TPM Resource Sharing for Embedded System Network Security

Da-Chuan Chen, Guan-Ruei Chen, Yu-Kai Lin, Yu-Ping Liao*

Department of Electrical Engineering, Chung Yuan Christian University
No. 200, Zhongbei Rd., Zhongli Dist., Taoyuan City 320314, Taiwan (R.O.C)
*Corresponding author: lyp@cycu.org.tw

2024 10th International Conference on Applied System Innovation (ICASI) | 979-8-3503-9492-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASI60819.2024.10547999

Abstract

Embedded systems have been widely deployed in various applications in the IoT field, making them vulnerable to security threats. The conventional approach of equipping each system with a discrete Trusted Platform Module (dTPM) chip can dramatically increase the mass deployment cost of hardware. To address this challenge, we advocate adopting a new architecture in which only the central node is equipped with dTPM and each node with Software TPM (SWTPM) by leveraging the IBM Attestation Client Server (IBMACS) to detect malicious system modification.

Keywords: dTPM, SWTPM, embedded system, remote attestation

Introduction

Embedded systems have been widely deployed in various applications in the IoT field, specializing in edge computing, controlling, and monitoring. They are also extended into other mission-critical fields like the medical field, which requires the system and handling information to be temper-proof. These embedded systems can be deployed on a large scale to ensure that information from a particular region is fully gathered and processed. It is essential to ensure mission-critical embedded systems are safe from malicious actors, achieved with reasonable expense, and simultaneously not exhausting most of their computational power to keep themselves safe.

TPM-based approach for network security is primarily due to its ability to act as the root of trust and store sets of cryptographic keys in its internal Non-Volatile Storage (NVS). This guarantees the security of encrypted data used for storing and transmitting data. Trusted Computing Group (TCG) published the specifications of TPM 1.2 and TPM 2.0 for all vendors to follow, an architecture overview for developers and researchers to understand specified architecture [1], including hardware functionalities, relationship with other parts of a computer system, and TPM Software Stack (TSS).

Modern TPM-based computer systems have two major phases: boot and post-boot. Activities in the boot phase are recorded in Platform Configuration Registers (PCRs) and Basic Input Output System (BIOS) log, and Integrity Measurement Architecture (IMA) log records events in the post-boot phase. While the study [2] proposed a general security architecture for embedded systems, other studies concentrate on securing the boot phase [3]−[4]. They aim to ensure that the booting process completes securely or terminates if compromised, preventing the loading of the operating system. However, embedded systems in real-world cases are deployed as a network, making not only the security of individual systems essential but also crucial to establish trust mechanisms between the entire network, which attestation protocols aim to help. In an attestation scheme, nodes in the network can act as either the attestor or challenger while all are equipped with discrete Trusted Platform Modules (dTPM). Studies in this field focus on creating protocols for nodes to trust each other [5]−[6], while [7] proposed an approach to enable attestation even when the challenger is not equipped with TPM. Other studies suggested an attestation protocol to achieve anonymous attestation, aiming to eliminate the need for challenger information [8]−[9], which can lead to security breaches. In this work, we propose the adoption of integrating Software Trusted Platform Modules (SWTPM) with IBM Attestation Client Server (IBMACS) to facilitate the attestation process in embedded system networks.

Embedded system hardware like Raspberry Pi and Jetson Nano are widely used to tackle tasks like audio and image processing, machine learning, and other edge computing. While it is possible to deploy the entire network of embedded system hardware with a dTPM chip, TPM hardware specially made for these popular platforms like Infineon TPM SLM 9670 can be expensive; a single one costs around 60 USD and is more expensive than the lower spec version of these popular options. If using available modules is not an option, adopting less expensive dTPM hardware with embedded system hardware requires custom hardware to be designed to fit individual use cases and extensive tests to ensure it meets the required quality. Adopting market-available TPM hardware and using custom TPM chips cannot be appropriate for large-scale deployment.

In this paper, we propose the adoption of an architecture on an embedded system network in which only the central node is equipped with dTPM. Other nodes in the network deployed with SWTPM need to regularly challenge the central node to verify that their system has not been tempered. This attestation process is achieved using IBMACS, which is responsible for transmitting PCRs, BIOS logs, and IMA logs securely.

The rest of the paper is organized as follows: Section 2 reviews the related work. Section 3 presents the system architecture for the proposed architecture. Section 4 demonstrates a real-world deployment scenario and our findings. Finally, we conclude our work in Section 5.

Overview of the Work

This section describes both the hardware and software components of our proposed architecture.

A. TPM

A Trusted Platform Module (TPM), according to the architecture overview [1], has multiple components, as shown in Fig 1. The following are the hardware components used in our proposed architecture:

ISBN 979-8-3503-9492-4

- *Non-Volatile Storage (NVS)*: A storage area for storing cryptographic keys and other sensitive data only accessible by TPM.
- *Platform Configuration Register (PCR)*: A set of registers used to store hash values of individual system booting processes. Linux systems have universally adopted the PCR usage regulations [10], as shown in Table I.
- *Random Number Generator (RNG)*: A random number generator used to generate nonces.
- *SHA-1 Engine*: A hash engine used to hash data in order to find modifications in the system.
- *Key Generation*: A cryptographic engine used to generate key pairs for attestation.
- *RSA Engine*: A cryptographic engine used to encrypt and decrypt data.
- *I/O*: An interface used to communicate with the host system. In the case of Infineon TPM SLM 9670, it uses a Serial Peripheral Interface (SPI).

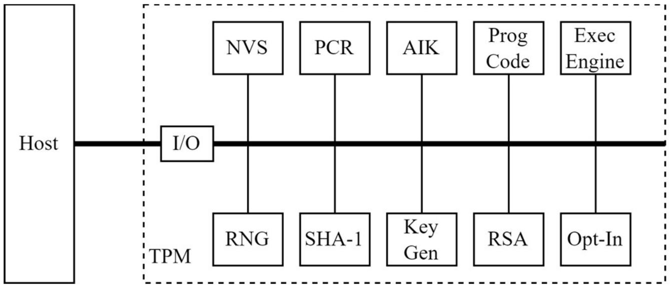


Fig. 1 TPM component architecture overview.

TABLE I
PCR USAGE REGULATIONS

PCR Index	PCR 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 and Boot Attempts
5	Boot Manger Code Configuration and 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

B. TSS

TPM Software Stack (TSS) is a bundle of software libraries used to communicate with TPM, removing the need for developers to understand the low-level details. TSS consists of Feature API (FAPI), Enhanced System API (ESAPI), System API (SAPI), Marshaling/Unmarshalling (MU), and TPM Command Transmission Interface (TCTI). Developers can choose different APIs considering storage sizes, computing power, and other platform-specific conditions. TPM 2.0 instead of TPM 1.2 is adopted in our architecture due to its additional functionalities and up-to-date specifications.

C. IBMTSS

IBM TPM Software Stack (IBMTSS) is a TSS implementation developed by IBM. While its functionality is compatible, it is not API-compatible with TCG TSS. Due to the software dependency of IBMACS, it is used in our proposed architecture.

D. SWTPM

Software TPM (SWTPM) is a software implementation of TPM that emulates TPM functionalities in software and can interact via socket interfaces.

E. IBMACS

IBM Attestation Client Server (IBMACS) is an attestation protocol developed and implemented by IBM. It can perform attestation and securely transmit PCRs, BIOS logs, and IMA logs from challenger to attestor, store them in local databases, and display entries on a webpage. The attestation scheme IBMACS provides ensures security between the attestor and challenger nodes, all of which are equipped with dTPM.

F. System Architecture

Our proposed architecture is shown in Fig 2. The central node is equipped with dTPM, while other nodes are deployed with SWTPM. We treat SWTPM similarly to dTPM during the boot phase when measured by the boot chain and recorded in PCRs. In the post-boot scenario, if all configuration and binary files of SWTPM are measured by IMA, any malicious modification to any SWTPM component will be recorded in IMA log. A daemon process on the challenger node will regularly issue challenges to interact with the attestor to ensure security. IBMACS server on the attestor node will compare entries stored in databases received from the client once a challenge is fired to detect an anomaly; any changes detected would indicate the challenger was modified. However, SWTPM can be compromised due to malicious actors potentially gaining access to SWTPM resources. Therefore, we need security mechanisms to encrypt critical components, such as NVS, using an attestor's dTPM.

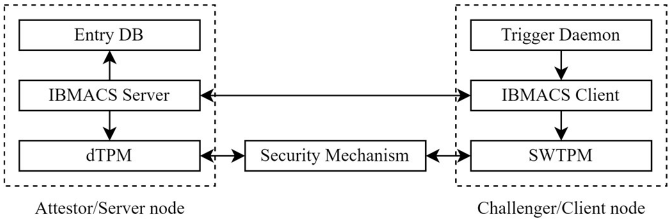


Fig. 2 Proposed system architecture.

Experimental Results

This section describes how we partially deploy our proposed architecture in real-world scenarios.

A. Hardware

We use Raspberry Pi 4 Model B as the attestor (server) node and Jetson Nano as the challenger (client) node. The attestor node is equipped with Infineon TPM SLM 9670, while the challenger node has no additional modules.

B. Attestation Software

We use IBMACS as an attestation tool in both the attestor and challenger nodes and SWTPM in the challenger node. Table 2 lists all of the software dependencies.

TABLE II
SOFTWARE DEPENDENCIES

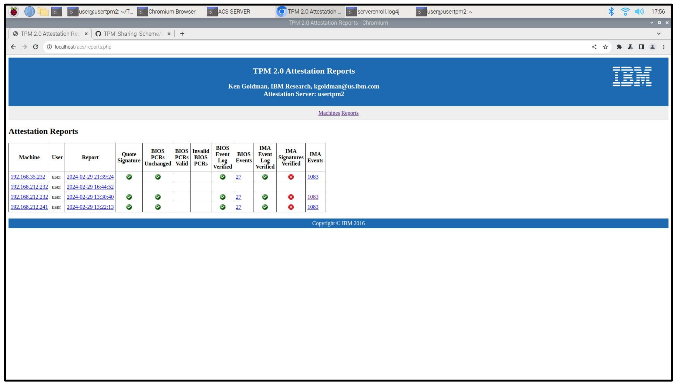
Software	Version
Raspberry Pi OS	2023-05-03-raspios-bullseye-arm64
Jetpack	jetson-nano-jp461
Tpm2-tools	5.2
Tpm2-abrmd	2.4.1
Tpm2-tss	3.2.0
Tpm2-tss-engine	1.1.0
Optiga-tpm-explorer	1.1.4
IBMTSS	1.6.0
SWTPM	1682
IBMACS	1658

C. Decoy Software

Our testing scenario focuses on a streetlight energy saving system [11], utilizing point cloud data derived from mmWave radar measurements to recognize incoming traffic. This decoy software requires the computation power embedded systems can provide and must be deployed on a massive scale to achieve its development purpose. Due to its capability to control streetlights, which greatly influence traffic once deployed, it is crucial to refrain from being maliciously modified. The mission-critical characteristic of this software makes it ideal to act as our decoy software.

D. Attestation

We have encountered issues in realizing IMA measuring SWTPM on the challenger node, so our experiments still need to implement monitoring and SWTPM security mechanism processes. However, we are able to successfully execute IBMACS on both the attestor and challenger nodes with SWTPM running on the challenger node. The results of successful attestation entries are shown in Fig 3.



TPM 2.0 Attestation Reports

Ken Goldman, IBM Research, kgoldman@us.ibm.com
Attestation Server: usertpm2

Machines Reports

Attestation Reports

Machine	User	Report	Quote Signature	BIOS PCRs Unchanged	BIOS PCRs Valid	Invalid BIOS PCRs	BIOS Event Log Verified	BIOS Events	IMA Event Log Verified	IMA Signatures Verified	IMA Events
192.168.35.232	user	2024-02-29 21:39:24	✓	✓			✓	27	✓	✗	1083
192.168.212.232	user	2024-02-29 16:44:52									
192.168.212.232	user	2024-02-29 13:30:40	✓	✓			✓	27	✓	✗	1083
192.168.212.241	user	2024-02-29 13:22:13	✓	✓			✓	27	✓	✗	1083

Copyright © IBM 2016

Fig. 3 Attestation result on attestor node web page.

Conclusion

This paper proposes an architecture for embedded system networks with only central nodes equipped with dTPM and other nodes deployed with SWTPM. We partially deployed our proposed architecture on our embedded experimental platforms and stood for the foundation of future research. Our proposed adoption of attestation architecture can serve as an essential security provider in embedded system networks and can be further integrated with other user applications in cases like cryptographic and hash engines provided with SWTPM to provide essential functionalities when processing critical data. Our proposed architecture can also integrate with trust management models from Collaborative Intrusion Detection Systems (CIDS) to further secure system networks. We plan to continue working on IMA monitoring and security mechanisms to realize our proposed architecture in our future work.

Acknowledgement

Research supported by National Science and Technology Council, R.O.C., under Grants NSTC 112-2622-E-033-001.

References

[1] Trusted Computing Group, "Tcg specification architecture overview revision 1.4," https://trustedcomputinggroup.org/wp-content/uploads/TCG_1_4_Architecture_Overview.pdf, 2007.

[2] O. Qingyu, L. Fang, and H. Kai, "High-security system primitive for embedded systems", *2009 International Conference on Multimedia Information Networking and Security*, vol. 2, pp. 319–321, 2009.

[3] K.-J. Lin and C.-Y. Wang, "Using tpm to improve boot security at bios layer", *2012 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 376–377, 2012.

[4] Promila, J. T., and S. Jain, "Tpm based secure boot in embedded systems", *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*, pp. 786–790, 2023.

[5] S. Cao, G. Zhang, P. Liu, X. Zhang, and F. Neri, "Cloud-assisted secure eHealth systems for tamper-proofing ehr via blockchain", *Information Sciences*, vol. 485, pp. 427–440, 2019.

[6] D. Lu *et al.*, "xtseh: A trusted platform module sharing scheme towards smart iot-ehealth devices", *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 370–383, 2021.

[7] R. Wang and Y. Yan, "A novel trusted boot model for embedded smart device without tpm", *2022 24th International Conference on Advanced Communication Technology (ICACT)*, pp. 228–233, 2022.

[8] A. Sedighi, D. Jacobson, and T. Daniels, "T-pki for anonymous attestation in tpm", *2021 IEEE 6th International Conference on Smart Cloud (SmartCloud)*, pp. 96–100, 2021.

[9] G. Bei and S. Guangyuan, "An anonymous attestation scheme of tpm", *2009 International Conference on Computational Intelligence and Security*, vol. 2, pp. 242–245, 2009.

[10] Trusted Computing Group, "Tcg pc client platform firmware profile specification," https://trustedcomputinggroup.org/wp-content/uploads/TCG_PCClient_PFP_r1p05_v23_pub.pdf, 2021.

[11] Y.-K. Lin, J.-C. Li, K.-L. Wang, and Y.-P. Liao, "Smart streetlight energy saving system based on mmWave radar," *2023 The 8th International Conference on Advanced Technology Innovation (ICATI2023)*, 2023.