

SCHOOL OF COMPUTATION,
INFORMATION AND TECHNOLOGY —
INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

**Establishing trust in an updatable fTPM
using remote attestation**

Andreas Korb

SCHOOL OF COMPUTATION,
INFORMATION AND TECHNOLOGY —
INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

**Establishing trust in an updatable fTPM
using remote attestation**

**Herstellung von Vertrauen in ein
aktualisierbares fTPM durch Remote
Attestierung**

Author:	Andreas Korb
Supervisor:	Prof. Claudia Eckert
Advisor:	Albert Stark
Submission Date:	15.11.2023

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 15.11.2023

Andreas Korb

Acknowledgments

Abstract

Contents

Acknowledgments	iii
Abstract	iv
1 Introduction	1
1.1 Motivation	1
1.2 Goal	1
1.3 Threat Model	1
1.4 Environment	1
1.5 Outline	1
2 Background	2
2.1 Trusted execution environment	2
2.2 Trusted Platform Module	3
2.2.1 Discrete TPM	3
2.2.2 Virtual TPM	3
2.2.3 Firmware TPM	3
2.3 Attestation	3
2.3.1 Local attestation	3
2.3.2 Remote attestation	3
2.4 Secure Boot and Measured Boot	4
Abbreviations	5
List of Figures	6
List of Tables	7
Bibliography	8

1 Introduction

1.1 Motivation

1.2 Goal

1.3 Threat Model

1.4 Environment

1.5 Outline

2 Background

This chapter discusses the relevant background knowledge required to understand the remainder of this work.

2.1 Trusted execution environment

One of the core security concepts of operating systems are the privilege levels of processes. Thereby, processes are protected against other processes with the same privilege level. However, they are not protected against more privileged processes. This bears problems for example for cloud computing and edge computing. In cloud computing, other services, the hypervisor, or the cloud provider in general could potentially access sensitive data of the cloud tenant. In edge computing, the edge applications deal with plain text data, while they are potentially running on insecure edge devices. Hence, protection against more privileged processes is desired.

A Trusted execution environment (TEE) is an integrated hardware extension to processors. Effectively, the execution environment is separated into the Rich execution environment (REE) and the TEE. The REE runs the common software, e.g., a Linux-based operating system and the applications. The TEE allows code to be executed and memory separately to be used on a device in a hardware-protected manner that ensures a high level of confidentiality and integrity. Therefore, it aims to protect data-in-use, instead of data-in-transit or data-at-rest. Since it is integrated into the processor, there is no separate chip required. However, it is common to give the TEE a dedicated volatile memory chip, namely the secure RAM (sRAM), which is ensured to be exclusively accessible to the TEE by hardware.

Moreover, the TEE commonly follows the same user and kernel space separation as REE operating systems. The kernel space is running a trusted OS kernel, and the user space is running the trusted applications.

One such TEE is ARM's TrustZone [1]. It partitions all software and hardware resources of the containing system into the Normal world (NW) and the Secure world (SW). While the SW can access the resources of the SW and the NW, the NW is restricted to its own resources. Since ARM is the dominant processor architectures for IoT devices with a market share of 86 % [2], many of the approaches in this field

of research rely on ARM technology such as TrustZone. Our approach also leverages TrustZone to enable the execution and the remote attestation of an fTPM.

Other common TEE technologies are Intel Software Guard Extensions (SGX), Intel Trusted Domain Extensions (TDX) and AMD Secure Encrypted Virtualization (SEV). Since we focus on the implementation of our concept with ARM TrustZone, we do not go into detail about these other technologies here. However, since our concept is not tied to ARM processors and can also be applied to others, they are mentioned for the sake of completeness.

2.2 Trusted Platform Module

TPMs support three main use-cases: secure key generation, remote system attestation, and secure storage.

There are various types of TPMs. They all offer the same functionality, but with different security guarantees and are also deployed differently.

2.2.1 Discrete TPM

2.2.2 Virtual TPM

2.2.3 Firmware TPM

2.3 Attestation

2.3.1 Local attestation

2.3.2 Remote attestation

Remote attestation is the process initiated by a remote trusted party (called “verifier”) to verify that an end-device (called “prover”) has not been tampered with. For detecting that, remote attestation generally inspects the following properties of a program: (i) its code and data has been correctly loaded into memory for execution, (ii) its execution has not been redirected in unintended ways at runtime, and (iii) its data has not been maliciously modified at runtime.

A trusted anchor is required on the device to be attested because at least one trusted component is necessary to extract the data from the remote device to be verified. In many cases, TEE’s act as a trust anchor because they are hardware-protected, making it an excellent candidate for a trust anchor.

2.4 Secure Boot and Measured Boot

Secure boot is a concept of UEFI doing local attestation of components directly at boot-time. Based on signatures of next-to-boot components. It cancels the boot process as soon as deviations are detected. Binaries of components are first signed and then, deployed universally. Hence, binaries are not bound to the platform and can be considered portable in this context.

Measured Boot is a concept that is often implemented in interplay with a TPM. Measured Boot allows remote attestation to a later time. Uses sealing functionality of TPMs, therefore, bound to the exact platform.

Both technologies are often used in conjunction.

Abbreviations

TEE Trusted execution environment

REE Rich execution environment

List of Figures

List of Tables

Bibliography

- [1] ARM Limited. *ARM Security Technology - Building a Secure System using TrustZone Technology*. Issue C. 2009.
- [2] E. foundation. *IoT & Edge Developer Survey Report*. (Accessed July 2022). 2022.