

A Survey of Secure Boot Schemes for Embedded Devices

Rui Wang*, Yonghang Yan**

*School of Computer Science and Technology, Xidian University, Xi'an, China

**School of Computer Science and Information Engineering, Henan University, Kaifeng, China, **corresponding author

ruigelwang@163.com, yanyonghang@henu.edu.cn

Abstract— With the rapid development of Internet of things and wireless communication technology, embedded devices are widely used in every aspect of our daily lives. Due to the lack of built-in security mechanism, the embedded devices are facing more and more threats from the malicious code attack, DDoS attack and so on. Secure boot technology can assure that the device boots up with an untampered and authorized software provided by a legitimate vendor and becomes a critical step in the device firmware and software security at the booting time. Aiming at designing and implementing the secure booting schemes, researchers have presented many effective approaches. This paper summarizes existing secure boot schemes for embedded devices and also compares and analyses the advantage and disadvantages of these existing schemes.

Keywords— secure boot; embedded devices; TPM; TrustZone; TEE

I. INTRODUCTION

Embedded devices have been present since almost the beginning of computing. In recent years, with the development of integrated circuit technology, Internet of Things (IoT) [1] and wireless communication technology, the embedded devices are becoming more popular and smarter. Embedded devices are widely used in various industrial control systems, power grid management systems, autonomous driving systems, smart home systems and military systems. Embedded devices in these systems are connected to each other through network technology for data and business interacting. As these devices often store large amounts of sensitive user data and operate in an open network environment, they are increasingly faced more and more security threats. Unfortunately, the embedded devices are brittle and vulnerable due to lack of in-built security mechanisms and special primitives for checking the authenticity and integrity of these devices. Common security attacks in insecure environments are as follows: Denial of Service (DoS), remote malware attacks, man-in-the-middle attacks, record and replay attacks and so on [2]. These attacks not only often cause that the embedded devices cannot work normally, but also access to sensitive user information illegally, which threaten the security of embedded devices seriously.

Secure boot is a process of measuring the integrity and authenticity of the operating system kernel and application software running on the device at the booting time. This secure

boot process can guarantee that the embedded devices boot up with an untampered and authorized software provided by a legitimate vendor. So, a secure boot process is the critical step for embedded devices to assure the security of device firmware, operating system kernel and application software at the booting time.

In order to realize the secure boot of computing system devices, some advanced techniques such as Trusted Platform Module (TPM) [3], Arm TrustZone [4], Microsoft's fTPM [5], Intel's SGX [6] have been designed and implemented to protect the operating system kernel and application software running on these computing system devices. The TPM presented by the Trusted Computing Group (TCG) has Platform Configuration Registers (PCRs) that cannot be overwritten and can be extended by hashing the new software measurements together with the previous PCR values. The hardware root-of-trust offered by TPM cannot be modified and can recursively construct a trusted chain from device hardware to application software to prove that the device is trusted. TrustZone uses two virtual processors called the secure and normal world respectively to enforce the hard-based isolation and ensure the secure boot of devices. The Microsoft's fTPM is a use case of the TrustZone for secure boot and attestation. The Intel's SGX provides instruction and memory access features which can ensure the trusted execution environment of the runtime, and malicious code can not access and tamper with the protected content of other programs, which further enhances the security of the system.

The above techniques are not completely suitable for embedded devices, and some researchers have proposed some specific secure boot methods to be suitable for different embedded devices based on the above techniques. The rest of this paper is organized as follows. In the next section, secure boot methods of embedded devices are presented. Section III gives the conclusion and introduces the research directions of secure boot schemes.

II. RELATED WORK

In this paper, we focus on the perspective of the problems to ensure the secure boot mechanism of embedded devices. Existing schemes can be divided into the following three categories: schemes based on hardware, schemes based on software and hybrid schemes, as shown in Figure 1. This section

describes the existing schemes and introduces the application scenarios for which each type of scheme is suitable.

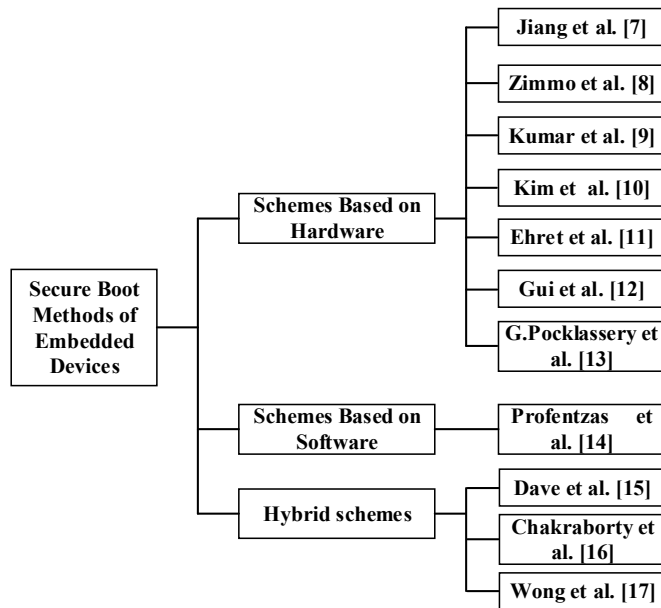


Figure 1. Classification of existing secure boot methods of embedded devices.

A. Schemes Based on Hardware

To ensure the secure boot of operating system for embedded devices, Jiang et al. [7] proposed an ARM-based secure boot scheme for the isolated execution environment, which is based on TrustZone technology. Firstly, the authors 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 they specify the hash value stored in eFuse as the root of trust and build the trusted chain using hardware encryption on zynq-zc702 FPGA to prevent malicious attacks in the start-up process of the device. Finally, the experimental results show that the processed scheme can prevent the malicious attack during the start-up process of the device. Moreover, the verification proved by Demper-Shafer theory demonstrates that their trust chain is trusted.

Zimmo et al. [8] proposed trusted boot method for embedded systems using hypothesis testing benchmark. This method contains two parts: achieving the hypothesis testing benchmark values and the trusted boot method based on the hypothesis testing benchmark values. To get the hypothesis testing benchmark values, the authors use a straight-line fitting technique to calculate the theoretical execution time of each boot sequence and record these computed execution time values. The hypothesis testing benchmark values are used to replace the hash value used in traditional chain of trust to ensure the trusted boot of the embedded device. The trusted boot process presented in this paper includes: preboot, bootloader, real-time operating system (RTOS) and application. The preboot is the root of trust (RoT), which is the verification of the first piece of code. If the integrity measurement of RoT is validated, then computing the integrity measurement of bootloader uses the hypothesis testing benchmark. If the measurement is validated, the trust is to be

passed on to RTOS. This process continues recursively till the trusted boot of application.

Kumar et al. [9] proposed a method called Post-Quantum Secure Boot (PQSB) that was implemented by fully hardware for reasons of security and performance consideration to ensure the integrity of the trusted computing base of a security system. The complete post-quantum secure boot solution based on the eXtended Merkle Signature Scheme (XMSS). XMSS is a post-quantum digital signature scheme that belongs to the family of hash-based signature schemes and uses hash chains and hash trees based on cryptographically secure hash functions. What's more, the authors integrated and validated this method on FPGA with a RISC-V based SoC platform system and compared with other hardware/software implementations of XMSS and also ECDSA. The experimental results indicate that this solution is more faster compared to corresponding hardware/software implementations and is also to compare competitively with a fully hardware elliptic curve DSA based solution.

Kim et al. [10] proposed a method based on platform integrity verification for activating a mobile trusted module (MTM). The MTM is a hardware security module based on tamper-proof Integrated Circuit (IC) card and it can be installed to embedded devices and also has the functions of platform integrity verification and cryptography management. This method stores integrity values of platform components in MTM-internal memory to protect the values from malicious attacks and verifies whether the components are illegally tampered through the comparison with the MTM-internal values. If the platform integrity verification is failed, MTM status is changed to a deactivation status although system booting is completed. The boot completion under the polluted platform is very important for platform maintenance and users. If the system booting is not completed, regular users cannot handle other critical works such as backups of encrypted data and recovery of violated platform. Moreover, on the side of system manufacturers, they need the completion of system booting for sale even to the custom OS users who don't use MTM to the system with MTM. Finally, this method can protect MTM cryptographic keys and security services although the system booting is completed, for the reason that MTM is a deactivation status under the violated platform.

Ehret et al. [11] designed a security-focused low-power SoC architecture with a hardware root-of-trust for use in edge devices. The architecture alleviates deployment-time threats faced by edge devices and meets the power and energy constraints. At the same time, it ensures the integrity of the execution environment when deployed in uncontrolled locations with the Root-of-Trust (RoT) Unit which is made up of several individual hardware modules. This architecture is named Reconfigurable Edge Computing for Optimum Resource Distribution (RECORD). This RECORD architecture is actually an accelerator-focused SoC design which includes a single RISC-V core and several reconfigurable hardware accelerators. These accelerators are designed to operate in a hierarchy, with each one consuming more resources than the last. When data must be processed, RECORD activates the lowest power accelerator first and evaluates the output to determine if further

processing is warranted. When additional processing is needed, more resource intensive accelerators are activated.

Gui et al. [12] proposed a trusted boot solution for vehicular electronic control units (ECU). This solution uses Trusted Platform Module (TPM) presented by Trusted Computing Group (TCG) to provide a secure key generation and storage cryptosystem for ECUs and validate the integrity and authenticity of the operating system and application software of ECUs.

G. Pocklassery et al. [13] proposed a Self-Authenticated Secure Boot method (SASB) for FPGAs. The process of SASB performs as follows: Firstly, at the beginning of enrollment of the secure facility, the PUF within SASB is configured to measure the path delays through corresponding functions of the SASB implementation as a tool to generate an encryption key, that is then used to encrypt the second stage boot image. Secondly, during regeneration in the field, the PUF regenerates the same key while at the same time self-authenticating the SASB image. This architecture is self-authenticating because any malicious modification to the existing SASB implementation will change the delay characteristics of one or more paths, which in turn, will introduce bit-flip errors in the regenerated key. If the process of regenerating the enrollment key is failure, then the booting of the system will be forbidden.

The above schemes depend on a hardware root of trust or a trusted execution environment. These schemes need to add additional hardware resources, which not only result in higher cost overhead, but also limit the scope of use of these schemes. The above schemes usually apply to embedded devices with large size, strong computing power and sufficient battery. The advantages of these schemes are as follows: the first is that these schemes are more security and the second is that the time required to secure boot is usually shorter.

B. Schemes Based on Software

Profentzas et al.[14] presented a software-based secure boot method for embedded devices. This method relies on several assumptions. The first is that the security of the boot stages before bootloader U-Boot cannot be modified. The second is that U-Boot has to be placed in read-only memory since there is no prior verification of the booting process. The third is that this design requires read-only storage of the cryptographic hashes used to verify the integrity of the operating system kernel. When to begin the secure boot process, the bootloader U-Boot verifies the integrity of the operating system kernel firstly. If the verification result is successfully, then the bootloader boots the operating system kernel and application. The experimental results show that the software-based secure boot method increases the overall boot-up time by 4%.

Software-based scheme typically lacks a hardware trusted of root, which makes secure boot scheme based on software more vulnerable to various attacks. Software-based scheme typically consumes fewer resources but consume more time to secure boot.

C. Hybrid Schemes

Dave et al. [15] proposed a framework of secure remote attestation with code authentication and resilience engine

(SRACARE) for trusted boot of untrusted embedded or IoT devices that verified by trusted devices that have trusted boot. The SRACARE used a custom communication protocol to measure integrity and authenticity of the device's software, which contains the following components: secure communication protocol (SCP), nonce generation technique (NGT), and remote attestation tools (RAT). SCP demonstrates the implementation of a lightweight secure authenticated communication protocol between the trusted device and untrusted device. NGT provide a lightweight novel nonce generation technique to ensure freshness for the untrusted device. RAT is used to verify the integrity and authenticity of the state of the software on the booting device by the remote booted trusted device. What's more, the SRACARE also presented a novel resilience and onboard recovery method to protect the device from the malicious code modification attacks. After malicious code modification attack, the recovery method provided by SRACARE will recover the device.

Chakraborty et al. [16] presented a software based trusted boot method called simTPM for mobile devices. The simTPM was implemented on a Hikey960 using a Gemalto Multos card as SIM card. The simTPM solves the technical challenge of implementing the compliant functionality of TPM2.0 on the SIM card, which does not require any additional hardware for the TPM. The authors propose a scheme to integrate this simTPM with the on-board TEE to solve the problem of binding the simTPM to the RTM and present an alternative solution based on distance bounding. Due to this binding, the trusted boot on the ARM Trusted Firmware (ATF) secure boot chain can be realized. The results of performance evaluation show that this simTPM is competitively fast to a reported fTPM implementation and is comparable with existing hardware TPMs. This approach also keeps the costs down and makes full use of dormant hardware capabilities of mobile devices.

Wong et al. [17] presented a hybrid approach called SMARTS which is actually a dynamic root-of-trust architecture for secure boot of low-end devices. This dynamic root-of-trust architecture relies on a Memory Protection Unit (MPU) which operates as an extension to the Memory Controller (MC) and is responsible to provide an authenticated encryption so as to guarantee confidentiality and integrity protections of the memory contents and prevent them from replay attack. The proposed MPU supports partial memory encryption (PME) mechanism with the additional support of dynamic and reconfigurable secure and trusted region. When the system boots up, DRAM allocation is freed and repartitioned into finer granularity regions where each can be assigned to: trusted region, non-trusted region or metadata region. In compliant with the secure and trusted boot up sequence, these regions allocation is protected from interception and modification by software.

Typically, hardware-based schemes consume more resources but take less time to boot securely. Software-based schemes consume fewer resources but take more time to boot securely. Hybrid solutions provide a good compromise between hardware-based schemes and software-based schemes, and they generally require less hardware resources and a longer time to

secure boot than hardware-based schemes but consume less time than software-based schemes.

III. CONCLUSIONS

The secure boot technology is very important for the security assurance of embedded devices. In this paper, the secure boot technology of embedded devices is studied deeply and the existing solutions are summarized. Then the advantages and disadvantages of these schemes are given. In the future, we will design effective solutions to the problem of secure boot of small embedded devices or IoT devices.

REFERENCES

- [1] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, Moussa Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," 2015 IEEE Communications Surveys & Tutorials Volume: 17, Issue: 4, Page(s): 2347 - 2376.
- [2] Jorge Granjal, Edmundo Monteiro, Jorge Sá Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," 2015 IEEE Communications Surveys & Tutorials, Volume: 17, Issue: 3, Page(s): 1294 - 1312.
- [3] Keith Mayes, Konstantinos Markantonakis, Smart Cards, Tokens, Security and Applications (second edition), Springer, 2016.
- [4] T. Alves and D. Felton, TrustZone: Integrated Hardware and Software Security-Enabling Trusted Computing in Embedded Systems (July 2004).
- [5] Himanshu Raj, Stefan Saroiu, Alec Wolman et al., "fTPM: A Software-Only Implementation of a TPM Chip", 25th USENIX Security Symposium, pp:841-856, Austin, TX, August 10-12, 2016.
- [6] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar, "Innovative instructions and software model for isolated execution," in Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy, ACM, 2013, pp. 1-1.
- [7] Hang Jiang et al. "Implementing a ARM-Based Secure Boot Scheme for the Isolated Execution Environment." 2017 13th International Conference on Computational Intelligence and Security (CIS).
- [8] Sara Zimmo et al. "Trusted Boot for Embedded Systems Using Hypothesis Testing Benchmark" 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE).
- [9] Vinay B. Y. Kumar et al. "Post-Quantum Secure Boot" 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE).
- [10] Daewon Kim et al. "A method based on platform integrity verification for activating a mobile trusted module" 2015 International Conference on Information and Communication Technology Convergence (ICTC).
- [11] Alan Ehret et al. "A Hardware Root-of-Trust Design for Low-Power SoC Edge Devices" 2020 IEEE High Performance Extreme Computing Conference (HPEC).
- [12] Yutian Gui et al. "Hardware Based Root of Trust for Electronic Control Units." SoutheastCon 2018.
- [13] G. Pocklassery et al. "Self-authenticating secure boot for FPGAs" 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST).
- [14] Christos Profentzas et al. "Performance of Secure Boot in Embedded Systems" 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS).
- [15] Avani Dave et al. "SRACARE: Secure Remote Attestation with Code Authentication and Resilience Engine" 2020 IEEE International Conference on Embedded Software and Systems (ICCESS).
- [16] Dhiman Chakraborty et al. "simTPM: User-centric TPM for Mobile Devices." 28th USENIX Security Symposium. page:533-550.
- [17] Ming Ming Wong et al. "SMARTS: Secure Memory Assurance of RISC-V Trusted SoC." In HASP'18: Hardware and Architectural Support for Security and Privacy, June 2, 2018.



Rui Wang (M'2021, Luoyang, 1995) received B.S. degree in computer science and technology from Henan University at 2018. And now she is a postgraduate in computer science and technology from Xidian University, Xi'an, China. Her research interest contains trusted computing.



Yonghang Yan (Zhoukou, 1981) received B.S. degree in computer science and technology from Zhengzhou University at 2004, M.S. degree in computer science and technology from Beijing Institute of Technology at 2007 and Ph.D degree in computer science and technology from Beijing Institute of Technology at 2014. His research interests include computer network, Mobile Ad hoc and Sensor Network, UAV network, mobile computing network and QoS.

Now he is an associate professor in the School of Computer and Information Engineering at Henan University, Kaifeng, China. He is the head of advanced network technology laboratory.