Estonian Electronic Identity Card: Security Flaws in Key Management

Arnis Parsovs, *Software Technology and Applications Competence Center and University of Tartu*

https://www.usenix.org/conference/usenixsecurity20/presentation/parsovs

This paper is included in the Proceedings of the 29th USENIX Security Symposium.

August 12–14, 2020

978-1-939133-17-5

Open access to the Proceedings of the 29th USENIX Security Symposium is sponsored by USENIX.

Estonian Electronic Identity Card: Security Flaws in Key Management

Arnis Parsovs[1,2]
[1]*Software Technology and Applications Competence Center, Estonia*
[2]*University of Tartu, Estonia*

Abstract

The Estonian electronic identity card (ID card) is considered to be one of the most successful deployments of smart card-based national ID card systems in the world. 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.

In this paper, we describe several security flaws found in the ID card manufacturing process. The flaws have been discovered by analyzing public-key certificates that have been collected from the public ID card certificate repository. In particular, we find that in some cases, contrary to the security requirements, the ID card manufacturer has generated private keys outside the chip. In several cases, copies of the same private key have been imported in the ID cards of different cardholders, allowing them to impersonate each other. In addition, as a result of a separate flaw in the manufacturing process, corrupted RSA public key moduli have been included in the certificates, which in one case led to the full recovery of the corresponding private key. This paper describes the discovery process of these findings and the incident response taken by the authorities.

1 Introduction

Estonia issues several types of credit card-sized identity documents (hereinafter – ID cards) that contain a smart card chip. The cryptographic functionality embedded in the chip enables secure authentication over the Internet and creation of legally binding digital signatures. The Estonian ID card roll-out started in 2002 and is considered to be one of the most successful in the world in 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 security of this electronic identity scheme depends on the secrecy of a cardholder's private keys. It is crucial for private keys to be generated in a secure manner and to be accessible only to the corresponding cardholder. In the Estonian ID card scheme, similarly as in many other countries, the key management (key generation, certificate issuance) is delegated to the ID card manufacturer. It is therefore essential to ensure that the manufacturer generates keys of good quality and does not store copies of the generated keys. Unfortunately, there are no effective controls to verify that the manufacturer is trustworthy and handles the key management correctly. The industry response to these concerns has been that manufacturers are in the business of trust and therefore they would never risk their reputation by engaging in sloppy security practices or malicious behavior.

Our contribution in this work is to show, by example of the Estonian ID card, that this trust model does not always work. We show that the ID card manufacturer has engaged in sloppy security practices, ignoring repeated signs of faults in the key management process, and has intentionally breached the ID card manufacturing contract in some cases creating copies of cardholders' private keys. While these findings have resulted in open litigation against ID card manufacturer Gemalto [2], there is no evidence that this loss of trust would have an impact on Gemalto's reputation or its business value and hence would have served as a deterring factor for such misbehavior.

Our findings are based on the analysis of the ID card public-key certificates collected over the years from the public ID card certificate repository. The findings are presented as three separate studies performed over different periods of time. For each study we present the context and describe the process of how the flaws were identified and handled.

First, we discovered that several ID card certificates shared the same RSA public keys. After further investigation we found that the affected ID cards also shared the same private keys. The discovery of duplicate private keys suggested that contrary to the security requirements, the ID card manufacturer had generated keys outside of the card. We obtained convincing evidence that most of the ID card keys had been generated in the card, while a specific set of keys produced in

the ID card renewal process had been generated outside the card. Our conclusion is that this violation was likely motivated by performance reasons.

We also found a separate fault in the ID card manufacturing process that resulted in corrupted RSA public key moduli being included in the certificates. In one instance we were able to fully factorize the affected key demonstrating the security impact of the fault. We analyzed the possible causes for the corruption and discussed prevention and detection measures.

The rest of the paper is organized as follows. Section 2 introduces the Estonian ID card ecosystem and smart card chip platforms used over the years. Section 3 gives an overview of related security flaws the Estonian ID card has experienced. The next three sections describe the main findings of this paper. Finally, Section 7 concludes the paper.

2 Estonian ID card

2.1 Cryptographic functionality

From its introduction in 2002 until now, the core cryptographic 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 with the card.

Authentication key. The authentication key is used to log into e-services by providing a signature in the TLS client certificate authentication process [3]. This key can also be used to decrypt documents encrypted for the cardholder [4]. Signature and decryption operations with this key have to be authorized using the 4-digit PIN1 code.

Digital signature key. The digital signature key is used to give legally binding digital signatures that under eIDAS [5] are recognized as qualified electronic signatures. Each signature operation with the key has to be authorized using the 5-digit PIN2 code.

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 allows to reset PIN codes in case the cardholder forgets them, generate new keys, write new certificates, and even reinstall the whole smart card applet if needed.

2.2 Parties involved

ID cards are identity documents issued by the state. The Police and Border Guard Board (Politsei- ja Piirivalveamet – PPA) is the authority responsible for procurement of ID card manufacturing services and the issuance of identity documents.

From the introduction of ID cards in 2002, the manufacturing and personalization of cards was performed by Trüb Baltic AS. In February 2015, Trüb Baltic AS with their parent company Trüb AG was acquired by Gemalto. As of the end of 2018, the ID cards have been manufactured by Oberthur (now known as IDEMIA).

The ID card certificates are issued by the privately-owned Estonian Certificate Authority (CA) SK ID Solutions AS (hereinafter – SK). According to eIDAS terminology, SK is a qualified trust service provider issuing qualified certificates. SK is a subcontractor of the card manufacturer.

The Estonian Information System Authority (Riigi Infosüsteemi Amet – RIA) is the state agency responsible for coordination and development of electronic identity and cyber security. Among other tasks, RIA organizes the development of ID card client-side software.

2.3 Chip platforms and document types

In this section, we chronologically introduce smart card platforms used over the years and the corresponding identity document types. We use the generic term ID card to refer to all identity document types covered. The SIM card-based digital identity card, in a Mobile-ID format, is not covered in this work.

2.3.1 MICARDO

In 2002, Estonia introduced the *identity card*, a mandatory identity document for all Estonian residents aged 15 and above. The electronic functionality of the card was implemented on top of smart card operating system MICARDO Public 2.1 [6]. The smart card interface is documented in the EstEID specification [7], which later became a national standard [8]. MICARDO-powered ID cards were issued from 2002 to 2011 (Figure 1). The platform is limited to 1024-bit RSA keys.

Figure 1: MICARDO-powered *identity card* issued from 2002-01-01 to 2010-12-31 [9]

2.3.2 MULTOS

In October 2010, a *digital identity card* 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 *digital identity card* is to provide a backup solution in the event the cardholder's *identity card* cannot be

used. The card is powered by MULTOS I4E platform by KeyCorp [10]. The MULTOS applet has been developed to mimic the MICARDO interface described in the EstEID specification. MULTOS-powered cards were issued until December 2014 (Figure 2). The platform is limited to 1024-bit RSA keys.

Figure 2: MULTOS-powered *digital identity card* issued from 2010-10-01 to 2014-11-30 [9]

2.3.3 jTOP SLE66

In 2011, the manufacturing of *identity cards* switched to a new chip platform implemented on top of Infineon's product JCLX80jTOP20ID masked on a SLE66CX800PE chip [11] (Figure 3). The card runs jTOP (Java Trusted Open Platform) JavaCard operating system developed by Trusted Logic. The EstEID functionality is implemented in the JavaCard applet. The platform uses 2048-bit RSA keys. With the introduction of the jTOP SLE66 platform, the *residence permit card* was introduced (Figure 4). This card is issued to non-EU third-country nationals residing in Estonia. The jTOP SLE66-powered ID cards were issued until the end of 2014.

Figure 3: jTOP SLE66/SLE78-powered *identity card* issued from 2011-01-01 [9]

Figure 4: jTOP SLE66/SLE78-powered *residence permit card* issued from 2011-01-01 [9]

2.3.4 jTOP SLE78

At the end of 2014, the production of *identity cards*, *residence permit cards* and *digital identity cards* switched to jTOP SLE78 platform. The visual design of *identity cards* and *residence permit cards* stayed the same (Figure 3 and 4), however, the visual appearance of *digital identity cards* became a bit more colorful (see Figure 5). The EstEID functionality was implemented in a JavaCard applet on top of Infineon's product SLJ52GCA080CL [12] masked on the SLE78CLX800P chip [13] that runs the jTOP JavaCard operating system developed by Trusted Logic. With the switch to jTOP SLE78 platform, the *e-resident's digital identity card* was introduced (Figure 5). This card is issued through the e-Residency program [14] to persons who are not residents of Estonia. In the beginning of 2017, the *diplomatic identity card* was introduced (Figure 6). This card is issued to persons with diplomatic status. Initially, the jTOP SLE78 platform used 2048-bit RSA keys, but due to the ROCA flaw (see Section 3), at the end of 2017, the switch to ECC keys using curve P-384 was made. The jTOP SLE78-powered ID cards were issued until the end of 2018. ID cards manufactured currently are powered by the chip platform supplied by IDEMIA (not covered in this work).

Figure 5: jTOP SLE78-powered *digital identity card* and *e-resident's digital identity card* issued from 2014-12-01 [9]

Figure 6: jTOP SLE78-powered *diplomatic identity card* issued from 2017 [15]

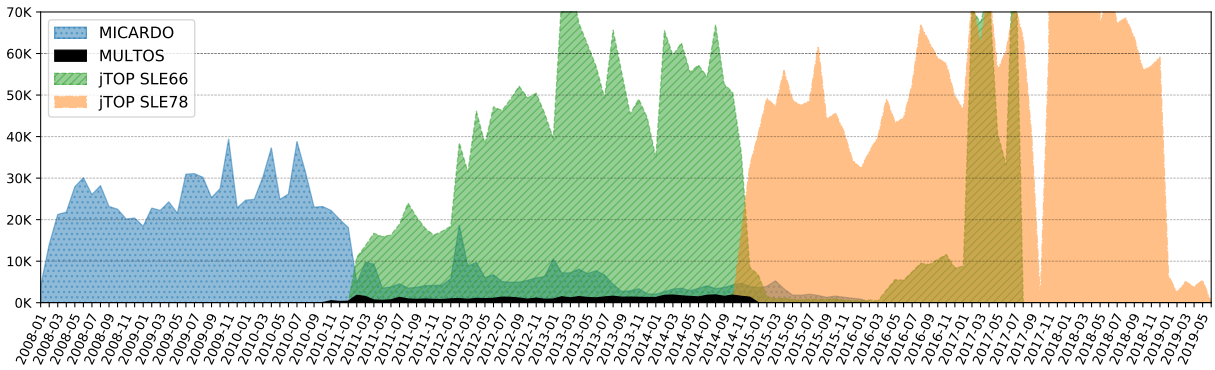


Figure 7: ID card certificates analyzed in this work (by issuance month)

2.4 Certificate repository

All valid ID card certificates issued by SK are available in the public LDAP directory `ldap://ldap.sk.ee` [16]. The publication of certificates is motivated by the document encryption use case, providing convenient means for senders to obtain public keys of recipients.

ID card 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 validity period of the certificate usually corresponds to the validity period of the identity document in which the corresponding private key resides.

2.5 Certificates analyzed in this work

Over the years, we have collected more than 7 million ID card certificates published in LDAP certificate repository. The certificate search in the repository is restricted to the personal ID code. However, since the search space for all possible personal ID codes is relatively small, over time certificates of all possible personal ID code holders could be crawled. Our certificate dataset is not complete, but we believe that it contains a representative sample of ID card certificates issued throughout the years. Figure 7 shows the distribution of ID card certificates in our dataset by issuance month (based on the certificate's `notBefore` field[1]) for different ID card platforms. The corresponding platforms have been determined by the certificate fields and properties of the public keys. 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 number of affected certificates.

We also collected certificate revocation information accumulated in publicly available CRLs [17]. The information 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 peculiarities of the ecosystem allowed us to deduce many important insights for this study.

[1]The `notBefore` field represents the time at which the certificate starts to be valid and usually corresponds to the time when the certificate was issued.

3 Related work

Over the 17 years of the Estonian ID card history, several ID card-related security flaws have been publicly disclosed.

More than 700 000 ID cards powered by the jTOP SLE78 platform were affected by Infineon's RSA key generation flaw (the ROCA flaw) [18]. The vulnerability in Infineon's proprietary RSA key generation algorithm allowed the factoring of 2048-bit RSA key in only 140.8 CPU-years. The discovery of this flaw in 2017 started the so-called Estonian ID card crisis, which was mitigated by switching to the ECC algorithm implemented by the platform and revoking vulnerable RSA certificates [19].

Publicly less noticed was a flaw in the jTOP SLE66 ID cards issued in 2011. Due to a publicly undisclosed flaw in EstEID JavaCard applet developed by the ID card manufacturer, 120 000 ID cards issued in 2011 were recalled [20]. While the authorities claimed that the card is secure and all transactions made with the card are fully reliable [20], later after the ROCA flaw broke out, it was disclosed in the media that the flaw in the 2011 ID cards was exploitable by having access to the card [21]. The context indicates that this may have been a type of PIN bypass flaw.

In 2002, it was discovered that PIN codes were printed in too dark, allowing for them to be seen through the PIN envelope [22]. Ironically, the same flaw in PIN envelopes was reintroduced by IDEMIA in 2018 after taking over the manufacturing of ID cards [23].

There have been incidents of including duplicate email addresses in certificates [24], issuing certificates with incorrectly encoded public keys [25], failing to revoke certificates of deceased persons [26] and others. Detailed analysis of these and other flaws related to the Estonian ID card are covered in [19].

4 Certificates with duplicate RSA public keys

In spring 2013, we discovered several certificate pairs in our dataset containing the same RSA public key. In most cases the public keys were shared between the authentication and digital signature certificates of the same ID card, however, in two occasions the same public key was shared between two different cardholders. The occurrence of such a fault could only have happened through a deep violation of the production processes, since each key pair is required to be unique even for the keys on the same ID card.

The set of 10 identified certificate pairs containing duplicate public key is listed in Table 1. All certificates have been issued for jTOP SLE66-powered ID cards. For each pair, the certificate issuance times have just a few seconds difference, indicating that the certificates were issued in parallel or close to each other. In most of the cases, the duplicate public keys were the result of the ID card renewal process, performed in the PPA customer service points, to replace the vulnerable applet for ID cards issued in 2011 (see Section 3).

No	Time of cert issuance	Type	Cardholder	Issuance	Expiry date	Revoked	Warranty
1	2012-11-06 15:35:09	sign	Ülle	PPA renewal	2016-07-07	2016-06-27	2014-10-09
	2012-11-06 15:35:46	auth	Toivo	PPA renewal	2016-07-04	2014-11-21	2014-10-09
2	2013-02-06 15:35:54	auth	Phillip	PPA renewal	2016-11-14	2015-05-04	2015-01-06
	2013-02-06 15:35:56	sign					
3	2013-02-07 12:18:34	auth	Sandra	PPA renewal	2016-01-02	expired	not issued
	2013-02-07 12:18:37	sign					
4	2013-02-19 09:09:58	auth	Nadiia	PPA renewal	2016-11-24	2016-11-08	2014-12-22
	2013-02-19 09:10:08	sign					
5	2013-02-25 09:33:17	auth	Moonika	PPA renewal	2016-08-22	2014-12-30	2014-12-22
	2013-02-25 09:33:29	sign					
6	2013-03-04 11:36:08	sign	Richard	PPA renewal	2016-11-30	2014-10-13	2014-10-09
	2013-03-04 11:36:38	auth	Anu	PPA renewal	2016-08-12	2014-10-23	2014-10-09
7	2013-03-30 13:40:38	auth	Leili	initial	2018-03-26	2015-05-14	2014-12-22
	2013-03-30 13:40:40	sign					
8	2013-03-30 13:42:03	auth	Jaan	initial	2018-03-26	2014-12-30	2014-12-22
	2013-03-30 13:42:05	sign					
9	2013-04-15 09:16:11	auth	Liis	PPA renewal	2016-05-06	expired	2014-12-22
	2013-04-15 09:16:28	sign					
10	2014-10-08 12:01:16	auth	Siim	initial	2019-10-07	2017-10-03	not issued
	2014-10-08 12:04:31	sign					

Table 1: Certificate pairs with duplicate public keys

4.1 Possible cause and impact

One explanation for these duplicate keys could be a poor source of randomness used in the on-card key generation process. However, we would expect such a failure to manifest randomly, independently of the time when the key is generated, since the ID card chip has no built-in time source that could be, for example, used to seed a pseudo-random number generator. Since the keys for the affected ID cards have been generated within an interval of a few seconds, this hypothesis can be safely rejected.

The close timing of the certificate issuance suggests that due to some software bug (such as race condition) a wrong public key was included in the certificate, i.e., the same public key was sent as a part of certificate signing request twice. This, however, would result in at least one of the certificates from the pair not being usable electronically, as the actual private key residing on the ID card would not correspond to the public key included in the certificate.

In the cases where the same public key is shared between the digital signature and authentication certificates of the same ID card, the risk is that the knowledge of only one PIN (PIN1 or PIN2 depending on which slot contains the corresponding private key) allows the card to be used for both purposes.

A more serious risk occurs in the two cases where the same public key is shared between different cardholders. For example, in case of pair 1, depending on whose ID card contains the corresponding private key, either Toivo can sign on behalf of Ülle, or Ülle can use her digital signature key to authenticate electronically as Toivo and decrypt files encrypted for Toivo (these use cases, however, would require the modification of the software).

It could not be excluded that the ID cards actually do contain duplicate private keys. However, if this was the case, the only credible explanation would be that contrary to the security requirements, the manufacturer had generated the keys outside the card and due to a flaw in the personalization process the same key was imported in two different ID cards/key slots.

4.2 Proof that ID cards share the same keys

Since we had a suspicion that private keys might be generated outside the ID card, we decided to investigate the shared public keys of the digital signature certificate of Ülle and authentication certificate of Toivo (pair 1).

In summer 2013, we were able to get in contact with Toivo, who informed us that his ID card was renewed in a PPA customer service point in Viljandi. He provided us cryptographic proof that both private keys in his ID card correspond to the public keys specified in the certificates. To demonstrate that Toivo's authentication private key can be successfully used to forge a digital signature of Ülle, with the assistance from Toivo[2], we created a proof-of-concept digital signature container in the name of Ülle (see Figure 8).

We did not manage to get in contact with Ülle to obtain a similar cryptographic proof from her ID card. In October 2014, we learned that the manufacturer had discovered the incident, since Toivo was invited to replace his ID card with a new one issued under warranty. The certificates of Ülle's ID card, however, remained valid. In spring 2015, we obtained confirmation from an Estonian service provider that Ülle had used the ID card for both authentication and signing in the e-service of the service provider. While this convinced us that her ID card contained the same private key, we still hoped to obtain cryptographic proof of that. In summer 2016, we managed to get in contact with Ülle's daughter who informed us that her mother used the card daily to sign banking transactions online, however, attempts to get in touch with Ülle herself did not succeed. Later we learned that her ID card was renewed in a PPA customer service point in Tallinn.

[2]Toivo was informed about the proof-of-concept signature forgery experiment, but not the nature of the flaw being exploited.

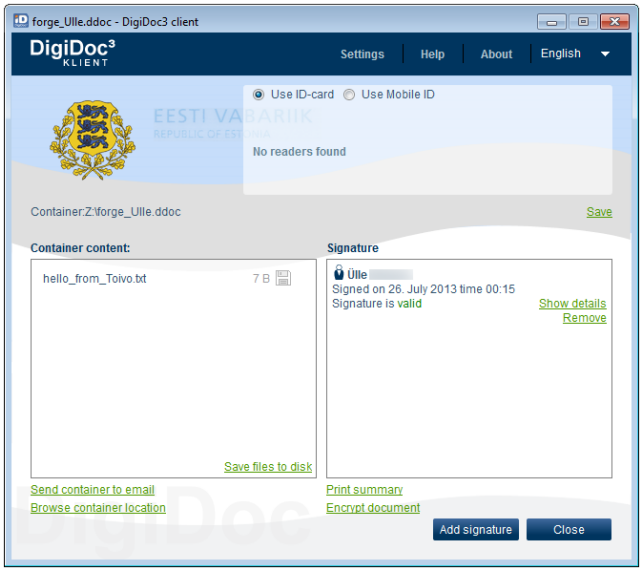


Figure 8: Digital signature of Ülle forged using the authentication key of Toivo

A similar (non-cryptographic) confirmation that both keys of the card are usable electronically was also obtained from Liis (pair 9).

The ability to successfully use both certificates involved in the duplicate certificate pair shows that the affected ID cards/key slots do share the same private keys that were apparently imported due to an error (e.g., race condition) in the ID card renewal process.

4.3 Incident response

In October 2014, at the latest, the manufacturer learned of the anomaly of duplicate keys. On 2014-10-09 a new ID card was produced for Toivo and on 2014-10-10 Toivo received an invitation from PPA to replace his ID card with a new one under warranty. The email stated that the ID card renewal on 2012-11-06 was unsuccessful and the card could not be used electronically (which actually was not true). For other cardholders the replacement cards were issued on 2014-10-09, 2014-12-22 and 2015-01-06 (the last column in Table 1). For unknown reasons the duplicate keys on the ID card of Sandra (pair 3) were missed, as for her the replacement ID card was not issued. Apparently, the cause of the flaw was not fully fixed and detection mechanisms were not implemented. As a result, a similar fault occured later again with the ID card of Siim (pair 10).

It is crucial to note that the incident was not handled as a security issue. The affected certificates were not revoked until the cardholders visited a PPA customer service point to receive the replacement card. Ülle was able to use her ID card until shortly before its expiration where it was then replaced. Liis informed us that the invitation from PPA did not reach her, therefore she kept using her ID card until its expiration.

In a meeting on 2017-02-06, we informed RIA about the case of Toivo and Ülle and the most likely explanation of keys being generated outside the ID card. At that time, we did not exclude the possibility that RIA and PPA may be well aware of the true reasons behind the flaw.

When approached by the authorities, the manufacturer responded that this was the old case already investigated in 2014 and that the mistake only occured with public keys. At the end of 2017, RIA ordered a follow-up study to determine whether any further evidence of key generation outside the ID card could be found [27]. Using statistical methods, strong evidence was found, that in the renewal process in PPA customer service points the keys were generated outside the ID card (see Section 5).

As we see in Table 1, the certificates with duplicate public keys were also found in 3 pairs of initially issued ID cards. These cases could be the result of a separate personalization fault where the cards actually do not contain duplicate keys. We urged RIA and PPA to investigate this, by using the database of OCSP certificate validity responses maintained by SK, to see whether the relying parties had requested validity confirmation of the involved certificates. From this it would be possible to infer whether the ID cards had been used successfully hence containing the keys specified in the certificates. We are not aware if this has been investigated.

5 Private keys generated outside the ID card

At the end of 2013, in the context of Snowden revelations, an opinion piece was published in Estonia [28] expressing concerns about authorities having copies of ID card private keys. The authorities rebutted the concerns [29], claiming that the recording of private keys is ruled out by the technological scheme used, i.e., the keys are generated inside the chip and the ID card is designed so that the private key itself never leaves the card.

Indeed, the security requirement of ID card key generation inside the chip has already been present in the ID card concept [30], has been documented in the EstEID technical specification (Section 4.1.5 in [7]), has been specified in SK certification policy according to which the CA is audited (Section 6.1.2 in [31]), and has also been present in the ID card manufacturing contract between the manufacturer and the state.

The rationale behind this requirement is that key generation inside the chip provides higher security. It is easier to ensure that copies are not created, rather than to make sure that all the copies have been irreversibly destroyed to eliminate potential misuse. For example, the Mobile-ID technology comes with extra risks, since it is documented that the keys for Mobile-ID are generated outside the chip (Section 6.1.1.3 in [32]).

In this section we describe our efforts to establish the true origins of the ID card private keys on each ID card platform.

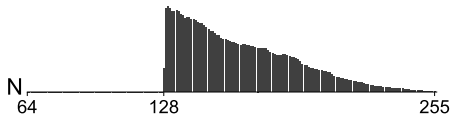
5.1 Finding the evidence

In 2016, Svenda et al. in their paper "The Million-Key Question – Investigating the Origins of RSA Public Keys" [33] described a method which can be used to infer from the RSA public key modulus some details about the algorithm used to generate the key. In particular, it was found that the most significant byte (MSB) of modulus N allows to establish the range from which primes p and q were selected. This range turned out to be different for different implementations of the RSA key generation algorithm. We used this and other techniques to verify whether the properties in the RSA keys from the ID card certificates match the properties of the key generation algorithm implemented by the ID card platform. To obtain reference keys, we generated and exported thousands of keys from each ID card platform (when it was possible), simultaneously measuring the time taken by the on-card key generation process.

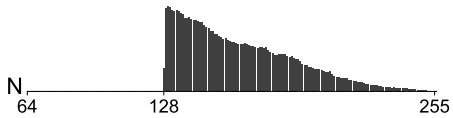
5.1.1 MICARDO

We found a configuration flaw in all MICARDO-powered ID cards that allowed us to perform card management operations with PIN2, without knowing the manufacturer's symmetric card management keys [19]. We used this to generate and export over a million 1024-bit RSA key pairs generated by the platform.

The MICARDO platform does not allow setting the value of the public exponent e. For each key the platform chooses a random public exponent e, either 2, 3 or 4 bytes in length, depending on the configuration. This peculiarity is visible in the certificates – for all, more than one million MICARDO-powered ID card certificates in our dataset, the public exponent value is random, no single value being over-represented.



(a) From the keys generated by the platform



(b) From the ID card certificates

Figure 9: MICARDO: distribution of the MSB of N

As we see in Figure 9, the distribution of the MSB of N from the keys generated by the MICARDO platform closely matches the keys from MICARDO-powered ID card certificates. Since the distribution of the MSB of N from the keys generated by MICARDO platform shows a unique pattern not observed in keys generated by any known software library (see Figure 12 in [33]), we can conclude that the keys in MICARDO-powered ID cards have been generated by the platform. We note, however, that our dataset does not have enough certificates issued in the period from 2002 to 2007 to draw definite conclusions about the keys generated in this period.

5.1.2 MULTOS

We did not have access to a non-personalized MULTOS platform, therefore we could not generate reference keys. In our dataset we have 29 262 certificates issued for MULTOS-powered ID cards. Figure 10 shows the distribution of the MSB values of these keys. The public keys have a random 4-byte public exponent, mimicking the non-standard behavior of MICARDO.

We cannot make conclusions about the origins of these keys. However, we see that these keys have not been generated by OpenSSL (non-FIPS), since moduli are not always congruent to 1 modulo 3 (see Section 4.2 in [33]).

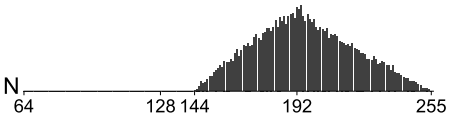


Figure 10: Distribution of the MSB of N from the MULTOS-powered ID card certificates

5.1.3 jTOP SLE66 (initially issued)

To export a million keys generated by jTOP SLE66 platform, we used blank jTOP SLE66 JavaCards. Since RSA key generation is implemented on the level of the JavaCard platform, access to the manufacturer's proprietary EstEID JavaCard applet was not required.

We observed that this CC certified [34] JavaCard platform has a functional bug. When asked to generate a 2048-bit RSA key, in 38% of the cases a 2047-bit key is returned. This is close to the theoretical ratio of 38.6294% when p and q are chosen uniformly from the distribution of 1024-bit primes. In order to generate an RSA modulus of required length, usually either the rejection sampling method is used to regenerate primes until their product is of the required length, or the primes are sampled making sure that k-bit prime is larger than $\sqrt{2} \cdot 2^{k-1}$ (see Section 3.2 in [33]). The distribution of the MSB of N from the keys generated by jTOP SLE66 platform is shown in Figure 11.

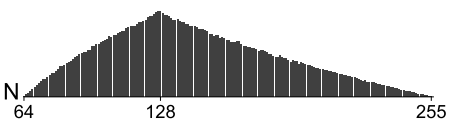
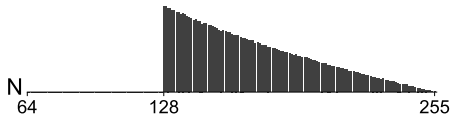


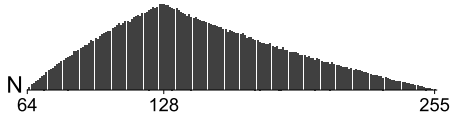
Figure 11: Distribution of the MSB of N for keys generated by jTOP SLE66 platform

The jTOP SLE66-powered ID cards were issued from 2011 until the end of 2014. All the certificates for initially issued ID cards contain public keys with random 4-byte public exponents, mimicking the non-standard behavior of MICARDO. JavaCard specification requires implementations to support arbitrary public exponent values for at least up to 4 bytes in length. We verified that jTOP SLE66 platform accepts and is able to generate RSA key pairs with any odd value e up to 4 bytes in length, therefore the keys from the certificates could have been generated by the platform.

We see that for ID cards issued in 2014 the distribution of the MSB matches the distribution as generated by the platform (Figure 12b). However, the ID cards issued before 2014 are missing the 2047-bit RSA keys (the MSB values smaller than 128) (Figure 12a). The exceptions are 3 cardholders who have been issued a certificate with a 2047-bit key in October 2013. These are two employees of SK and a person related to the manufacturer. We hypothesize that these cards were issued to test the changes in the manufacturing process before going into production.



(a) ID cards issued before 2014



(b) ID cards issued in 2014

Figure 12: Distribution of the MSB of N from initially issued jTOP SLE66-powered ID cards

Since the generation of 2047-bit RSA keys is an anomaly peculiar only to the jTOP SLE66 platform, we can conclude that for the ID cards issued in 2014 the keys have been generated by the platform.

By analyzing the time difference between the `notBefore` fields of the authentication and digital signature certificates, we found convincing evidence that both the keys for the ID cards issued before 2014 and for the ID cards issued in 2014 have been generated by the platform (see Section 5.2).

Apparently, the ID card manufacturing process before 2014 rejected 2047-bit keys to ensure that the certificates contained standards-compliant 2048-bit keys. Such a rejection of 2047-bit keys increased the key generation time by a factor of 1.63, hence increasing the average time of key generation (in case of random e) from 87 to 141 seconds. The slower key generation time may have been the cause for ending the practice of 2047-bit key rejection in 2014.

5.1.4 jTOP SLE66 (PPA renewal)

To fix the flaw in 2011 ID cards (see Section 3), the ID card manufacturer introduced the ID card renewal procedure which can be performed in the PPA customer service points. In the renewal process the old EstEID JavaCard applet was removed and a new applet with new keys and certificates was installed. The renewal was reused later in 2015 to fix an incident with duplicate email addresses specified in the certificates and in 2016 to fix certificates with incorrectly encoded public keys (see Section 3). The renewal of jTOP SLE66-powered ID cards was terminated on 2017-07-01. In total, more than 74 000 jTOP SLE66-powered ID cards were renewed in PPA customer service points.

In contrast to initially issued ID cards, the keys renewed in PPA customer service points have public exponent e set to 65537. These keys show an MSB distribution that is completely different from the keys generated by jTOP SLE66 platform (see Figure 13). Such a distribution is the result of setting the two most significant bits of p and q to 11_2 (see Section 3.2.2 in [33]).

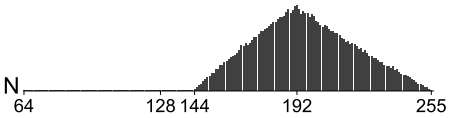


Figure 13: Distribution of the MSB of N from jTOP SLE66-powered ID cards renewed in PPA customer service points

In theory, the EstEID applet version installed in the PPA customer service points could have regenerated the keys until the two most significant bits of p and q were 11_2. This, however, would have increased the key generation time by a factor of 4, increasing the average time of key generation (in case of $e = 65537$) from 33 to 132 seconds. We see no legitimate explanation why this would be done, hence we conclude that these keys were generated outside the smart card. This was likely done to increase the key generation speed and hence the throughput of the PPA renewal service. In fact, the authorities could verify this by looking at the average time that was required to renew jTOP SLE66-powered ID card in PPA customer service point.

According to Table 7 in [33], there are several software libraries which generate keys by setting the two most significant bits of p and q to 11_2. These are: Botan 1.11.29, cryptlib 3.4.3, GPG Libgcrypt 1.6.5, LibTomCrypt 1.17, Nettle 3.2, OpenSSL FIPS 2.0.12, PGP SDK 4 and WolfSSL 3.9.0. OpenSSL 1.0.2g is excluded as the moduli generated by OpenSSL (non-FIPS) are always congruent to 1 modulo 3, which is not the case for the moduli observed in the certificates.

5.1.5 jTOP SLE78

Since the jTOP SLE78 platform was affected by the ROCA flaw (Section 3), it is possible to use the method published in [18] to verify whether the certificates issued for jTOP SLE78-powered ID cards contain keys affected by the ROCA flaw. The method has no false negatives, and the rate of false positives for 2048-bit RSA key is negligible (1 in 2^{713}).

Verification showed that the RSA keys have indeed been generated by the platform. This includes all keys – initially issued, remotely renewed and the keys renewed in PPA customer service points. There were, however, 23 keys that did not have the structure of the vulnerable keys. The possible causes for these anomalous keys are analyzed in Section 6.

5.2 Inferring key generation time from certificate issuance time

While modern computers are able to generate 2048-bit RSA keys in less than a second, RSA key generation in smart card chips requires tens of seconds on average. Since the time spent for key generation can be used to deduce whether the keys have been generated by the slow on-card key generation process, we decided to investigate whether the time spent to generate the keys can be observed from the timing of the certificate issuance.

During the ID card personalization process, if the certificate signing request is submitted to the CA right after the particular (authentication or digital signature) key pair is generated, the time difference between the `notBefore` field of the first and the second ID card certificate will include the time spent on the generation of the second key pair. On the other hand, if in the personalization process the certificate signing requests are submitted together after both key pairs have been generated, the difference in the `notBefore` dates of the certificates will not include the key generation time. To our knowledge this is the first work proposing the use of certificate validity dates as a side-channel to infer key generation time.

We grouped the certificates into pairs belonging to the same ID card if they were issued to the same cardholder in a 24-hour window for the same type of identity document, and looked at the distribution of time differences in `notBefore` validity date.

5.2.1 MICARDO

For all initially issued MICARDO certificates the time part of the `notBefore` validity date in the certificates is set to '00:00:00'. For certificates issued in the certificate renewal process, the `notBefore` field contains different values which seem to correspond to the actual time when the certificates were issued by the CA. The generation of a 1024-bit RSA key on MICARDO platform takes around 15 seconds on average. However, the average time difference between certificate issuance in each month is below 4 seconds. This is, however, expected as certificates are issued after both key pairs have been generated in the MICARDO certificate renewal process.

5.2.2 MULTOS

All certificates for MULTOS-powered ID cards have different values in the `notBefore` field, which likely correspond to the time the certificates were issued. However, the time difference between certificate issuance is a few seconds at best. This is expected because the MULTOS platform was used solely for *digital identity cards*, which are distributed to PPA customer service points with the keys pre-generated (see Section 6.1.2.1 in [35]).

5.2.3 jTOP SLE66 (initially issued)

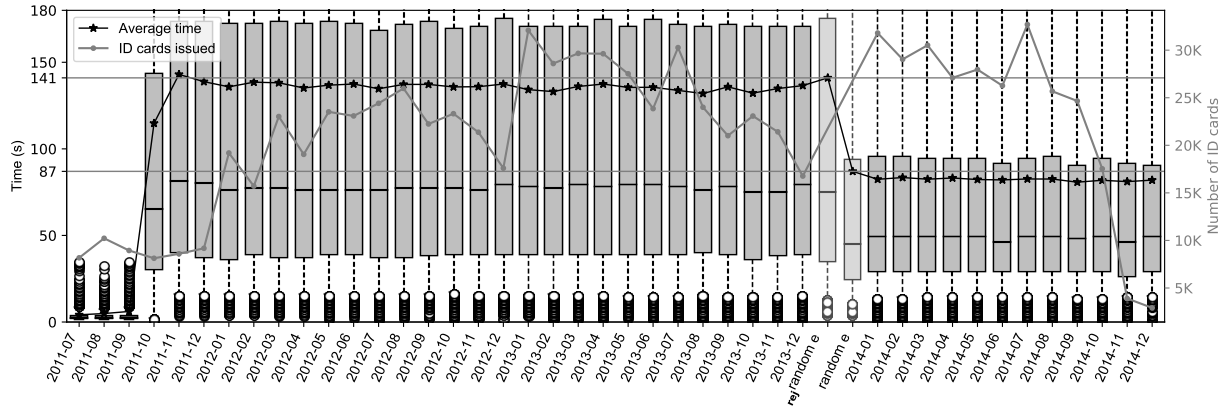
For the jTOP SLE66-powered ID cards issued up to 2011-07-09 the time part of the `notBefore` validity date in the certificates is set to '00:00:00'. However, starting from 2011-07-11, the `notBefore` date contains different time values which seem to correspond to the time the certificates were issued.

The ID cards with a certificate issuance time difference larger than 2 hours were excluded from the analysis. There were less than 0.32% of such ID cards in each month. These cases are possibly the result of an interrupted card personalization process that was completed at a later time.

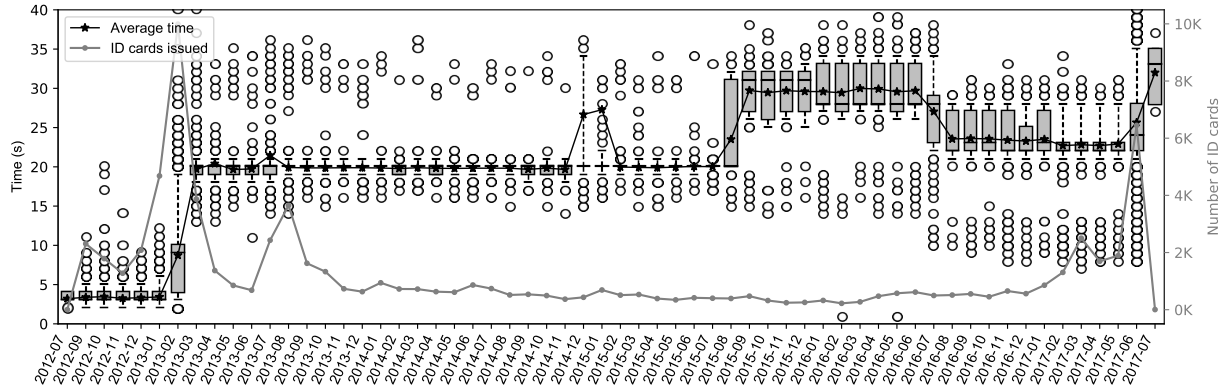
The distributions of time differences between issuance of the first and the second certificate grouped by month are shown in Figure 14a (the outliers in the box plots cover $< 5\%$ and $> 95\%$ percentiles). Before 2011-10-06, the time difference between certificate issuance is minimal with the authentication certificate being the first issued certificate close to half of the time. Starting from 2011-10-06, the authentication certificate is the first issued certificate at least 99.88% of the time and the average time difference between the certificate issuance increases significantly.

We see that the distribution of time differences very closely matches the key generation time distribution of the jTOP SLE66 on-card key generation. That is, the distributions observed from November 2011 to January 2014 match the distribution of the RSA on-card key generation when a random public exponent e is used and the key is regenerated when the produced modulus is 2047 bits long (average time 141 seconds). The distributions observed from January 2014 in turn match the distribution of the RSA on-card key generation when a random public exponent e is used, but no rejection sampling method is applied (average time 87 seconds).

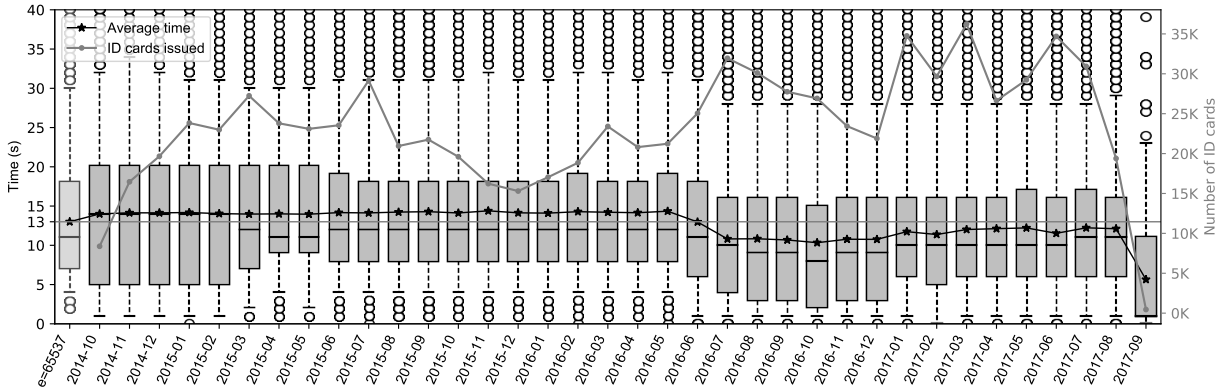
The timing observed supports the hypothesis that the keys on the ID cards issued before 2014 and in 2014 have been generated by the jTOP SLE66 platform. The small time differences observed before 2011-10-06 do not allow us to make definitive conclusions about the origins of these keys, however, as the properties of these keys match the properties of keys issued after 2011-10-06, we are inclined to conclude that these keys have also been generated by the platform.



(a) jTOP SLE66-powered ID cards (initially issued)



(b) jTOP SLE66-powered ID cards (PPA renewal)



(c) jTOP SLE78-powered ID cards

Figure 14: Certificate issuance time differences for certificates from the same ID card (by month)

5.2.4 jTOP SLE66 (PPA renewal)

In Section 5.1.4 we already found that the keys for jTOP SLE66-powered ID cards renewed in PPA were not generated by the card. However, to not disregard any possible counterevidence, we also looked at the timing between the authentication and digital signature certificate issuance also for these ID cards (see Figure 14b). We see that the time difference between the issuance of the first and the second certificate varies only slightly. We see that in 2013-02, 2015-08 and 2016-07, some changes were introduced in the PPA renewal process which caused a change in the certificate issuance time differences. Since the time difference is not close to zero and the authentication certificate is the first issued certificate 99.79% of the time, we tend to conclude that the time differences observed include the time spent on key generation and import, and possibly certificate loading in the ID card.

5.2.5 jTOP SLE78

The timing between the authentication and digital signature certificate issuance for jTOP SLE78-powered ID cards is shown in Figure 14c. The *digital identity card* certificates were excluded from the analysis. We see that the timing of certificate issuance matches the distribution of key generation time by jTOP SLE78 platform when $e = 65537$ (average time 13 seconds). This confirms the findings of Section 5.1.5. The average time below 13 seconds, starting from 2016-06, is explained by the introduction of remote ID card renewal on 2016-06-22. In the remote renewal process the certificates are issued after both key pairs have been generated by the card.

5.3 Discussion

The illicit practice of key importing in jTOP SLE66-powered ID card renewals could not have been accident. The EstEID applet had to be specially programmed to implement such a key import functionality.

The fact that the ID card manufacturer was able to use this forbidden feature without it being discovered for years, leads us to the corollary that in an analogous manner the manufacturer could have used the key export feature, retrieving the private keys after they were generated by the chip. It is not clear to what extent the strict industry rules could have been violated.

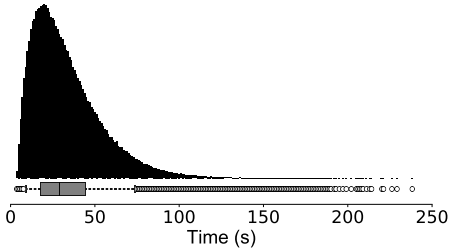
Large scale abuse of signature keys would be hard to keep secret, while abuse of decryption keys would not. We hope that the intent of the manufacturer was not malicious, and this illicit practice was motivated only by the need to increase the throughput of the PPA renewal service.

It is not clear whether the manufacturer initially understood that generating keys with a random public exponent increases the average key generation time from 33 to 87 seconds (see Figure 15 for distribution). The increase is due to the candidate primes p and q having a larger probability of not being suitable, since a randomly selected public exponent e is likely to have small prime divisors. The rejection of 2047-bit RSA keys increased the average key generation time even more – to 141 seconds. The generation of both ID card key pairs alone, would have extended the renewal process by approximately five minutes on average, and, in worst-case scenarios, even more time as shown in Figure 15b.

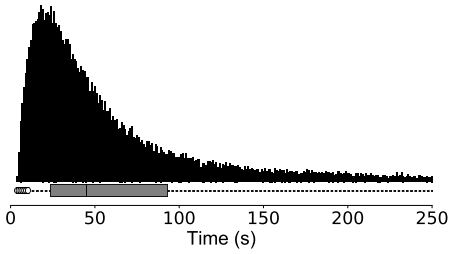
For the jTOP SLE78-powered ID cards the worthless practice of using a random public exponent was ended. The average time of 13 seconds (see Figure 16 for distribution) was deemed to be acceptable in the initial key generation process as well as for ID card renewal in PPA customer service points. Later, the switch to ECC using curve P-384 decreased the on-card key generation time to 0.37 seconds on average.

The fact that the same key was imported in two different ID cards renewed in different PPA customer service points suggests that the keys were generated in the manufacturer's backend and imported in the ID card over the Internet. Even if the keys were sent over an end-to-end encrypted channel, the logs and the symmetric card management keys could be used by the manufacturer to recover imported private keys.

The manufacturer's unauthorized modification of the EstEID applet also has far-reaching implications on the validity of digital signatures made with the affected platform. Since this modified version of the EstEID applet never passed the secure-signature-creation device (SSCD) conformity assessment as required by the eSignature Directive 1999/93/EC [36], this ID card platform never had the SSCD status, which is the legal prerequisite for a digital signature to have handwritten signature status.



(a) $e = 65537$



(b) Random 4-byte e

Figure 15: jTOP SLE66: time distribution of 2048-bit RSA key generation (CRT form)

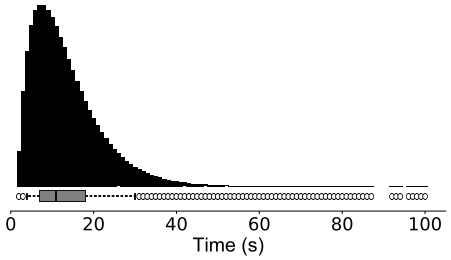


Figure 16: jTOP SLE78: time distribution of 2048-bit RSA key generation (CRT form, $e = 65537$)

5.4 Incident response

After receiving our analysis, the authorities decided to recall the jTOP SLE66-powered ID cards renewed in PPA customer service points. From more than 74 000 renewed ID cards, only 12 500 were still valid.

On 2018-05-17, PPA went public announcing that 12 500 ID cards did not meet the security requirements, because their private keys had been generated outside the chip. These cards would be replaced under warranty and on 2018-06-01 the affected certificates would be revoked. [37]

On the same day, the affected cardholders received email notification to apply for the replacement. The cardholders had to respond, specifying the PPA customer service point where they would collect the new card. As a replacement, the cardholders received jTOP SLE78-powered ID cards with the same expiration date as the original. The replacement card, however, was not issued if the original expiration date was in less than three months. [37, 38]

On 2018-06-01, the certificates of 11 100 non-replaced ID cards were revoked, with 3 300 cardholders waiting to receive the replacement card [39]. The legal basis for certificate revocation was the EITSETA act [40], clause 19 (4) 2): "a possibility of using the private key corresponding to a public key contained in the certificate without the consent of the certificate holder" [41].

We note that even if the authorities had not considered this to be a security issue, there was a non-compliance issue, and hence the certificates could also have been revoked based on the EITSETA act clause 19 (4) 12): "appearance of an error in the certificate or in the data entered in the certificate", as the certificates had not been issued in accordance with the CA's certificate policy referenced in the certificate.

5.5 Claim against the manufacturer

According to PPA, in the internal audit it was found that the state had not asked and was not aware that Gemalto was generating keys outside the card [41]. After receiving our initial analysis, PPA submitted a claim to Gemalto. A response from Gemalto denying violation, however, was only received the night before the announcement for the ID card recall [42].

On 2018-05-18, the day after PPA's announcement, Gemalto announced that PPA's statements were a surprise, and that it had fulfilled the ID card contract and the obligations agreed therein [42]. The state was then put in an unfortunate situation. It was evident that the ID card manufacturer could not be trusted, but contractually they had to produce ID cards until the end of 2018, when the new manufacturer IDEMIA would take over.

On 2018-09-26, after failing to reach an agreement, PPA brought Gemalto to court demanding a contractual penalty in the amount of 152 million EUR for generating keys outside the chip [2]. This claim, however, has to be viewed in context with other ongoing litigations with Gemalto – the PPA's claim of 300 000 EUR from Gemalto for their failure to inform the state about the ROCA flaw [43] and Gemalto's appeal about the results of ID card procurement [44]. The court decisions on these cases are yet to be seen.

6 Certificates with corrupted RSA public keys

In 2012, Heninger et al. [45] published an efficient method for testing RSA public keys for shared prime factors. This method was used to find that 103 RSA keys from Taiwan's Citizen Digital Certificates share prime factors [46]. We used the same method to test the RSA public keys from Estonian ID card certificates for shared prime factors and found several small common factors (e.g., 3, 5, 7) in the output of pairwise GCD computation. By using trial division with small primes we found 14 certificates whose public key moduli could be divided by one or several small factors. Since the public key modulus of 2048-bit RSA is generated by multiplying two distinct random 1024-bit primes, the public key moduli included in the certificates evidently had been corrupted. This corruption seemed to only affect the jTOP SLE78 platform, as all the certificates with the corrupted moduli had been issued for ID cards powered by the jTOP SLE78 platform.

We used the software utility YAFU [47] with the GMP-ECM implementation of the elliptic curve method (ECM) to test all RSA keys in our dataset for small factors. The keys were tested up to t-level[3] t20. This, however, did not find any additional corrupted keys. Two of the corrupted keys had an obvious anomaly – the length of the modulus was 2040 bits. We found one more anomalous 2040-bit modulus in our dataset and by applying more ECM testing to it (about t40) we were able to find a 132-bit prime factor. Later, when Nemec et al. [18] published a method to detect moduli generated by the vulnerable Infineon's key generation algorithm, we were able to identify 8 more presumably corrupted moduli. These were discovered when we observed that these certificates, which according to the certificate revocation date, had been revoked due to the ROCA flaw and hence had been issued for jTOP SLE78-powered ID cards, did not have the structure of ROCA keys. The full set of 23 identified certificates is listed in Table 2.

6.1 Full factorization

The issuance of ID card certificates with corrupted public key moduli means that the cardholders of these ID cards will not be able to use the cryptographic functionality, since the private key that resides in their ID card does not correspond to the public key in the certificate. The corruption of the public key, however, also has critical security consequences. By recovering all the prime factors from the corrupted modulus, it is possible to calculate the corresponding private exponent and perform private key operations with the key. If the modulus has 2048 bits, we can expect to factorize the corrupted modulus efficiently with a probability of 12 – 22% for an arbitrary corruption [48].

[3] T-level is the terminology used to express how much ECM testing the number has received. For instance, the work of t20 implies that the probability of a 20-digit factor being missed by ECM is about $exp(-1) = 37\%$.

No	Date of cert issuance	Cardholder (cert type)	N	Work	N-res	Factors (min / max)	Date of revocation	Corruption of N
1	2014-12-30 08:41:14	Toomas (auth)	2048	t45.76	2048	0	2017-11-03 23:59:59	?
2	2014-12-30 09:57:22	Raja (auth)	*2040*	t54.58	1713	3 (132-bit / 196-bit)	2015-08-26 16:37:53	117th byte missing
3	2014-12-30 16:03:43	Valentina (auth)	2048	t45.76	2048	0	2017-11-03 23:59:59	?
4	2014-12-30 16:05:23	Valentina (sign)	2048	t47.06	2048	0	2017-11-03 23:59:59	?
5	2015-01-05 11:25:19	Raisa (auth)	*2040*	t54.52	1958	4 (3-bit / 38-bit)	2017-06-09 14:07:57	27th byte missing
6	2015-01-27 13:48:40	Lennart (auth)	2048	t54.70	1937	4 (2-bit / 56-bit)	2016-07-01 09:36:57	64th byte changed
7	2015-02-19 09:19:21	Svetlana B. (sign)	2048	t47.47	–	7 (9-bit / 1762-bit)	2017-02-22 10:35:49	160th byte changed
8	2015-03-13 12:27:40	Imre (auth)	2048	t54.55	1895	6 (2-bit / 81-bit)	2015-04-06 13:54:33	?
9	2015-03-13 12:27:45	Imre (sign)	2048	t54.86	1757	7 (2-bit / 133-bit)	2015-04-06 13:54:33	?
10	2015-03-27 09:21:51	Vyacheslav (sign)	2048	t54.75	1808	9 (7-bit / 110-bit)	2017-06-09 14:17:20	71st byte changed
11	2015-06-01 12:07:45	Svetlana S. (auth)	*2040*	t54.54	1924	2 (25-bit / 92-bit)	2017-06-09 14:18:39	254th byte missing
12	2015-07-21 12:52:10	Rasmus (auth)	2048	t56.46	1844	4 (3-bit / 161-bit)	2017-06-09 14:21:50	254th byte changed
13	2015-08-06 14:18:44	Armand (sign)	2048	t54.42	1884	7 (11-bit / 50-bit)	2016-01-07 13:54:10	254th byte changed
14	2015-09-11 12:30:06	Paul (sign)	2048	t54.29	1973	4 (2-bit / 69-bit)	2017-06-09 14:23:09	230th byte changed
15	2015-11-04 11:27:25	Vambola (auth)	2048	t55.00	1604	6 (2-bit / 172-bit)	2017-06-09 14:50:32	87th byte changed
16	2015-12-02 10:10:37	Erki (sign)	2048	t54.34	2011	2 (2-bit / 35-bit)	2017-06-09 14:51:51	254th byte changed
17	2016-01-18 09:07:15	Pentti (auth)	2048	t46.44	2048	0	2017-11-03 23:59:59	?
18	2016-05-10 10:13:54	Laura (auth)	2048	t56.49	2002	5 (3-bit / 17-bit)	2017-06-09 14:53:29	92nd byte changed
19	2016-06-20 10:29:55	Ilja (auth)	2048	t54.58	1819	9 (2-bit / 124-bit)	2017-06-09 14:54:41	128th byte changed
20	2017-06-16 14:13:04	Vladislav (auth)	2048	t45.76	2048	0	2017-11-03 23:59:59	MSB as a minimum
21	2017-06-16 14:13:26	Vladislav (sign)	2048	t45.99	2048	0	2017-11-03 23:59:59	?
22	2017-06-16 16:28:30	Pirgit (auth)	2048	t45.86	2048	0	2017-11-03 23:59:59	MSB as a minimum
23	2017-06-16 16:28:55	Pirgit (sign)	2048	t45.73	2048	0	2017-11-03 23:59:59	MSB as a minimum

Table 2: Corrupted public keys from jTOP SLE78-powered ID card certificates. *N:* modulus length in bits. *Work:* amount of work done to factorize modulus. *N-res:* residual length of modulus after known factors removed. *Factors:* number of factors found and length of minimal / maximal factor found.

We were able to fully factorize one of these corrupted public keys – the key issued in digital signature certificate to Svetlana B. The modulus consisted of 7 factors (9-bit, 15-bit, 21-bit, 39-bit, 53-bit, 153-bit and 1762-bit). The probabilistic YAFU ECM factorization process took 60 hours (work t40.80) on a Core i5-6260U@1.8GHz CPU using 2 cores. We calculated the private exponent d in the RSA multi-prime setting and, as a proof-of-concept, successfully forged a digital signature on an empty file. The digital signature, as expected, passed validation by the state-provided digital signature verification software (see Figure 17).

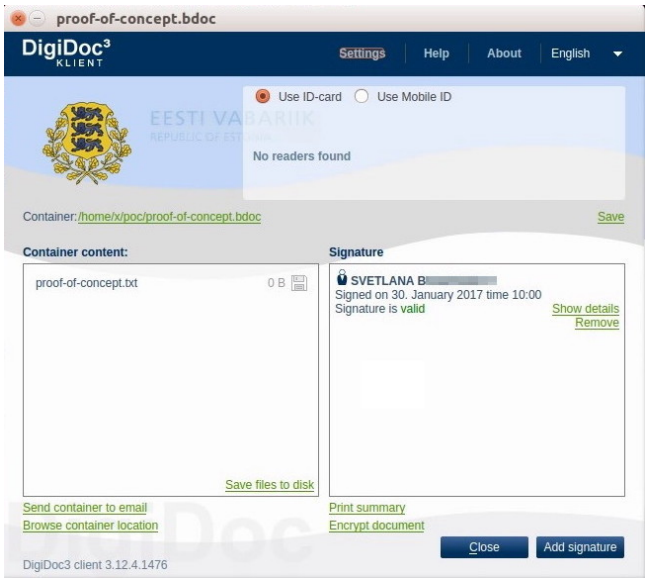


Figure 17: Digital signature forged using factorized key of Svetlana B.

6.2 Incident response

We informed RIA about the corrupted public keys and the successful factorization of Svetlana's key in the meeting on 2017-02-06. At that time, 8 out of 15 initially identified certificates were already revoked, possibly because the cardholders found that the cryptographic functionality did not work and applied for a new card.

On 2017-02-22, the certificates of Svetlana's ID card were suspended. In the meantime, RIA performed computations using their resources to verify our findings and to identify more corrupted keys in the full certificate database.

Only on 2017-06-09, the certificates of affected ID cards (including those of Svetlana) were revoked and PPA, under warranty, issued the cardholders new ID cards. Since the defect in the chip could not be excluded, the replacement ID card was also issued to Lennart, who on 2016-07-01, in a PPA customer service point, had already successfully renewed his keys using the renewal procedure intended for replacement of certificates with incorrectly encoded public keys.

Since the source of the corruption was not known, as a measure, trial division with small primes was implemented to discover corrupted moduli in the ID card production process. Unfortunately, as the corrupted key of Raja shows, the smallest factor of the corrupted key can be quite large and hence cannot be discovered by this method.

The risk was finally mitigated on 2017-11-03, when all RSA keys of jTOP SLE78-powered ID cards were revoked due to the discovery of the ROCA flaw, and manufacturing of jTOP SLE78-powered ID cards switched to ECC keys. A similar corruption cannot also be excluded for ECC keys, however, we have verified that all ECC keys in our dataset have EC points that are on the curve, and a random corruption resulting in the EC point that is on the curve will not provide advantage in deriving the corresponding private key.

It is important to note that the anomaly of 2040-bit RSA moduli had already been discovered by the manufacturer in August 2015, as new ID cards, with the expiration date of the original cards, were produced for 2 out of the 3 cardholders (Raja and Svetlana S.) on 2015-08-24 and 2015-09-04. For unknown reasons the case of Raisa was missed by the manufacturer. For her, the replacement ID card was only issued on 2017-06-09 after we informed the authorities of the corrupted public keys.

In 2015 the case was not handled as a security issue, since the certificates containing the corrupted keys were revoked only after the cardholders visited PPA to obtain the replacement card. This is yet another example of a serious anomaly in the ID card production process being mitigated by simply issuing a replacement ID card, without finding the root cause and without analyzing its scale and security impact.

6.3 Cause of data corruption

After Nemec et al. [18] published a method to detect moduli vulnerable to the ROCA attack, we tried to recover the corrupted moduli by modifying the modulus until the ROCA key detection test returned a positive result. We were able to successfully recover the corruption for 13 keys. We found that in the case of 2040-bit RSA moduli, the byte `0x81` (10000001_2) was missing in different positions for each modulus. In the case of 2048-bit RSA moduli, the byte `0x80` (10000000_2) in different positions for each modulus, was replaced with byte `0x00` (00000000_2). We did an exhaustive search modifying up to 4 bits in any bit position and modifying up to 3 bytes in any byte position, but were not able to recover corruption for any additional keys.

The corruption of the public key could have occured at any point up to its inclusion in the certificate. The corruption could have also occured due to a fault in the chip, for example, the chip failing to generate or correctly store the generated key under some specific operational conditions (such as temperature or voltage). We note, however, that these security chips are claimed to implement a set of measures to detect and prevent corruption even when the chip is under hostile environmental conditions [49].

We contacted Lennart, the owner of the affected ID card, who then shared a screenshot he had sent to the ID card customer support on 2016-01-15, showing a Mozilla Firefox 43.0.4 error message "Peer reports failure of signature verification or key exchange (`SSL_ERROR_DECRYPT_ERROR_ALERT`)" that appeared after trying to perform TLS client certificate authentication to a server. This error means that the ID card was able to produce a signature, but the server failed to verify the signature using the corrupted public key from the authentication certificate.

The signature was likely created using a valid private key, since the private key operations in CRT form do not use the modulus, but p and q. Had p or q been corrupted, the modulus (which is the product of p and q) would be more severely corrupted than a single bit change as we found above. The existence of valid RSA private keys on these cards does not exclude the possibility that the corruption of the modulus occured while the modulus was being read or written in the memory. The lost byte in the 2040-bit moduli case, however, is difficult to explain by memory corruption inside the chip.

In summer 2018, we contacted Infineon to ask whether they had heard of similar incidents with the product, and if not, would they completely rule out the possibility that the corruption could have ocurred due to a fault in the chip. To cite Infineon: "We are not aware of any process within our system (neither software nor hardware) that could result in such a change." [50].

Without any additional evidence available, we put forward the hypothesis that the corruption occured in the manufacturer's personalization line during the communication between the card and the reader. The lost byte in the case of 2040-bit moduli could be explained by retransmission failure of an incorrectly received byte in the APDU transmission over byte-oriented T=0 protocol. For 4 out of the 13 moduli, for which the corruption was recovered, we see that the 254th byte (the second most significant byte) of the moduli had been corrupted. In case of T=0 protocol, this would correspond to the second character transmitted after the procedure byte, assuming that a 256-byte modulus was returned by the chip in a single APDU response. Since the manufacturer's personalization line uses special-purpose hardware, such faults cannot be ruled out.

6.4 Prevention and detection measures

In traditional PKI deployments, the risk of including a corrupted public key in the certificate is mitigated by employing the PKCS#10 [51] standard that requires the certificate signing request (CSR) to be signed using the corresponding private key. In this case, the CA considered this requirement un-

necessary, relying on publicly undocumented organizational measures, which the manufacturer is required to implement to ensure the manufacturer's possession of the corresponding private key (Section 3.2.1.1 in [32]). As we now see, these unknown organizational measures, in practice, proved to be insufficient to provide the assurance a signed CSR would have provided.

Regardless of whether the moduli were corrupted inside the chip or in the transmission from the chip, the lesson here is that even for personalization performed in a trusted environment, the integrity of critical APDU data should be protected by transmitting it over a MAC-protected secure channel. Had this been the case, the source of the corruption would have been located with cryptographic precision.

To avoid this and other personalization faults where a wrong certificate or a certificate with an incorrect public key is loaded into the card, the card should perform an internal sign-verify sanity check to verify that the public key in the loaded certificate corresponds to the private key the card stores.

6.5 Valid RSA moduli from unknown source

We put forward the hypothesis that the 4 certificates issued for the ID cards of Vladislav and Pirgit, actually do contain valid RSA keys, but these keys have not been generated by the corresponding ID cards.

We base this hypothesis on the fact that contrary to all other certificates from Table 2, these certificates have been issued in the certificate renewal process in a PPA customer service point and not in the initial ID card personalization process. We see that 3 of these keys have MSBs of modulus that are not in the range 144–168 generated by jTOP SLE78 platform, but all 4 are in the range 144–255, which corresponds to the range for RSA keys generated by the manufacturer outside the ID card (Section 5.1.4).

It seems that due to some unknown failure, for these ID cards, the manufacturer's backend performed the renewal process assuming that they are powered by the jTOP SLE66 platform. The keys were generated and corresponding certificates were activated without detecting that the renewal process (including the key import) was not successful. Without any other evidence available, we will only be able to prove or disprove this hypothesis once factorization of these moduli becomes feasible.

7 Discussion and conclusions

All the issues, except for the manufacturer's decision to breach the security requirements by generating keys outside the ID card, could have been avoided by improved security engineering practices. While the flaws of duplicate public keys and corrupted public keys were discovered by the manufacturer, they were not sufficiently investigated and led to repeated incidents.

In the context of eIDAS, key management is the responsibility of the CA. The fact that the manufacturer's malpractice was not discovered in the internal and external audits of the CA shows the limited level of assurance these audits provide.

Compliance violations are also frequent issues among web browser CAs [52]. The browser vendors, however, require CAs to publish detailed reports of discovered violations thereby forcing CAs to investigate the incidents and improve their practices [53, 54]. In the event CAs show lack of trustworthiness, they can be distrusted by the browsers [55].

Similarly, the EU member states are required to establish supervisory bodies exercising state supervision over trust service providers' compliance to the requirements of eIDAS. In the case of the Estonian ID card, applying coercive measures might be hindered by the fact that the ID card manufacturer (and hence the CA) is the government's contractual partner on which the state is dependent until at least the 5-year ID card manufacturing contract expires. Nevertheless, the findings of this work show that the state cannot rely on the security guarantees provided in the ID card manufacturing contract and instead should seek effective means of oversight, either through public policy or the terms of ID card manufacturing contract.

Overall, the findings of this paper provide yet another example (see [18, 46] for others) that it is not sustainable to blindly trust the security of the manufacturing process. From the technical perspective, we suggest looking for fault-tolerant designs, for example, those involving threshold cryptography [56–58]. These designs should seek to provide effective means to prevent accidental failures and ensure that intentional malice would require higher conspiracy from the manufacturer and hence increase the risk of detection and attribution.

Unfortunately, we have not seen fundamental changes in the organization and execution of the Estonian ID card manufacturing process, therefore incidents like these, in one form or another, are destined to happen 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.

Acknowledgments

We thank Arne Ansper for the idea to use ROCA vulnerable moduli detection tests to recover the corrupted public keys, Alex Halderman for the initial ID card certificate dataset (December 2012), owners of the affected ID cards who provided information and participated in the experiments, and those persons who provided comments and feedback for this paper. This research was supported by the European Regional Development Fund through the Estonian Centre of Excellence in ICT Research under grant number EU48684.

References

[1] Estonian Information System Authority. ID card usage statistics inferred from queries to OCSP service, 2019.

[2] ERR News. Police claim 152 million from ID card producer Gemalto, September 2018. https://news.err.ee/864523/police-claim-152-million-from-id-card-producer-gemalto.

[3] 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.

[4] ID Help Centre. I've received encrypted document, how can I decrypt it?, October 2018. https://www.id.ee/index.php?id=38893.

[5] 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.

[6] ORGA Kartensysteme GmbH. MICARDO Public Chip Card Operating System Version 2.1 User Manual, September 2001. https://cybersec.ee/storage/mic21_druck.pdf.

[7] ID Süsteemide AS. EstEID card specification v2.01 (in Estonian), November 2002. http://www.id.ee/public/EstEID_Spetsifikatsioon_v2.01.pdf.

[8] Estonian Centre for Standardisation. EVS 827:2004 – Security chip – Application and interface, 2009. https://www.evs.ee/products/evs-827-2004.

[9] Estonian Police and Border Guard Board. Document descriptions issued by Police and Border Guard Board, November 2017. https://www2.politsei.ee/en/nouanded/dokumentide-naidised/.

[10] MULTOS. MULTOS Implementation Reports: Multos International I4E, December 2017. https://www.multos.com/products/approved_platforms/MIR/multos_international/i4e.

[11] Infineon. Product Brief: JCLX80JTOP20ID: Java Card™ Open Platform for Identification, 2008. https://cybersec.ee/storage/infineon_JCLX80JTOP20ID_product_brief.pdf.

[12] 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.

[13] Infineon. SLE 78CLX800P: Dual-interface and contactless security cryptocontroller, July 2012. https://cybersec.ee/storage/SPO_SLE%2078CLX800P_2012-07.pdf.

[14] Republic of Estonia. e-Residency, May 2019. https://e-resident.gov.ee/.

[15] 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 2017. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.C_.2017.279.01.0005.01.ENG.

[16] SK ID Solutions AS. LDAP directory service, May 2019. https://www.sk.ee/en/repository/ldap/.

[17] SK ID Solutions AS. Certificate Revocation Lists, May 2019. https://www.sk.ee/en/repository/CRL/.

[18] 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.

[19] Arnis Parsovs. *Estonian Electronic Identity Card and its Security Challenges*. PhD thesis (to be completed), University of Tartu, 2020.

[20] Police and Border Guard Board. The Police and Border Guard Board is renewing ID-Cards issued in 2011, September 2012. http://www.id.ee/index.php?id=35927.

[21] Delfi.ee. Estonia's largest PR operation: Saving the ID card (in Estonian), September 2017. https://ekspress.delfi.ee/kuum/eesti-suurim-pr-operatsioon-id-kaardi-paastmine?id=79478038.

[22] Delfi.ee. Security hole found in ID-card (in Estonian), May 2002. http://epl.delfi.ee/news/eesti/id-kaardis-leiti-turvaauk?id=50922213.

[23] ERR News. New ID card issue: Codes can be read using torch, without opening envelope, December 2018. https://news.err.ee/886313/new-id-card-issue-codes-can-be-read-using-torch-without-opening-envelope.

[24] Police and Border Guard Board. Eesti.ee email addresses of four thousand documents must be renewed (in Estonian), September 2015. https://sk.ee/uudised/neljal-tuhandel-dokumendil-tuleb-uuendada-eestiee-meiliaadressi.

[25] ERR News. 250,000 Estonian ID cards could be faulty, September 2015. https://news.err.ee/116849/250-000-estonian-id-cards-could-be-faulty.

[26] Geenius. The police discovered 15,000 faulty ID cards, over 300 have been used (in Estonian), June 2019. https://digi.geenius.ee/rubriik/uudis/politsei-avastas-15-000-veaga-id-kaarti-ule-300-on-kasutatud/.

[27] Postimees. New ID-card fault could have been intentional, May 2018. https://news.postimees.ee/4491312/new-id-card-fault-could-have-been-intentional.

[28] Otto de Voogd. The Flaw in the Estonian ID Card, October 2013. https://news.err.ee/108556/the-flaw-in-the-estonian-id-card.

[29] Agu Kivimägi. Rebuttal: Estonian ID Card Secure, Says Rep, November 2013. https://news.err.ee/108797/rebuttal-estonian-id-card-secure-says-rep.

[30] AS Sertifitseerimiskeskus. The Estonian ID Card and Digital Signature Concept: Principles and Solutions, March 2003. https://www.id.ee/public/The_Estonian_ID_Card_and_Digital_Signature_Concept.pdf.

[31] AS Sertifitseerimiskeskus. ESTEID Card Certification Policy, Version 5.0, January 2016. https://sk.ee/upload/files/SK-CP-ESTEID-20160125v5_0_en.pdf.

[32] AS Sertifitseerimiskeskus. ESTEID-SK Certification Practice Statement, Version 1.0, November 2016. https://sk.ee/upload/files/SK-CPS-ESTEID-EN-v1_0_20161101.pdf.

[33] 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.

[34] 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 2009. https://www.ssi.gouv.fr/certification_cc/carte-a-puce-jclx80jtop20id-java-trusted-open-platform-ifxv42-avec-patch-en-version-2-0-masquee-sur-composants-sle66clx800pe-et-sle66clx360pe/.

[35] AS Sertifitseerimiskeskus. ESTEID Card Certification Policy, Version 3.3, September 2012. https://sk.ee/upload/files/SK-CP-ESTEID-20120901v3_3_en.pdf.

[36] 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.

[37] Estonian Information System Authority. The Police and Border Guard Board will replace nearly 12,500 ID-cards which do not meet the security requirements, May 2018. https://www.ria.ee/en/news/police-and-border-guard-board-will-replace-nearly-12500-id-cards-which-do-not-meet-security.html.

[38] Police and Border Guard Board. Email notification from id@politsei.ee: Important information for ID card user, May 2018. https://cybersec.ee/storage/20180517_PPA_notification.txt.

[39] Postimees. Estonia cancels security certificates of 11,100 electronic ID-cards, June 2018. https://news.postimees.ee/4498133/estonia-cancels-security-certificates-of-11-100-electronic-id-cards.

[40] 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/ee/527102016001.

[41] Police and Border Guard Board. FAQ: ID cards not corresponding to security requirements (in Estonian), May 2018. http://web.archive.org/web/20180517170408/https://www.id.ee/?id=38558.

[42] Postimees. Gemalto denies breach of contract (in Estonian), May 2018. https://tehnika.postimees.ee/4490902/gemalto-eitab-lepingu-rikkumist.

[43] ERR News. PPA seeking EUR 300,000 from Gemalto, November 2018. https://news.err.ee/874973/ppa-seeking-300-000-from-gemalto.

[44] Postimees. Gemalto and PPA are carrying tens of millions after the war (in Estonian), March 2018. https://tehnika.postimees.ee/4432811/gemalto-ja-eesti-politsei-veavad-kumnete-miljonite-parast-vagikaigast.

[45] Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices. In *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, pages 205–220, Bellevue, WA, 2012. USENIX.

[46] Daniel J. Bernstein, Yun-An Chang, Chen-Mou Cheng, Li-Ping Chou, Nadia Heninger, Tanja Lange, and Nicko van Someren. Factoring RSA Keys from Certified Smart Cards: Coppersmith in the Wild. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, pages 341–360, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[47] Ben Buhrow. Yet Another Factorization Utility (YAFU), 2016. http://yafu.sourceforge.net/.

[48] Kaveh Razavi, Ben Gras, Erik Bosman, Bart Preneel, Cristiano Giuffrida, and Herbert Bos. Flip Feng Shui: Hammering a Needle in the Software Stack. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 1–18, Austin, TX, August 2016. USENIX Association.

[49] Infineon Technologies AG. Security Target M7820 A11 and M11 including optional Software Libraries RSA – EC – SHA-2 – Toolbox, Version 1.6, August 2012. https://www.commoncriteriaportal.org/files/epfiles/0829b_pdf.pdf.

[50] Wieland Fischer, Infineon Technologies AG. Personal communication, September 2018.

[51] M. Nystrom and B. Kaliski. PKCS #10: Certification Request Syntax Specification Version 1.7. RFC 2986 (Proposed Standard), November 2000. http://www.ietf.org/rfc/rfc2986.txt.

[52] Nicolas Serrano, Hilda Hadan, and L. Jean Camp. A Complete Study of P.K.I. (PKI's Known Incidents), July 2019. http://dx.doi.org/10.2139/ssrn.3425554.

[53] Mozilla. CA/Responding To An Incident, July 2019. https://wiki.mozilla.org/CA/Responding_To_An_Incident.

[54] Chromium. Root Certificate Policy, September 2019. https://www.chromium.org/Home/chromium-security/root-ca-policy.

[55] Mozilla. Distrust of Symantec TLS Certificates, March 2018. https://blog.mozilla.org/security/2018/03/12/distrust-symantec-tls-certificates/.

[56] 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.

[57] 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.

[58] 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.