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

1University of Toulouse; INP; IRIT
2University of Limoges, XLIM

Journée de présentations des doctorants - Chantier Sécurité et Vie Privée du RTRA STAE
Introduction and Context

Trusted Execution Environment

OP-TEE

Identity Based Encryption

Our proposition
Introduction and context (1)

- Sustainable and smart campus.
- Deployment of sensors and actuators.
- Increase users’ quality of life.
NFC-based contactless cards.

- Using UUID to authenticate the card.
- Many vulnerabilities allowing cloning.
- DESFIRE cards proposed (izly) -> Still vulnerable.
- No user authentication.
Dematerialization

Solution and definition

Dematerialization means replacing the physical card with a smartphone application that provides at least the same features and security level as this physical card. This process allows us to:

- Save money since no physical card is needed.
- Deploy more complex security and authentication mechanisms (user authentication for instance).
- Combine several features into one secure mobile application.
Dematerialization of contactless cards (2)

Card data
Access Rights

Smartphone acts like a regular card

NFC Communications

Dematerialization process
Dematerialization vs Regular cards

Dematerialization

- No need for a physical smartcard.
- Ability to deploy more complex algorithms.
- Mutual authentication between the user and the reader.
- More storage capacity for different applications.

Regular cards

- A physical smartcard is needed.
- The processing power is limited.
- No user authentication can be performed.
- Limited storage capacity -> limited number of applications.
Introduction and Context
Trusted Execution Environment
OP-TEE
Identity Based Encryption
Our proposition

Stockage et traitement des données d’authentification ?

User's Smartphone

Protocole d'authentification ?
Certificats/PKI ?

Reader

Ouverture de la porte

Access Door
Problematic : How to secure data in the mobile world?

The native OS of a mobile is not a suitable environment to deploy our secure application. Indeed:

- The mobile OS is an untrusted software, we have to consider it as an enemy.
- Many applications are harmful for personal data on a mobile.
- The I/O operations are not secure.

Requirement

These potential vulnerabilities on the native OS led us to use a Trusted Execution Environment.
A Trusted Execution Environment (TEE) is a secure environment allowing the secure storage and processing of sensitive data. The characteristics of such an environment are:

- **Isolated execution**: Every application should run independently from the other applications.
- **Secure storage**: Guarantee the integrity and secrecy of all data including the binaries (applications).
- **Secure provisioning**: The ability to install securely applications and to send securely data to a specific application.
TEE features

- A trusted part of the processor or an entire processor dedicated to execute sensitive operations.
- A secure OS with a minimal set of instruction to allow a formal verification of its security level.
- A secure boot to verify the mobile OS and load sensitive modules.
TEE features

- A trusted part of the processor or an entire processor dedicated to execute sensitive operations.
- A secure OS with a minimal set of instruction to allow a formal verification of its security level.
- A secure boot to verify the mobile OS and load sensitive modules.

- To deploy an application on a TEE, a developer should have the agreements of the manufacturer.
Motivation

The major obstacle to the use of the TEE is that the secure OS needed is proprietary and any application we want to execute on it has to be signed by the Secure OS provider.

OP-TEE:

- was designed to propose to developers an open source secure OS.
- can be executed on a physical device (ARM Juno).
- can be executed in a virtual environment.
OP-TEE components

- Two main parts: Client Application and Trusted Application.
- An internal API to facilitate the development and the communication between the two parts.
- Isolation between the Client and the Trusted Application.
Identity Based Encryption (IBE)

Definition

The Identity Based Encryption mechanism is based on the feature of a cryptographical system to use an identity as a public key. Then, no need for certificate to authenticate the owner of a public key.

- The identity is used by the cryptographical system for ciphering and generating the private key.
- The process need to be deterministic.
- The IBE should be at least as secure as a classical assymetric encryption.
This is a full online solution.
Removing the step 3 led us to an offline solution.
Online vs Offline: no one wins.
This is a full online solution.

Removing the step 3 led us to an offline solution.

Online vs Offline: no one wins.
A Decentralized OP-TEE server (Virtualized or Hardware) hosts an IBE system.

2. An identity-based authentication protocol is deployed to secure the access control.

3. The identities are used in an IBE to encrypt/decrypt messages.

4. The communications need to be secure (SSL-based solutions for instance.)
Challenges

To design and implement such an architecture, it is necessary to overcome some challenges:

- **Identity management**: In an IBE-based system, the management of the identities is critical.
- **User authentication with the server**.
- **Scaling problem**: Deal with thousands of simultaneous users.
Authentication Protocol

1- Initiate authentication
2- Request Challenge
3- IBAKE $E_{id}(x \times P)$
4- IBAKE $E_{id}(x \times P, y \times P)$
5- IBAKE $E_{id}(y \times P)$ or KO
6- Session key OK or KO
7- Request Access
8- Access Response

Smartphone’s Cloud

Smartphone

Reader

Reader’s Cloud

Access Control Server
First Results

IBAKE Execution time according to chosen random number

IBAK Execution time Virtualized
IBAKE Execution time hardware

IBE key generation time according to key length

Generation time in Virtualized environment
Generation time in hardware environment
First Results (2)

AES256 Ciphering time according to data length in bytes

IBE Ciphering time according to data length

\[ \text{Duration (s)} \]
Contact-less cards are not secure enough for trusted access control.
Dematerialization can be a credible solution to replace physical cards.
TEE are needed to secure data and execution on the smartphone.
An identity-based authentication protocol was designed, implemented and evaluated.
Perspectives

- Take into account the offline scenario with authentication tokens.
- Use alternative TEE like Intel SGX for instance.
- Scaling problem: Deal with thousands of simultaneous users.