IST 597 Project Report

Yufei Jiang, Xiao Liu

Abstract

This report summarizes the main difference between two trust computing techniques, Intel SGX and ARM TrustZone. With respect the mentioned difference, we also present the extensions on ARM TrustZone that achieves same functionality as Intel SGX.

1 The Difference between Intel SGX and ARM TrustZone

1.1 Overview

Trusted Computing (TC) is a concept developed by the Trusted Computing Group. It has been discussed for years. Regarding the meaning of "trusted", there are a lot of controversy. Some people also challenge the idea of Trusted Computing fundamentally since it may be harmful to keep anonymity over Internet and avoid vendor-lock-in. Despite the buzz, some players in the technical industry, including Dell, HP, Microsoft, Intel, and ARM have made solid progress on realizing this concept.

In this report, we will investigate two concrete Trusted Computing solutions. They are Intel SGX and ARM TrustZone. According to the definitions given by Trusted Computing Group. A complete Trusted system must have the following 6 elements: endorsement key, secure input and output, memory curtaining, sealed storage, remote attestation, and trusted third party. More specifically, both Intel SGX and ARM TrustZone are two measurements to fulfill some parts in the whole Trusted system.

In the following subsections, we will first briefly introduce Intel SGX and ARM TrustZone respectively. Then we will compare them two from certain aspects.

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

1.2 Intel SGX

The full name of Intel SGX is Intel Software Guard Extensions. Intel SGX is a technology to help developers better protect the confidentiality and integrity

of selected code and data of their applications from those rogue software that is running in higher level including the operating system. It was implemented in 6th generation Intel Core microprocessors in 2015 for the first time. Its basic mechanism is to create some protected areas in memory which are called "enclaves". Developers can explicitly use Intel SGX SDK during the development to put the data and code of interests into enclaves.

1.3 ARM TrustZone

According to ARM, ARM TrustZone technology "is a System on Chip (SoC) and CPU system-wide approach to security". Its key idea is to provide two virtual processors to create two hardware-separated worlds, the secure world and the insecure world. Any information flows from the secure world to the insecure world must go through the secure monitor. Besides this difference, the two worlds have the same capabilities so each side can operate independently.

1.4 Comparison

1.4.1 Architecture

Both Intel SGX and ARM TrustZone are security extensions to their existing CPU architectures. Conceptually, Intel SGX is more lightweight than ARM TrustZone. Intel SGX essentially is an extension to Intel instruction sets. Only with those special instructions, data can be read from written to the enclaves. ARM TrustZone, from our point of view, is a hardware-assisted virtualization. It provides two completely separate virtual running environment. Each side has the complete computation capacity. To this extent, Intel SGX and ARM TrustZone have different security models.

1.4.2 Security

Intel SGX and ARM TrustZone follows the different security models. The security provided by Intel SGX are from following two aspects. First, only several specific instructions, which requires explicitly to be called by developers, can access enclaves. Second, the cryptographic key is only available to the Intel processors. In short, Intel SGX tries to help developers to achieve information hiding. ARM TrustZone, from our perspective, provides isolation, which does not grant security automatically. The key insights of ARM TrustZone are to reduce attack surface and isolation. It assumes that developers will put almost all code in the insecure world. Those code, such as the operating system, is complex, hard to be audited and analysed. They are doomed to have exploitable vulnerabilities. On the other hand, even the secure world has the capability of running any code, we usually only put some small pieces of code in the secure world. Those code has simpler logic and performs some critical function, like signing or making a transaction. So we can review, audit, perform model checking, analyze those code to make sure it is very likely to be bug free. A malicious user may temper the operating system in the insecure world easily, however,

he still cannot perform those critical functions in the secure world arbitrarily. However, if the code in the secure world has vulnerabilities, TrustZone cannot prevent a malicious user from exploiting it.

1.4.3 Development

Development in Intel SGX is easier. Intel SGX provides enclave binding API in C and C++. To utilize Intel SGX in an application, a developer can use the API provided by Intel SGX SDK. A developer can explicitly put the data of interests into enclaves. The two most important APIs are for creating and destroying enclaves in the memory. Conceptually, they are just like special "malloc" and "free". ARM TrustZone can be applied to different layers. One way is just to use ARM TrustZone APIs to invoke SMC calls in application levels. The other way is so called "co-operative OSes". The operating system resides in both the secure world and the insecure world. The OS on the two sides must handle interrupts and scheduling properly. Besides the SMC calls, semaphores, lock-free algorithms, and shared memory may also be used.

1.4.4 Invocation

During runtime, the SGX APIs called by the application will invoke the driver of SGX to perform encryption, decryption, read, and write operations. In ARM TrustZone, when the insecure world requests a service in the secure world, it makes a SMC call to transfer the control to the secure world.

1.4.5 Encryption

Intel SGX uses symmetric encryption to encrypt the data in enclaves. The key will be refreshed in every boot. ARM TrustZone, on the other hand, does not encrypt the data in the secure world, since it follows a different security model.

1.4.6 Adoption

WolfSSL is a "small, portable, embedded, SSL/TLS library targeted for use by embedded system developers" [1]. It supports Intel SGX in its product. Bromium has prototyped an extension to their product which takes advantage of SGX to protect the online credentials [2]. Samsung Knox [3] might be a killer application that utilizes ARM TrustZone technology. It is widely deployed in Samsung mobile phones.

1.5 Takeaway

Intel SGX is more lightweight security extension than ARM TrustZone. They are different regarding their architectures, security models, ways to develop, and many other aspects. Due to ARM TrustZone is released earlier than Intel SGX, now ARM TrustZone is being used more widely.

2 Extension of ARM TrustZone

With respect to the discussed differences between Intel SGX and ARM TrustZone, in this section, we presents the extension techniques on TrustZone that allows developers to use it, as in SGX.

To summarize, the main differences of the two techniques are:

- Architecture: With TrustZone we often think of a CPU which is in two halves i.e. the insecure world and the secure world. In Intel SGX model we have one CPU which can have many secure enclaves (islands).
- Transparency: TrustZone technology is programmed into the hardware, enabling the protection of memory and peripherals. But SGX is a set of instructions and mechanisms for memory accesses added to Intel architecture processors.
- Process: Due to the difference in architecture design, the service in Intel SGX have additional verified launching, licensing, and attestations.

With respect to these main differences, we are going to explicitly explain the enabling extensions to ARM TrustZone.

2.1 Isolation

ARM TrustZone is a hardware isolation technology. Each processor mode has its own memory access region and privilege. The code running in the normal world cannot access the memory in the secure world, while the program executed in the secure world can access the memory in normal world. The secure and normal worlds can be identified by reading the NS bit in the Secure Configuration Register (SCR), which can only be modified in the secure world. Due to the distinction in archietecture, the functionalities are different for these two technologies. TrustZone does not distinguish between different secure application processes in hardware, in another word, the privilege are not explicitly specified.

Each processor mode has its own memory access region and privilege. The code running in the normal world cannot access the memory in the secure world, while the program executed in the secure world can access the memory in normal To enable confidentiality between different applications, we should extend the current system with isolations, either hardware-based or virtually. In addition, for specifying priviliges between applications, it requires a controller which is similar to management engines in Intel SGX. The Intel Management Engine (ME) is a micro-computer embedded inside of all recent Intel processors, and it exists on Intel products including servers, workstations, desktops. It will schedule the application process and control the invocation privileges as well. In order to assess security among different application processes, it should also provide attestation mechanisms. Local attestation enables an process to verify another running on the same physical CPU. Remote attestation can be used by

a remote party to check if the attested process is indeed running on a genuine CPU. It allows initial provisioning of keys and secrets.

2.2 Abstraction

The development environment in Intel SGX is more mature than ARM TrustZone, which provide abundant APIs for developers to use. For developers, Intel SGX is more transparent. It is a set of instructions which developers can rely on. Developers can take advantage of the enclaves without understanding the detailed implementations on hardware enforcement. However, in the design of ARM TrustZone, only hardware environment is divided which requires additional effort of the developer.

To achieve a same development transparency for ARM TrustZone, we should extend current design with more abstracted implementation choices. That is, to encapsulate possible process initializations, and privilege specifications. By abstract the hardware functionalities into software-level, developers can easily access a secure development by merely changing the APIs but the service logic remains the same and security maintainance is done by the trusted computing platform.

2.3 Additional Processes

Because of the different archietectures of Intel SGX and ARM TrustZone, the processes to set up a secure application and the principle to enforce the overall integrity is different. Therefore, to enable a same or similar functionality of ARM TrustZone, it requires additional processes in the development workflow. In this part, we will explain these processes one after another.

2.3.1 Verified Process Launch

For Intel SGX, when loading an enclave into memory, the CPU measures its content in a chained cryptographic hash log, stored in a register called MRENCLAVE. This is equivalent to a Platform Configuration Register (PCR) in a TPM. Before running the enclave, the CPU verifies MRENCLAVE against a vendor-signed version and aborts when finding any mismatches. The verified enclave launch in Intel SGX enforces the isolation of application processes. Thus, when isolating different processes in ARM TrustZone, it is still required that we extend the verified launching into regular setup procedures.

2.3.2 Licensing

SGX has a feature, called launch enclave. During enclave initialization, the CPU verifies a so-called EINITTOKEN, which contains several enclave attributes to enforce, including the debug mode flag. The EINITTOKEN has to be signed by a special launch key, which is owned by so-called launch enclaves, is sued by Intel. Hence, by issuing proper launch enclaves, Intel has full control over

which enclaves are to be executed in debug or production mode. The SGX evaluation SDK is shipped with a launch enclave issuing EINITTOKENs for debug enclaves only. In order to run an enclave in production mode, one needs to obtain a proper license from Intel. This process ensures the confidentiality within any enclaves. Therefore, to achieve a similar functionality for ARM TrustZone with isolation, we should add additional licensing process.

2.3.3 Attestation

In Intel SGX, because the running enclave is more than one, the enclave contacts the service provider to have its sensitive data provisioned to the enclave. The platform produces a secure assertion that identifies the hardware environment and the enclave. It has dedicated sealing and attestation components that uphold the security model assertions below to prove to the service provider that the secrets will be protected according to the intended security level. These components enforce a set of designed privileges among enclaves which is not maintained in ARM TrustZone. Therefore, to ensure the security of application processes in ARM TrustZone, we should also maintain a corresponding attestation mechanism including local attestation and remote attestation.

References

[1] "Wolfssl," https://www.wolfssl.com/wolfSSL/Home.html, accessed: 2017-05-25.

[2] "Using intel sgx to protect on-line credentials," https://blogs.bromium.com/using-intel-sgx-to-protect-on-line-credentials/, accessed: 2015-05-25.

[3] "Samsung knox," https://www.samsungknox.com/en, accessed: 2017-05-25.