Solving Terminal Revocation in EAC by Augmenting Terminal Authentication*

Rafik Chaabouni[1,2]

[1]EPFL
CH-1015 Lausanne, Switzerland
rafik.chaabouni@epfl.ch

[2]University of Tartu
Ülikooli 18, 50090 Tartu, ESTONIA
rafik@ut.ee

Abstract: In this paper we propose a solution to enable an accurate terminal revocation in the Extended Access Control (EAC). Chaabouni and Vaudenay in [CV09] pointed out the need for an accurate revocation procedure, but failed to provide a complete solution description. We aim at filling this gap. Our solution relies on augmenting terminal authentication with a t-out-of-ℓ threshold signature provided by neighboring terminals. These terminals will be in charge of checking the revocation status of the requested terminal. As Terminals have a real clock embedded and more computational power than Machine Readable Travel Documents (MRTDs), they are better suited for checking revocation status.

1 Introduction

In response to the initial weak standard for Machine Readable Travel Documents (MRTDs), produced by the International Civil Aviation Organization (ICAO), the European Union has mandated the Federal Office for Information Security (BSI) to provide and maintain a stronger standard for MRTDs. In that regard, the BSI has issued the Extended Access Control (EAC) which provides a stronger privacy protection for MRTDs. Its first initial release [BfSidI12a] was made in 2006, while the last version [BfSidI12b, BfSidI12c, BfSidI12d] was published in 2012. It was believed that with the introduction of EACv2 in 2009, the majority of threats were solved. Unfortunately, Chaabouni and Vaudenay [CV09] pointed out several remaining flaws and threats. The major flaw pointed out was the absence of a good terminal revocation. The other issues are now considered marginal as they are or will be gradually solved with the evolution of previous standards (notably the one from the ICAO [ICAO08, ICAO13]). However, no progress has been made regarding terminal revocation nor terminal authentication. Chaabouni and Vaudenay

*This work has been supported from research theme IUT2-1 and European Regional Development Fund through the Estonian Center of Excellence in Computer Science, EXCS.

suggested a solution for terminal revocation but they omitted to give a detailed description. We aim at filling this gap by providing an efficient and secure solution.

Our concern in this paper targets two types of threats. We are first concerned by the threat of a stolen integrated terminal device. These are considered to be Portable Computing Devices (PCD) in the Technical Guideline TR-03110 [BfSidI12b, BfSidI12c, BfSidI12d], when terminal key pairs are explained. An integrated terminal, as explained in [BfSidI09], consists of a single reader with an integrated hardware security module and a proximity coupling device. Moreover a stolen integrated terminal could still be used to read MRTD, as long as its certificate is not expired. This threat even applies with an expired certificate if the date approximated in the MRTD is outdated. Hence there is no real revocation system present for terminals. This is a known problem and is even mentioned by the BSI in [BfSidI09], section 1.2.1:

> The disadvantage of this architecture is, that a stolen reader can be used to perform Terminal Authentication at least as long as the current CV certificate is valid.

Secondly, we have to keep in mind that threats come often from an inside attack. This pushes us to study the threat of a compromised terminal that is remained in place, acting maliciously. With the actual standard, a stolen or compromised terminal could be used to target a group of person (e.g. by nationality), or a specific person (e.g. important politicians).

Furthermore, we need to take into account efficiency. In [Fri], it is mentioned that more than 56 millions passengers traveled through Frankfurt airport in 2011. As around half of them are only transfer passengers, and thus do not necessarily need a passport control, we can see that big hubs need to process more than 2 millions passport checks per month.

1.1 Prior and Related Work

Terminal revocation has received little amount of interest as the BSI community is convinced that the Password Authenticated Connection Establishment (PACE) protocol mitigate this threat, as explained in [BDFK12]. Indeed, when executing EACv2, PACE is the initial phase before Terminal Authentication. After its successful completion, the MRTD is ensured that the terminal has knowledge of a shared password, and can proceed with Terminal Authentication. Moreover, the ISO/IEC JTC1 SC17 WG3 mentioned in [ICAO13] that:

> In its meeting on 19-21 February 2013 the NTWG concluded that as of the date 01 January 2018 eMRTDs supporting only PACE will be considered to be ICAO compliant.

However no guarantees are provided in the obtention of this password. If the shared password has been obtained by social engineering, or read directly by eavesdropping on the

MRTD, then a successful terminal authentication will allow the stolen terminal to access all sensitive data contained in the MRTD. This issue has been raised by Belguechi et al. in [BLR12]. Unfortunately they concentrate on the protection of biometric data and do not provide a solution for terminal revocation. Li et al. in [LZJX10] also mention the threat of terminal revocation. However they concentrate on presenting the Singapore solution that implicates Authorized Smartcard with Identity Based Cryptography. Hence to solve the terminal revocation issue they require heavy hardware modifications.

1.2 Contribution

Our main idea is to introduce terminal collaboration in order to achieve terminal authentication. Terminal revocation will thus be verified with the help of neighboring terminals. We make use of threshold cryptography to enforce terminal collaboration. We assume that Document Verifiers (DVs) in the EAC standard are trusted participants. In our general case, several terminals are present. If the number of terminals is considered too low, our scheme can easily be modified to provide equivalent properties. Moreover the required modifications to enable this method are solely software upgrades and the existence assumption of a communication channel between terminals. Hence no hardware modification is needed in MRTDs. Due to space limitation, we assume that readers are familiar with several topics: the EAC standard [BfSidI12b, BfSidI12c, BfSidI12d], Terminal Authentication, Terminal Revocation, Shamir's secret sharing scheme [Sha79], Non-Interactive Zero-Knowledge (NIZK) Proofs, Threshold Signature schemes and more specifically the Threshold RSA signature scheme presented by Shoup in [Sho00]. For the sake of completness, a full version of this paper explaining these topics as well can be found in [Cha13]

1.3 Organization

Section 2 will precise our security assumptions. In section 3, we explain how terminal authentication should be augmented to achieve a realistic terminal revocation. Section 4 will provide the security outcomes and we will finish by some closing remarks in section 5.

2 Security Assumptions

We assume the same structure of participants than the EAC model. However we make some precisions. Each DV is responsible for ℓ terminals (ℓ differs from one DV to the other). DVs play the role of trusted authority amongst their terminals. We assume the existence of secure and authenticated channels between all ℓ terminals. This is easily achieved with public key encryption as it is the same DV, i.e. a trusted party, that issued every terminal key pairs. When a terminal is stolen, its certificate will be revoked. This revocation will disable its use. Moreover, the lack of online connectivity should affect

only CVCAs and DVs as they are Public Key Generators. As such they should be turned offline once their keys setup generation has been achieved ([Sha84]). This is not the case for terminals.
Furthermore, we assume attackers to be *computationally bounded*. We will focus on threats targeting terminals, as they are somehow neglected in the current EAC. Nevertheless, we assume CVCAs and DVs to be honest. We consider a threshold security assumption, i.e. cases where the adversary can corrupt up to t terminals among $\ell \geqslant 2t + 1$. We will expect adversaries to be either *passive adversaries*, where attackers corrupt targets by reading their contents and secrets, or *active adversaries*, where attackers will additionally change the behavior of corrupted terminals. Lastly we restrict ourselves to *static adversaries*, meaning that the adversary will select which terminals to corrupt before the start of the protocol. Moreover, the adversary is free to corrupt them when he wants to. When a terminal is corrupted, all his communications will be revealed to the adversary. We set aside cases of dynamic adversaries as the corresponding solutions will induce a high loss in efficiency.

3 Augmented Terminal Authentication

Figure 1 gives a sketch of the general structure of our additional part to the current terminal authentication protocol. Our Setup phase is very similar to the original EAC one. DVs have to contact CVCAs from every other country, in order to obtain their DV certificate. The main difference is that now, certificates will contain an additional public key PK_{DV} corresponding to a secret key SK_{DV} only known by the DV and that will be shared among terminals. Moreover, certificates will contain additional information regarding how many terminals are required to collaborate in order to authenticate themselves (parameters t and ℓ). When a DV will set up his terminals, he will additionally give them a share d_i of his secret such that every terminal authentication will require the collaboration of at least $t+1$ of them. Hence our scheme tolerates up to t corrupted terminals. As long as $t + 1$ honest terminals are available, terminal authentication will be able to proceed. Once the Setup phase has been completed, only terminals and MRTDs are present in the interactions. Hence the DV can be used offline as described in the EAC standard.

DVs are in charge of the setup phase. They will run the key generation algorithm and distribute to each terminal its corresponding secret key, the public key pk of the system and the verification keys of all participants. After this step, DVs can be turned offline.

During the terminal authentication and just after the Certificate Chain Validation process, a MRTD will first select a random challenge M in the message space $\mathcal{M}$. He will then challenge the terminal with $(M||\widetilde{date})$ where $||$ denotes concatenation and $\widetilde{date}$ is the approximation of the current date stored in the MRTD. Moreover M must be independent from the MRTD identity, otherwise a tracking privacy threat would rise. Indeed, in this case the signature will prove that a given identity was at a given specific location and time. In order to sign the challenge, the terminal will have to collaborate with at least t other terminals. The revocation process takes place during the terminal collaboration. It will be

the role of other terminals to determine whether the requesting terminal T_r is revoked or not. As terminals have real clocks and better computation capabilities than MRTDs, they will be able to check this revocation status much more efficiently. Any standard strong revocation mechanism can then be used here. The basic solution is to apply Certificate Validation as described in section 2.5 of [BfSidI12d], but with a real clock. More complex solution can also be used such as Certificate Revocation Lists (CRL) or with an Online Certificate Status Protocol (OCSP) if an OCSP responder is set up for terminals. If the requesting terminal is revoked, then his request can be simply ignored. If T_r status is not revoked, then a partial signature σ_i can be computed from the *partial signature algorithm* Σ_i and sent to him, possibly with a verification proof π_i. At this stage, T_r will check, with the *partial signature verification algorithm* Σ_v, the validity of each σ_i. Then, T_r will combine with the *combining share algorithm* Σ_c, t valid partial signatures together with its own to create a global signature σ on the MRTD challenge. The latter will be sent to the MRTD as a proof of authenticity and non-revocation.

Once the MRTD receives the global threshold signature, he will have to verify it with the global public key of the DV. If the check is successful, he can be ensured that either the terminal knows the DV secret or that he has gone through a threshold signature involving some revocation checks. As we assume the DV to have correctly achieved the initial setup, the MRTD is ensured on the non-revocation status of the terminal.

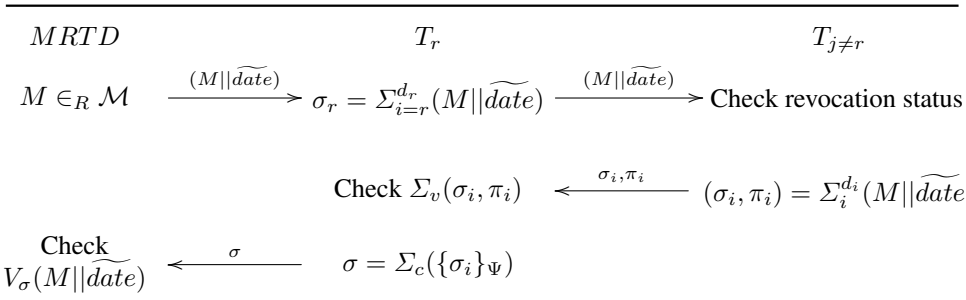

Figure 1: Terminal Authentication with Revocation

At this point, any efficient and secure threshold signature scheme can be used. In that regard, we suggest to use Shoup's threshold RSA signature [Sho00]. In this case, the MRTD computation will be dominated by one single exponentiation. The terminal communicating directly with the MRTD and in charge of combining the partial signatures, will have a computational complexity dominated by $(5t + 4)$ exponentiations. However this computational cost can be reduced to $(t + 5)$ exponentiations as explained in section 5.1. For the collaborating terminals, the computational cost is dominated by 3 exponentiations. More details can be found in [Cha13].

4 Security Outcome

Due to space limitation, we refer to [Cha13] for a complete analysis of the security outcomes. Following are the main conclusions.

As our augmented terminal authentication is enforced with threshold signatures, the security achieved is highly dependent on the security of the threshold signature scheme used. We assume a threshold signature scheme that is robust, unforgeable and threshold secure, as the one from [Sho00]. Hence any computationally bounded adversary corrupting at most t terminals will not be able to learn the master secret of the threshold signature scheme ($sk = d_0 = f(0)$). Moreover adversaries will not be able to forge valid signatures on chosen messages.

A stolen terminal will not be able to authenticate itself. A corrupted collaborating terminal will learn no information except that a MRTD with some approximation date has requested an authentication process. However, a corrupted requesting terminal interacting with a MRTD will be granted access to the MRTD sensitive data if the terminal behaves honestly with the other collaborating terminals. As long as at most t terminals are corrupted, the secret key used to authenticate terminals remains protected. Furthermore, the leakage of the secret key can be achieved only if at least $t + 1$ key shares are compromised. These security properties are desirable as they improve the current state of the EAC. By lowering the trust in terminals, we increase the DV level of trust. This is an acceptable change as DVs are less exposed than terminals.

Proactive security can be achieved by frequently renewing the global secret of the threshold signature scheme. This can be done efficiently by resharing the same secret with the means of sharing the "secret" value '0' and adding the obtained partial secrets to the previous ones. This method reduces the threat of terminal keys being exposed. In order to compromise the general secret key, an adversary will have to obtain $t+1$ key shares during the same time frame of a resharing phase. This allows DV certificates to protect their general secret used for threshold signature throughout their entire time validity. Notice that this step is highly efficient if performed by the DV, i.e. the DV generates the additional secret key shares and distribute them to their corresponding terminal. Verification keys will also have to be redistributed to every participants. However, this can be achieved without the need of the DV with secure multiparty computation.

5 Closing Remarks

5.1 Efficiency Tuning

Regarding computational costs, several modifications can be brought to reduce them. We refer readers to [Cha13] for complete details. First, the terminal in charge of combining partial signatures could perform the robustness checks solely if the resulting combined signature is not valid. Hence instead of computing $4t$ exponentiations he would first check the validity of the signature with one exponentiation. Furthermore, minor enhancements

are possible by letting the DV perform some precomputations and storing results in each terminals during the setup phase. The drawback of this method is that it will require a storage space in terminals. In the case of a large ℓ (e.g. $\ell > 100$), the threshold signature scheme of Gennaro et al. [GHKR08] will be preferable than the one from Shoup [Sho00] as it will be more efficient.

Furthermore, a small efficiency gain could be obtained by using the threshold signatures of King [Kin00] which is itself derived from the Desmedt-Frankel [DF94] scheme. However, the gain in efficiency is achieved by an increased difficulty to implement them and a higher storage requirement.

5.2 Remarks

Let us mention the existence of *multisignatures*. These are a type of threshold signature where the identity of signers is provided in the general signature. However, even the latest result in multisignatures that we could use, namely the scheme from Boldyreva [Bol03], would imply an important efficiency decrease.

The overhead in time of our suggested solution should be less than 0.1 seconds, assuming 30 MHz CPU for MRTDs, 520 MHz CPU for terminals, 802.11g wireless communication between terminals (net average of 22 Mbit/s) and 200 Kbit/s communication speed between MRTDs and terminals.

References

[BDFK12] Jens Bender, Özgür Dagdelen, Marc Fischlin, and Dennis Kügler. The PACE—AA Protocol for Machine Readable Travel Documents, and Its Security. In Angelos D. Keromytis, editor, *Financial Cryptography*, volume 7397 of *Lecture Notes in Computer Science*, pages 344–358. Springer, 2012.

[BfSidI09] Bundesamt für Sicherheit in der Informationstechnik. PKIs for Machine Readable Travel Documents – Protocols for the Management of Certificates and CRLs. Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2009. Technical Guideline TR-03129, Version 1.10.

[BfSidI12a] Bundesamt für Sicherheit in der Informationstechnik. Advanced Security Mechanisms for Machine Readable Travel Documents – Extended Access Control (EAC). Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2012. Technical Guideline TR-03110, Version 1.00.

[BfSidI12b] Bundesamt für Sicherheit in der Informationstechnik. Advanced Security Mechanisms for Machine Readable Travel Documents – Part 1. Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2012. Technical Guideline TR-03110-1, Version 2.10.

[BfSidI12c] Bundesamt für Sicherheit in der Informationstechnik. Advanced Security Mechanisms for Machine Readable Travel Documents – Part 2. Technical report, Federal Office for

Information Security, 53133 Bonn, Germany, 2012. Technical Guideline TR-03110-2, Version 2.10.

[BfSidI12d] Bundesamt für Sicherheit in der Informationstechnik. Advanced Security Mechanisms for Machine Readable Travel Documents – Part 3. Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2012. Technical Guideline TR-03110-3, Version 2.10.

[BLR12] Rima Belguechi, Patrick Lacharme, and Christophe Rosenberger. Enhancing the privacy of electronic passports. *IJITM*, 11(1/2):122–137, 2012.

[Bol03] Alexandra Boldyreva. Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. In Yvo Desmedt, editor, *Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2003.

[Cha13] Rafik Chaabouni. Solving Terminal Revocation in EAC by Augmenting Terminal Authentication. Cryptology ePrint Archive, Report 2013/460, 2013. `http://eprint.iacr.org/2013/460`.

[CV09] Rafik Chaabouni and Serge Vaudenay. The Extended Access Control for Machine Readable Travel Documents. In Arslan Brömme, Christoph Busch, and Detlef Hühnlein, editors, *BIOSIG*, volume 155 of *LNI*, pages 93–103. GI, 2009.

[DF94] Yvo Desmedt and Yair Frankel. Perfect Homomorphic Zero-Knowledge Threshold Schemes over any Finite Abelian Group. *SIAM J. Discrete Math.*, 7(4):667–679, 1994.

[Fri] Friedhelm. 2012 Facts and Figures on Frankfurt Airport. `http://www.frankfurt-airport.com/content/frankfurt_airport/en/misc/container/facts-and-figures-2011/jcr:content.file/zadafa-2012_e_lowres.pdf`.

[GHKR08] Rosario Gennaro, Shai Halevi, Hugo Krawczyk, and Tal Rabin. Threshold RSA for Dynamic and Ad-Hoc Groups. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 88–107. Springer, 2008.

[ICAO08] International Civil Aviation Organization. Machine Readable Travel Documents – Document 9303. Technical report, ICAO, 2005-2008. `http://www.icao.int/Security/mrtd/Pages/Document9303.aspx`.

[ICAO13] International Civil Aviation Organization. Machine Readable Travel Documents – SUPPLEMENT to Document 9303. Technical report, ICAO, 2013. `http://www.icao.int/Security/mrtd/Pages/Document9303.aspx`.

[Kin00] Brian King. Improved Methods to Perform Threshold RSA. In Tatsuaki Okamoto, editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 359–372. Springer, 2000.

[LZJX10] C. H. Li, X. F. Zhang, H. Jin, and W. Xiang. E-passport EAC scheme based on Identity-Based Cryptography. *Inf. Process. Lett.*, 111(1):26–30, 2010.

[Sha79] Adi Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sha84] Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.

[Sho00] Victor Shoup. Practical Threshold Signatures. In Bart Preneel, editor, *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 207–220. Springer, 2000.