Security of the proposed Mobile-ID document decryption feature

Arnis Parsovs
University of Tartu

July 22, 2022

1 Introduction

A company, B.EST Solutions, has proposed a concept for using Mobile-ID to decrypt documents [1]. The proposed solution is based on the design of the Estonian CDOC file format that is currently used to carry encrypted files that can be only decrypted by the respective holder of the Estonian ID card (see Section 2.9.1 in [2]). The CDOC encryption file format relies on a hybrid encryption scheme, where a symmetric transport key is used to encrypt the files stored in a CDOC container. The transport key is encrypted for each CDOC recipient with their public key. This ensures that only the holder of the corresponding private key can decrypt the transport key and hence the encrypted files stored in the CDOC file container.

The document of the proposed concept describes a method of how the encrypted transport key is delivered to the Mobile-ID SIM card for decryption and how the decrypted transport key is delivered back to the CDOC decryption software running on the Mobile-ID holder's computer. The central challenge addressed by the proposed scheme is how to securely deliver the decrypted transport key from the Mobile-ID holder's SIM card back to the Mobile-ID holder's computer, such that no third parties (i.e., the Mobile-ID service provider or mobile operator) could learn the decrypted transport key. In the proposed scheme this is achieved by the Mobile-ID SIM card re-encrypting the decrypted transport key using a public key of the CDOC decryption software. The CDOC decryption software generates a fresh ephemeral key pair for each Mobile-ID decryption transaction and sends the public key to the Mobile-ID SIM card together with the encrypted transport key. The core of the proposed scheme is depicted in Figure 1.

In this report we discuss the security aspects of the proposed solution.

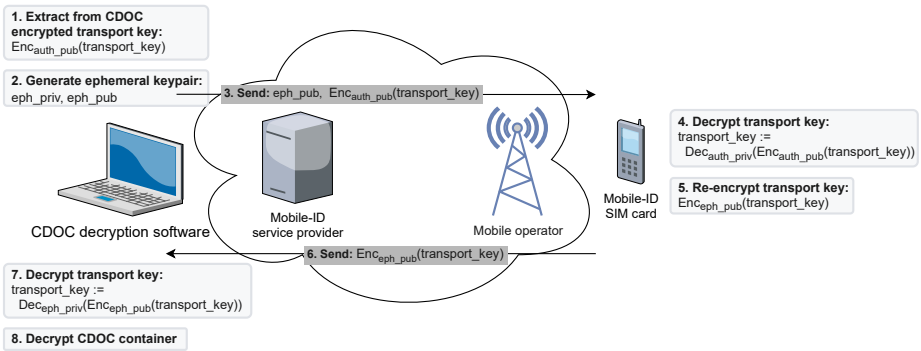

Figure 1: The proposed Mobile-ID document decryption scheme

2 Discussion

This section discusses our observations, security issues and possible improvements that we find important to highlight. To avoid analyzing different versions of the proposed scheme, each section is built on the assumption that the protocol changes suggested in the preceding sections have been implemented.

2.1 Security strength of the proposed schemes

The concept document proposes two schemes – one using the RSA cryptosystem and another using the ECC cryptosystem. The RSA-based scheme is proposed with 1024-bit and 2048-bit keys, while the ECC-based scheme is proposed with the curves P-256 and P-384. The security strength provided by the corresponding asymmetric keys is listed in Table 1.

Table 1: Security strength of the asymmetric keys (based on [3])

Key	Strength (in bits)
RSA-1024	80
RSA-2048	112
P-256	128
P-384	192

We note that data inside a CDOC container that confirms to the latest CDOC specification [4] is encrypted using 256-bit AES. Therefore, the CDOC file encryption format provides up to 256 bits of security, depending on the security strength of the asymmetric keys of the recipients.

We would like to point out that today 1024-bit RSA is estimated to provide only 80 bits of security and therefore has been deprecated. If two RSA key lengths have to be supported, we instead suggest supporting 3072-bit RSA that is estimated to provide 128 bits of security.

2.2 Size of the transmitted data

The concept document specifies the size of the data that must be transmitted in Mobile-ID service SMS messages of the decryption transaction (see Section 7 in [1]). In regard to the data sizes provided, we point out the following:

1. In the case of RSA-2048, the data returned will be 256 bytes, as the ciphertext for 2048-bit RSA will be 256 bytes long.

2. In the case of ECC, the public key is 64 and 96 bytes long for P-256 and P-384, respectively. The additional byte specified in the document encodes the public key encoding type that is explicitly known by the implementation and hence can be omitted.

3. In the case of ECC, the public keys can be transmitted in a compressed form using the EC point compression method. This will reduce the size of the public keys to 33 and 49 bytes for P-256 and P-384, respectively. This, however, requires the SIM card to implement EC point decompression.

4. In the case of ECC, the size of the data that must be returned to the CDOC decryption software will be 32 and 48 bytes for P-256 and P-384, respectively. This is a half of the size than specified in the document, because in ECDH only the x coordinate of the obtained EC point is used as the shared secret.

2.3 Differences in the RSA and ECC-based schemes

While the proposed RSA and ECC-based Mobile-ID document decryption schemes seem to be similar, they provide slightly different security properties. In the ECC-based scheme, the Mobile-ID SIM card encrypts the shared secret of the CDOC container using a symmetric key that is derived from the Mobile-ID holder's ECC key and the ephemeral ECC key generated by the CDOC decryption software. This provides an assurance to the CDOC decryption software that the encrypted shared secret of the CDOC container has been encrypted by someone who has access to the Mobile-ID holder's ECC private key. Such an assurance to the CDOC decryption software is not provided in the RSA-based scheme. (An implication of this will be evident in Section 2.5.)

2.4 Optimization of the RSA-based scheme

Compared to the ECC-based scheme, the RSA-based scheme provides reduced security strength and is less efficient as the amount of data that must be transmitted in the Mobile-ID decryption transaction is large and therefore has to be split over several SMS messages.

We suggest using symmetric cryptography for the re-encryption of the transport key to optimize the RSA-based scheme. This means that instead of generating an ephemeral RSA key pair, the CDOC decryption software would generate a random 32-byte shared secret and send it to the Mobile-ID SIM card encrypted with the Mobile-ID holder's RSA key.

First of all, this would make the RSA-based scheme more similar to the ECC-based scheme. This would simplify the implementation and would ensure that both schemes provide similar security properties (see Section 2.3). Secondly, and perhaps even more important, this would significantly reduce the size of the data that needs to be transmitted over SMS messages.

Instead of sending an ephemeral RSA public key whose size is the byte length of the RSA modulus used, an encryption of a 32-byte shared secret would have to be sent. The ciphertext containing the shared secret will also occupy the same length as the byte length of the RSA modulus. However, since the RSA PKCS#1 v1.5 encryption scheme can be used to encrypt up to 11 bytes less than the size of the RSA modulus, there will be a lot of unused space in the encryption of the 32-byte shared secret. This remaining space can be used to transmit most of the encrypted transport key.

In the case of RSA-2048, this would reduce the size of the incoming Mobile-ID service SMS from 512 to 299 bytes and the size of the outgoing message from 256 to 32 bytes. We note that the size of the outgoing message would be 32 bytes regardless of the size of the RSA modulus used in the RSA-based scheme.

2.5 Susceptibility to active MITM attacks

The proposed solution provides end-to-end encryption between the Mobile-ID holder's SIM card and the CDOC decryption software that runs on the Mobile-ID holder's computer (or a trused remote service). The proposed solution, however, is secure only in the case of passive attacks, where an attacker just passively observes the messages exchanged between the CDOC decryption software and the Mobile-ID holder's SIM card. The solution is not secure against active man-in-the-middle (MITM) attacks by an attacker who can modify Mobile-ID service SMS messages exchanged. The attacker can acquire such capability by obtaining control over the Mobile-ID service provider or mobile operator.

According to the concept document, the verification code displayed to the user is derived from the ephemeral public key generated by the CDOC decryption software. We note, however, that in an active MITM attack the attacker can modify the transaction by replacing the ephemeral public key generated by the CDOC decryption software with an ephemeral public key that will produce the same verification code, but whose corresponding private key is known by the attacker. Generating such a colliding key pair is trivial and the key generation step can be moved to the attack preparation phase by pre-computing a dataset of 8192 key pairs where the public key of each key pair corresponds to a unique verification code.

In the case of the initially proposed RSA-based scheme, such a MITM attack can be performed in a fully stealthy manner as the attacker can re-encrypt the transport key without having access to the Mobile-ID holder's private key (see Section 2.3). This allows the attacker to successfully complete the original decryption transaction initiated by the Mobile-ID holder. In the case of the ECC-based scheme or in case the attacker needs to decrypt some other CDOC container than the one requested by the Mobile-ID holder, after obtaining the result, the attacker would have to simulate some type of transaction failure.

2.5.1 Calculation of the verification code

To prevent the attacker from replacing the ephemeral public key of the CDOC decryption software with another public key that produces the same verification code, the CDOC decryption software and the Mobile-ID SIM card should derive the verification code from the data that is known only to these two parties.

We suggest deriving the verification code from the shared secret, which in the optimized RSA-based scheme is generated by the CDOC decryption software and in the ECC-based scheme is computed using ECDH between the Mobile-ID holder's ECC key and the ephemeral key of the CDOC decryption software. Since the value of the shared secret is known only to the CDOC decryption software and the Mobile-ID holder's SIM card, the attacker will not be able to predict the expected verification code and hence will *not* be able:

1. In the case of the optimized RSA-based scheme: to replace the encryption of the shared secret with the encryption of another shared secret that will produce the same verification code.

2. In the case of the ECC-based scheme: to replace the ephemeral public key of the CDOC decryption software with another public key that will produce the same verification code.

We note that the attacker will be able to replace the encrypted transport key (in the case of the RSA-based scheme) or the CDOC container ephemeral key (in the case of the ECC-based scheme) contained in the incoming Mobile-ID service SMS message without affecting the verification code of the transaction. This, however, will not provide any benefit to the attacker as the result of the decryption will be re-encrypted using the shared secret that is not known to the attacker.

It is important to note that the calculation of the verification code should be done on a value *derived* from the shared secret. Otherwise, the value of the verification code will leak 13 bits of the shared secret. A safe method for deriving a value from the shared secret is proposed in Section 2.8.2.

2.6 Linkability to the encrypted document

The proposed scheme allows an attacker (the Mobile-ID service provider or mobile operator) to identify the CDOC container that corresponds to the Mobile-ID decryption transaction. The encrypted transport key (in the case of the RSA-based scheme) and the CDOC container ephemeral key (in the case of the ECC-based scheme) are unique to the CDOC container. Therefore, this data can be used to identify the CDOC container that the Mobile-ID decryption transaction aims to decrypt.

To provide privacy in this regard, the CDOC decryption software should send this identifying data to the Mobile-ID SIM card in an encrypted form. This data can be encrypted using a symmetric key derived from the shared secret that is known only to the Mobile-ID SIM card and the CDOC decryption software. In addition, this measure would also prevent an attacker from replacing this data with other meaningful data in an active MITM attack (see Section 2.5.1).

2.7 Protection against mobile communication attacks

The concept document does not mention how and whether the Mobile-ID decryption transactions will be protected against MITM attacks in the communication between the Mobile-ID holder's mobile phone and the mobile operator's network. We note that communication protocols used by mobile phones to communicate with mobile operators are susceptible to attacks that may allow third parties to capture and modify Mobile-ID service SMS messages.

If no layer of security is added to the Mobile-ID service SMS messages, malicious third parties could execute the same attacks as the Mobile-ID service provider and mobile operator (see Section 2.5 and 2.6). To prevent such attacks by third parties, the incoming Mobile-ID service SMS message of the decryption transaction should be protected by similar cryptographic measures as currently employed by the Estonian Mobile-ID SIM card implementation to protect the authentication and digital signature transactions (see Section 5.2 in [5]). Such an additional security layer may not be required if the security measures discussed in Section 2.5 and 2.6 are implemented.

2.8 Unspecified cryptographic primitives

The proposed concept document in some parts is vague in describing the specifics of the cryptographic primitives that will be used and their security parameters.

Since these details are important from the engineering and security perspective, we propose specific primitives and their security parameters below.

2.8.1 Symmetric cipher used for encryption

The concept document mentions that AES-128 or AES-256 could be used for symmetric encryption. We note that in the proposed scheme, the following data needs to be encrypted using a symmetric cipher:

1. In the case of the optimized RSA-based scheme:
   (1) the encrypted transport key (128 and 256 bytes for RSA-1024 and RSA-2048, respectively);
   (2) the decrypted transport key (32 bytes).
2. In the case of the ECC-based scheme:
   (1) the CDOC container ephemeral key (64 and 96 bytes for P-256 and P-384, respectively);
   (2) the shared secret derived from the CDOC container ephemeral key (32 and 48 bytes for P-256 and P-384, respectively).

Since the block size of AES is 16 bytes, some block cipher mode of operation will have to be used to encrypt data larger than 16 bytes. We suggest encrypting the data using the CBC mode of operation with a random initialization vector (IV). Since the size of the data that needs to be symmetrically encrypted is a multiple of 16, no padding needs to be applied. This guarantees that the size of the encrypted data will be the same as the size of the plaintext version of the data.

In order to achieve the same security strength as provided by the asymmetric key used, the AES key length must correspond to (at least) the key length specified in Table 2.

Table 2: The required minimal AES keylength (based on [3])

Asymmetric	Symmetric
RSA-1024	AES-128
RSA-2048	AES-128
RSA-3072	AES-128
P-256	AES-128
P-384	AES-192
P-521	AES-256

However, to simplify the implementation, we suggest using AES-256 in all cases, as this provides up to 256 bits of security strength and is the same symmetric cipher that is used to encrypt data stored in the CDOC container.

2.8.2 Key derivation function used to derive secrets

The concept document suggests that a HMAC-based Key Derivation Function (HKDF) could be used to derive a symmetric encryption key from the ECDH

shared secret. It is mentioned that a HKDF function could be implemented on the SIM card to support any SHA2 family hash function internally.

From the engineering perspective, we suggest using the same KDF function that is used to derive keys in the CDOC container decryption process. The CDOC specification defines the use of the Concatenation Key Derivation Function (ConcatKDF) with SHA-384 internally. The use of this KDF will simplify the implementation of the CDOC decryption software as then the same cryptographic primitive can be reused. Furthermore, the implementation of ConcatKDF on the SIM card is trivial as it can be implemented as a single invocation of SHA-384:

```
SHA384("\x00\x00\x00\x01" + shared_secret + otherinfo)
```

The first 32 bytes of the resulting hash value can be used as the symmetric key for AES-256 and the next 16 bytes as an IV for the CBC mode of operation.

From the perspective of cryptographic resilience, the `otherinfo` parameter of ConcatKDF should be set to a unique value providing derivation-specific context information. This ensures that keys derived from the same shared secret in different contexts will be different. For example:

1. When used to derive a symmetric key for encryption of the encrypted transport key (for the RSA-based scheme) or the CDOC container ephemeral key (for the ECC-based scheme), the `otherinfo` parameter can be set to value "Mobile-ID incoming cryptogram".

2. When used to derive a symmetric key for encryption of the decrypted transport key (for the RSA-based scheme) or the derived shared secret of the CDOC container (for the ECC-based scheme), the `otherinfo` parameter can be set to value "Mobile-ID outgoing cryptogram".

3. When used to derive a value for the calculation of the verification code, the `otherinfo` parameter can be set to value "Mobile-ID verification code".

2.9 The use of PIN1 for decryption

In a similar fashion as the Estonian ID card, the concept proposes to reuse the Mobile-ID holder's authentication key pair and PIN1 for the document decryption use case. However, in the case of the Mobile-ID solution, this introduces an additional phishing attack vector, as a Mobile-ID holder may fail to notice the difference between a Mobile-ID prompt asking PIN1 for authentication, versus a prompt asking PIN1 for decryption.

A malicious e-service provider could exploit this by initiating a Mobile-ID decryption transaction against a victim's Mobile-ID instance at the same time when the victim tries to authenticate to the malicious e-service. Since the verification codes will match, the victim is likely to enter PIN1 considering it to be an authentication transaction, while actually it is a decryption transaction. This will allow a malicious e-service provider to decrypt arbitrary data that has been encrypted for the victim.

If introducing a separate PIN code (e.g., PIN3) for authorizing Mobile-ID decryption transactions is not a viable option, the Mobile-ID service provider should consider restricting access to the decryption requests, such that decryption requests could be initiated only from specific (trusted) relying parties that Mobile-ID holders are able to identify in the decryption prompt.

3 Recommendations

From the issues discussed above, we have extracted a list of specific recommendations that we have ordered based on our opinion of their significance:

1. To prevent stealthy MITM attacks by the Mobile-ID service provider and mobile operator, the verification code should be calculated from secret data that is only known by the Mobile-ID SIM card and the CDOC decryption software (Section 2.5).
2. To reduce the potential of successful phishing attacks, consider introducing a separate PIN3 for authorizing decryption transactions (Section 2.9).
3. Consider optimizing the RSA-based scheme by using symmetric cryptography for the encryption of the decrypted transport key (Section 2.4).
4. Consider improving the privacy by encrypting the data that allows the Mobile-ID service provider and mobile operator to identify the CDOC container that is being decrypted (Section 2.6).
5. Consider introducing an additional layer of encryption to protect Mobile-ID-specific communication between the mobile phone and the mobile operator's network (Section 2.7).
6. Consider removing the obsolete RSA-1024 cryptosystem from the concept and consider defining the RSA-3072 cryptosystem instead (Section 2.1).
7. Consider using the AES-256 cryptosystem in the CBC mode with a random IV for symmetric encryption of data (Section 2.8.1).
8. Consider using ConcatKDF for key derivation from the shared seceret (Section 2.8.2).

4 Conclusion

We praise the authors of the Mobile-ID document decryption concept as they have come up with a smart solution that provides a secure way of decrypting documents using the Mobile-ID solution. Thereby, the authors have made possible what was believed to be impossible due to the architecture of the Mobile-ID solution (see Section 2 in [6]).

However, the proposed concept has two security issues that we strongly advise addressing: (1) the MITM attack vector for which we have proposed a protocol change that effectively eliminates the success of such MITM attacks; and (2) the phishing attack vector when the same PIN code is used for both authentication and document decryption.

Other than that, the solution has been well designed – it provides strong security guarantees and we do not see any fundamental security issues that would prevent a large-scale deployment of the proposed concept.

References

[1] B.EST Solutions. Mobile-ID document decryption concept. Public v1.0, May 17, 2022. https://cybersec.ee/storage/Decryption%20with%20Mobile-ID.pdf.

[2] Arnis Parsovs. *Estonian Electronic Identity Card and its Security Challenges*. PhD thesis, University of Tartu, 2021. https://dspace.ut.ee/handle/10062/71481.

[3] BlueKrypt. Cryptographic Key Length Recommendation: NIST Recommendations (2020), 2020. https://www.keylength.com/en/4/.

[4] Cybernetica AS. Required modifications to CDOC for elliptic curve support, September 27, 2017. https://www.ria.ee/sites/default/files/content-editors/EID/cdoc.pdf.

[5] Semjon Kravtšenko. *The Estonian Mobile-ID Implementation on the SIM Card*. BSc thesis, University of Tartu, 2022. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=74566&year=2022&language=en.

[6] Mart Oruaas and Jan Willemson. Developing Requirements for the New Encryption Mechanisms in the Estonian eID Infrastructure. In Tarmo Robal, Hele-Mai Haav, Jaan Penjam, and Raimundas Matulevičius, editors, *Databases and Information Systems*, pages 13–20, Cham, 2020. Springer International Publishing.