Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 4, No. 2, 2023

Overview of System-Level Security Technologies based on ARM TrustZone

Pengfei Deng [1, 2], Xiyin Liang [1, 2], Peirong Pan [1, 2], Xu Pan [1, 2, *]

[1] Department of Physics and Electronic Engineering, Northwest Normal University, Lanzhou, Gansu 730070, China
[2] Gansu Province Intelligent Information Technology and Application Engineering Research Center, Lanzhou, Gansu 730070, China
[*] **Corresponding author**: Xu Pan (Email: 1009350770@qq.com)

Abstract: With the rapid development of embedded technology and the increasing complexity of system functionality, there is a growing need for a trusted computing environment to ensure the security, integrity, and reliability of sensitive information. Systems not only need to protect the security of sensitive application code but also ensure the isolation of their execution process to prevent attacks and data theft. Traditional system protection is achieved by using security mechanisms that run in the same address space and privilege level as the kernel. However, this approach is not sufficiently secure as attackers who compromise the kernel can also compromise these security mechanisms. To achieve true kernel and critical data protection, security mechanisms need to be isolated. Therefore, building a trusted isolation runtime environment in the system is crucial for system security. TrustZone technology, developed by ARM, is a system-level security isolation framework capable of defending against various potential attacks. this paper provides an overall overview of different security isolation technologies. By concentrating on the principles and characteristics of ARM TrustZone, the paper conducts an in-depth analysis of system security isolation technology based on TrustZone. Finally, considering the existing security issues in the field of trusted execution environments, the paper presents prospects for the future development of this technology.

Keywords: ARM TrustZone; Safe Isolation; Trusted Execution Environment.

1. Introduction

With the booming development and wide application of information technology, people are enjoying the convenience of network resources, but also have to deal with many potential insecurity factors in the network. Embedded devices such as smartphones, as an indispensable part of life, play the role of accessing, storing, manipulating, and transmitting sensitive information, so how to ensure the security of the end devices and their applications and services becomes particularly important.

System designers usually add specific cryptographic algorithms and security protocols to the system to ensure the deployment of more secure software systems. In addition, for the current processor and chipset hardware, design vendors can also take more measures, such as the use of TPM (Trusted Platform Module) metrics software to ensure that the original software has not been maliciously tampered with; the use of the MMU (Memory Management Unit) and the addition of a number of new registers can be very good is the implementation of the security domain segregation and the new access control primitives; and storing device keys and trusted roots in the internal memory of the chip processor to protect them from offline attacks. However, these software-only or hardware-only security measures reveal their own shortcomings in terms of security design.

In order to enable security throughout the system design process and to protect the data stored in mobile devices, the industry has proposed the Trusted Execution Environment (TEE) technology for private computing, which is a technology that allows data to be computed in a way that ensures that the data is "available but invisible", and that is a segregated processing environment where the code and data are protected during execution and the memory area is separated from the rest of the processor and provides confidentiality and integrity properties [1]. and data are protected during execution, with memory areas separated from the rest of the processor and provided with confidentiality and integrity properties [1].

ARM TrustZone, as a TEE solution introduced by ARM, is a hardware-based System on Chip (SoC) security architecture that is heavily used in smart mobile devices [2]. The advantage of ARM TrustZone technology is that it protects the security and integrity of data from malicious attacks. Based on this technology, ARM creates a TEE for security-sensitive code and data to achieve system-level isolation. A typical TrustZone-based system divides the resources of the entire SoC, including hardware components and running software, into two worlds with different privilege settings and protections. Of these two worlds, the common world is responsible for running the operating system as well as all the regular applications, called the Client Application (CA). In contrast, the other isolated security world performs a small, trusted and secure operating system, as well as some advanced security management tasks developed by Trusted Application (TA). Within the TEE there can be TrustZone-based operating systems such as Qualcomm's QSEE [3], the open-source OPTEE [4], etc. that provide security services for security-sensitive applications. All operations in the system that do not require protection are performed in the ordinary world.

As can be seen, security isolation technology is a very important means of ensuring the security and reliability of a system by preventing threats arising from mutual interference between different components of the system. To this end, the construction of a secure isolated operating environment in the system, so that it cannot only ensure the normal execution of sensitive applications, protect private data and sensitive information, but also able to detect, monitor and protect the system from malicious behavior, thus improving the ability of

the system to resist security threats. In short, in the protection of system security isolation technology has become a trend, to a certain extent, to break the bottleneck of the current system security, it has become an important system security technology.

This paper firstly gives an overall overview of various security isolation technologies, then focuses on analyzing the basic principle of ARM TrustZone and its framework, and further discusses the security isolation technology based on ARM TrustZone. Finally, it looks forward to the future development direction and application requirements of the security isolation technology by combining the current security problems in the field of trusted execution environment and the advantages of this technology in security.

2. Overview of Security Isolation Techniques

2.1. Hardware Isolation Technology

In order to ensure the security of sensitive information in the system, the designer considers providing a relatively secure hardware isolation environment by designing a dedicated security hardware module, which is used to implement access control and make it difficult for the running software to bypass this isolation mechanism. This allows critical data, keys, or encryption and decryption services of the system to be stored in the module and restricts access to other illegal software [5]. This type of isolation is typically provided by the processor or a specialized device connected to the main processor. Most processors provide MMUs to assign different virtual addresses to different processes, thus performing process isolation.

There are two more mainstream solutions to realize hardware isolation: one solution is to design a dedicated hardware security module outside the SoC during chip design; the other is to integrate a hardware security module inside the SoC during chip design. The first of these is more widely used in SIM cards and smart cards in cell phones [6]. The second one consists of two main categories: hardware security modules that manage cryptographic operations and key storage, and general-purpose processors designed specifically for security subsystems.

2.2. Software Isolation Technology

Software isolation technology is the construction of a trusted and isolated operating environment at the software layer, thereby limiting the spread of malicious code or protecting trusted software, trusted code or sensitive data in that isolated environment. Typical software isolation techniques include virtualization techniques. The function of virtualization technology is to abstract a virtual software or hardware interface to ensure that the software modules on it can run on a virtualized environment. Virtualization essentially reproduces an entire physical server and runs an application as a virtual machine, with a virtualization monitor abstracting server resources and allocating resources to virtual machines (VMs).

Virtualization technology can be implemented at all levels of the system, which mainly includes hardware virtualization, OS virtualization and application virtualization. The typical hardware virtualization is introduced here as an example, and its system virtualization architecture is shown in Figure 1. The system virtualization architecture is shown in Figure 1. By using virtualization technology, multiple virtual hardware abstraction layers can be abstracted, thereby isolate multiple client operating systems. Typical virtual machines are KVM (Kernel-based Virtual Machine) and Xen, the former as a kernel-based virtual machine, which is a very small module of the Linux kernel; the latter mainly runs on bare metal.

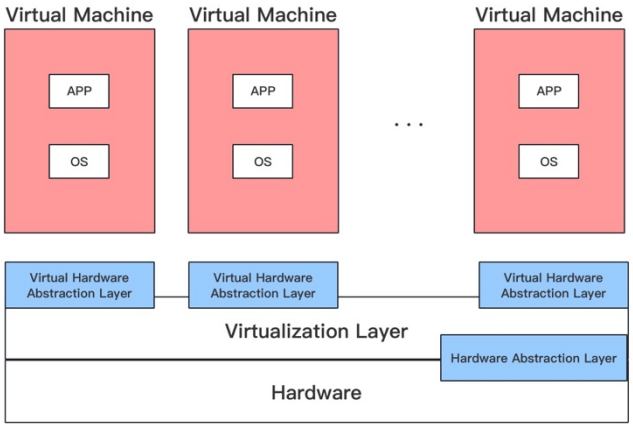

Figure 1. System Virtualization Architecture

Specifically, hardware virtualization abstracts the entire hardware layer of the system, with the processor instruction set as its interface to the guest operating system. The virtual machine runs as a higher privileged software isolation layer between the hardware and the operating system. It abstracts the entire hardware layer and implements a virtual-to-physical resource mapping mechanism, and therefore can assist the guest operating system in mapping to specific hardware devices. When the guest operating system needs to use sensitive instructions to access system resources, the virtual machine can also intercept this operation and process it by emulating the corresponding instructions. This mechanism of trapping sensitive instructions and then having the VM simulate the processing of those instructions and return the results to the guest OS effectively prevents the execution of illegal operations. At the same time, the virtual machine is able to save and switch multiple guest operating systems, which effectively ensures that each virtual system is securely isolated from each other, independent of each other, and does not affect each other [7].

2.3. System-level Isolation Technology

System-level isolation technology is mainly through the security expansion of hardware, and with the corresponding trusted software to build a relatively safe and reliable trusted execution environment (Trusted Execution Environment, TEE) in the system [8].

TEE is a secure area on the main processor, which provides an isolated execution environment that can guarantee the isolated execution of programs, the integrity of trusted applications, the confidentiality and secure storage of trusted data, etc., and can guarantee the security, confidentiality, and integrity of the code and data loaded into this environment. TEE is an independent execution environment that runs in parallel with the ROS (Rich OS) to provide security services for the feature-rich ROS. TEE is an independent execution environment running in parallel with ROS (Rich OS), which provides security services for the feature-rich ROS, and contains a set of application program interfaces to satisfy the communication between REE (Rich Execution Environment) and TEE, which consists of Trusted OS (TOS) and the trusted applications running on it. The REE communication agent

provides support for messaging between client applications and trusted applications. Trusted applications running on the TEE have access to the full functionality of the device's main processor and memory, and hardware isolation technology protects them from malicious attacks from the ROS. Software and cryptographic isolation within the TEE ensure that trusted applications within the TEE are isolated from each other. both GlobalPlatform and TCG have been working on TEE in recent years, with the former focusing on the development of a standard specification for the TEE, and the latter trying to combine the TEE specification with its Trusted Platform Module (TPM) specification to enhance the security and trustworthiness of the device and releasing its newest specification, the TPM2.0 specification. TPM2.0 specification [9].

In July 2010, Global Platform first announced their own TEE standard, as shown in Figure 2, proposing a TEE Client API (an interface to interact with the TEE), which was later expanded to include a TEE Internal API and a complete set of TEE system specifications. The TEE Functional API is a wrapper around the Client API to allow developers to use the TEE Functional API as an interface to the TEE Functional API and the TEE Functional API. The TEE Functional API is an encapsulation of the Client API so that developers can use a standardized interface model to develop applications that invoke security services in ROS.

The specifications included in Global Platform are summarized as follows: TEE Systems Architecture specification describes the hardware and software architecture under TEE; TEE Client API Specification defines how ROS applications communicate with TEE trusted applications; TEE Internal API Specification defines how to develop trusted applications that can run inside TEE, and provides programming interfaces for trusted applications; in order to have friendly interactions with users, TEE can also provide Trusted GUI user interface TGUI (Trusted GUI), which can provide a secure display of user's private data in TEE, and it is completely independent of ROS running in the ordinary execution environment. It is completely independent of ROS running in the normal execution environment, and therefore is not vulnerable to attacks from malicious ROS systems. And TEE allows client applications and trusted applications to communicate data efficiently in the form of shared memory, and this mechanism well ensures that client applications can call all kinds of trusted applications designed within the trusted execution environment. Generally, TEE has access to resources in the REE and vice versa. Because Global-Platform's TEE specification forms the basis of the TEE environment, general commercial or open-source products will refer to the specification, and in accordance with its definition of a variety of functional interfaces for the specification of the implementation, thus ensuring the consistency of the Trusted Execution Environment developed by different vendors.

A number of TEEs have been proposed in both the commercial and academic worlds, with T-base developed by Trustonic and MobiCore developed by JITEH. Abroad, Andreas Fitzek of the Technical University of Graz has also developed an ANDIX OS [10], which is a multitasking-supporting, nonpreemptive operating system. Jitesh of North Carolina State University developed a secure OS using ARM TrustZone technology [11]. Domestically, Shanghai Jiao Tong University developed T6 [12], which is a micro-trusted OS based on the ARM TrustZone Secure Extension Processor with a common execution environment that supports running systems such as Linux and Android. In addition, the Linux community developed an open-source TEE called OPEN-TEE [13].

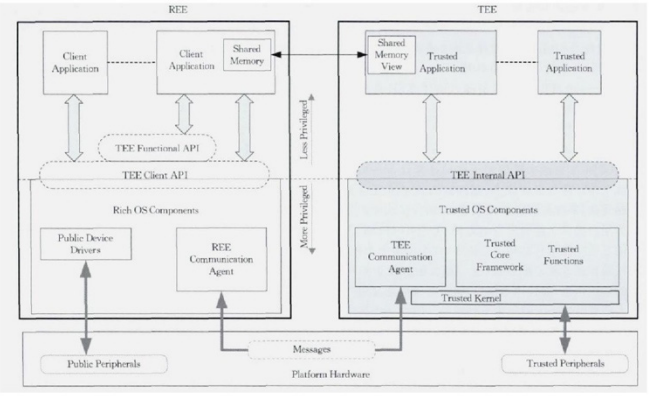

Figure 2. TCG TEE System Architecture

3. Overview of the ARM TrustZone security architecture

3.1. Hardware Isolation Technology

ARM processors supporting TrustZone technology are divided into two regions, i.e., a single physical processor contains two virtual processor cores: a secure processor core and a normal processor core [14]. Both regions have their own user space and kernel space, as well as caches, memory, and other resources, and their hardware framework is shown in Figure 3. The secure processor can access the resources of the normal processor and vice versa, which is the fundamental reason why the ARM TrustZone technology is able to achieve hardware-level protection and isolation of system resources. At the same time, to improve system security based on ARM TrustZone technology, it is necessary to make corresponding extensions to the system hardware and processor cores. These extensions guarantee safe memory and safe peripherals and can deny access to non-secure transactions. Therefore, they can hide and isolate themselves well in the normal operating system, thus realizing the true sense of system security.

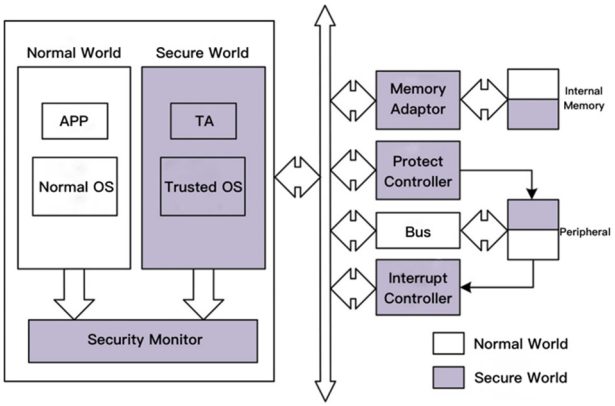

Figure 3. ARM TrustZone Hardware Architecture

3.2. Hardware Isolation Technology

The typical software architecture of ARM TrustZone is shown in Figure 4. the untrusted OS runs in the normal world, also called Rich Execution Environment (REE). the TEE runs in the secure world, the protection ring EL1 is the trusted OS and provides runtime support, and the protection ring EL0 running in user mode is able to the protection ring EL0,

running in user mode, is able to maintain the lifecycle of the TA. At the heart of the Trusted OS is the Trusted Kernel, which provides the basic OS primitives for scheduling and managing TAs. The Trusted OS also implements device drivers for accessing trusted peripherals, handles cross-world requests by switching security monitoring modes to invoke SMC instructions and shared memory, and implements shared libraries (e.g., encryption) and TEE primitives, i.e., Remote Authentication, Trusted I/O, and Secure Storage.

In addition to the trusted operating system, the TrustZone software architecture includes two basic software components: the security monitor and the TEE boot loader. The security monitor implements the mechanism for switching security contexts between worlds and runs with the highest privileges in the protection ring EL3. The TEE bootloader boots the TEE system to a secure state, and it is critical to implement the trusted boot primitive, which is split into two steps, running first in EL3 and then in EL1. Together, the trusted operating system, security monitoring, and the TEE bootloader form the Trusted Computing Base (TCB) software for a typical TEE system.

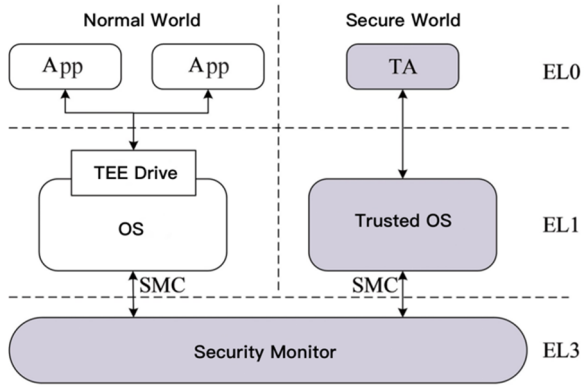

Figure 4. ARM TrustZone Software Architecture

4. Conclusion

4.1. Strengths and Weaknesses of TrustZone Technology

TrustZone, an embedded platform security technology proposed by ARM, protects peripherals such as secure memory, cryptographic blocks, keyboards, and displays through physical isolation with minimal impact on power consumption, performance, and area while minimizing the impact on the original processor design. Therefore, this technology has certain advantages in improving the security of embedded systems. First, it provides a physically isolated environment that can provide secure storage space for private data and keys for encrypted and decrypted data, which is a natural way to maintain confidentiality. Second, TrustZone can guarantee the bus bandwidth of the entire storage space, which makes it different from other security technologies and has minimal impact on performance. In addition, the TrustZone system architecture is a combination of hardware and software. With the hardware architecture fixed, users can meet the needs of customizing and upgrading the security system by developing the corresponding TA in TEE.

But TrustZone technology is not omnipotent, its own shortcomings, mainly reflected in the following aspects. First, it can only defend against various software attacks, it is difficult to defend against physical attacks, such as physical tampering with the device's main memory. Second, it only provides an isolated execution environment, and does not prove to the user or the remote person that this environment is trustworthy. In addition, although users can meet the needs of customizing and upgrading the confidential system by developing appropriate TAs in the TEE, some new hardware-dependent technologies may not be applied to old hardware architectures, thus failing to defend against new types of attacks. Finally, providing a reliable trusted root for the system platform is the cornerstone of guaranteeing the security of the whole system, and the current technology is to solidify the device key as the trusted root through the system-on-a-chip, which has the drawbacks that the key is difficult to be updated, and once leaked, it will cause the whole platform to be unusable, and this method requires the device key to be stored in the device for a long period of time, whose security is difficult to be guaranteed, for example, how to defend against side-channel attacks, fault attacks, and reverse engineering and other types of attacks.

4.2. Future Outlook

In the future, TrustZone technology is used in emerging fields such as IoT and blockchain. Wearables and IoT devices are becoming more and more common in modern society. These devices continue to generate privacy-related data from a variety of sensors. The problem is that securing IoT devices can be a challenge as hardware requirements and cost constraints drive different design directions. To address this, ARM has extended TrustZone to a new generation of microcontrollers (ARMv8-M), making security practical at scale and across the value chain. By building in TrustZone, ARM mitigates the economics of security, reducing the risk, cost and complexity of implementing robust security measures.

Blockchain technology is expected to drive a revolution across manufacturing. For example, some supply chain use cases may benefit from transparent asset tracking and automated processes using smart contracts. However, in real-world deployments, blockchain's transparency is both a benefit and a drawback. Exposure of assets and business interactions can raise serious security risks. However, there is usually no confidentiality scheme to protect smart contract logic as well as processed data.TZ4Fabric is an extension of Hyperledger Fabric that utilizes ARM TrustZone to securely execute smart contracts. The design minimizes the size of the trusted computation for execution by avoiding executing the entire Hyperledger Fabric node in a TEE (which continues to run in an untrusted environment). At the same time TZ4Fabric restricts it to only executing smart contracts, a design that not only leverages ARM TrustZone, but also takes advantage of the modular architecture to extend TZ4Fabric to future TEEs.

Acknowledgments

This work was supported in part by a grant from Education technology innovation project of Gansu Province (2021CY ZC-22).

References

[1] Jang J , Kong S , Kim M ,et al. SeCReT: Secure Channel between Rich Execution Environment and Trusted Execution Environment[C]//Network & Distributed System Security Symposium.2015.

[2] Demigha O , Larguet R .Hardware-based solutions for trusted cloud computing – Science Direct[J].Computers & Security, 103[2023-07-13].

[3] Ji D , Zhang Q , Zhao S ,et al.MicroTEE: Designing TEE OS Based on the Microkernel Architecture[J].IEEE, 2019.

[4] Fabien M .The Bureau of Reclamation and Practical Applications in Desalination and Membrane Separation Technologies [J]. 2000.

[5] Ravi S , Raghunathan A , Kocher P C ,et al.Security in Embedded Systems: Design Challenges[J].ACM Transactions on Embedded Computing Systems, 2004, 3(3):461-491.

[6] Moghimi D , Sunar B , Eisenbarth T ,et al.TPM-FAIL: TPM meets Timing and Lattice Attacks[J]. 2019.

[7] Yang X , Shi P , Tian B ,et al.Trust-E: A Trusted Embedded Operating System Based on the ARM Trustzone[C]//IEEE International Conference on Autonomic and Trusted Computing; IEEE International Conference on Ubiquitous Intelligence and Computing; IEEE International Conference on Scalable Computing and Communications and Associated Symposia/Workshops.0[2023-07-13].

[8] Shepherd C , Arfaoui G , Gurulian I ,et al. Secure and Trusted Execution: Past, Present and Future -- A Critical Review in the Context of the Internet of Things and Cyber-Physical Systems[C]//Trustcom/bigdatase/ispa.IEEE, 2017.

[9] Jinwen W , Yong J , Qi L I ,et al.Survey of Research on SGX Technology Application[J].Journal of Network New Media, 2017.

[10] Benhani E M , Marchand C , Aubert A ,et al.On the security evaluation of the ARM TrustZone extension in a heterogeneous SoC [C]//2017 30th IEEE International System-on-Chip Conference (SOCC).IEEE, 2017.

[11] Johnson W , Eizirik E , Pecon-Slattery J ,et al.The late Miocene radiation of modern Felidae: a genetic assessment.[J].Science (New York, N.Y.), 2006, 311(5757):73.

[12] Winter J .Trusted computing building blocks for embedded linux-based ARM trustzone platforms[C]//Acm Workshop on Scalable Trusted Computing.ACM, 2008.

[13] Sarker A K , Islam M K , Tian Y .MVAM: Multi-variant Attacks on Memory for IoT Trust Computing[J]. 2023.

[14] Huang Q X , Chiu M Y , Yeh C S ,et al.STBEAT: Software Update on Trusted Environment Based on ARM TrustZone [J]. Sustainability, 2022, 14.