A Secure Platform for IoT Devices based on ARM Platform Security Architecture

Junyoung Jung and Jinsung Cho
Department of Computer Engineering
Kyung Hee University
Yongin 17104, Korea
{jjy920517, chojs}@khu.ac.kr

Ben Lee
School of Electrical Engineering and Computer Science
Oregon State University
Corvallis 97331, OR, USA
benl@eecs.orst.edu

***Abstract*—Recent IoT services are being used in various fields such as smart homes, smart factories, smart cars and industrial systems. These various IoT services are implemented through hyper-connected IoT devices, and accordingly, security requirements of these devices are being highlighted. In order to satisfy the security requirements of IoT devices, various studies have been conducted such as HSM, Security SoC, and TrustZone. In particular, ARM proposed Platform Security Architecture (PSA), which is a security architecture that provide execution isolation to safely manage and protect the computing resources of low-end IoT devices. PSA can ensure confidentiality and integrity of IoT devices based on its structural features, but conversely, it has the problem of increasing development difficulty in using the security functions of PSA. To solve this problem, this paper analyzes the security requirements of an IoT platform and proposes secure platform based on PSA. To evaluate the proposed secure platform, a PoC implementation is provided based on hardware prototype consisting of FPGA. Our experiments with the PoC implementation verify that the proposed secure platform offers not only high security but also convenience of application development for IoT devices.**

***Index Terms*—IoT, Security, PSA, TrustZone, Security service**

I. Introduction

Advances in communications and computer technologies have lead to the emergence of a new computing paradigm called *Internet of Things* (IoT), which is rapidly spreading to different fields in the form of various services. However, unsecure IoT devices can be targets of security attacks such as Man-In-The-Middle (MITM), device hijacking, and Distributed Denial of Service (DDoS). In particular, *Mirai botnet* is a representative example of recent attacks on IoT devices, and it strongly suggests that there is a serious problem with IoT devices exposed to security vulnerabilities [1]. Therefore, there is an urgent need for analysis of security requirements and research on enhancing security of IoT devices.

Meanwhile, since most of IoT devices have hardware limitations such as low computational power and leakage of memory, it is difficult to apply existing security solutions to these devices. In order to solve this problem, various discussions have been carried out by standards organizations and industries. OneM2M, a global standards organization for Machine-to-Machine (M2M) services, defines the security requirements for M2M services covering IoT and recommends the use of hardware Root of Trust (RoT) for security enhancement of IoT devices. In addition, ARM proposed TrustZone as security solution for IoT devices. The TrustZone is an ARM processor-based commercial Trusted Execution Environment (TEE) that improves security reliability of IoT devices by isolating hardware and software resources of SoC into Secure world and Non-secure world [2]. ARM also proposed Platform Security Architecture (PSA), which is a security platform applicable to Cortex-M series consisting of low-end ARM processors [3]. However, only platform developers with a Non-Disclosure Agreement (NDA) with ARM can implement an IoT platform based on PSA. In addition, it is difficult for application developers to develop secure IoT services because application developers are not familiar with using PSA-enabled hardware.

In order to solve this problem, this paper proposes a secure platform for IoT devices based on ARM PSA. To provide convenience to platform and application developers as well as ensure the integrity and confidentiality of IoT devices, the proposed secure platform includes core security services and provides interfaces for developers to easily utilize these security services. In addition, the proposed secure platform was validated through a Proof of Concepts (PoC) implementation on FPGA hardware prototype.

II. Analysis of Security Requirements for IoT Devices

OneM2M defines an informative functional role models and normative technical requirements for providing M2M services [4]. Details of the Secure Requirements (SERs) are highlighted below.

- **SER-002**: The oneM2M system shall be able to ensure the confidentiality of data.
- **SER-003**: The oneM2M system shall be able to ensure the integrity of data.
- **SER-013**: The oneM2M system shall be able to provide the mechanism for integrity-checking on boot, periodically on run-time, and on software upgrades for

This research was supported by Basic Science Research Program through National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2017R1D1A1B04035914).

978-1-7281-5453-4/20/31.00©2020$IEEE$

TABLE I
SECURITY REQUIREMENTS FOR IOT DEVICES

Number	Requirement	Corresponding SERs
R1	IoT devices should be able to ensure the confidentiality and integrity of the security sensitive data, and ensure the safe management of cryptography key.	SER-002, SER-003, SER-068, SER-069
R2	IoT devices should be able to ensure the integrity of the boot components at device boot time.	SER-003, SER-013
R3	IoT devices should only update reliable and the latest firmware.	SER-003, SER-064
R4	IoT devices should be able to attest the verification of the device components at run-time and boot time.	SER-003, SER-013, SER-064, SER-069

software/hardware/firmware component(s) on M2M Device(s).

- **SER-064**: The M2M devices shall provide a mechanism to prevent installation or modification of the software/middlewre/firmware which run on M2M devices, unless it is authorized by an allowed stakeholder.
- **SER-065**: The oneM2M system shall be able to detect installation or modification of the software/middleware/ firmware of M2M devices that has not been authorized by an allowed stakeholder.
- **SER-068**: The information exchanged within the oneM2M system shall use cryptographic technology to ensure information integrity.
- **SER-069**: The oneM2M system shall be able to securely transfer information by using an appropriate method such as digital signature.

In general, the security requirements for all M2M services defined in [4] are at an outline level, which may require a lot of hardware resources to develop devices that satisfy all SERs. Therefore, these requirements are not suitable for applying to IoT devices. To solve this problem, this paper defines security requirements for IoT devices as shown in TABLE I.

III. PROPOSED SECURE PLATFORM

This section describes the proposed secure platform to satisfy the requirements in TABLE I. Secure platform is designed with an ARMv8-M SoC based on PSA that supports TrustZone-based RoT. In secure platform, there are two participants in mind: *Platform provider* and *Application developer*. When an application developer want to develop IoT services based on proposed secure platform, the Platform provider provisions the secure platform to the ARMv8-M SoC-based device.

A. ARM PSA Hardware

ARM designed the PSA to reflect the proactive investigation of threat models and security analysis of IoT devices [5]. The trusted platform resource of the PSA is called *PSA RoT*, and one of the most secure isolation methods to ensure PSA RoT is *TrustZone-based RoT* [6]. The ARM processors that support TrustZone-based RoT are the ARMv8-M family, which provides completely secure isolation methods.

As shown in Fig. 1, the ARMv8-M processor is connected to Processing Element (PE), memories, and peripherals via a bus. Memories consist Non-Volatile Memory (NVM), SRAM, and BootROM. NVM is divided into embedded Flash memory (eFlash), One-Time-Programmable (OTP) memory, and Multi-Time-Programmable (MTP) memory. The eFlash stores firmware and the *second bootloader* (2^{nd} BL), and the OTP memory provides secure storage for *Provisioning data*. Encrypted data, called *Secure data*, are stored in the MTP memory. SRAM stores the code to be executed and BootROM is used to store the *first bootloader* (1^{st} BL).

In the ARMv8-M architecture, memory map of NVM and SRAM is divided into secure and non-secure regions according to the security importance of the code to be executed, and BootROM is defined only as a secure region because it is RoT. These secure and non-secure regions can be accessed through Memory Processing Unit (MPU) which can be switched between Secure and Non-secure, respectively. Meanwhile, for memory protection, PE distinguishes states of the system as secure state and non-secure state according to memory regions, and Security Attribution Unit (SAU) within the PE defines Secure or Non-secure MPU to isolate memory access in each state. In addition, peripherals are supervised by the PE. A non-secure peripheral operates only in non-secure state while a secure peripheral can operate in both states.

There are two cases of system state transitions. State transition from Non-secure state to Secure state is performed when codes in the non-secure region requests call a secure function in the secure region. When a call is performed, it is necessary to securely store an entry point for correctly returning after executing the secure function. To securely store the entry point, PE allocates Non-Secure Callable (NSC) in secure memory region and provides the Secure Gateway (`SG`) instruction for storing an entry point in the NSC. When the secure function completes, the Secure state is switched to the Non-secure state using the `BXNS` instruction that returns to the entry point stored in the NSC. In contrast, when codes in the secure region call a non-secure function in the non-secure region, the state transitions from the Secure state to the Non-secure state. During the state transition, it is necessary to push the return address and the processor state information onto the Secure stack, and the return address on the Link Register (`LR`) should be set to a special value called `FNC_RETURN`. For these purposes, PE provides the `BLXNS` instruction, and the current system state is switched to the Non-secure state

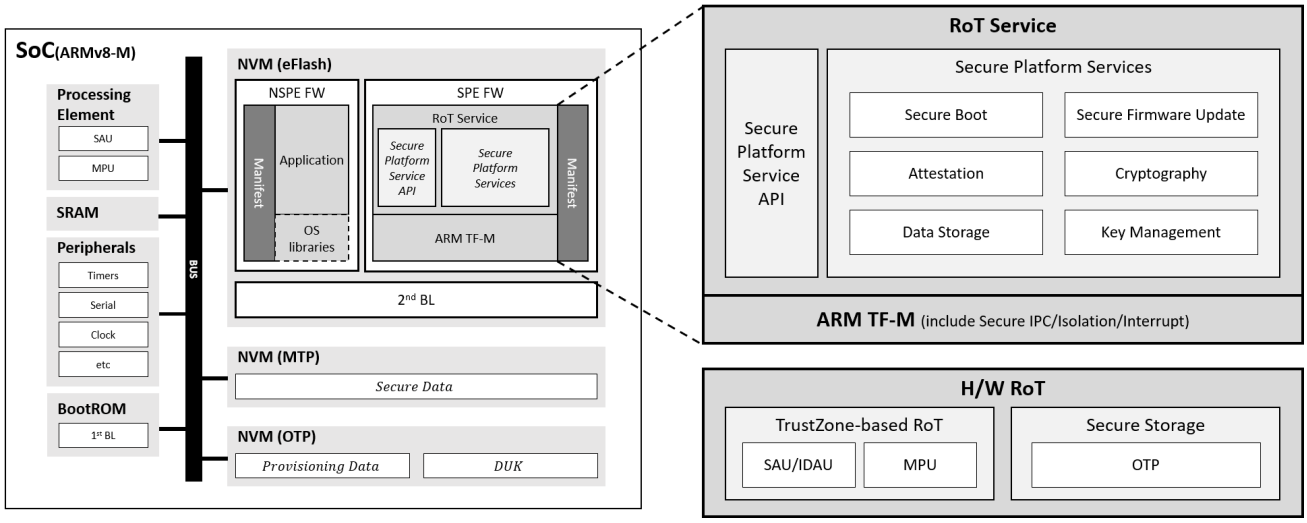

Fig. 1. System architecture of the proposed Secure Platform

after the `BLXNS` instruction is executed. When the execution of the non-secure function is completed, the branch to the `LR` set to the `FNC_RETURN` address is executed. This causes the Non-secure function to return by unstacking the return address and the processor state information stored in the Secure stack.

B. Overall System Architecture

The secure platform firmware is divided into Non-Secure Processing Environment Firmware (NSPE FW) and Secure Processing Environment Firmware (SPE FW) by providing isolation between Non-secure state and Secure state. NSPE FW is an application that is developed by the application developer. SPE FW provides secure services called *RoT Service* that is developed by the platform provider. NSPE FW and SPE FW include a manifest with firmware version, firmware hash and digital signature. The manifest of NSPE FW is signed with the application developer's private key (*PriK.App*), and the manifest of SPE FW is signed with the platform provider's private key (*PriK.Platform*). The 2nd BL contains a digital signature signed with *PriK.Platform* and is responsible for verifying the integrity of the firmware during device booting.

Provisioning data such as platform provider's public key (*PubK.Platform*), application developer's public key (*PubK.App*), and Device Unique Key (*DUK*) are provisioned by the platform provider during the device provisioning phase. *PubK.Platform* and *PubK.App* are used by the 2nd BL to verify the digital signatures of the manifests of SPE FW and NSPE FW. *DUK* is used to generate a derived key that will be store in SRAM's secure region for encryption and decryption.

C. Secure Platform Services

RoT Service belonging to SPE FW operates on the basis of ARM Trusted Firmware for Cortex-M architecture (*ARM TF-M*) as shown in Fig. 1. Due to the fact that ARM TF-M is based on H/W RoT, such as TrustZone-based RoT and Secure Storage, it can provide secure Inter Process Communication (IPC), isolation, and interrupt. Therefore, ARM TF-M can trigger both Secure and Non-secure state switches.

RoT Service based on ARM TF-M has six *Secure Platform services* and *Secure Platform Service APIs* that can be called within the NSPE FW application. Details of Secure Platform Services are described below.

- **Secure Boot**: This service checks the integrity of the boot components during the booting of secure platform. When the reset signal is applied to the SoC, the device boot begins and the 1st BL operates as the basis for Secure Boot. To form a trust of chain, the 1st BL verifies the integrity of 2nd BL using *PubK.Platform* and the 2nd BL verifies the integrity of SPE FW and NSPE FW using *PubK.Platform* and *PubK.App*. Secure Boot satisfies *R2* in TABLE I.
- **Secure Firmware Update**: This service checks the integrity of both the current and the new NSPE FW developed by the application developer. When the firmware update starts, the new NSPE FW and the manifest are stored in an empty area of eFlash. If the result of comparing the current and the new manifest is valid, the current NSPE FW is replaced with the new NSPE FW. Secure Firmware Update satisfies the the *R3* in TABLE I.
- **Attestation**: This service measures the components of platform. The results of Attestation is used for the device verification during run-time and boot time. Remote attestation is a typical use case for attestation [7]. Attestation satisfies the *R4* in TABLE I.
- **Cryptography**: This service provides hash, symmetric, and asymmetric cryptographic algorithms and the Pseudo Random Number Generator (PRNG) algorithm. Cryptography satisfies the the *R1* in TABLE I..
- **Data Storage**: This is a service to read Provisioning data from the OTP memory and to read/write Secure data from/to the MTP memory. Data Storage satisfies the the *R1* in TABLE I..

TABLE II
SECURE PLATFORM SERVICE APIS

Service	Function	Parameter(s)	Return
Attestation	int getMeasure(measureInfo_t data, measureInfo_t *result)	Measured information of device data, Result of measurement	1(Success) or 0(Failure)
Cryptography	int getRand( )	-	Random integer value by PRNG
	int hash(char *plain, int plainSize, char *digest, int *digestSize)	Plain data, Size of plain data, Digest data, Size of digest data	1(Success) or 0(Failure)
	int symmetric(int keyID, type algorithm, int enc_dec, char *in, int inSize, char *out, int outSize)	ID of derived key, Algorithm type, Encryption or decryption, Input data, Size of input data, Output data, Size of output data,	1(Success) or 0(Failure)
	int asymmetric(int keyID, type algorithm, int sign_verify, char *in, int inSize, char *out, int outSize)	ID of derived key, Algorithm type, Signing or verification, Input data, Size of input data, Output data, Size of output data,	1(Success) or 0(Failure)
Data Storage	int storeData(char *cipher, int size)	Cipher data, Size of cipher data	Page ID where data are stored (pageID)
	void loadData(int pageID, int size)	Page ID where data are stored, Size of cipher data	-
Key Management	int genKey()	-	ID of the derived key (keyID)
	void delKey(int keyID)	ID of the derived key	-

- **Key Management**: This service manages provisioned cryptography key and derived key generated by DUK. Key Management satisfies the the *R1* in TABLE I..

Meanwhile, the proposed secure platform provides *Platform Service APIs*, which provide convenience and rapid development to application developers as shown in TABLE II. The secure boot and secure firmware update services operate as system security service on the proposed Secure Platform, therefore application developers do not need to consider them.

IV. POC IMPLEMENTATION

This section presents the PoC implementation of the proposed secure platform. As shown in Fig. 2, the PoC implementation uses *ARM V2M-MPS2+ FPGA prototyping board* [8] and *CoreLink SSE-200 subsystem* [9]. V2M-MPS2+ is a development board for evaluating and developing ARM Cortex-M, and can operate with the CoreLink SSE-200 subsystem, which includes ARM Cortex-M33 processor that supports TrustZone-based RoT. Therefore, it was possible to develop PSA-based secure platform that provides isolation into NSPE FW and SPE FW.

Meanwhile, most of the proposed secure platform was implemented on the V2M-MPS2+, but this prototyping board does not have a memory that can support a secure storage such as OTP memory. Therefore, *DUK* is hard-coded to implement the 2nd BL. For this reason, future research will implement secure platform with other boards that can use tamper-resistant secure storage.

Fig. 2. ARM V2M-MPS2+ FPGA prototyping board

REFERENCES

[1] E. Bertino and N. Islam, "Botnets and Internet of Things Security," in Computer, vol. 50, no. 2, pp. 76-79, Feb. 2017.
[2] Arm Limited, "Arm Security Technology Building a Secure System using TrustZone Technology," 2009.
[3] Arm Limited, "Arm Platform Security Architecture Security Model 1.0 Alpha-2," 2019.
[4] oneM2M Partners, "TS-0002-Requirements-V4.6.0," 2019.
[5] Arm Limited, "Water Meter Threat Model and Security Analysis (English language Protection Profile) Beta-1," 2018.
[6] Arm Limited, "Arm Platform Security Architecture Trusted Base System Architecture for ARMv6-M, ARMv7-M and ARMv8-M 1.0 Beta-1," 2019.
[7] C. Kil, E. C. Sezer, A. M. Azab, P. Ning and X. Zhang, "Remote attestation to dynamic system properties: Towards providing complete system integrity evidence," 2009 IEEE/IFIP International Conference on Dependable Systems & Networks, Lisbon, 2009, pp. 115-124.
[8] Arm Limited, "Arm Versatile Express Cortex-M Prototyping Systems (V2M-MPS2 and V2M-MPS2+) Technical Reference Manual," 2016.
[9] Arm Limited, "Application Note AN521 Example CoreLink SSE-200 Subsystem for MPS2+," 2018.