# Developing requirements for the new encryption mechanisms in the Estonian eID infrastructure

Mart Oruaas[1] and Jan Willemson[1,2]

[1] Cybernetica AS, Mäealuse 2/1, 12618, Tallinn, Estonia
{mart.oruaas,jan.willemson}@cyber.ee
[2] STACC, Narva mnt 20, 51009 Tartu, Estonia

**Abstract.** After the Estonian ID-card crisis in 2017, it became apparent that the document encryption-decryption functionality built on top of the ID-card infrastructure requires major rethinking. This paper describes the starting points of this process and reviews some of the requirements that have been identified in the first phase. We study the main usage scenarios and process flows, and discuss the need to distinguish transport encryption from storage encryption.

**Keywords:** Electronic identity, Document encryption, Requirements engineering

## 1 Introduction

Digital communication has become an integral part of our everyday life. However, as such communication is mostly happening over open Internet, extra measures need to be taken if the messages require security in some sense.

In this paper, we will be concentrating on one aspect of security, namely confidentiality. Confidentiality protection is built into many communication protocols, and is typically achieved via encryption. Encryption, however, does not solve the message protection problem cleanly, but rather reduces it to key management. The latter, in turn, is not necessarily an easy task to solve.

One option for key distribution is to use cryptographic chip cards. In Estonia, for example, a specific instance of such a card (called ID-card) is even made mandatory since 2002, so virtually all the Estonian residents have one.

Even though message encryption was originally not a planned usage scenario of the ID-cards, it was added later. The current paper deals with some of the issues that the present solution (code-named CDOC) has, and discusses requirements for the next-generation message encryption framework (code-named CDOC2.0).

## 2 Background

In Estonia, there are currently three main electronic identity solutions.

– **ID-card**, first launched in 2002, was historically the first one and is the only compulsory form of identification in Estonia. It features a smart card chip capable of asymmetric cryptographic operations, and it comes with pre-generated and certified public-private key pairs. There is also digi-ID card that is digitally equivalent to ID-card, but can not be used for physical identification. For the sake of this paper we will use the name ID-card for both of them.
– **Mobile-ID**, first launched in 2007, relies on the mobile phone SIM card as the key storage and cryptographic coprocessor.
– **Smart-ID**, first launched in 2016, is a software-only solution making use of a specific cryptographic scheme [5] where the signature key is split between the mobile device and server.

As of May 2020, there were about 1.35 million active ID-cards, more than 230,000 mobile-ID users and more than 500,000 Smart-ID users[3].

All of the three solutions listed above provide two basic functionalities, *authentication* and *digital signature*. From the cryptographic point of view, both of them actually rely on digital signature primitive, but since their legal context is different, authentication and digital signing make use of different key pairs.

Originally, ID-cards and mobile-ID SIM cards used 1024-bit RSA, but around 2010 it became clear that advances in computing power would make it too weak at some point in foreseeable future [2], so it was gradually phased out in favour of RSA2048 on ID-cards and elliptic curve P256 on mobile-ID. After the ID-card crisis in 2017, ID-cards started to make use of the elliptic curve P384. Smart-ID originally used 4-kilobit multi-prime RSA, which is today replaced by 6-kilobit keys.

Besides digital signatures, there is another major application of public key cryptography, namely public key encryption. Since more and more Estonian citizens had access to strong cryptographic tokens in the form of ID-cards, the idea of providing encryption functionality emerged naturally. Along with the general growth of digital data exchange, also the need for private communication increased, so there was also demand for such a service.

Thus, in 2005, encryption-decryption functionality was added to the base software package of the Estonian ID-card [8]. Originally, the package featured a standalone application *DigiDoc Krüpto*, but today this functionality is integrated into the main DigiDoc4 Client (see Figure 1).

In order to send an encrypted file, recipient's authentication public key certificate is required. This is fetched from the LDAP server of Estonia's main Certification Authority by supplying the recipient's (public) Personal Identity Code. In fact, supporting the encryption service is the only reason why all the public key certificates are available in a public LDAP server in the first place [7].

Out of the three eID solutions available in Estonia, only the ID-card can be used for decryption, and there is a good technical reason for that. Namely, ID-card is the only form of eID directly connected to the user's computer via

[3] `https://www.id.ee/?lang=en`, last accessed May 7th, 2020.

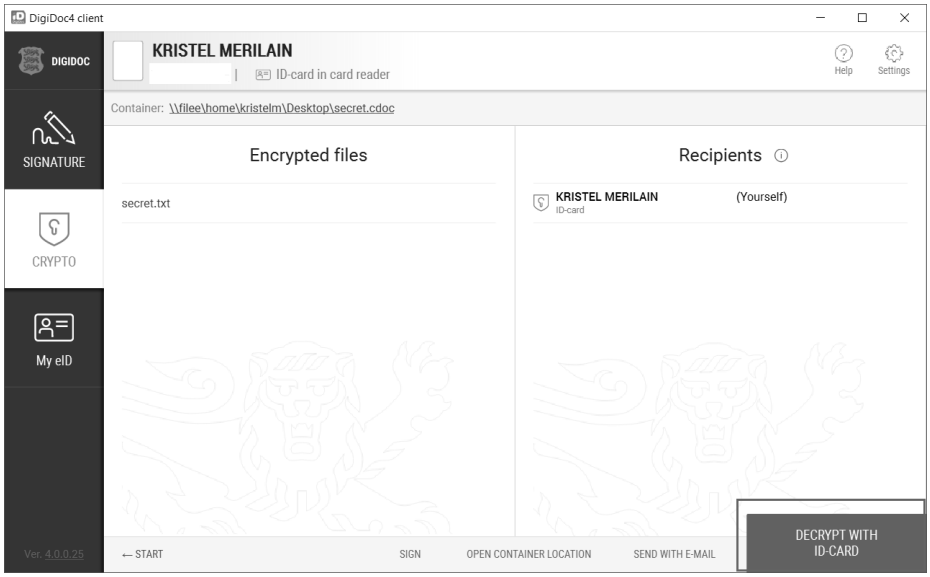


**Fig. 1.** DigiDoc4 Client with decryption functionality (Image source: `https://www.id.ee/`)

a wired card reader.[4] Mobile-ID and Smart-ID requests are, on the other hand, server-mediated. This is OK for signature-based applications (including authentication), but transferring decrypted material (essentially symmetric transport keys) via a server would break message confidentiality.

Estonian eID ecosystem has had a number of challenges throughout the years [7], but by far the most severe of them was the ROCA vulnerability found by a group of Czech researchers in 2017 [6]. It turned out that prime number generation algorithm used to create the RSA public moduli was using a very small seed space, hence it became feasible to factor the moduli by full inspection of that space. The lab experiments conducted by the Czech researchers were also independently verified in Estonia on a real ID-card belonging to Margus Arm, the leader of Estonian eID project[5].

As an aftermath of the ROCA case, several analyses were conducted [1, 4]. One of the conclusions drawn in these analyses was that out of the three main application scenarios of the ID-card – authentications, signing and encryption – it was encryption that was hit the hardest. Indeed, authentication is a short-lived protocol anyway, and once the affected keys are revoked, breaking them becomes useless. Digital signatures, on the other hand, are protected by time-

[4] As of 2019, also ID-cards with NFC interface started to roll out, but the support for NFC operations is still very limited

[5] `https://epl.delfi.ee/eesti/id-kaart-murti-lahti-ria-toestas-et-kara-id-kaardi-turvanorkuse-parast-polnud-asjata?id=81807683`, last accessed May 11th, 2020.

stamps. Thus, even if factoring some RSA modulus becomes possible in the future, we can still reasonably argue that at the time of signing such an attack was infeasible.

However, encryption is a scenario requiring long-term secrecy. In the most common use case, a document is encrypted using the recipient's public key, the cryptogram is saved into a special `.cdoc` container file and sent as an email attachment. The problem is that, from the end user perspective, it is very hard to control the actual chain of intermediate servers that the email together with its attachments travels through. Every one of them is in principle capable of retaining a copy of the `.cdoc` file until the time when breaking the underlying cryptography becomes feasible.

## 3 Setting requirements for the new encryption solution(s)

After the ROCA crisis, it became clear that, in order to prevent similar problems in future, better-designed document encryption features were needed for the Estonian eID ecosystem. Of course, the design phase needs to be preceded by an analysis phase. The respective project was initiated by the Estonian Information System Authority, seeking questions to the following broad questions.

- What are the common usage scenarios where CDOC infrastructure is used, and what are the corresponding user requirements?
- What are the core shortcomings of the present CDOC infrastructure, and how should a new system be built to avoid them?

The authors of the current paper were involved in this analysis project, and this paper presents a high-level summary of the final report of the project.

### 3.1 Usage scenarios

In order to map the main usage scenarios of document/data encryption in Estonian public and private sectors, a series of interviews were conducted with eight large organisations, presumably having the need for secure communication. The organisations included law enforcement agencies, state defence structures, health information management, certification authority and one of the main banks active in Estonia.

As a result, we identified the following scenarios.

- Inter-organisational communication requiring an extra layer of confidentiality
  - System administrators transferring passwords
  - Handling sensitive incidents
- Intra-organisational confidential communication
  - Law enforcement data exchange
  - Restricted access data exchange
  - Time-critical information required by courts and criminal investigations

    - Contracts, official data queries
- Confidential communication between the citizens and organisations
    - Some *ad hoc* health data transfer
    - Whistle blowing (from citizen to law enforcement)
    - Contracts, official data queries
    - Fines (from law enforcement to citizens)
    - Getting excerpts from bank statements
- Confidential document exchange between citizens
- Confidential document exchange with foreign partners

The last scenario was not directly in scope of the study, but potential interest was expressed by several interviewees. E.g., after establishment of the eIDAS framework in the European Union, it is a natural question to ask whether it could also be extended to confidential data exchange, much like a similar question arose in Estonia after the introduction of ID-card.

## 4 Transport cryptography *vs* storage cryptography

One of the problems identified during the analysis was that the same `.cdoc` files were used both for transport and long-term storage of confidential documents. Such usage brings along several issues.

1. Relying on the implicit heuristic assumption that CDOC is secure (without actually specifying the security definition), users kept the `.cdoc` files in their mailboxes, on unencrypted hard drives, removable media, etc. After the underlying cryptography turned out to be weak, it was very hard to control all the copies of the `.cdoc` files.
2. ID-card was efficiently used as the private decryption key management token. However, the ID-card does not (and should not) provide any key back-up mechanism, so when it becomes dysfunctional (e.g. due to physical ageing, being lost, etc.), all the `.cdoc` containers are left un-decryptable.

To address these issues, it is necessary to make a more clear distinction between transport and storage encryption. The first item above translates to the requirement that transport encryption should provide *forward-secrecy* [3], i.e. that compromise of the long-lived keys should not cause breaching the secrecy of all the exchanged messages. There are a few ways how this property can be achieved, but the most popular approach in practice is using ephemeral (i.e. short lived, temporary) encryption keys. For example, in the latest TLS version 1.3, only key agreement protocols resulting in ephemeral transport keys are allowed[6].

On the other hand, if a confidential document sent via the CDOC infrastructure needs to be accessed for a longer period of time, ephemeral encryption keys are not a good solution. Another aspect of information access arises from a number of organisational scenarios listed in Section 3.1. Currently, documents

[6] `https://tools.ietf.org/html/rfc8446`, last accessed May 14th, 2020.

can only be encrypted for decryption with ID-cards of physical persons. However, in many occasions, the intended recipient is not a person, but rather an organisation or a specific role within the organisation.

This aspect can be observed also in the long-term storage scenario. There are documents with legally required confidentiality periods lasting for decades. The person who once will possibly be filling the role that needs access to the document is perhaps not even born yet at the time the document gets stored.

Thus, the new CDOC2.0 infrastructure needs to make a clear distinction between the transport and storage encryption. The component where this distinction manifests itself most clearly is the key management. Ephemeral keys will be generated and maintained just for one session or for a short period of time, and after expiry they should be deleted. If a document needs longer-term access, it needs to be re-encrypted with a different key that is managed according to the needs and regulations of the specific organisation.

This re-encryption step is something that probably requires a paradigm shift from the users who are currently used to saving `.cdoc` files on their hard-drives, or just leaving them in mailboxes. In order to break this habit, CDOC2.0 should change the process flow in a way that such insecure behaviour would become impossible, or at least quite inconvenient.

## 5 Main classes of process flow

In fact, CDOC2.0 would anyway need to support several process for transport encryption, as the requirements of different usage scenarios differ. We have identified the following main types of process flows.

- **Ad hoc data exchange with low confidentiality requirements** where data exchange is not agreed upon beforehand, the data itself is not very sensitive and it may expire quite fast (e.g. situational updates).
- **Synchronous data exchange** where both of the communicating parties can be expected to be online at the same time.
- **Information exchange within an organisation**, where we can typically also assume some perimeter protection measures.
- **Highly confidential data exchange between different organisations**, where the communication occurs over open Internet, but on the other hand, the sender and recipient may be motivated to use methods that require extra efforts (say, specific ephemeral key generation and distribution).
- **Confidential data exchange between citizen and an organisation**, e.g. a bank or a law enforcement agency.

On one hand, the requirements of different scenarios are different, but on the other hand, so do the motivation and technical capabilities of the communicating parties. Thus, future DigiDoc client will need to provide a list of encryption options, and the sender will need to choose between them based on the concrete scenario.

Also, the long-term storage processes can be divided into two large categories depending on the needs and capabilities of the recipients.

- **Personal data access**, where just one person needs the decryption capability which is the case with personal data or the data for a one-man-organisation.
- **Role-based data access**, which is the case for larger organisations where different physical persons may be fulfilling a certain role at different points of time.

The needs of the first category can be satisfied with a number of readily-available solutions like encrypting the hard-drive of the user's main computer and backing up the key in one's physical safe deposit box.

For the second scenario, however, extra development is necessary. Large organisations already have existing document management systems (DMS), possibly also integrated with email exchange functionality. Assuming that CDOC2.0 containers are still mostly transferred via email, the following additional activities need to be enabled for a typical DMS.

1. The DMS must be capable of decrypting the CDOC2.0 container from the transport encryption scheme.
2. The person fulfilling the respective role must have access to the encryption functionality using an in-house long-term secure storage solution.
3. Later, a (possibly different) person fulfilling the role must have access to the in-house solution's decryption functionality.

There are a number of ways the encryption-decryption key management can be implemented in a company. There can be employee badges in the form of cryptographic chip cards, or there can be a dedicated in-house key distribution server. The specific implementation depends on the needs and capabilities of the organisation.

One interesting aspect to note is that the CDOC2.0 container format can be the same for both transport and storage process flows, but it is the key management process that makes all the difference.

## 6 Conclusions

Being initiated about 20 years ago, Estonian eID infrastructure has matured. It is used every day to provide authentication and signature functionality for public and private e-services.

However, there is also a third kind of cryptographic e-service required by the end users, namely document encryption. For the last 15 years it has been technically implemented as an add-on to authentication, but this has caused several problems (with the ROCA case of 2017 being the best known one).

The root of these problems lies within different requirements that authentication and encryption keys have. Authentication keys can be instantly revoked and hence potential vulnerabilities due to, say, key compromise are easy to limit. Decryption functionality, on the other hand, has to be maintained over a long period of time (years, perhaps even decades), and thus it is impossible to mitigate key compromises by just revoking them.

This means that encryption-decryption key management has to be designed much more thoroughly, and this is a process that the Estonian Information System Authority has initiated in 2019. The current paper summarised the main findings of the first, analysis and requirement gathering phase. The next steps will include design, development and deployment. All of these remain the subject of further work, parts of which have already started by the time of this writing (May 2020).

**Acknowledgements.** We are grateful to the Estonian Information System Authority for their initiative on the CDOC2.0 process. This paper has been supported by the Estonian Personal Research Grant number 920 and European Regional Development Fund through the grant number EU48684.

## References

1. Ansper, A., Buldas, A., Willemson, J.: Cryptographic algorithms lifecycle report (2018), Cybernetica research report A-101-9, https://www.ria.ee/sites/default/files/content-editors/publikatsioonid/cryptographic-algorithms-lifecycle-report-2017.pdf, last accessed May 11th 2020
2. Babbage, S., Catalano, D., Cid, C., de Weger, B., Dunkelman, O., Gehrmannand, C., Granboulan, L., Güneysu, T., Lange, T., Lenstra, A., Mitchell, C., Näslund, M., Nguyen, P., Paar, C., Paterson, K., Pelzl, J., Pornin, T., Preneel, B., Rechberger, C., Rijmen, V., Robshaw, M., Rupp, A., Schläffer, M., Vaudenay, S., Vercauteren, F., Ward, M.: ECRYPT II Yearly Report on Algorithms and Keysizes (2009-2010) (2010), https://www.ecrypt.eu.org/ecrypt2/documents/D.SPA.13.pdf, last accessed May 11th 2020
3. Boyd, C., Mathuria, A., Stebila, D.: Protocols for Authentication and Key Establishment, Second Edition. Information Security and Cryptography, Springer (2020). https://doi.org/10.1007/978-3-662-58146-9, https://doi.org/10.1007/978-3-662-58146-9
4. Buldas, A., Jung, M., Kuivjõgi, K., Tallinn, L., Osula, A.M., Ottis, R., Priisalu, J., Vaks, T.: ID-kaardi kaasuse õppetunnid. https://www.ria.ee/sites/default/files/content-editors/EID/id-kaardi_oppetunnid.pdf, in Estonian, last accessed May 11th 2020 (2018)
5. Buldas, A., Kalu, A., Laud, P., Oruaas, M.: Server-Supported RSA Signatures for Mobile Devices. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10492, pp. 315–333. Springer (2017). https://doi.org/10.1007/978-3-319-66402-6_19, https://doi.org/10.1007/978-3-319-66402-6_19
6. Nemec, M., Sýs, M., Svenda, P., Klinec, D., Matyas, V.: The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017. pp. 1631–1648. ACM (2017). https://doi.org/10.1145/3133956.3133969, https://doi.org/10.1145/3133956.3133969

7. Paršovs, A.: Estonian Electronic Identity Card and its Security Challenges. PhD thesis draft (2020)
8. Sinivee, V., Uukkivi, K.: Encrypted DigiDoc Format Specification (2012), https://www.id.ee/public/SK-CDOC-1.0-20120625_EN.pdf