Implementing a ARM-based Secure Boot Scheme for the Isolated Execution Environment

Hang Jiang, Rui Chang, Lu Ren, Weiyu Dong
State Key Laboratory of Mathematical Engineering and Advanced Computing
Zhengzhou, China
crix1021@meac-skl.cn

***Abstract*—In recent years, mobile terminals and other smart devices are an indispensable part of daily life, and a large amount of personal privacy data is stored on the smart devices. Because security threats has been moved from the traditional desktop computers to the mobile terminals, it is imperative to protect the security of the sensitive data on the devices. A variety of security defense solutions based on ARM TrustZone are proposed by academic and industry in last ten years, which are used to prevent system kernel and stored data being attacked. In these solutions, the devices need to have the ability to resist attacks in the boot phase.**

In this paper, we propose a scheme of secure boot based on TrustZone, in order to ensure that the security of operating system and prevent the device from being attacked during start-up process. We build an isolated execution environment on Xilinx zynq-zc702 evaluation board which runs the OP-TEE kernel in the secure world and Linux in the normal world. Then we build the trust chain using hardware encryption on zynq-zc702 FPGA, so as to prevent malicious attacks in the start-up process of the device. Finally, the experimental results show that the proposed scheme can prevent the malicious attack during the start-up process of the device. Moreover, the verification proved by Demper-Shafer theory demonstrates that our trust chain is more trusted than the trust chain defined by TCG.

Keywords-secure boot; ARM TrustZone; Trusted execution environment

I. INTRODUCTION

The emerging technology promotes the intelligence and miniaturization of the computer equipments. Various intelligent terminals (e.g., the smartphone, wearable devices) are playing an important role in daily life. With a large number of privacy data stored on the devices, the information security faces an increasingly complex environment. In such circumstance, how to ensure the security of intelligent devices has become an urgent problem in both academia and industry.

There are some isolation-based solutions have been put forward, including the virtualization, Trusted Platform Module (TPM), Intel SGX and ARM TrustZone technology, etc. At present, the ARM series chips are dominant in smart devices. As a hardware module within the ARM chip, compared with other isolation-based solutions, the TrustZone has a competitive advantage of the high performance, compact volume and low power consumption.

Recently, many researchers have built multiple trusted execution environment based on the TrustZone, such as OP-TEE[1], Open-TEE[2], Open Virtualization and T6. All of them follow the GlobalPlatform (GP) standard. Weihao Li [3] et al. developed the AdAtteseter on the basis of T6. Ahmed [4] et al. proposed the TZ-RKP, providing real-time protection with the normal world kernel. Ning Zhang[5] et al. applied the TrustZone and Cache-as-RAM technology to establish a cache-based isolation environment, which protected the sensitive data and code. All the secure applications are based on the assumption of a secure execution environment. However, the isolation environment created by TrustZone is not a complete secure environment. It is possible that the device is attacked during start-up process. If the attacker modifies the system image in the external memory and obtains system privileges, the security of applications in the attacked system will be unknown.

In this paper, we propose a secure boot scheme based on the ARM TrustZone, which assures the integrity and credibility of the isolated execution environment and the applications. First, we establish an isolated environment on the Xilinx zynq-zc702 evaluation board, which runs the OP-TEE in the secure world and Linux in the normal world. Then, we utilize the hardware-based cryptographic technology to construct a trust chain, and implement the proposed secure boot scheme on the device. Finally, the security of the proposed scheme is proved to be effective, and the experimental results show that it can prevent the malicious attack during the start-up process of the device. Moreover, the verification proved by Demper-Shafer theory demonstrates that our trust chain is more trusted than the trust chain defined by TCG.

The main contributions of this paper are:

- We propose a secure boot scheme based ARM TrustZone to ensure the security of operating system and prevent the device from being attacked during start-up process.
- We implement the scheme on Xilinx zynq-zc702 evaluation board, and present the experimental results and theoretical verification of the trust chain.

II. BACKGROUND

The TrustZone technology has been integrated into the ARM architecture since ARMv7. The Cortex-A8 and subsequent Cortex-A series chips are all equipped with

0-7695-6341-4/17/31.00 ©2017 IEEE
DOI 10.1109/CIS.2017.00079

TrustZone as well. TrustZone is used to establish a trusted execution environment to against the attack from both software and low-budget hardware effectively.

A. Hardware-based ARM TrustZone Architeture

According to a series of hardware extensions, ARM TrustZone divides all sources of the SoC's hardware and software into two isolated worlds, i.e., the normal world and the secure world. The AXI system bus contains a non-secure (NS) bit which indicates an operation coming from secure or normal world. The TrustZone Address Space Controller (TZASC) and the TrustZone Memory Adapter (TZMA) divide physical memory and on-chip memory into two worlds separately, which guarantee the security of the memory in the secure world. The TrustZone Protection Controller and the TZPCDECPORT in the AXI-to-APB Bridge are used to configure the security of peripherals dynamically or statically. The NS bit of SCR register in the CP15 indicates the current active world. The TrustZone also introduces a new processor mode, called monitor mode, to switch the two worlds. Based on the hardware isolation technology, the operating system and software in normal world cannot access the sensitive asset in the secure world, but the secure applications running in the secure world are able to access the resource of normal world.

B. Software-based Security Extension Technology

As described in Figure 1, the normal world runs non-secure operating system (e.g., Linux, Android), the secure world is running the secure kernel (e.g., OP-TEE, SirreTEE) and trusted service (e.g., Trustlets). Generally, the processor executes in normal world, while security service and sensitive data is stored in secure world. When the applications in the normal world need the resources of secure world, the processor will switch to the secure world and perform the corresponding service, then return to the normal world in the end. When the processor switches the state, it will trap to monitor mode first. As a part of secure world, the software in monitor mode manages the switches between the secure and non-secure processor states. Normal world traps to monitor mode is tightly controlled. It is impossible to switch the two worlds unless the following exceptions: Secure Monitor Call (SMC), interruption and external abort.

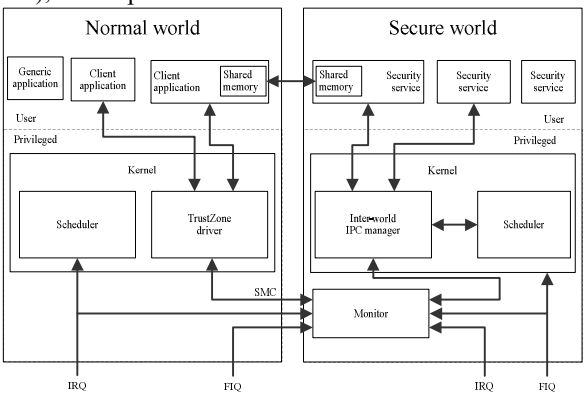

Figure 1. The architecture of the isolated exection envrionment

III. Design and Implementation

OP-TEE is a popular open source framework of trusted execution environment, and it supports several kinds of platforms including Xilinx zynq-zc702 evaluation board, QEMU and so on. We port OP-TEE to the zynq-zc702 to build an isolated execution environment in which the secure world runs OP-TEE and the normal world runs Linux kernel. As mentioned above, the built execution environment is an isolated execution environment rather than trusted. In order to build a truly trusted execution environment, secure boot is adopted through structuring root of trust and trust chain which ensures the device boot is secure.

A. Boot sequence

To establish a trusted execution environment using TrustZone technology, the Xilinx zynq-zc702 needs to load and run the secure world and the normal world kernel successively during the boot phase[6]. Zynq-zc702 boot involves several systems within the SoC device. The RootROM code will be executed when the device is power-on or reset. It initializes basic peripherals such as NAND FLASH, NOR FLASH and the processor configuration access port (PACP). Then, the BootROM loads First Stage Bootloader (FSBL) into the on-chip memory (OCM). The FSBL is provided by the Xilinx SDK, which is responsible for initializing the Processing System (PS) with configuration data. It also programs the Programmable Logic (PL) using bitstream file, and loads second stage bootloader (SSBL) into DDR memory. When the above work is finished, FSBL hands the control over to the SSBL. The U-Boot is used as SSBL, it initializes peripherals, loads and runs the OP-TEE monitor mode code. The monitor code divides the hardware and software resources of SoC, then initializes OP-TEE security kernel. Finally, OP-TEE kernel context is saved by the monitor code and the status of processor is switched to normal world to boot Linux. The main process is shown in Figure 2.

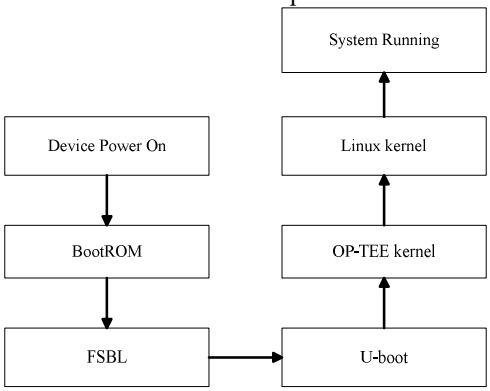

Figure 2. Boot sequence

B. Building root of trust

In this section, we specify the hash value stored in eFuse as the root of trust. In order to check FSBL whether it was tampered or attacked, the BootROM preforms RAS authentication, Advanced Encryption Standard (AES) decryption and Keyed-hash message authentication code (HMAC) authentication in sequence.

The Xilinx zynq-zc702 evaluation board provides several kinds of secure boot features, including AES, HMAC and RSA public key authentication[7]. In addition, the zynq-zc702 provides two eFuse arrays to store the RSA public key hash and AES key. As a result, the device provides adequate security authentication for secure boot.

By default, the zynq-zc702 always adopts secure boot, but it will switch to a non-secure boot if the BootROM detects that FSBL is unencrypted. After device is power-on or reset, the on-chip BootROM begins to execute to perform 128KB CRC for its own integrity. Next, the BootROM loads the boot image header and FSBL into the on-chip memory from the external memory. Then, the BootROM reads primary public key (PPK) from the boot image, and calculates SHA-256 signature which is compared with the hash value stored in the eFuse. Before decrypting and executing FSBL, the BootROM verifies FSBL by RSA authentication. The structure of its encrypted image is shown in Figure 3.

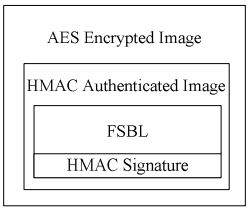

Figure 3. The structure of FSBL image

Because the AES and HMAC engines are located in PL, it must be powered up for secure boot. BootROM sends the encrypted FSBL image to AES and HMAC engines via the PCAP until PL is powered up. After decrypting and verifying FSBL, the decrypted FSBL image is sent to the PS and loaded into the on-chip memory. If all of authentication are passed, the control will be turned over to the decrypted FSBL. Figure 4 shows the process.

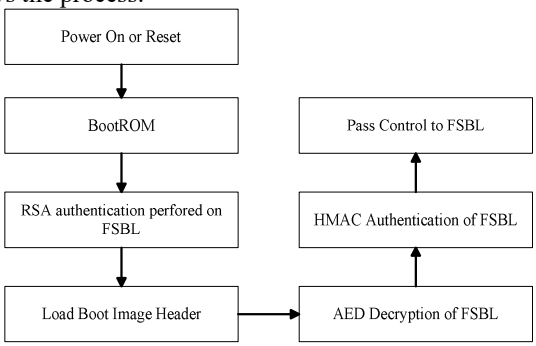

Figure 4. The process of secure boot

C. The trust chain

The main purpose of the secure boot is to establish a secure and trusted execution environment. Since the secure world stores sensitive data such as keys and certificates, it is necessary to make sure that the images loaded from the device are complete, secure and not be tampered. Generally, the bootloader and system images are stored in the external memory. Attackers can read and modify these images through tools such as JTAG. If an attacker reads images from the device and modifies malicious codes, he will control the device. Therefore the integrity of images should be verified before it executed at each stage of device boot. The device could build a trust chain through validation of each stage, and it is ensure that the execution environment is trusted.

In the previous section, we guarantee the security of FSBL through verification. In order to build a complete trust chain, it is necessary to ensure the security of the other stages of the device boot. As described in Figure 5, before the FSBL passes the processor control to U-Boot, AES and HMAC authentication is performed on the U-Boot image to ensure that the U-Boot image is safe. At each boot stage, the important thing is that the image of next stage should be verified. In this way, the entire trust chain is created. If the verification of every stage are successful, it is ensure that the device boots is secure, trusted, and not be modified by attackers.

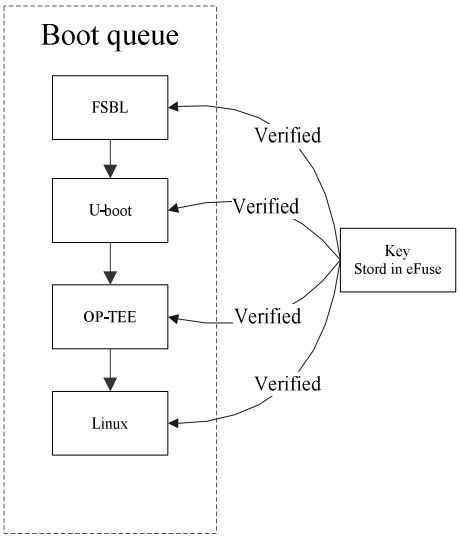

Figure 5. The structure of the trust chain

IV. EVALUATION

A. Experimental evaluation

In this experiment, we build an isolated execution environment on Xilinx zynq-zc702 evaluation board using U-Boot, Linux and OP-TEE, as showed in Table I. The U-Boot and Linux are provided by Xilinx github[1]，and the OP-TEE kernel is optee_os_2.3.0[2]. The zynq-zc702 board includes a zynq-7000 XC7Z020-1CLG484C AP SoC which consists of a SoC-style integrated PS and PL. As showed in Table II, the PS integrates two ARM Cortex-A9 application processors, AMBA interconnect, internal memories, external memory interfaces and peripherals. OP-TEE is an open source framework for the TEE using ARM TrustZone technology, which is maintained by Linaro. It meets the GlobalPlatform TEE System Architecture specification. Compared with the non-secure boot, secure boot will increase the performance overhead due to the image decryption and verification. However, it is acceptable in terms of the hardware-based encryption.

1 Xilinx github https://github.com/Xilinx
2 optee_os: Trusted side of the TEE https://github.com/OP-TEE/optee_os

TABLE I. THE SOFTWARE IN THE EXPERIMENT

Software platform	Version	Size
OP-TEE	2.3.0	217 KB
Linux	4.9.0	3.8 MB
U-Boot	2017.01	483 KB

TABLE II. THE BASIC CONFIGURATION OF ZYNQ-ZC702

Hardware platform	Configuration
Processor	ARM Cortex-A9
Memory	1 GB DDR3((four 256 Mb x 8)
On-chip SRAM	256KB
Quad-SPI Flash Memory	128 MB

B. Security evaluation

The secure boot process is initiated by the BootROM. If the attacker wants to read the chip's internal information during the execution process of BootROM, it will fail. The reason is that both the debug access port controller in PS and the test access port controller in PL are disabled with the secure boot mode. Therefore, attacker cannot access the internal registers and memory via JTAG during start-up process. In addition, the BootROM is provided by Xilinx, and it cannot be modified. Thus, BootROM is secure.

We assume that the attacker tries to read and tamper the bootloader and kernel images which are stored in the external memory. The images are checked by the RSA authentication, AES encryption and HMAC authentication. AES key is generated by the Xilinx BootGEM tool and written in the eFuse located in PL. The AES key is secure, because the written key is unreadable. The AES key cannot be rewritten unless clearing all FPGA memory. The RSA authentication has two pairs of keys, primary public key/primary secret key (PPK/PSK) and secondary public key/secondary secret key (SPK/SSK). The PPK/PSK is used to sign and verify SPK/SSK. The SPK/SSK is used to sign and verify FSBL. Both keys are generated by the openSSL. PPK is stored in the boot image, and it is easy to be read and modified by attackers. However, BootROM will calculate the hash of PPK's SHA-256 signature and compare it with the one stored in eFuse. Even if the attackers read the images, they cannot decrypt and temper the images without keys.

During the secure boot process, the images are verified by RSA authentication, AES encryption and HMAC authentication to ensure the security and prevent against the tamper. Therefore, the scheme of secure boot is feasible and effective.

C. Verification of trust chain

Demper-Shafer theory is adopted to analysis the process of trust transfer in trust chain, which has two principles.

1) Trust attenuation principle

If the trust value of node A and node B is $T(A,B)$, the trust value of node B and C is $T(B,C)$, the trust value of node A and C through B is $T_B(A,C)$, then there exists:

$$T_B(A,C) \le min\big(T(A,B), T(B,C)\big) \quad (1)$$

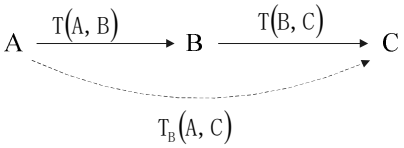

Figure 6. Trust attenuation principle

2) Trust aggregation principle

The trust value of node A and D through B is $T_B(A,D)$, and the trust value of node A and D through C is $T_C(A,D)$. As described in Figure 7, if the trust value of node A and node D is $T(A,D)$, then there exists:

$$T(A,D) \ge \max\big(T_B(A,D), T_C(A,D)\big) \quad (2)$$

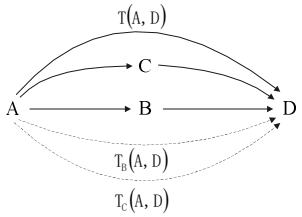

Figure 7. Trust polymerization principle

The nodes A, B, C, D and K represent FSBL, U-Boot, OP-TEE, Linux and key respectively. In the specification defined by TCG, trust transfer is a chain. In the process of trust transfer, the trust value is delivered through nodes one by one. According to Demper-Shafer trust attenuation principle, there exists:

$$T_{ABC}(K,D) \le \min\big(T(K,A), T(A,B), T(B,C), T(C,D)\big) \quad (3)$$

Furthermore, the destruction of a node in the middle will destroy the entire chain of trust.

In the structure of trust chain proposed in this paper, the final trust value $T(K,D)$ is generated by node K and node D directly. According to the Demper-Shafer trust polymerization principle, there exists $T(K,D) \ge T_{ABC}(K,D)$. The method of trust transfer can maintain trust value without loss. As described in security evaluation, the root of trust is stored in eFuse and protected by hardware, and it can guarantee the security of root of trust. Moreover, the proposed trust chain has the following advantages:

- It reduces the path of trust transfer and the loss of trust value, and increases the credibility of the system.
- Each node is independent in process of trust transfer. The update or replacement of each node does not affect the entire trust chain, and it only needs to modify itself. Comparing with the trust chain TCG specified, the proposed trust chain is easy to update.

V. RELATED WORK

The purpose of secure boot is to ensure the security and trustworthy of isolated execution environment running on the device. Through verifying each stage of the device boot, the loaded image is not tampered by attacker and keeps its integrality.

Javier González[8] et al. presented a trusted boot based on secure boot. Their method was divided into two stages. In the

first stage, the integrity of the bootloader and OS images were verified and logged at each stage of device boot with cryptographic algorithms. The key was stored in Secure Element. In the second stage, the secure service applications checked the log to verify whether the secure requirements of system was satisfied before running. In the model, the device started up normally even if the check failed at boot time, but the security service did not work well. The root of trust was the beginning and foundation of building the trust chain, and it must meet the conditions of secure storage and re-use. In above boot model, they used additional hardware Secure Element as the root of trust. Similar to Secure Element, eFuse and BBRAM were utilized as the root of trust.

SRAM Physical Unclonable Functions (PUFs) is robust and unique. Shijun Zhao[9] et al. solved the problem of hardware-relying by using SRAM PUFs. They built a building block in the on-chip memory which provided the foundations for the trust root to support the secure boot. The building block generated a unique sequence as the device key. The generated key was applied to derive public/private key pair and encrypt the kernel image. When device was powered-on or reset, the device key would reproduce the device key by SRAM PUFs and fuzzy extractor. Then it derived decryption key and decrypted the image.

In addition, the secure boot of Samsung KNOX[10, 11] used X.509 certificates and public keys embedded into the bootloader. The hash of the certificates was fused into the hardware Read-Only Memory (ROM). The secure boot would check the legitimacy of the certificates and verify the cryptographic signature of images before handing the control to the OS.

All above methods used cryptography to ensure the security of kernel image, which was also applied in our scheme. Moreover, the hardware-assisted encryption technology is used to speed up boot process.

VI. CONCLUSION

For the sake of making a truly trusted execution environment, we propose a secure boot scheme and establish a complete trust chain, which takes advantage of the hardware source provided by the zynq-zc702 evaluation board. In the boot phase, we verify the security and integrity of images through the cryptographic algorithm, which causes extra performance overhead due to the decryption and verification of images. However, we use hardware-assisted encryption technology to reduce the performance overhead.

We establish an isolated environment on the Xilinx zynq-zc702 board, which runs the OP-TEE in the secure world and Linux in the normal world. Then we utilize the hardware-based cryptographic technology to construct a trust chain, and we prove the validity of trust chain with Demper-Shafer theory. In addition, we implement the proposed secure boot scheme on the device and the experimental results show that it can prevent the malicious attack during the start-up process of the device.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 61572516).

REFERENCES

[1] Linaro.OP-TEE, "https://wiki.linaro.org/WorkingGroups/Security/%20OP-TEE,"

[2] B. Mcgillion, T. Dettenborn, T. Nyman, N. Asokan, “Open-TEE -- An Open Virtual Trusted Execution Environment, ” the 2015 IEEE Trustcom/BigDataSE/ISPA (TRUSTCOM 15), IEEE Computer Society, Aug.2015, pp.400-407, dio:10.1109/Trustcom-BigDataSe-ISPA.2015.400

[3] W. Li, H. Li, H. Chen, Y. Xia. “AdAttester: Secure Online Mobile Advertisement Attestation Using TrustZone,” the 13th Annual International Conference on Mobile Systems, Applications, and Services(MobiSys 15), ACM, May.2015, pp.75-88. doi: 10.1145/2742647.2742676

[4] A. M. Azab, P. Ning, J. Shah, Q. Chen, R. Bhutkar, G. Ganesh et al. “Hypervision Across Worlds: Real-time Kernel Protection from the ARM TrustZone Secure World,” the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS 14), ACM, Nov.2014 , pp.90-102, doi: 10.1145/2660267.2660350

[5] N. Zhang, and K. Sun, and L. Wenjing, and Y. Thomas Hou, “CaSE: Cache-Assisted Secure Execution on ARM Processors,” 2016 IEEE Symposium on Security and Privacy (SP), IEEE Computer Society, pp.72-90, doi:10.1109/SP.2016.13

[6] ARM, “ARM Security Technology Building a Secure System using TrustZone® Technology”

[7] Xilinx, “Zynq-7000 All Programmable SoC Secure Boot Getting Started Guide”

[8] J. González, M. Hölzl, P. Riedl, P. Bonnet, R.Mayrhofer, “A Practical Hardware-Assisted Approach to Customize Trusted Boot for Mobile Devices,” Information Security (ISC 2014), Springer International Publishing, 2014, pp.542-554,doi: 10.1007/978-3-319-13257-0_35

[9] Z. Shijun, Z. Qianying, H. Guangyao, Q.Yu, F.Dengguo, “Providing Root of Trust for ARM TrustZone using On-Chip SRAM,” the 4th International Workshop on Trustworthy Embedded Devices (TrustED 14) ACM,2014,pp.25-36,doi: 10.1145/2666141.2666145

[10] Samsung Electronics , “White Paper : An Overview of Samsung KNOX™,”2013

[11] K. Uri and W. Avishai,“Secure Containers in Android: The Samsung KNOX Case Study, ” the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM 16). ACM,Oct.2016 , pp.3-12, doi:10.1145/2994459.299