A Card-less TEE-based Solution for Trusted Access Control

Mohamed Amine Bouazzouni[1], Emmanuel Conchon[2], Fabrice Peyrard[1]
and Pierre-François Bonnefoi[2]
[1]*University of Toulouse, INP Toulouse, Department IRIT, 2, rue Charles Camichel BP 7122, 31400, Toulouse, France*
[2]*University of Limoges, XLIM, 123, avenue Albert Thomas, 87060, Limoges, France*

Keywords: Secure Access Control, Trusted Execution Environments (TEE), Identity Based Encryption, IBAKE.

Abstract: In this paper, we present a new card-less access control system aiming to replace existing systems based on vulnerable contact-less cards. These existing systems have many vulnerabilities which makes them not secure enough to be deployed to protect restricted areas. We propose to deploy a new access control architecture based on the use of a smartphone to remove the physical card. Our secure access control system is based on Trusted Execution Environment (TEE) in the cloud and Identity Based Encryption (IBE) mechanisms. The authentication protocol deployed on our architecture is based on IBAKE. Finally, a performance evaluation of the protocol is provided.

1 INTRODUCTION

Contact-less tags using Near Field Communication (NFC) or Radio Frequency IDentification (RFID) technologies are used for access control for several years now. However, the first generation of these tags comes with a very small set of security mechanisms and the confidence they can provide is then limited. Indeed, the main security aspect they provide is their capacity to provide a unique identifier that is stored in a non-rewritable memory. But, as presented in (Mitrokotsa et al., 2011) these contact-less tags can be cloned. To address these security issues, the cards manufacturers developed a new generation of cards with more security capabilities. For instance, the MIFARE DESFIRE card supports Advanced Encryption Standard (AES) encryption algorithm that allows to deploy authentication algorithms between the card and the reader for secure communication. However, almost all these authentication algorithms are based on the *shared secret* paradigm, which means that the reader and the card are sharing a secret (a pool of encryption keys in some implementations) prior to the communication. Furthermore, some DESFIRE cards are sensitive to side channel attacks like exposed in (Oswald and Paar, 2011).

To overcome these issues, we propose a new authentication scheme based on the emulation of the physical card on a smartphone. The smartphone storage and computation capabilities allow to deploy complex authentication algorithms avoiding the use of a shared secret. However, smartphones can not be considered as trusted devices. Thereby, all sensitive operations such as key generation, key storage and encryption/decryption operations need to be supported by secure components.

In order to satisfy these security constraints, several trusted mobile computing solutions have been proposed (Asokan et al., 2014; Bouazzouni et al., 2016) among which is the Trusted Execution Environment (TEE): a combination of a hardware part (processor) and a software part (Secure Operating System).

A main drawback of TEE is the use of a hardware component that is fully controlled by the manufacturer. Indeed, it does not allow the user to deploy trusted application on this component in an easy way and often requires a manufacturer agreement prior deployment. However, new fully software based solutions such as OP-TEE can be used to bypass this agreement.

Section 2 presents the smart campus context in which this works is performed. In section 3, we present the identity-based authentication schemes that will be used in the proposition. In section 5, we present and discuss our solution to address access control on a smart-campus. Then, an evaluation is given in section 6 before concluding.

Bouazzouni, M., Conchon, E., Peyrard, F. and Bonnefoi, P-F.
A Card-less TEE-based Solution for Trusted Access Control.
DOI: 10.5220/0006431704560461
In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - Volume 4: SECRYPT*, pages 456-461
ISBN: 978-989-758-259-2

2 CONTEXT AND PROBLEM STATEMENT

The work presented in this paper is a part of a global smart campus project called neoCampus that aims at designing a modern and green campus thanks to a variety of new sensors and softwares. So far, both campus employees and students have Mifare classic 1K NFC cards that they use to access various services. For instance, to access a restricted area such as a lab or a meeting room, the user has to authenticate using a contact-less card to a reader. If he is authorized, the system opens the door otherwise the door remains closed.

Since these cards have vulnerabilities allowing an attacker to clone it (see (Mitrokotsa et al., 2011; Mitrokotsa et al., 2010), the university is now switching these cards for DESFIRE NFC cards. These cards are more secure than the previous one and can perform symmetric cryptographic operations and to deploy simple authentication algorithms based on the shared secret paradigm. However, there are still two major drawbacks: deployment costs and user authentication. Indeed, when a Mifare 1K costs in average 0.02$, a DESFIRE card is three times more expensive. Also, Oswalds et al. pointed in (Oswald and Paar, 2011) that these cards are vulnerable to a low cost side channel attack allowing the attacker to recover the encryption key.

In this paper, we propose a secure access control system for a smart campus to replace the existing one that relies on vulnerable MIFARE cards without disrupting the platform currently deployed.

3 ID-BASED SOLUTIONS

The current access control systems based on contactless cards are using the UUID of the card in the authentication process. In order to remove the physical card and deploy a more secure access control system using the identity, it is necessary to rely on cryptographic mechanisms dealing with the identity of a user.

The authentication protocols based on identity mainly rely on a Public Key Infrastructure. The PKI is an organization that can register users and provide them a key pair composed of a private and a public one that can be used in the authentication process. If a user wants to get a key pair for a HTTPS connection for instance, he has to register his identity in the PKI. He has also to provide many information about him and his organization. At the end of the process, he will get his key pair and will be able to execute authentication protocols based on asymmetric cryptography.

However, this process is heavy and expensive and requires the user to give a lot of information about him and his organization. Moreover, this system relies on the use of certificates to prove the identity of the key owner. Exchanging, storing and processing certificates induce an overhead in the system performances. Indeed, the number of certificates to store grows according to the number of users. To deal with these issues, a new mechanism was introduced by Shamir (Shamir, 1984): The Identity Based Encryption (IBE) in which the identity can be viewed as a public key in public key cryptography.

This approach is a trade-off between a straightforward process of binding an identity to an asymmetric pair of keys and the usage of a new kind of trusted third-party acting as a mandatory key escrow (i.e. capable of producing every pair of cryptographic keys).

The first IBE scheme fully satisfactory in terms of security and performances has been proposed by Boneh and Franklin (Boneh and Franklin, 2001) in 2001. Therefore, IBE schemes based on existing cryptographic standards have emerged such as the one proposed by Callas (Callas, 2005) that is based on RSA.

4 TRUSTED EXECUTION ENVIRONMENT

As the smartphones are considered untrusted, the Trusted Execution Environment (TEE) was developed to secure the data storage and to process sensitive operation such as cryptographic operations. The TEE consists of a hardware part and a software part. The hardware part is embedded in the processor and provides a set of secure operations. The software part is split into two separate execution environments: REE and TEE. The Rich Execution Environment (REE) (a.k.a Normal World) represents the standard operating system (Rich OS) of the smartphone such as Android for instance. The TEE represents the Secure OS and is responsible for performing sensitive processing and secure Input/Output.

OP-TEE (Linaro And STMicroelectronics, 2017) is a Linaro/STMicroelectronics project with the objective to release a totally virtualized and open source TEE. It also can be used as a secure OS on the top of a physical TEE. OP-TEE provides an Application Programming Interface (API) to invoke certified cryptographic operations[1] in the Secure World side (TEE).

[1] Libtomcrypt : http://www.libtom.com/

In this paper, for our implementation, we use a JUNO board which is a physical TEE and a virtualized TEE using FVP which is an hypervisor running on the top of a Debian-based OS.

5 TRUSTED ACCESS CONTROL ARCHITECTURE

In this section we present the solution that we propose. We first start by presenting the general architecture, we follow by exposing the threats we identified and our security hypothesis and finish by describing the authentication protocol.

5.1 General Architecture

To perform a secure access control on a smart campus, we propose the architecture presented in Figure 1 where a user tries to access a restricted area.

This architecture is composed of a smartphone with NFC capabilities, a NFC reader and for each device, a remote TEE deployed in a dedicated secure cloud and an access control server.

The smartphone is used to initiate the authentication process by taping a NFC reader. We suppose that the smartphone has no TEE capabilities. It communicates with a remote TEE in the cloud to perform the authentication process and the cryptographic operation.

The NFC reader represents a wall reader installed close to each door. It is a cheap device with no secure processing capabilities that is already present in every campus. It is assumed that all readers have an Ethernet connection with an access control server thereby our solution does not modify the current infrastructure. The only modification to existing infrastructures is that now all NFC reader will communicate with a secure cloud to perform the authentication process.

The secure cloud consists of one or several TEE accessible through the Internet. This cloud receives messages from the two previous devices and is in charge of executing sensitive operations such as encryption, decryption and signatures for instance. From a practical standpoint, there are two separate secure cloud: one for the user, one for the smart campus infrastructure which can be viewed as a private cloud.

The access control server is in charge of checking the user's credentials in order to grant the access or not. It's already a part of the existing infrastructure of most university campus and can be viewed as a service offered by the private cloud of the university. No modifications are made on this part.

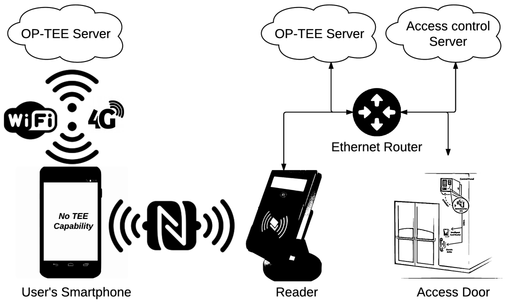

Figure 1: Trusted access control architecture based on IBE.

The use of a TEE in the cloud is based on the consideration that at the time of writing, only few smartphones have TEE capabilities. Therefore, to provide the TEE secure storage and secure processing capabilities for any user, a cloud-based solution seems to be the best candidate.

Indeed, several alternatives to the architecture proposed in Figure 1 have been considered. First was investigated the possibility to use TEE on the smartphone TEE for the end user and a private cloud-based TEE for the reader. Even if the overall performances are better, this solution has not been selected for the above reasons. Second, is the possibility to rely on a single cloud-based TEE that limits the overall complexity of the architecture but with less flexibility for the end user. Indeed, the readers secure cloud is a part of the overall smart campus and can be viewed as a private cloud. In order to allow the use of the proposed solution in other contexts beside smart campus it seems more relevant to have a dedicated cloud-base TEE for end users.

5.2 Threats Identification

In the proposed architecture, every parties involved are assumed to be honest and will not deviate from the proposed protocol. Likewise, end-users' smartphones, readers and cloud servers only exchange valid information. Furthermore, cloud servers are assumed to be secure and no information can be leaked from a user to another. Cloud security is out of the scope of this paper. Smartphones are assumed to have NFC capabilities to emulate contact-less cards. No assumption are made on the communication links.

However, several threats can be identified. First is the classical passive attacker that will eavesdrop communications to discover access credentials. To avoid this, every communication has to be cyphered and have to provide mechanisms to avoid replay. Second are threats targeting the smartphone of the end-user. Indeed, these terminals are vulnerable to malwares or trojans that can intercept sensitive information during both communications and processing. Rooted terminals are even weaker in terms of security as malicious softwares can potentially have access to the overall file system . Therefore, no sensitive operation neither critical credentials can be performed or stored on the smartphone. Every smartphone is considered as an untrusted device.

In the proposed architecture, every cryptographic operation are assumed to be realized in the secure cloud. A secure identification between the end-user and the secure cloud has then to be performed. In the remaining of the paper, we will focus on the authentication of the user against the access control server and therefore consider that this cloud authentication has already been realized. Please also note that this cloud authentication can be performed prior to the access control and does not have to be performed again as long as the connectivity with the secure cloud is active.

5.3 Protocol Description

The overall protocol is composed of an ID-Based authentication protocol IBAKE (Cakulev and Sundaram, 2012), that is adapted to the cloud-based environment. In our architecture, every cryptographic operation and every secure storage are performed in TEEs that are hosted in the secure cloud and please note that, as discussed previously, the user is securely authenticated in the cloud prior to the proposed protocol.

The IBAKE protocol allows both entities (i.e. the user and the reader) to agree on a session key that will then be used to encrypt the subsequent exchanges. The IBAKE protocol has been implemented in OP-TEE based on Callas's IBE in which the identity is hashed so that it can be used with a regular RSA.

The overall protocol, depicted in Figure 2, is composed of 8 steps as given in the following:

- **1 - Initiate the Authentication Process.** When a user wants to get access to a restricted area, he has to tap his smartphone against the wall-reader.
- **2 - Request Challenge.** When a user requests an access to the restricted area, the reader communicates with its cloud to request an encrypted challenge.
- **3-6 - IBAKE Messages.**[2] This part of corresponds to the IBAKE key exchange protocol execution. At the end of this process, the two devices agree on a session key $x * y * P$. From this session key, an AES symmetric key is derived. An AES encrypted channel communication is then set up where every other exchanges will take place.
- **7-8 - Authenticated Access Control.** The user requests an access to a restricted area and sends his credentials encrypted with the session key to the reader which forwards them to the access control server. Upon reception, the server uses the OP-TEE server to decipher the message and grants or refuses the access based on the credentials.

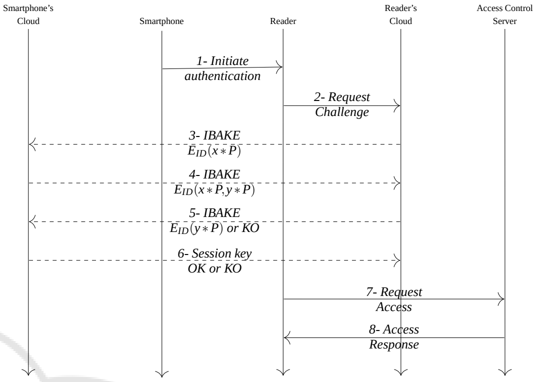

Figure 2: Chronogram of the proposed access control protocol.

6 PERFORMANCE EVALUATION

In this performance evaluation, we will focus on the time taken by the cryptographic operations as it will be the most time consuming part of the process. Indeed, since the size of the IBAKE packet payload does not exceed 64 bytes, the time taken by network communications is not significant. The most time consuming communication part is the NFC link between the smartphone and the reader (i.e.: 53 kBps) which takes around 1ms for a single packet and can therefore be neglected from a usability standpoint.

6.1 Experimental Setup

To evaluate the performance of the cryptographic operation in the secure cloud, experiments have been made with both virtualized and hardware TEE.

The virtualized TEE consists of a virtual machine with an Intel I7 processor with 1 GByte of dedicated RAM (Random Access Memory) hosting the OP-TEE emulated environment.

The hardware TEE consists of an ARM development card called the ARM JUNO Board: a development card provided by ARM to develop applications that need to use some proprietary features that are usually locked in smarpthones. In our case, it enables

[2] Dashed lines are indirect messages.

the deployment of OP-TEE granting it access to the TrustZone features provided by the embedded processor. From a technical standpoint, the board consists of a complete development kit with a dual core Cortex-A57 MPCore processors, Random Access Memory (RAM) and input/output peripherals such as USB and Ethernet ports. For comparison, the provided processors are the same as the one that equip the Samsung Galaxy S6 and note 5, the LG G4 and the HTC M8.

6.2 Experimental Results

The first thing to evaluate is the time taken by the IBE key generation process to evaluate if it's possible to do it online or if the key pair has to be generated and stored prior to the authentication.

Figure 3 shows the time taken by the virtualized and the hardware TEE to perform an IBE key pair generation. We observe that this time grows exponentially according to the key size. For instance, for a 1024 bits key-pair, the virtualized TEE takes about 16 seconds when the hardware TEE takes 2.09 seconds. The performances are 8 times better in average between the two environments. However, the time taken by this process is not in adequacy with the constraint of a real time authentication protocol. For these reasons, the generation process have to be done offline and the key pair has then to be stored in the secure cloud.

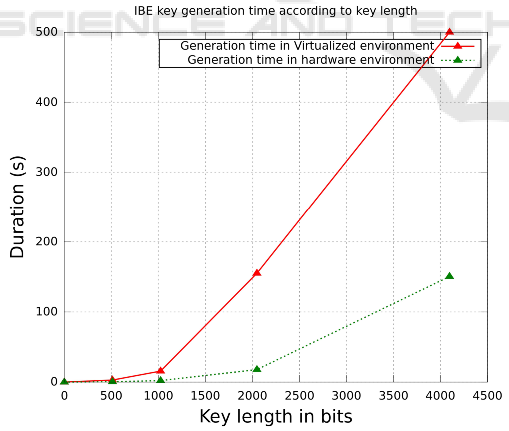

Figure 3: RSA-based IBE key pair generation time.

Figure 4 shows the time taken by both virtualized OP-TEE and hardware OP-TEE to compute $x*P$ and $y*P$ of the IBAKE authentication process until the agreement on the session key. With the virtualized OP-TEE, we observe that the authentication time evolves linearly according to the random numbers generated. Below a value of 400, the processing time is under 1 second. With the hardware OP-TEE, the performances are 2 times better comparing to the virtualized environment. For instance, with a random value $x = 500$, the virtualized OP-TEE takes 1.1 seconds where the hardware environment takes 0.59 seconds.

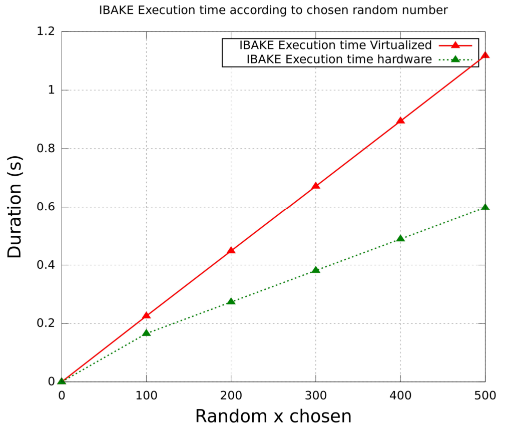

Figure 4: IBAKE computation time.

Every IBAKE exchanged messages are ciphered with an IBE algorithm. Figure 5 depicts the time taken by both virtualized and hardware TEE to cipher such messages with a Callas IBE. We observe that the hardware TEE performance are times better than the one of the virtualized one. For instance, with 1024 bits of data (128 Bytes), which is more than our packet size, the hardware TEE takes less than 0.1 seconds. It is a very acceptable performance for our protocol.

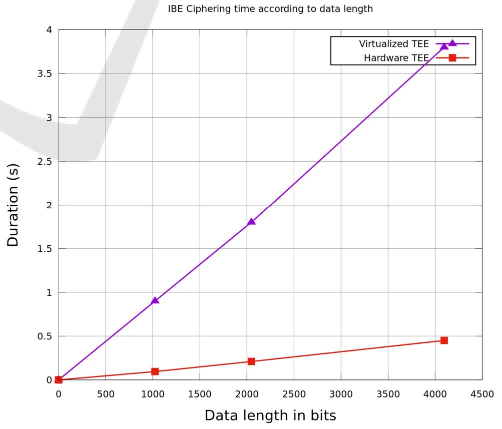

Figure 5: RSA-based IBE ciphering time.

After agreeing on a session key $(x*y*P)$, the user sends his credentials to the access control server to request the access to the restricted area. The credentials are encrypted with AES-256 to avoid eavesdropping. The AES key used is derived from IBAKE session key.

Figure 6 shows the time taken by AES-256 to encrypt different size of messages. We observe that the ciphering time is less than 0.14 seconds with virtualized OP-TEE for 4096 bytes of data which is very quick regarding to the overall protocol time. As expected, the hardware environment performs better as, for the same amount of data (4096 bytes in this case), the gain is about 10 times regarding to the virtualized environment (less than 0.02 seconds).

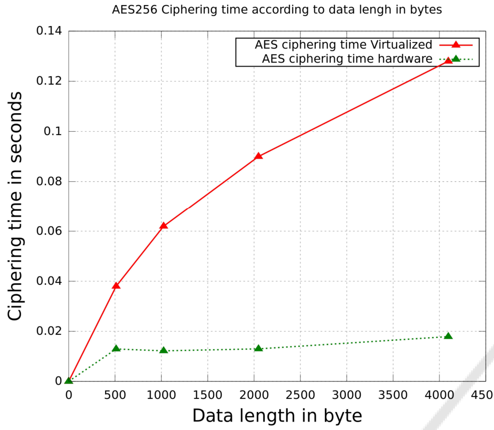

Figure 6: AES ciphering time.

To conclude, the most efficient solution is to use an hardware based TEE in the secure cloud and to rely on symmetric algorithm for the secure exchanges that follow the IBAKE authentication. Based on the provided measurements, it is possible to estimate the time taken by the whole protocol. Indeed, we have 6 ciphering operations with IBE that takes 0.05 seconds (i.e. for a 64 byte packet) each giving a total of 0.30 seconds. We have also to add the IBAKE key agreement according to the random number choice. In the worst case , this operation takes 0.6 seconds. Finally, the AES-256 encryption of the credential take less than 0.01 seconds to perform which gives a total of 0.9 seconds for the authentication process.

7 CONCLUSION

In this paper, we propose and evaluate a secure access control protocol based on TEE and IBE. First, we described the current access control systems based on contact-less cards and pointed out their vulnerabilities. Then, we described identity-based authentication schemes followed by a focus on ID-Based cryptographic solutions. We also exposed our solution based on IBAKE and on the use of a cloud-based TEE to secure the overall access control protocol and gave a performance evaluation of the TEE in terms of time to perform the different cryptographic operations involved in the protocol. This evaluation highlights the viability of the solution.

In future works, it is investigated to test to evaluate the use of another TEE: Intel SGX to compare the performance between this technology and our remote TEE. A scalability evaluation of our protocol will be performed to evaluate the impact of the number of students on the system and a real deployment will be performed in the framework of neOCampus.

REFERENCES

Asokan, N., Ekberg, J.-E., Kostiainen, K., Rajan, A., Rozas, C. V., Sadeghi, A.-R., Schulz, S., and Wachsmann, C. (2014). Mobile trusted computing. *Proceedings of the IEEE*, 102(8):1189–1206.

Boneh, D. and Franklin, M. K. (2001). Identity-based encryption from the weil pairing. In *Advances in Cryptology, Proceedings of CRYPTO 2001, Santa Barbara, California, USA, August 19-23, 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer.

Bouazzouni, M. A., Conchon, E., and Peyrard, F. (2016). Trusted mobile computing: An overview of existing solutions. *Future Generation Computer Systems*.

Cakulev, V. and Sundaram, G. (2012). Ibake: Identity-based authenticated key exchange. https://tools.ietf.org/html/rfc6539.html.

Callas, J. (2005). Identity-based encryption with conventional public-key infrastructure. *PGP Corporation Palo Alto, California, USA jon@ pgp. com*.

Linaro And STMicroelectronics (2017). OP-TEE official wiki page. https://wiki.linaro.org/WorkingGroups/Security/OP-TEE. Online; accessed 29 January 2017.

Mitrokotsa, A., Beye, M., and Peris-Lopez, P. (2011). *Unique Radio Innovation for the 21st Century*, chapter Security Primitive Classification of RFID Attacks, pages 39–63. Springer.

Mitrokotsa, A., Rieback, M. R., and Tanenbaum, A. S. (2010). Classifying RFID attacks and defenses. *Information Systems Frontiers*, 12(5):491–505.

Oswald, D. and Paar, C. (2011). Breaking mifare desfire mf3icd40: Power analysis and templates in the real world. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 207–222. Springer.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer.