Trusted Execution Environments on Mobile Devices

Jan-Erik Ekberg
Trustonic
jan-erik.ekberg@
trustonic.com

Kari Kostiainen
ETH Zurich
kari.kostiainen@inf.ethz.ch

N. Asokan
University of Helsinki
asokan@acm.org

ABSTRACT

A trusted execution environment (TEE) is a secure processing environment that is isolated from the "normal" processing environment where the device operating system and applications run. The first mobile phones with hardware-based TEEs appeared almost a decade ago, and today almost every smartphone and tablet contains a TEE like ARM TrustZone. Despite such a large-scale deployment, the use of TEE functionality has been limited for developers. With emerging standardization this situation is about to change. In this tutorial, we explain the security features provided by mobile TEEs and describe On-board Credentials (ObC) system that enables third-party TEE development. We discuss ongoing TEE standardization activities, including the recent Global Platform standards and the Trusted Platform Module (TPM) 2.0 specification, and identify open problems for the near future of mobile hardware security.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

Keywords

Trusted execution environments; mobile devices

1. INTRODCUTION

A *trusted execution environment* (TEE) is a secure, integrity-protected processing environment, consisting of processing, memory and storage capabilities. It is *isolated* from the "normal" processing environment, sometimes called the *rich execution environment* (REE), where the device operating system and applications run. TEEs can make it possible to build REE applications and services with better security and usability by partitioning them so that sensitive operations are restricted to the TEE and sensitive data never leave the TEE. In our daily lives, we encounter more and more services that use dedicated hardware tokens to improve their security: one-time code tokens for two-factor authentication, wireless tokens for opening doors in buildings or cars, tickets for public transport, and so on. Mobile devices equipped with TEEs have the potential for replacing these many tokens thereby improving the usability for users while also reducing the cost for the service providers without hampering security.

Chances are that the mobile device in your pocket sports a hardware-based TEE. Chances are, too, that you have not come across too many applications that actually make use of TEE functionality. In this tutorial, we explain why this situation came to pass and what the future may hold.

Security in mobile world had a very different trajectory compared to the world of personal computers. Various stakeholders had strict security requirements, some of which date back to two decades ago, right at the beginning of the explosion of personal mobile communications. Standardization requirements like ensuring that the device identifier will resist manipulation and change, regulatory requirements like ensuring secure storage for radio frequency parameters, business requirements like ensuring that subsidy locks, and end user expectations (e.g., no blue screen of death) incentivized mobile device manufacturers, chip vendors and platform providers to design and deploy hardware and platform security mechanisms for mobile platforms from early on. Hardware-based TEEs were seen as essential building blocks in meeting these requirements. The first mobile phones with hardware-based TEEs appeared almost a decade ago, and today almost every smartphone and tablet contains a TEE like ARM TrustZone, along with software platform security mechanisms.

Despite such a large-scale deployment, the use of TEE functionality has been largely restricted to its original intended uses. There has been no widely available means for application developers to benefit from existing TEE functionality. Fortunately, with emerging standardization this situation is about to change.

In this tutorial, we explain the security features provided by TEEs and describe *On-board Credentials* (ObC), a system that we developed at Nokia Researcher Center for safely opening up access to TEE functionality for application developers. We demonstrate the possibilities of TEEs with a programming example using the ObC system, discuss ongoing TEE standardization activities, including the recent Global Platform standards and the Trusted Platform Module (TPM) 2.0 specification, and identify open research questions for the near future of mobile hardware security.

CCS'13, November 4–8, 2013, Berlin, Germany.
ACM 978-1-4503-2477-9/13/11.
http://dx.doi.org/10.1145/2508859.2516758.

2. TUTORIAL DESCRIPTION

2.1 Structure and Length

The total length of the tutorial is *three hours.* The tutorial consists of the following parts:

1. **A look back.** We start the tutorial with a brief historical background on development of mobile security. We explain the business, regulatory, and end user requirements that have steered introduction of platform security features to mobile devices over the past two decades. (15 minutes)

2. **Mobile hardware security.** We proceed by explaining typical mobile hardware security functionality in a conceptual level. We describe mechanisms present in a typical mobile TEE, including secure boot, secure storage, isolated execution, immutable device identifiers and devices authentication. We introduce ARM TrustZone architecture [?] that is widely deployed in a majority of existing smartphones, and explain how TrustZone can be used to implement many of the previously described hardware security mechanisms in practice. (30 minutes)

3. **Application development and ObC.** The current standardized and de-facto hardware security APIs (e.g., PKCS-11 [?] and JSR-177 [?]) allow secure storage of cryptographic keys and execution of pre-defined cryptographic computation. To take full advantage of the isolated execution capabilities of mobile TEEs different kinds of API abstraction are needed. We describe On-board Credentials (ObC) system [?] that we have developed at Nokia Research Center. The ObC system serves as an example of an architecture that allows third-parties to develop and deploy credentials on TrustZone enabled devices. We explain the ObC architecture, its open provisioning model, developer interfaces, and a few example applications. Both the ObC system and applications built on top of it have been deployed on recent Nokia Lumia device models. We provide a brief demonstration of ObC application development in practice. (40 minutes)

   Break. (15 minutes)

4. **Emerging standardization.** We explain recent and on-going standardization activities regarding mobile TEEs. We briefly explain the set of APIs that Global Platform [?] has specified for usage of hardware-security functionality in mobile devices. After that, we focus in more detail on the recently published Trusted Platform Module (TPM) 2.0 standard [?]. We especially explore the new authorization model introduced in TPM 2.0 and examine its potential in use cases like secure boot and provisioning. (60 minutes)

5. **Summary and a look ahead.** We end the tutorial we a brief summary and identify open issues and research questions for mobile hardware security in near future. (10 minutes)

2.2 Level of Detail

The first two parts of the tutorial will provide the audience a background information on mobile hardware security. We describe the historical development, the common hardware-security features, and the TrustZone architecture in a fairly high level. After that we provide mode detailed treatment on two topics: application development and standardization. In these parts the tutorial gets more technical: we explain the ObC system architecture and provisioning model in detail, and, for example, cover individual TPM 2 instructions and data structures to explain the functionality of this emerging security standard. The ObC demo part illustrates practical TEE development on a programming level.

2.3 Intended Audience

This tutorial is intended for researchers and practitioners interested in mobile security in general and development of novel hardware-security services in particular. Successful following of the first half (parts 1-3) of the tutorial does not require any prior knowledge of mobile hardware security architectures. Basic background knowledge on mobile operating system security (e.g., Android OS security model) or embedded system security (e.g., smart card security) will help the listener to place some of the the explained concepts to context, but such background knowledge is not mandatory for following the tutorial. Basic background knowledge of mobile application development is useful for following the ObC demo part, but not mandatory.

In the second half of the tutorial (part 4) we discuss emerging hardware security standards, especially the recently introduced TPM 2.0 specification. To follow the standardization part, some background knowledge of basic trusted computing basic concepts, such as TPM commands and data structures, is useful. Our intention is to focus on the concepts and features that are new in TPM 2.0 specification, and we do not to give a complete description of TPM 1.2 as a background.

After the tutorial, the audience should have a good overall understanding on the security mechanisms that TEEs provide in current mobile devices, and what type of applications one can develop for TEEs. The audience should also have a basic understanding on recent and currently on-going standardization regarding mobile TEEs. These standardized interfaces are likely to make TEE programming possible for third-party developers, and the purpose of our tutorial is to raise the awareness of such emerging TEE development possibilities both in the research community as well as among practitioners.

3. REFERENCES

[1] ARM. Trustzone technology overview. `http://www.arm.com/products/security/trustzone/index.html`, 2009.

[2] GlobalPlatform. GlobalPlatform's GPD/STIP Solution for Mobile Security, August 2007. GlobalPlatform white paper. `http://www.globalplatform.org/uploads/STIP_WhitePaper.pdf`.

[3] Kari Kostiainen, Jan-Erik Ekberg, N. Asokan, and Aarne Rantala. On-board credentials with open provisioning. In *Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, 2009.

[4] RSA Laboratories. PKCS # 11 v2.20: Cryptographic Token Interface Standard, 2004. `ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/v2-20/pkcs-11v2-20.pdf`.

[5] Oracle. JSR 177: Security and trust services API for J2ME, 2007. Available at: `http://jcp.org/en/jsr/detail?id=177`.

[6] Trusted Computing Group. TPM 2.0 library specification, parts 1-4, Level 00, Rev. 00.96, March 2013. `http://www.trustedcomputinggroup.org/resources/tpm_library_specification`.