Enhancing IoT Security Through Trusted Execution Environments

Zheng Zhang

College of Big Data and Information Engineering, Guizhou University, Guiyang, 550025, China
cse.zzhang21@gzu.edu.cn

Abstract. As the Internet of Things (IoT) proliferates, securing these interconnected devices has become a critical concern. Trusted Execution Environments (TEEs) offer a crucial mechanism for bolstering IoT device security. This paper delves deeply into the application of TEEs within the IoT ecosystem to protect sensitive data and operations. It begins by discussing the necessity of hardware support in implementing TEEs, followed by an examination of the core security principles essential to their functioning. A systematic analysis then explores successful deployments of TEEs in IoT devices, with case studies illustrating their effectiveness in enhancing security. Further, this study reviews current IoT security regulations, providing insights into how TEEs can aid in compliance. The discussion transitions to the challenges associated with integrating TEEs into IoT devices, focusing on scalability, network complexity, and specific security vulnerabilities inherent to TEEs. This research highlights the significant role of TEEs in strengthening IoT devices against an evolving landscape of cyber threats, while also recognizing the complexities involved in their implementation.

Keywords: IoT Security, Trusted Execution Environments, Hardware Support, Security principles.

1 Introduction

The Internet of Things (IoT) signifies a fundamental shift in the interaction between the world and technology, imbuing everyday objects with intelligence and facilitating their ability to communicate and collaborate. As IoT rapidly transforms industries, forges smart cities, and revolutionizes healthcare systems, it also introduces significant security challenges. The diverse and distributed nature of IoT devices renders them susceptible to cyber-attacks, threatening personal privacy, corporate data, and national security. A promising approach to addressing these security challenges involves the adoption of Trusted Execution Environments (TEEs) [1]. TEEs provide a secure area within the main processor where sensitive code and data are isolated, processed, and protected, increasingly crucial for IoT devices requiring robust security measures against sophisticated threats [2].

Y. Wang (ed.), *Proceedings of the 2024 2nd International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2024)*, Advances in Computer Science Research 115,
https://doi.org/10.2991/978-94-6463-540-9_58

This paper explores the role of TEEs in enhancing IoT security. It begins by emphasizing the necessity of hardware support in implementing TEEs, which forms the basis of a solid security framework. It then examines the security principles essential for TEE operation, ensuring data confidentiality and integrity during execution. A detailed system analysis offers empirical evidence from successful TEE deployments in IoT devices, highlighting their efficacy in mitigating security risks. Additionally, this study evaluates how TEEs assist IoT devices in adhering to stringent security regulations and standards, which are tightening as digital and physical realms merge. Despite the advantages, integrating TEEs into IoT devices faces challenges, including scalability, cost implications, and the emergence of new security vulnerabilities and attack vectors specific to TEEs.

The conclusion synthesizes these insights and provides a prospective outlook on the future of TEEs in IoT security, taking into account technological progress and the changing threat landscape. This research contributes to the ongoing discourse on IoT security and offers valuable insights for stakeholders aiming to bolster the security posture of IoT devices through TEE implementation.

2 Layers of TEE Application in IoT

2.1 Hardware Support

ARM's TrustZone. TrustZone, ARM's implementation of a Trusted Execution Environment (TEE), is integrated into Cortex-A range processors. It delineates a secure world (or secure mode) and a non-secure world (or normal mode), effectively allowing the system to function as though it possesses two distinct processors, as depicted in Fig.1 [3]. TrustZone's architecture extends through the system via the AMBA AXI bus, which facilitates secure boot and provides hardware support for secure peripherals. It employs two virtual processors, each supported by hardware-based access control measures [4, 5]. This architecture ensures that sensitive data can be exclusively processed in the secure world, completely isolated from the regular operations of the device. Code executed within the secure world benefits from dedicated resources and remains inaccessible unless the processor operates in secure mode. This design critically prevents any interference from the non-secure world with secure world operations, bolstering the overall security framework of the device.

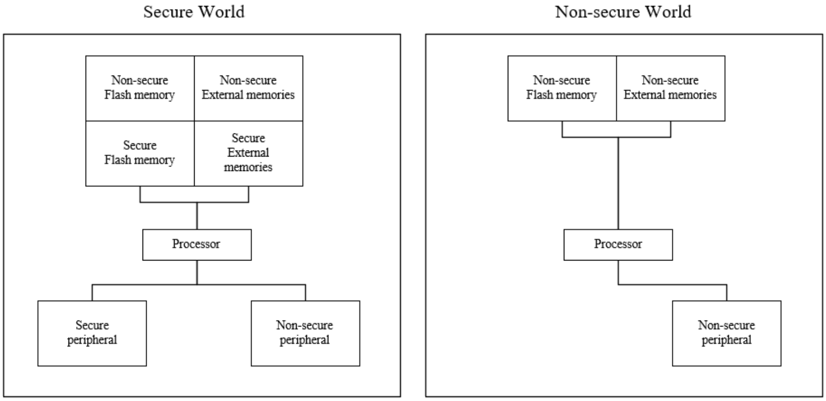

Fig. 1. ARM's TrustZone secure world and non-secure world .

Intel's Software Guard Extensions (SGX). SGX is Intel's set of CPU instruction codes that allows user-level code to allocate private regions of memory, called enclave. SGX allows applications to create enclaves, which are protected areas in the application’s address space [6].

The content within an enclave is encrypted and inaccessible to any code outside the enclave, including code running at higher privilege levels. In Fig.2, Applications need to call trusted functions to access sensitive data which are stored in enclave. Enclaves are designed to be protected from processes running at higher privilege levels, such as the operating system or the hypervisor. The CPU protects the enclave memory, ensuring that sensitive code and data remain confidential and maintain their integrity.

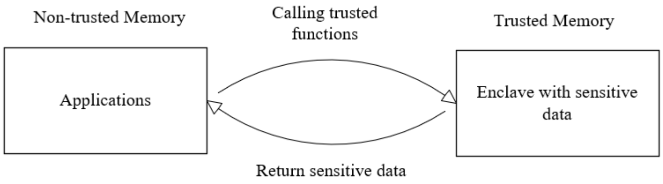

Fig. 2. Intel's Software Guard Extensions (SGX) separate Memory into Trusted memory and Non-trusted memory.

RISC-V's MultiZone Security. MultiZone Security is an open standard framework for RISC-V processors that enables the creation of secure zones on the chip. MultiZone relies on the RISC-V hardware features to partition memory into separate zones. Each zone can be configured with different permissions and can be used to isolate sensitive tasks from each other. With MultiZone, each zone acts as a separate TEE.

The hardware ensures that code and data within a zone cannot be accessed by other zones, unless explicitly allowed by the zone's configuration. This is similar to having multiple separate secure processors on a single chip, allowing for a scalable and flexible approach to security.

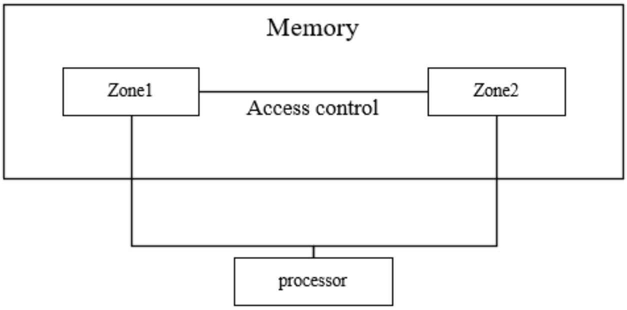

Fig. 3. RISC-V's MultiZone Security different memory zone.

2.2 Security Principles in TEE

While the specifics of TEE security protocols can vary depending on the implementation and the hardware vendor, there are several key principles and components that are generally part of TEE's security approach.

Memory Isolation: Memory regions are specifically configured as secure or non-secure, which is fundamental in maintaining the isolation between the two worlds. Non-secure transactions are prevented from accessing memory regions that have been configured as secure. This ensures that the normal world cannot access the secure memory areas designated for the secure world [7].

Cryptographic Operations: TEEs usually offer a set of cryptographic operations that can be used by applications, such as hashing, encryption/decryption, and digital signature creation/verification. The library contains a pair of public and private keys for RSA encryption and offers system functions to handle RSA, AEC-GCM, DSA, ECC, along with hashing algorithms SHA-1 and SHA-2. These operations are done within the secure environment to prevent leakage of sensitive data [8].

Attestation: TEEs often support remote attestation (RA), which allows a device to prove to a remote party that it is running in a secure state. Trusted Execution Environments (TEEs) like Intel SGX (Software Guard Extensions), remote attestation allows a third party, called a Relying Party (RP), to verify that the software running in a secure enclave on a remote machine is genuine and has not been tampered with. The chain of trust for remote attestation typically begins with the manufacturer (Mfr) who provides the Root of Trust (RoT) through provisioned keys within the hardware. This Root of Trust is considered the foundation upon which the security of the system is built. The Verifier (Vrf) is a service, like Intel's Attestation Service (IAS) or Microsoft's Azure Attestation Service, that validates the integrity of the remote enclave [9].

API Security: TEE SDKs like those for Intel SGX and RISC-V Keystone assume the use of EDL, which generates glue code for secure communication between a normal application and a trusted application. EDL ensures the security of the communication by verifying pointers' regions and buffer sizes. a library implementation of Global

Platform TEE Internal APIs that aims to maintain interoperability and security across different TEE architectures. This library fits with each TEE's EDL and aims to keep communication security intact[10].

3 System Analysis and Application Research

3.1 Successful Deployment of TEEs in IoT Devices

ARM TrustZone in Smartphones and Tablets. TrustZone creates an isolated secure world that is separate from the normal world, where the main operating system (such as Android or iOS) resides. The secure world can run security-sensitive operations, like cryptographic functions or secure boot, without risk of interference from the normal world [11].

Many smartphones and tablets use ARM's TrustZone technology which provides a TEE by partitioning the system-on-chip into secure and non-secure worlds. Brands like Samsung with their Knox security platform use TrustZone to protect critical functions and sensitive data on their devices [12] .

Qualcomm Secure Processing Unit in Mobile Devices. Qualcomm's Snapdragon automotive processors feature called Secure Processing Unit (SPU). When a user engages in activities like saving a document or capturing a photo, the SPU generates a unique cryptographic key. Additionally, apps like WeChat and Facebook can leverage the SPU to create keys when necessary [13].

The SPU operates independently and is segregated from the rest of the system. Although it does not have overarching control to access or manage other systems, it can independently handle information from them. The SPU is set to play an integral part in the management of biometric information. The goal is to confine biometric data within the SPU, process authentication within this secure environment, and ensure that the data does not leave this secure space.

3.2 Review of IoT Security Regulations

The regulatory landscape for IoT security is constantly evolving as the proliferation of IoT devices has necessitated the establishment of robust security frameworks. ENISA (European Union Agency for Cybersecurity) provides comprehensive guidelines and standards aimed at securing IoT ecosystems. ENISA has published a set of baseline security recommendations for IoT which encompasses technical guidelines and measures to protect against threats and to handle security breaches. The use of a Trusted Execution Environment (TEE) in IoT is crucial because it addresses multiple aspects of device security and integrity. It provides a protected area for sensitive operations, enhances identity management, ensures the integrity of software and firmware, and enables secure updates and authentication mechanisms. All of these capabilities are essential for maintaining the security and functionality of IoT devices in the face of evolving threats and the need for robust, scalable security solutions.

4 Challenges

4.1 Scalability Issues

Resource Constraints. Trust Execution Environments (TEEs) provide a secure area within a processor ensuring that code and data loaded inside to it are protected with respect to confidentiality and integrity. Implementing TEEs in IoT devices poses significant challenges due to the resource constraints inherent in many such devices.

IoT devices often have just enough processing capability to perform their required tasks. TEEs require additional computational overhead for security measures such as encryption, secure boot, and runtime integrity checks. These operations may strain the limited CPU resources of low-power IoT devices. TEEs require memory to execute secure operations. Many IoT devices have minimal RAM and storage, which limits the size and complexity of the applications that can be run in a TEE. This can restrict the functionality that can be moved into the secure environment. To address these constraints, solutions may involve optimizing TEE implementations for low-resource environments or adopting lightweight cryptographic and security principles. The security requirements need to be balanced with the device capabilities and the threat model specific to the IoT application.

Network Complexity. As the IoT ecosystem grows, the complexity of managing TEEs across a multitude of devices becomes another critical concern.

The IoT ecosystem comprises a wide range of devices with different hardware capabilities and software stacks. Ensuring compatibility and seamless integration of TEEs across this heterogeneous landscape is challenging. The foundation of the Internet of Things (IoT) is composed of a complex assortment of both longstanding and newly developed technologies. Remarkably, a wave of fresh standards bodies has surfaced, offering promising ideas for shaping IoT frameworks. The expectation is that the diverse elements of the IoT landscape will eventually converge, adopting a shared suite of web-based technologies, which collectively resemble the structure of an hourglass [14]. As more devices are added, establishing and maintaining a root of trust throughout the network becomes more difficult. There must be a scalable way to enroll new devices and securely manage their TEEs over their lifecycle.

4.2 Security Vulnerabilities and Attack Vectors Specific to TEEs

Trusted Execution Environments (TEEs) are designed to provide secure areas within a device's main processor. They aim to protect sensitive operations from the rest of the device's environment, which could be compromised. However, TEEs are not impervious to attacks. Many types of vulnerabilities that can affect TEEs:

Implementation Bugs: A long history of critical implementation bugs in TEE systems, particularly in Trusted Applications (TAs) and the TEE kernel, poses a major security risk. Classic input validation errors such as buffer overflows are common and can be used to compromise both the Android OS and the TEE kernel.

Architectural Deficiencies: TrustZone-assisted TEE systems suffer from architectural weaknesses. For example, memory protection mechanisms like Address Space Layout Randomization (ASLR) or page guards are either poorly implemented or missing. This makes it easier for attackers to exploit vulnerabilities.

Exposed Attack Surface: TEE systems tend to expose a large attack surface, including system calls within the TEE kernel that can be invoked by TAs. This design allows for TAs to potentially manipulate memory regions of the host OS, leading to control over the Android environment if a TA is compromised.

Hardware Vulnerabilities: There are vulnerabilities at the hardware level, with some arising from microarchitectural side-channels such as cache attacks. Additionally, the ability to exploit hardware components like FPGAs to exfiltrate data from TEE-protected memory is concerning.

Defense Mechanisms Lagging: The defense mechanisms in place within studied TEE systems are not on par with the state-of-the-art defenses seen in mainstream operating systems. This suggests that by leveraging up-to-date defensive technologies, the security of commercial TEEs could be significantly improved.

Potential for Mitigation: The paper suggests that adopting modern defense techniques could enable TEEs to better counter prevalent vulnerabilities. Creating a taxonomy for classifying implementation bugs, being aware of hardware components that could be leveraged by attackers, and analyzing current research community proposals for defense are steps toward mitigating these risks [15].

TEEs are designed to enhance security by providing isolated execution environments, they are not impervious to attacks. The documented security vulnerabilities and attack vectors indicate that there is a pressing need for improved design, implementation, and adoption of advanced defensive strategies to bolster the security of TEE systems.

Side-Channel Attacks. Side-channel attacks are techniques that exploit information gained from the physical implementation of a computer system, rather than weaknesses in the implemented algorithms themselves. In the context of TEEs, these attacks can be quite pertinent, as they are often not fully mitigated by traditional software-based security measures [16, 17].

To counter this threat, TEE designers need to consider side-channel resistant hardware, constant-time algorithms, and other mitigations that make it difficult for attackers to extract useful information through side channels.

Software Bugs. Software within TEEs, like all software, can contain bugs or vulnerabilities that, if exploited, could lead to security breaches. These can range from simple coding errors to complex design flaws. Like buffer overflows, A classic software vulnerability, a buffer overflow in TEE code could allow an attacker to corrupt memory and potentially execute arbitrary code within the TEE. Failing to properly validate inputs from untrusted sources could also lead to a range of attacks, including privilege escalation or execution of unauthorized commands within the TEE.

To mitigate the risks associated with software bugs, TEEs must be developed with security best practices in mind. This includes regular code audits, employing static and dynamic analysis tools, fuzzing, and keeping the TEE software up to date with the latest security patches.

5 Conclusion

This paper provides a detailed overview that lays the theoretical groundwork for Trusted Execution Environments (TEEs) and showcases their practical applications across various case studies. The discussion of hardware support mechanisms, such as ARM's TrustZone, Intel's SGX, and RISC-V's MultiZone Security, underlines the versatility and adaptability of TEEs across different technological platforms. The analysis elucidates that TEEs offer a robust solution to the escalating security challenges posed by the widespread adoption of IoT devices. By establishing isolated, secure domains within processors, TEEs effectively protect sensitive operations from the myriad threats prevalent in the cyber landscape. The successful implementation of TEEs across diverse chip manufacturers underscores their effectiveness in securing critical functions and sensitive data. The paper also recognizes the dynamic nature of IoT security standards and regulations, positioning TEEs as a key technology in achieving compliance. Discussions on security principles such as hardware-based attestation, secure boot, and secure communication channels emphasize the comprehensive approach that TEEs take to strengthen the IoT ecosystem.

However, the path toward widespread adoption of TEE technology is fraught with challenges. Issues such as scalability, cost, and the emergence of TEE-specific vulnerabilities pose significant hurdles that both industry stakeholders and academic researchers must overcome. Collaboration between these entities is essential to foster innovation and develop strategies to address these potential risks. Looking forward, the IoT security landscape is expected to continually evolve, driven by technological advancements and an ever-changing threat environment. TEEs will remain a crucial component of this evolving field, acting as a strong defense against cyber threats. Continued research is encouraged to further refine TEE technology, enhance integration ease, and improve the scalability of TEE-based security solutions, ensuring that they remain effective in the face of new challenges.

References

1. J.L. Hernández Ramos and A. Skármeta, eds., Security and privacy in the internet of things: Challenges and solutions, vol. 27, IOS Press, (2020).
2. M. Sabt, M. Achemlal, and A. Bouabdallah, Trusted execution environment: What it is, and what it is not, IEEE Trustcom/BigDataSE/Ispa, 1 (2015) 57-64.
3. S. Pinto, H. Araujo, D. Oliveira, J. Martins, and A. Tavares, Virtualization on TrustZone-enabled microcontrollers? Voilà, IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) (2019) 293-304.

4. Q. Zhu, Q. Chen, Y. Liu, Z. Akhtar, and K. Siddique, Investigating TrustZone: A Comprehensive Analysis, Security and Communication Networks (2023).
5. W. Zheng, Y. Wu, X. Wu, C. Feng, Y. Sui, X. Luo, and Y. Zhou, A survey of Intel SGX and its applications, Frontiers of Computer Science, 15 (2021) 1-15.
6. G.S. Nicholas, Y. Gui, and F. Saqib, A survey and analysis on soc platform security in arm, intel and risc-v architecture, IEEE International Midwest Symposium on Circuits and Systems (MWSCAS) (2020) 718-721.
7. Y. Ma, Q. Zhang, S. Zhao, G. Wang, X. Li, and Z. Shi, Formal verification of memory isolation for the trustzone-based tee, Asia-Pacific Software Engineering Conference (APSEC) (2020) 149-158.
8. D.J. Sebastian, U. Agrawal, A. Tamimi, and A. Hahn, DER-TEE: Secure distributed energy resource operations through trusted execution environments, IEEE Internet of Things Journal, 6(4) (2019) 6476-6486.
9. X. Zhang, K. Qin, S. Qu, T. Wang, C. Zhang, and D. Gu, Teamwork Makes TEE Work: Open and Resilient Remote Attestation on Decentralized Trust, arXiv preprint arXiv:2402.08908 (2024).
10. K. Suzaki, K. Nakajima, T. Oi, and A. Tsukamoto, Library implementation and performance analysis of GlobalPlatform TEE Internal API for Intel SGX and RISC-V Keystone, IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) (2020) 1200-1208.
11. W. Li, Y. Xia, and H. Chen, Research on arm trustzone, GetMobile: Mobile Computing and Communications, 22(3) (2019) 17-22.
12. M. Dorjmyagmar, M. Kim, and H. Kim, Security analysis of samsung knox, International Conference on Advanced Communication Technology (ICACT) (2017) 550-553.
13. P. Čisar and I. Rudas, Overview of Some Security Aspects of Smart Phones, Archibald Reiss Days, 2 (2018) 383-393.
14. X. Zhu, H. Xu, Z. Zhao, and others, An Environmental Intrusion Detection Technology Based on WiFi, Wireless Personal Communications, 119(2) (2021) 1425-1436.
15. D. Cerdeira, N. Santos, P. Fonseca, and S. Pinto, Sok: Understanding the prevailing security vulnerabilities in trustzone-assisted tee systems, IEEE Symposium on Security and Privacy (SP) (2020) 1416-1432.
16. A. Zankl, H. Seuschek, G. Irazoqui, and B. Gulmezoglu, Side-channel attacks in the Internet of Things: threats and challenges, Research Anthology on Artificial Intelligence Applications in Security (2021) 2058-2090.
17. X. Zhang, J. Wang, Y. Cheng, Q. Li, K. Sun, Y. Zheng, and X. Li, Interface-Based Side Channel in TEE-Assisted Networked Services, IEEE/ACM Transactions on Networking, (2023).