Tartu, 2000, 116 p.
24. **Olga Sokratova.** Ω-rings, their flat and projective acts with some applications. Tartu, 2000, 120 p.
27. **Tiina Puolakainen.** Eesti keele arvutigrammatika: morfoloogiline ühestamine. Tartu, 2001, 138 lk.
29. **Jan Villemson.** Size-efficient interval time stamps. Tartu, 2002, 82 p.
45. **Kristo Heero.** Path planning and learning strategies for mobile robots in dynamic partially unknown environments. Tartu 2006, 123 p.
49. **Härmel Nestra.** Iteratively defined transfinite trace semantics and program slicing with respect to them. Tartu 2006, 116 p.
53. **Marina Issakova.** Solving of linear equations, linear inequalities and systems of linear equations in interactive learning environment. Tartu 2007, 170 p.
55. **Kaarel Kaljurand.** Attempto controlled English as a Semantic Web language. Tartu 2007, 162 p.
56. **Mart Anton.** Mechanical modeling of IPMC actuators at large deformations. Tartu 2008, 123 p.
59. **Reimo Palm.** Numerical Comparison of Regularization Algorithms for Solving Ill-Posed Problems. Tartu 2010, 105 p.
61. **Jüri Reimand.** Functional analysis of gene lists, networks and regulatory systems. Tartu 2010, 153 p.
62. **Ahti Peder.** Superpositional Graphs and Finding the Description of Structure by Counting Method. Tartu 2010, 87 p.
64. **Vesal Vojdani.** Static Data Race Analysis of Heap-Manipulating C Programs. Tartu 2010, 137 p.
66. **Mark Fišel.** Optimizing Statistical Machine Translation via Input Modification. Tartu 2011, 104 p.
67. **Margus Niitsoo**. Black-box Oracle Separation Techniques with Applications in Time-stamping. Tartu 2011, 174 p.
71. **Siim Karus.** Maintainability of XML Transformations. Tartu 2011, 142 p.
72. **Margus Treumuth.** A Framework for Asynchronous Dialogue Systems: Concepts, Issues and Design Aspects. Tartu 2011, 95 p.
73. **Dmitri Lepp.** Solving simplification problems in the domain of exponents, monomials and polynomials in interactive learning environment T-algebra. Tartu 2011, 202 p.

74. **Meelis Kull.** Statistical enrichment analysis in algorithms for studying gene regulation. Tartu 2011, 151 p.
77. **Bingsheng Zhang.** Efficient cryptographic protocols for secure and private remote databases. Tartu 2011, 206 p.
78. **Reina Uba.** Merging business process models. Tartu 2011, 166 p.
79. **Uuno Puus.** Structural performance as a success factor in software development projects – Estonian experience. Tartu 2012, 106 p.
81. **Georg Singer.** Web search engines and complex information needs. Tartu 2012, 218 p.
83. **Dan Bogdanov.** Sharemind: programmable secure computations with practical applications. Tartu 2013, 191 p.
84. **Jevgeni Kabanov.** Towards a more productive Java EE ecosystem. Tartu 2013, 151 p.
87. **Margus Freudenthal.** Simpl: A toolkit for Domain-Specific Language development in enterprise information systems. Tartu, 2013, 151 p.
90. **Raivo Kolde.** Methods for re-using public gene expression data. Tartu, 2014, 121 p.
91. **Vladimir Šor.** Statistical Approach for Memory Leak Detection in Java Applications. Tartu, 2014, 155 p.
92. **Naved Ahmed.** Deriving Security Requirements from Business Process Models. Tartu, 2014, 171 p.
94. **Liina Kamm.** Privacy-preserving statistical analysis using secure multi-party computation. Tartu, 2015, 201 p.
100. **Abel Armas Cervantes.** Diagnosing Behavioral Differences between Business Process Models. Tartu, 2015, 193 p.
101. **Fredrik Milani.** On Sub-Processes, Process Variation and their Interplay: An Integrated Divide-and-Conquer Method for Modeling Business Processes with Variation. Tartu, 2015, 164 p.
102. **Huber Raul Flores Macario.** Service-Oriented and Evidence-aware Mobile Cloud Computing. Tartu, 2015, 163 p.
103. **Tauno Metsalu.** Statistical analysis of multivariate data in bioinformatics. Tartu, 2016, 197 p.
104. **Riivo Talviste.** Applying Secure Multi-party Computation in Practice. Tartu, 2016, 144 p.
108. **Siim Orasmaa.** Explorations of the Problem of Broad-coverage and General Domain Event Analysis: The Estonian Experience. Tartu, 2016, 186 p.
109. **Prastudy Mungkas Fauzi.** Efficient Non-interactive Zero-knowledge Protocols in the CRS Model. Tartu, 2017, 193 p.
110. **Pelle Jakovits.** Adapting Scientific Computing Algorithms to Distributed Computing Frameworks. Tartu, 2017, 168 p.
111. **Anna Leontjeva.** Using Generative Models to Combine Static and Sequential Features for Classification. Tartu, 2017, 167 p.
112. **Mozhgan Pourmoradnasseri.** Some Problems Related to Extensions of Polytopes. Tartu, 2017, 168 p.

113. **Jaak Randmets.** Programming Languages for Secure Multi-party Computation Application Development. Tartu, 2017, 172 p.
114. **Alisa Pankova.** Efficient Multiparty Computation Secure against Covert and Active Adversaries. Tartu, 2017, 316 p.
116. **Toomas Saarsen.** On the Structure and Use of Process Models and Their Interplay. Tartu, 2017, 123 p.
121. **Kristjan Korjus.** Analyzing EEG Data and Improving Data Partitioning for Machine Learning Algorithms. Tartu, 2017, 106 p.
122. **Eno Tõnisson.** Differences between Expected Answers and the Answers Offered by Computer Algebra Systems to School Mathematics Equations. Tartu, 2017, 195 p.

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS

1. **Abdullah Makkeh**. Applications of Optimization in Some Complex Systems. Tartu 2018, 179 p.
2. **Riivo Kikas**. Analysis of Issue and Dependency Management in Open-Source Software Projects. Tartu 2018, 115 p.
3. **Ehsan Ebrahimi**. Post-Quantum Security in the Presence of Superposition Queries. Tartu 2018, 200 p.
4. **Ilya Verenich**. Explainable Predictive Monitoring of Temporal Measures of Business Processes. Tartu 2019, 151 p.
5. **Yauhen Yakimenka**. Failure Structures of Message-Passing Algorithms in Erasure Decoding and Compressed Sensing. Tartu 2019, 134 p.
6. **Irene Teinemaa**. Predictive and Prescriptive Monitoring of Business Process Outcomes. Tartu 2019, 196 p.
7. **Mohan Liyanage.** A Framework for Mobile Web of Things. Tartu 2019, 131 p.
8. **Toomas Krips.** Improving performance of secure real-number operations. Tartu 2019, 146 p.
9. **Vijayachitra Modhukur.** Profiling of DNA methylation patterns as biomarkers of human disease. Tartu 2019, 134 p.
10. **Elena Sügis.** Integration Methods for Heterogeneous Biological Data. Tartu 2019, 250 p.
11. **Tõnis Tasa.** Bioinformatics Approaches in Personalised Pharmacotherapy. Tartu 2019, 150 p.
12. **Sulev Reisberg.** Developing Computational Solutions for Personalized Medicine. Tartu 2019, 126 p.
13. **Huishi Yin.** Using a Kano-like Model to Facilitate Open Innovation in Requirements Engineering. Tartu 2019, 129 p.
14. **Faiz Ali Shah.** Extracting Information from App Reviews to Facilitate Software Development Activities. Tartu 2020, 149 p.
15. **Adriano Augusto**. Accurate and Efficient Discovery of Process Models from Event Logs. Tartu 2020, 194 p.
16. **Karim Baghery.** Reducing Trust and Improving Security in zk-SNARKs and Commitments. Tartu 2020, 245 p.
17. **Behzad Abdolmaleki.** On Succinct Non-Interactive Zero-Knowledge Protocols Under Weaker Trust Assumptions. Tartu 2020, 209 p.
18. **Janno Siim.** Non-Interactive Shuffle Arguments. Tartu 2020, 154 p.
19. **Ilya Kuzovkin.** Understanding Information Processing in Human Brain by Interpreting Machine Learning Models. Tartu 2020, 149 p.
20. **Orlenys López Pintado.** Collaborative Business Process Execution on the Blockchain: The Caterpillar System. Tartu 2020, 170 p.
21. **Ardi Tampuu.** Neural Networks for Analyzing Biological Data. Tartu 2020, 152 p.

22. **Madis Vasser.** Testing a Computational Theory of Brain Functioning with Virtual Reality. Tartu 2020, 106 p.
23. **Ljubov Jaanuska.** Haar Wavelet Method for Vibration Analysis of Beams and Parameter Quantification. Tartu 2021, 192 p.