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

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 Attestation

2.2.1 Local attestation

2.2.2 Remote attestation

Remote attestation is a challenge-response protocol initiated by a remote attestor. The challenge contains a nonce, enforcing a fresh response. The response must be a proof of the challenged system that it is trustworthy.

TPMs send PCR values in the form of a digitally signed quote to a remote attestor.

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.

2.3 Trusted Platform Module

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

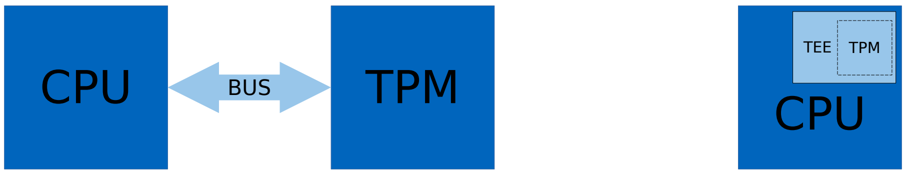
There are various types of TPMs. They all offer the same functionality, but with different security guarantees and are also deployed differently.

Allows remote attestation with PCR values in the form of a digitally signed quote to a remote attestor.

$PCR(i)_{t+1} = hash(PCR(i)_t \mid new\ measurement),\ PCR(i)_0 = 0$

Table 2.1: The PCR register usages as defined by the TPM PC Client specification [3].

PCR Index	Usage
0	SRTM, BIOS, host platform extensions, embedded Option ROMs and PI Drivers
1	Host platform configuration
2	UEFI driver and application code
3	UEFI driver and application configuration and data
4	UEFI Boot Manager code (usually the MBR) and boot attempts
5	Boot Manager Code configuration/data and GPT/partition table
6	Host platform manufacturer specific
7	Secure boot policy
8-15	Defined for use by the static OS
16	Debug
23	Application support



(a) Discrete TPM (b) Firmware TPM

Figure 2.1: Schematic illustration of the different TPM types.

2.3.1 Discrete TPM

This is the classical form of a TPM. It is a dedicated piece of hardware, connected to the CPU via a bus. It is designed and manufactured to be highly temper-resistant against hardware attacks.

The well known TPM Reset Attack was independently described in [4, 5]. It requires minimal hardware, precisely only a wire connecting the data signal of the LPC bus to ground. This results in a reset signal for the TPM, yielding predictable values for the PCR registers, i.e., 0. This allows an attacker to replay arbitrary register extends.

Winter and Dietrich [6] show attacks against the bus between a TPM and the CPU. They built-up onto the previously described TPM Reset Attack. It focuses on resetting the PCR values by active hardware attacks, while the running platform does not change. For that, a valid boot procedure is required to retrieve the valid PCR values. However, it also introduces the opposite, where the PC is reset, and an evil operating system/firmware booted, but the TPM is not aware of this, keeping the valid PCR values of the previous boot procedure.

[7] also attacks LPC bus against TPM 1.1. They observed that the data of some operations like unsealing are transmitted via the bus in plain text.

Invasive hardware attacks on dTPMs have already been shown in 2010 by Tarnovsky [8]. However, this requires a lot of time, knowledge, and resources, i.e., hardware and money.

Also, I2C common for bus system. Hardware interface even standardized by TCG.

2.3.2 Firmware TPM

As seen in the previous section, the bus between the CPU and a TPM is the biggest attack vector. An fTPM circumvents this by being directly executed within the CPU, revealing no easily accessible bus.

2.3.3 Virtual TPM

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. URL: `https://outreach.eclipse.foundation/iot-edge-developer-2021`.

[3] Trusted Computing Group. *TCG PC Client Platform Firmware Profile Specification*. Level 00 Version 1.05 Revision 23. May 2021.

[4] B. Kauer. "OSLO: Improving the Security of Trusted Computing." In: *16th USENIX Security Symposium (USENIX Security 07)*. Boston, MA: USENIX Association, Aug. 2007. URL: `https://www.usenix.org/conference/16th-usenix-security-symposium/oslo-improving-security-trusted-computing`.

[5] E. R. Sparks. "A Security Assessment of Trusted Platform Modules." In: *Dartmouth College Undergraduate Theses* (2007).

[6] J. Winter and K. Dietrich. "A hijacker's guide to communication interfaces of the trusted platform module." In: *Computers & Mathematics with Applications* 65.5 (Mar. 2013), pp. 748–761. DOI: `10.1016/j.camwa.2012.06.018`.

[7] K. Kursawe, D. Schellekens, and B. Preneel. "Analyzing trusted platform communication." In: 2005.

[8] C. Tarnovsky. *Hacking the smartcard chip*. 2010. URL: `http://www.blackhat.com/html/bh-dc-10/bh-dc-10-archives.html`.