A Survey and Analysis on SoC Platform Security in ARM, Intel and RISC-V Architecture

Geraldine Shirley Nicholas, Yutian Gui, Fareena Saqib
Dept. of Electrical and Computer Engineering, University of North Carolina at Charlotte, Charlotte, North Carolina
gnichola@uncc.edu, ygui@uncc.edu, fasqib@uncc.edu

Abstract— **Modern heterogeneous computing including IoT devices and Networks deliver optimized and enhanced performance along with high speed but rely on an increased number of components to achieve the desired results. The design productivity for hardware accelerators with machine learning platforms for various application has significant progress on system-on-chip architectures. Most of these technologies provide the desired performance, however, there is always a tradeoff between security and performance. The major role in developing frameworks for hardware security attacks depends on the IP and system architecture. RISC-V provides a platform for custom implementation of security extensions when compared to other traditional architectures. This paper provides a brief survey of different hardware/software security attacks and summarizes a comparison of security features in RISC-V and other traditional architectures along with security extensions that can be achieved by RISC-V.**

Keywords—RISC-V, ARM TrustZone, Intel SGX, Trusted Execution Environment (TEE)

I. Introduction

Electronic systems in emerging technology are susceptible to several security threats. Any active device connected to a network is vulnerable to Firmware and Hardware attacks. Firmware attacks involve different components and techniques that are not present on traditional software-based attacks. Some of the firmware-level threats today are network attacks, Denial of Service, Trojan insertion etc. These attacks play a major role in the system components, wherein an adversary takes control over the whole system by exploiting vulnerabilities in the system. In terms of hardware attacks, the SoC designs must be protected from any unauthorized access. Some of the threat sources are IC supply chain, side-channel attacks, reverse engineering, cloning etc.

For an SoC platform with different levels of abstraction, the security vulnerabilities of that system must be determined to develop a secured platform. Key factors of SoC Platform Security are 1) Root of Trust, 2) Secure Boot, and 3) Trusted Execution Environment (TEE) to run the system. In the connected world, root of trust is a set of modules with different security features that are trusted by the system, to monitor the functionality and provide secure authentication with protection to each component in the system along with securing the data and resources by executing secure algorithms and functions. A programmable hardware-based root of trust design offers the most efficient secured system. Secure Boot on the other hand is a mechanism that can be used for the integrity of the firmware, building trust between the system and the firmware before the execution of the application by providing authentication and validation. Device Enrollment, Attestation, and key exchanges compose a chain of trust in Secure Boot.

Trusted Execution Environment (TEE) is an isolated execution environment providing security features where the software and the data is protected through isolation [1]. The ARM TrustZone based TEE technology provides a methodology to isolate security-critical components in a system [2]. Intel Software Guard Extensions (SGX) enclave is supported in modern processors to protect the privilege levels by certain authorized functions [3]. Some other architectures for security-critical applications are AMD Platform Secure Processor, AMD Memory Encryption Technologies, and Intel Management Engine (ME), Open Portable TEE, and different Platform Security Architectures (PSA). Though these traditional architectures provide a secure environment to a certain level, they fail to ensure isolation due to the separation of different stacks of libraries.

RISC-V, an open-source architecture provides the platform to implement different levels of security in a system unlike the SoC with 3PIP proprietary architecture and integration of securing the system. RISC-V Multizone Security by Hex Five provides a hardware-enforced software defined separation with multiple TEEs [4]. With the benefits of RISC-V being open-source, especially in the security context, different modules can be implemented to secure the system from any kind of attacks [5].

This paper focusses on the major threats in an SoC Design and Architectural Platform and points out the different features and vulnerabilities present in the ARM TrustZone and Intel SGX platforms. Finally, it provides a solution for security enhancement using RISC-V architecture.

II. Threat Model

Most of the implementations aim at securing only the firmware and data in critical applications. However, an adversary can find a link through other backdoor channels to penetrate the system. This can be done through a micro-architectural event of the hardware, privileged software, or physical hardware probing. Considering some of the reference models for security and challenges in the SoC Design and Architectural Platform, four major types of scenarios are reviewed.

i. *Insertion of Malware/Unwanted Application gaining access:* In a connected network, the adversary can insert a hidden functionality that can track the data and

978-1-7281-8058-8/20/$31.00 ©2020 IEEE

critical information or one that gets triggered to deliver disruptive outcomes. This can be achieved using system software privilege levels. In addition, malicious modification to the circuit can be done by bypassing the security fence of the system.

ii. *Side-channel attacks:* It is one of the most used security exploitation methods, to obtain information on crypto engines running in the system through communication channels. In this method, the adversary can reverse engineer the functions to gain access to the system and network by monitoring the power consumption or the electromagnetic fields associated with the hardware.

iii. *Supply Chain Attacks:* Globalized IC supply chain can result in malicious design modification or IP theft through reverse engineering [6]. In the software supply chain attacks, an unsecured network or infrastructure is targeted, where malicious code can compromise the build tools.

iv. *Network Attacks:* Denial of Service in a distributed network result in bridging through the system and gaining access over it. Figure 1 illustrates the different security challenges in a connected network.

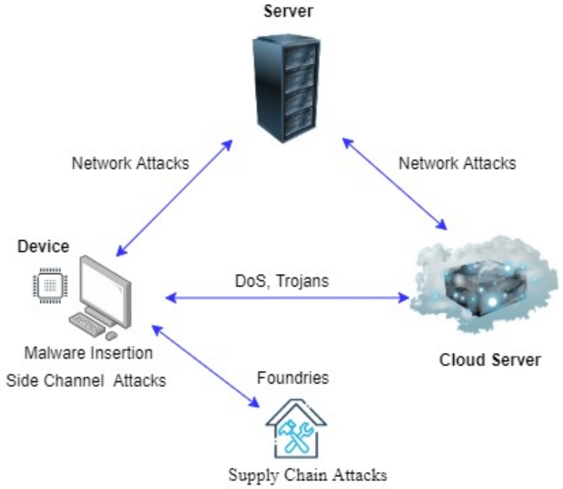

Fig. 1. Security Challenges in a Connected Network

Most of the traditional architectures have limited capability to implement security features necessary to secure devices from the above-mentioned attacks. The main reason for this is the lack of flexibility as the developers of proprietary architectures do not offer security enhancements due to the associated performance tradeoffs. RISC-V, on the other hand, provides this flexibility to customize the design of a system with reconfigurability and system security.

III. Security Models, Features and Vulnerabilities

A. ARM TrustZone

TEEs secure the data from unauthorized access by isolating the secured and non-secured applications. ARM TrustZone is an implementation of the TEE standard. The intellectual property cores are partitioned into secure and non-secure world. Figure 2 shows the ARM TrustZone implementation consisting of the core along with different units. The TrustZone Memory Adapter (TZMA) and TrustZone Address Space Controller (TZASC) are used to provide partitions between the memory and peripheral units for both the worlds. A detailed description of the ARM TrustZone and building a secure system using TrustZone Technology is provided in [7]. Hardware Isolation is achieved by the ARM TrustZone, where separation in the model secures the critical data but in modern design, the system tends to enlarge with large stacks of libraries, and optimized functions. Hence, it fails in the design point of modern security systems.

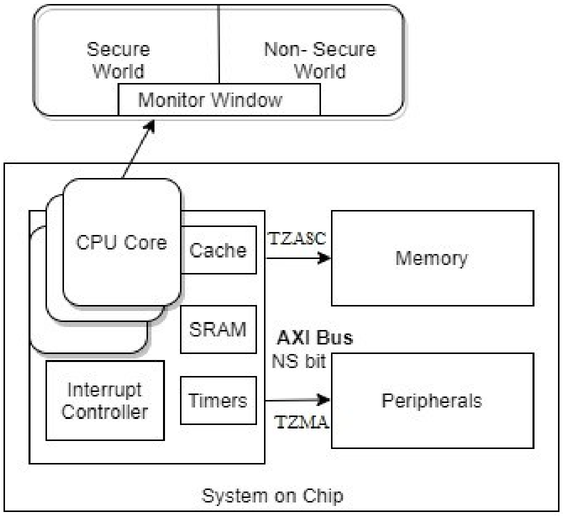

Fig. 2. ARM TrustZone Implementation

This model is vulnerable to cache-based side-channel attacks. Extraction of keys from any crypto engines running in the secure world is possible by compromising the non-secure world OS or by tracking the power or EMF signals during key exchanges between the two worlds [8]. Malicious modification of the secure IP, denial of service by prohibiting access to any secure IP, resource denial, port attack, etc. are some of the hardware attacks performed on the ARM TrustZone in Zynq-7010 SoC platform [9]. Configuring the NS bit along the AXI bus in two different worlds is simple and effective but it becomes difficult to manage the structure in a multi-core environment. Also, additional security enforcements are to be incorporated for optional memory controller present outside the Cortex-M TrustZone Architecture.

B. Intel SGX

Intel SGX offers enclave memory access semantics and protection of ad dress mappings of the application [10]. The enclaves are a region of memory, protected from any access or modifications. Encrypted and decrypted on the fly, these enclaves are hardware isolated trusted environments. Figure 3 shows the basic operation done in the SGX Model. An untrusted application invokes a trusted function inside the enclave which cannot be accessed by any application. Integrity violation from software attacks, confidentiality of the code with isolation can be achieved by this model. The Processor Reserved Memory

(PRM) holds the enclave page cache and is protected from any non-enclave memory accesses. The downside of this model is, the enclave gains full access to the entire address space of the untrusted application, which makes it vulnerable to enclave malwares.

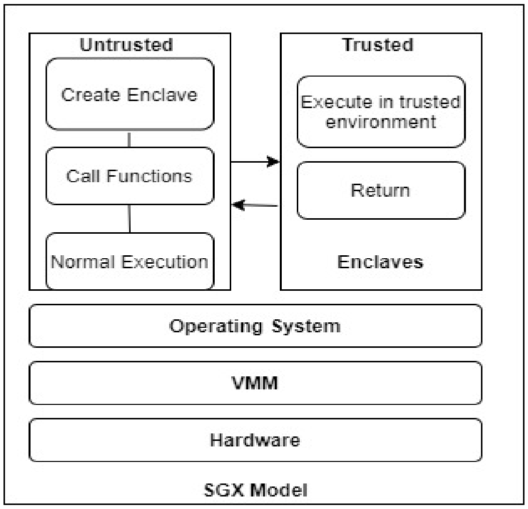

Fig. 3. SGX Model

Since the enclave uses the same cache architecture, it is also vulnerable to cache-based side-channel attacks. Fine-grained software-based side-channel attacks were targeted on co-located SGX enclaves to extract the RSA private keys [11]. Though most of the modern Intel processors feature Hyper-threading, SGX does not prevent it, causing malicious software to execute system threads.

By gaining the enclaves memory access patterns, any critical data can be extracted. Intel-specific architectural and microarchitectural details with SGX's security features are summarized in [12]. It is also impossible to reason about SGX's security features as they are not publicly available. Therefore, to develop a trust model by customizing the TEEs involves the provider's architectural modification which requires proprietary rights. Bridging the support for more than one hardware-enforced isolated domain is not possible with the traditional architectures.

To restrict such vulnerabilities, a hardware/software-based co-design resilience model must be implemented to provide maximal protection to the system from any kind of attacks.

C. *RISC-V Implementations*

An open-source framework for building a customizable multi-domain Trusted Execution Environment is achieved by RISC-V for various applications. RISC-V has different privilege modes to operate and can be configured easily to manage the TEE. The physical memory protection provided by this architecture is used for authenticating the execution of trusted nodes. Integration of hardware cryptographic accelerators, key management, and security extensions are made simple using available open-source frameworks. Multi-threaded enclaves with memory-mapped resource protection are achieved by different RISC-V security implementations module.

One among them is *Sanctum*: *Minimal Hardware Extensions for strong Software Isolation,* implemented with Rocket RISC-V core, is a strong provable isolation module that protects against different software attacks related to memory access patterns [13]. The microarchitectural state, caches, data structures managed by the OS along with memory and interrupts are protected. Privileged enclave and signing enclave can invoke a secure inter-enclave service for attestation. Figure 4 shows the basic software stack in the Sanctum Model with different modes of operation.

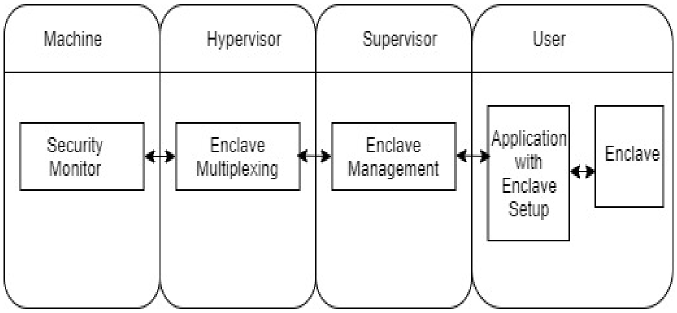

Fig. 4. Software stack in the Sanctum Model

RISC-V MultiZone Security by HEX-Five provides a trusted environment to shield critical functionality from untrusted components with the freedom to multi-source open source software's and third-party libraries [14]. The main feature in this model consists of equal memory-mapped resources per zone with multi-tasking functionalities. Similar to the previous model, the MultiZone takes the advantages of RISC-V privilege levels and offers a platform to compile and link each zone separately with its own protected features. Keystone based TEEs on unmodified RISC-V hardware is another framework for building customizable TEEs [15]. This provides a new programmable layer and isolation primitives below the untrusted code with decoupled resource management. Each layer is independent with secure abstraction awareness to make it compatible with all the existing privilege levels.

In addition, a hardware approach using hardware accelerators compatible with RISC-V architecture for memory protection is designed for the Keystone Framework [16]. This design proves that RISC-V architecture is structured in a way that is compatible and user friendly with numerous open-source frameworks. Designing a security extension for a model with RISC-V architecture helps in implementing multi-layered security protection against different attacks. The limitations in Keystone was enhanced by this model by adding security features to the existing model which is impossible in traditional architectures.

Existing RISC-V TEEs which provide different levels of protection but are vulnerable to side-channel attacks can be improvised to a resilient model with the open-source features available and by implementing different countermeasures in the existing framework. Table 1 provides RISC-V compatible

TABLE I

Threat Models	RISC-V compatible countermeasure models
Cache-Timing Attacks	Transparent Hardware-Protection Layers with memory access leakage protection
Side-Channel Attacks	Core hardened resilient models with hardware Accelerators and virtual TEEs
Denial of Service Attacks	Information flow tracking models tracking the flow of the data to protect memory corruption and mitigate DoS attacks by attestation models [18]
Malware Insertion	Multilayer camouflaged secure boot for SoCs along with data tracking models [17]
Supply Chain Attacks	Logic Obfuscation with SAT attack resilient model for the SoC platform

countermeasures by taking into consideration the possible threats in the Trusted Execution Environment, and the SOC design and Architecture Platform.

Finally, the open-source nature of RISC-V facilitates the open community including researchers to participate in finding and fixing security vulnerabilities, thereby bolstering the overall security of the architecture. There have been proven track records of this approach, as manifested by the success of other open-source initiatives.

IV. Conclusion

This paper summaries the major challenges in an SoC Design and Architectural Platform along with the security features, implementation, and shortcomings in the hardware-isolated trusted environments such as ARM Trust Zone and Intel SGX. It also provides countermeasures for the vulnerabilities present in this domain by utilizing the features in RISC-V architecture. RISC-V provides security enhancement and unique models that are resilient to side-channel attacks along with MultiZone models to a multi-world running with multiple cores and engines.

V. References

[1] M. Sabt, M. Achemlal and A. Bouabdallah, "Trusted Execution Environment: What It is, and What It is Not," *2015 IEEE Trustcom/BigDataSE/ISPA*, Helsinki, 2015, pp. 57-64, doi: *10.1109/Trustcom.2015.357.*

[2] TrustZone Technology for the ARMv8-M Architecture, ARM, Cambridge, U.K., 2017. [Online]. Available: https://developer.arm.com/

docs/100690/0200.

[3] M. A. Mukhtar, M. K. Bhatti and G. Gogniat, "Architectures for Security: A comparative analysis of hardware security features in Intel SGX and ARM TrustZone," *2019 2nd International Conference on Communication, Computing and Digital systems (C-CODE)*, Islamabad, Pakistan, 2019, pp. 299-304, doi: 10.1109/C-CODE.2019.8680982.

[4] https://content.riscv.org/wp-content/uploads/2019/03/15.05-RISC-V-Security-Multizone-v-TrustZone-3-12-19.pdf

[5] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanoviä, "The RISC-V instruction set manual, volume I: User-level ISA, version 2.0," *EECS Dept., Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2014-54, May 2014*.

[6] M. Yasin and O. Sinanoglu, "Evolution of logic locking," *2017 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Abu Dhabi, 2017, pp. 1-6, doi: 10.1109/VLSI-SoC.2017.8203496.

[7] http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf

[8] Ning Zhang, Kun Sun, Deborah Shands, Wenjing Lou, and Yiwei Thomas Hou. Truspy: Cache side-channel information leakage from the secure world on arm devices. IACR Cryptology ePrint Archive, 2016:980, 2016.

[9] E. M. Benhani, C. Marchand, A. Aubert and L. Bossuet, "On the security evaluation of the ARM TrustZone extension in a heterogeneous SoC," *2017 30th IEEE International System-on-Chip Conference (SOCC)*, Munich, 2017, pp. 108-113, doi: 10.1109/SOCC.2017.8226018.

[10] Intel Corporation, "Software Guard Extensions Programming Reference", 329298-002US October 2014.

[11] Schwarz, M., Weiser, S., Gruss, D., Maurice, C., & Mangard, S. (2017). Malware Guard Extension: Using SGX to Conceal Cache Attacks. Lecture Notes in Computer Science, 3–24. doi:10.1007/978-3-319-60876-1_1

[12] Costan, V., & Devadas, S. (2016). Intel SGX Explained. *IACR Cryptol. ePrint Arch., 2016*, 86.

[13] Costan, Victor et al. "Sanctum: Minimal Hardware Extensions for Strong Software Isolation." *USENIX Security Symposium* (2016).

[14] https://hex-five.com/wp-content/uploads/2020/01/multizone-datasheet-20200109.pdf

[15] Lee, Dayeol et al. "Keystone: A Framework for Architecting TEEs." *ArXiv* abs/1907.10119 (2019): n. pag.

[16] T. Hoang *et al.*, "Quick Boot of Trusted Execution Environment With Hardware Accelerators," in *IEEE Access*, vol. 8, pp. 74015-74023, 2020, doi: 10.1109/ACCESS.2020.2987617.

[17] A. S. Siddiqui *et al.*, "Multilayer Camouflaged Secure Boot for SoCs," *2019 20th International Workshop on Microprocessor/SoC Test, Security and Verification (MTV)*, Austin, TX, USA, 2019, pp. 56-61, doi: 10.1109/MTV48867.2019.00019.

[18] A. S. Siddiqui, G. Shirley, S. Bendre, G. Bhagwat, J. Plusquellic and F. Saqib, "Secure Design Flow of FPGA Based RISC-V Implementation," *2019 IEEE 4th International Verification and Security Workshop (IVSW)*, Rhodes Island, Greece, 2019, pp. 37-42, doi: 10.1109/IVSW.2019.8854418.

[19] I. Lebedev, K. Hogan and S. Devadas, "Invited Paper: Secure Boot and Remote Attestation in the Sanctum Processor," 2018 IEEE 31st Computer Security Foundations Symposium (CSF), Oxford, 2018, pp. 46-60, doi: 10.1109/CSF.2018.00011.

[20] D. Hwang, M. Yang, S. Jeon, Y. Lee, D. Kwon and Y. Paek, "RiskiM: Toward Complete Kernel Protection with Hardware Support," 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), Florence, Italy, 2019, pp. 740-745, doi: 10.23919/DATE.2019.8715277.