Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Hardware-based solutions for trusted cloud computing

*Oualid Demigha**, *Ramzi Larguet*

Ecole Militaire Polytechnique. PO Box 17, Bordj El-Bahri, Algiers. 16111 Algeria

ARTICLE INFO

Article history:
Received 30 March 2020
Revised 2 November 2020
Accepted 12 November 2020
Available online 16 January 2021

Keywords:
Trusted cloud computing
Hardware-assisted security
Trusted execution environment
Intel TXT
AMD SEV
ARM TrustZone
Intel SGX

ABSTRACT

The increasing number of threats targeting cloud computing and the exploitation of specifically privileged software vulnerabilities have pushed the security managers of cloud service providers to deploy hardware-based solutions. These solutions can offer better hardware-assisted security features for a broad range of computing platforms including both CISC and RISC architecture families in datacenters. Their goal is to reduce the attack surface by rooting the trust into the hardware instead of some high-privileged pieces of system software such as the operating system or the hypervisor which have been demonstrated that they include severe security vulnerabilities, thus limiting the adoption of the cloud computing model for some security-skeptical users. In this paper, we give cloud users and customers, application developers and security managers a comprehensive overview of four major industrial-scale commercial hardware-based solutions brought by major vendors in the cloud market. We present, analyze and compare Intel TXT, ARM TrustZone, AMD SEV, and Intel SGX technologies with respect to more than twenty criteria fitting within three categories: security, functional and deployment. We discuss each of these technologies and show the cases where they particularly excel. Our comparison can help IT managers to take the right decision about which better industrial technology to adopt for their particular security requirements and future cloud migrations.

Contents

* Corresponding author.
E-mail address: o_demigha@esi.dz (O. Demigha).
https://doi.org/10.1016/j.cose.2020.102117

1. Introduction

Cloud computing has been widely adopted during the last decade thanks to its many advantages such as deployment cost reduction, physical resource sharing between VMs (Virtual Machines) using virtualization, and flexible and elastic on-demand services. However, besides the traditional threats present in every computing environment, cloud computing has opened the door for many additional threats as customers share the same physical resources due to the physical co-location paradigm (Singh, 2014; Zoltn dm Mann, 2017).

Given the increasing number of attacks targeting cloud environments (Distributed Denial of Service (DDoS) or DoS attacks, Man-in-the-Cloud attacks (Jabir et al., 2016), rootkit attacks (Modi et al., 2012), replay code attacks (Hetzelt and Buhren, 2017), code injection attacks, etc.), customers need more guarantees to adopt cloud services. Therefore, many solutions have been proposed to face these threats which can be divided into two main families: software-based solutions and hardware-based solutions.

On one hand, software-based solutions are the first security solutions to show up in the market. They are relatively easy to implement, inexpensive and offer the ability to review and upgrade implementations. Indeed, they improve cloud system security but may be insufficient to protect VMs (Szefer and Lee, 2012) because they require a trusted hypervisor with millions of lines of code (LOC) to be present in the TCB (Trust Computing Base). That induces a large attack surface because the probability that vulnerabilities exist in a huge code increases with the number of LOC, and the task of verifying complex software is still very hard (Kaplan et al., 2016). These vulnerabilities can be exploited by hackers to make possible attacks such as code injection, code reuse, fork and rollback, colluding and rootkits. Moreover, the privileged nature of the software present in these solutions makes it a preferred attackers' target as recent research reports indicate that the OS (Operating System), the hypervisor (Asvija et al., 2019), the BIOS and the SMM (System Management Mode) layers can be easily targeted to exploit their known vulnerabilities and compromise the cloud computing infrastructure.

On the other hand, hardware-based solutions use dedicated Integrated Circuits (IC), or a completely separate processor designed specifically to provide security operations with a specialized hardware architecture. The sensitive information such as cryptographic keys, biometric information, passwords, and critical parameters of the system configurations are protected by the hardware thanks to silicon-implemented cryp-

tographic primitives, random number generators, tamper detection algorithms, etc. that are relatively less complex compared to software implementations, which make the verification task relatively easy.

Therefore, with a hardware-based solution, it maybe more difficult and expensive to alter such a security setup. Beside their high costs, hardware-based solutions provide a big security proof against unauthorized access to the cloud infrastructure, embedded devices, and peripherals with optimized system performances.

Several hardware-based solutions have been proposed in recent years by both academia and industry. At the industrial scale, we can find the following: TPM (Trust Processor Module), Intel TXT (Trust eXecution Technology), ARM TrustZone, AMD SEV (Secure Execution Virtualization), and Intel SGX (Software Guard eXtention) (Lie and Maniatis, 2017; Mofrad et al., 2018; Pinto and Santos, 2019; Zhang and Zhang, 2016). These security solutions can be leveraged by CSPs (Cloud Service Provider) to offer a TEE (Trusted Execution Environment) for end-users to execute their applications in a trusted cloud environment.

As cloud tenants are concerned about protecting their sensitive data and ensuring the security of their VMs, and since there are many industrial-scale hardware-based security solutions available for customers in the market, we propose in this paper a deep analysis and a comprehensive comparison study between them to help clients, application designers, and security professionals to decide about the technology to adopt for their future cloud migration.

We compare four industrial-scale hardware-based security solutions, namely: Intel TXT, ARM TrustZone, AMD SEV, and Intel SGX. We do not include TPM because it is most commonly implemented as a dedicated, special-purpose chip that can be coupled with Intel TXT for instance. To the best of our knowledge, this paper is the first to give a comprehensive overview and an up-to-date exhaustive comparison of the major industrial hardware-based trust solutions available for end-users.

The rest of the paper is organized as follows: in Section 2, we describe the cloud environment setup to which the studied solutions are projected to be deployed, alongside with the common attacker model with which they deal. In Section 3, we give a brief description of each of the above-mentioned solutions. In Section 4, we go in-depth of our comparison according to the criteria we define. In Section 5, we discuss the different solutions with a special emphasis on particular cases where each one of them excels. In Section 6, we present related work that follows a similar approach as ours but differ in many other aspects. In Section 7, we conclude the paper and give some future work.

2. Cloud setup and assumptions

The multitenant nature of cloud computing makes it challenging to ensure data security and access control when users outsource their data for remote computation on cloud servers with a potentially compromised cloud software stack and malicious cloud operators (Gu et al., 2020).

Although the deployment of internal private cloud is safer when compared to hybrid and public ones,[1] the security problem remains entire because of the critical vulnerabilities discovered every time in the server hardware and the system software, that is opened the door for multiple severe attacks such as Meltdown (Lipp et al., 2018), Spectre (Kocher et al., 2019), and RowHammer (Mutlu and Kim, 2019).

CSPs either for IaaS (Infrastrcure-as-a-Service), Paas (Platform-as-a-Service) or SaaS (Software-as-a-Service), or any other kind of services share all common concerns about protecting data of cloud tenants. Instead of using software-based solutions with large TCB and large attack surface, they can leverage hardware-based TEE to give more security guarantees than software-based solutions using system software. Hardware-assisted TEE are secure isolation technologies that have been engineered to serve as efficient defense mechanisms to provide a security boundary at the system level by isolating software execution at runtime so that sensitive data is processed in a trusted environment (Gu et al., 2020) even in the infrastructure of public cloud providers with unreliable/vulnerable system software.

As trust is rooted in the hardware in these solutions, we consider a large cloud model that includes almost all cloud types with every service that manipulates end-users' secret data/code. The latter may not trust the CSPs at all levels of the software stack except the software that they use/craft themselves as will be described shortly in the attacker model.

2.1. *Attacker model*

We consider an attacker with the capabilities of the complete control of the software. This means that he/she can tamper with any privileged software, and manipulate it, insert a malware into the system or even deploy malicious software components. This includes the OS, BIOS, SMM, and VMM, which are considered untrusted.

The attacker is considered be able to read the memory and capable of performing physical attacks but not on the CPU.[2] That means he/she does not have physical access to the system's CPU but can access any other hardware component of the platform including the memory controller or the buses interconnecting platform components.

The attacker may be an insider or a remote adversary but he/she is assumed to be unable to break cryptographic primitives. In this model, we do not consider denial of service attacks, physical-based attacks,[3] side channel and timing attacks, and network-based attacks (Atamli-Reineh and Martin, 2017; Kocher et al., 2019; Lipp et al., 2018; Maene et al., 2018; Mutlu and Kim, 2019; Van Bulck et al., 2018).

[1] that indeed provide more professional equipment, employ high skilled security managers and offer balanced business plans for a variety of IT industry segments.

[2] This means that he/she cannot unpackage the CPU and inspect/analyze it using electronic means.

[3] Those that exploit vulnerabilities in the hardware to alter its normal functioning and operate on it outside its specifications provided by the vendors, such as rowhammer attack.

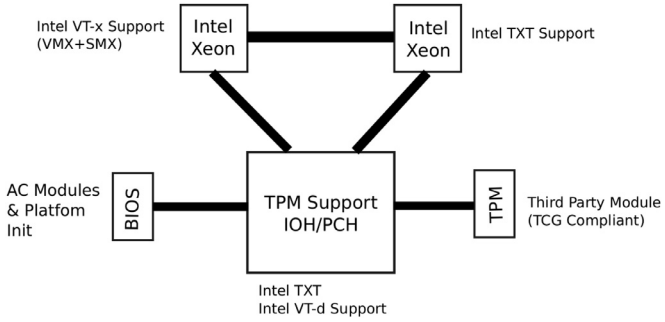


Fig. 1 – Intel TXT components.

3. Hardware-based solutions

Currently, the market of datacenters is dominated by Xeon and Opteron microprocessor architectures manufactured by Intel and AMD, respectively. Nevertheless, some have argued that ARM can become a viable alternative to Intel and AMD architectures for servers due to the reduced size, energy efficiency, flexibility, and low cost of ARM processors (Pinto and Santos, 2019). Based on this, we present in this section hardware-based solutions come from these three major vendors that we compare in the next section.

3.1. *Intel TXT*

Intel TXT was formerly known as LaGrande Technology. It was announced in 2005 and renamed later as Intel TXT. It aimed at creating a trusted platform using a TPM i.e., a dedicated microprocessor formally specified by the TCG (Trusted Computing Group) (Maene et al., 2018). A TPM is designed to secure and integrate cryptographic primitives and keys within the hardware devices. As shown in Fig. 1, TXT is not an alternative to TPM but heavily relies on to provide basic security services (Costan et al., 2017; Wojtczuk and Rutkowska, 2009). It is integrated in the chipset that supports the IOH/PCH (I/O Hub for Input/Output leveraging Intel VT-d technology and Platform Controller Hub for the processors), and interacts with the BIOS and one or more Intel Xeon processors with potential virtualization technology (supported by Intel VT-x) to offer measurement and code authentication features. As Fig. 1 shows, the BIOS includes an ACM (Authenticated Code Module) created and signed by Intel to enable an isolated execution environment within the processor for the system launch (Authenticated Code Mode). Based on the above hardware setup, a MLE (Measured Launch Environment) is created to verify and attest the system software.

Intel TXT provides a measured and controlled boot of the system software that leads to establish a trusted environment (Intel Corporation, 2017). It provides discrete integrity measurements that can prove or disprove a software component's integrity, so it can detect unauthorized changes in the boot sequence, the BIOS, the OS, the software configuration and the policies.

Additionally, Intel TXT provides an attestation mechanism to verify that the system has correctly invoked Intel TXT and make sure that the code is executing in a protected environment. It is especially well suited for cloud computing and other use cases where data integrity is paramount (Greene, 2013b). It is used by leading cloud infrastructure such as Amazon AWS, VMWare vCloud and OpenStack.

As trust is rooted in the processor in Intel TXT, this solution provides a better security protection for sensitive information on the servers. It creates an isolated execution environment by specifying some regions of memory where secret data are stored and manipulated inside. It also offers a sealing service to store encryption keys securely in hard drives.

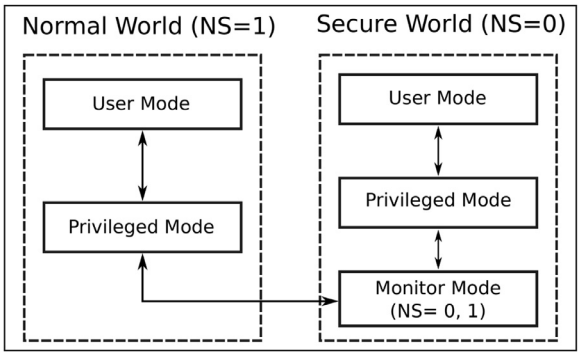


Fig. 2 – Processor modes in ARM TrustZone.

3.2. *ARM TrustZone*

TrustZone is an optional hardware security extension of the ARM processor architectures that provides a secure execution environment by splitting computer resources into two execution worlds: the normal world and the secure world (Ngabonziza et al., 2016). It is a hardware-based security architecture for SoC (System on Chip) that is currently used in a large number of smartphones.

TrustZone technology was announced in 2002 but did not get widely used until 2009. ARM integrated it in ARM64 and ARMv8-M architectures to support a broader range of platforms including servers and IoT devices (Li et al., 2019).

With TrustZone, the processor can execute instructions in one of two possible security worlds: the normal world where untrusted code executes and the secure world where security services run (Costan et al., 2017). As illustrated in Fig. 2, these processor modes have independent memory address spaces and different privileges. In addition to the user mode

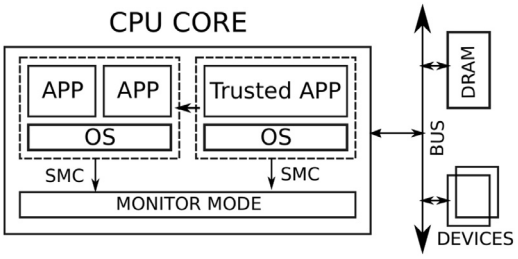


Fig. 3 – Hardware components interaction in ARM TrustZone.

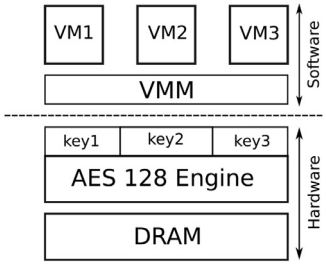


Fig. 4 – AMD SEV architecture.

(in which user applications run) and the privileged mode (in which system software run) which are commonly present in both worlds, Fig. 2 shows that the Secure World can run in a third mode called the monitor mode. This mode is supported by a special part of the ARM processor, and the software running inside has control over all the other software. The switching from this mode to the privileged mode and vice-versa is done via the usual processor interface, i.e., the ISA (Instruction Set Architecture).

In ARM TrustZone and under certain conditions, the code in secure world can access the address space of the normal world, but the reverse is not possible. This mechanism is controlled by a hard-coded bit in the processor called the NS bit; it specifies the current execution world of the processor and it is communicated to the peripherals using some I/O buses (Santos et al., 2014). Access to the NS bit is protected by the secure monitor, which is triggered by the SMC (System Monitor Call) as shown in Fig. 3 (Gonzalez, 2015). The down arrow lines inside the CPU core represent the software-hardware interaction via the SMC, and the other left arrow line represents the direct access from the software running in the secure world address space to the software running in the normal world address space. The other lines outside the CPU core represents the different buses connecting it to the memory and the I/O devices.

3.3. *AMD SEV*

AMD SEV is the unique technology among the four studied and compared that is directly and natively addressing trust issues in a cloud environment. The Secure Encrypted Virtualization (SEV) feature encrypts transparently the memory contents of a VM with a unique key for each guest VM. As Fig. 4 shows, the memory controller contains a high-performance encryption engine (AES-128 hardware encryption engine) that can be programmed with multiple keys for use by different VMs in the system. The management of these keys and the secure data transfer between the host hypervisor and the guest VM memory is handled by the SEV firmware running on the AMD Secure Processor (Advanced Micro Devices, 2018b).

AMD Secure Processor is a dedicated security processor that provides cryptographic functionalities for secure key generation and management. It was incorporated into AMD microprocessors since 2013.

Encrypting VMs can help protecting them from physical threats, other VMs threats or even the VMM itself. SEV represents a new virtualization security solution where the hypervisor or the administrator of the host system is not trusted. AMD SME (Secure Memory Encryption) encrypts all the system memory with a single key that is generated by the AMD Secure Processor at boot (Advanced Micro Devices, 2018a).

SEV based on AMD-V technology can be used in cloud computing to provide a new security model for virtualized environments. It does not require any application software changes, and the VMs' encryption is performed quickly and transparently thanks to dedicated hardware engines (Kaplan et al., 2016).

3.4. *Intel SGX*

Intel SGX was announced by Intel Corporation in 2013 as an extension to the IA-64 ISA. It was introduced in 2015 with the sixth generation of Intel Core processors, that are based on *Skylake* micro-architecture. It gives applications the ability to protect a portion of their address spaces and secure their code and data within containers called enclaves (Wang et al., 2018).

It is an architectural feature that introduces a new set of CPU instructions allowing a user application to create and use enclaves as a hardware-assisted TEE. An enclave is defined as a protected area in the application address space which cannot be altered by a code outside it, not even by higher privileged code (SMM, BIOS, VMM, OS, etc.) (Gonzalez, 2015).

Intel SGX guarantees the confidentiality and the integrity of the enclave code and data at runtime (Costan et al., 2017; Mofrad et al., 2018), even in a fully compromised environment, and the secrets remain protected even when the attacker has full control of the platform. It prevents memory bus snooping and memory tampering and provides hardware-based attestation capabilities to measure and verify valid code and data signatures.

With Intel SGX, cloud tenants can rely only on the CPU hardware for protecting their data and IP (Intellectual Property) against curious or malicious cloud providers. Besides, this technology is also used to protect copyrighted material from piracy via DRM enclaves (Digital Right Management), and various further use cases such as protecting secrets, password managers and messengers, encryption/decryption keys, etc.

Fig. 5 illustrates an example of how Intel SGX protects the user secrets. First, it is worth noting that the application code is explicitly split into two distinct sections in the same address space: the trusted section and the untrusted section. Secret data/code are located in the trusted section, and are accessed from the untrusted section through SGX call gates.

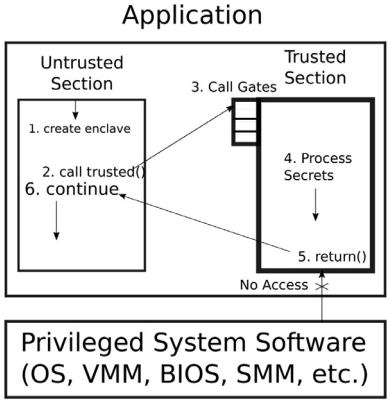


Fig. 5 – Intel SGX trusted execution path.

The first step in this process is to create one or more enclaves via SGX API then make a call to the trusted code. The security of the control flow transfer is ensured by Intel SGX hardware. After finishing secret data processing within the trusted section, the control flow is returned back to normal execution via SGX instructions that guarantee processor state cleaning upon switching the execution mode. All privileged software have no access to the trusted section even if their execution level is higher.

With the upcoming version SGX v2, a new feature called *oversubscribing* will allow multiple VMs to share the EPC (Enclave Page Cache), i.e., making explicit support for cloud computing (Chakrabarti et al., 2017). Intel will also open its technology for the open-source community allowing them to bypass Intel's strict enclave signing policy via their own key infrastructure (Schwarz et al., 2019). Applications running in Intel SGX trusted execution environments could be enriched with some security options such as identity privacy, secure browsing, and DRM. Moreover, Developers could also harden the endpoint protection for some security-demanding applications that must process and store sensitive data securely.

4. Comparison

In this section, we adopt the three categories of comparison criteria defined by Sabt et al. (2015), namely: security, functional, and deployability, with a specific customization of each category.

4.1. *Criteria definition*

The criteria we adopt in our comparison can be found in different execution environments. Nevertheless, we present them in this subsection in the perspective of cloud computing usage.

Also, it is worthy to notice here that the far distance between the two abstraction levels of cloud computing and hardware-based solutions for trust makes non-obvious such a presentation. Secret data/code are defined regarding the end-users not the CSPs or the hardware. Therefore, the CSPs interact with the cloud infrastructure via a stack of cloud management software that run in remote VMs with a different semantics of trust. Moreover, the domains where target applications run are located in different and possibly remote VMs too, that themselves use a stack of privileged system software. This latter can be brought by the same CSP or another CSP via a PaaS. All this stack may run above a host OS that runs on top of the target hardware via IaaS. In the hardware, secret data/code access semantics change drastically when compared to a normal execution context.

A detailed full presentation of how hardware-based solutions offer trust to cloud services from end-user perspective, and how end-user secret data/code protection semantics cross the boundary of different interpretation domains are out-of-scope of this paper. Nonetheless, we try in this subsection to demystify this relationship in the light of the simple cloud setup we defined in Section 2.

4.1.1. *Security criteria*

This category focuses on the mechanisms for avoiding specific attacks on the cloud according to the attacker model defined in Section 2.1. They represent the security features implemented by the TEE mechanisms for execution environment isolation.

- **Isolation level:** isolation of a cloud execution environment is defined as the property by which VMs, processes and threads executing inside it do not interfere with untrusted software and/or hardware parts outside it. Outsider unauthorized software cannot read nor modify it. The modification is controlled and permitted for authorized parts only by the hardware-based solution via strict call gates. The isolation level shows the area where the isolation is done by the hardware-based solution e.g., Memory, CPU, ROM, etc. From cloud users point of view, isolation at all these levels is hidden by the system software and driver modules. It is transparent to application developers and security managers too, thanks to the compilers, SDKs (Standard Development Kit) and programming languages. However, OS/VMM editors must carefully deal with the isolation level established by the respective hardware solutions as all the execution model changes from one level to another.
- **Memory confidentiality and integrity protection:** the hardware-based solution provides confidentiality protection by encrypting the memory secret code/data. Memory integrity protection guarantees the detection of any unauthorized modification or alteration of these code/data. These features are required in a cloud environment as VMs with sensitive data are often owned by unique cloud tenant, and shared VMs needs to be modified only by authorized parties.
- **Protection against compromised OS/VMM/BIOS/SMM:** the hardware-based solution provides protection against privileged software as the latter can be compromised and used to retrieve secret data. In addition, most of the system software stack in a cloud environment, as described in Section 2, is not controlled by the end-users, and the CSPs are untrusted.
- **Protection against physical attacks:** a physical attack is an unauthorized physical access to the hardware where the attacker can *physically* take off a device or a component from the platform for later analysis in order to get some

secrets. The hardware-based solution that provides a protection against physical attacks must protect the data that are stored in the specified component, generally by cryptographic methods. Usually, cloud infrastructure is protected against such a type of attacks. Even though, hardware-based solutions may offer this protection especially for high-security applications.

- **Protection against memory snooping:** snooping is a mechanism used to keep the cache memory coherency. Attackers can get information from sniffing the bus between the cache memory and the main memory. With regards to the attacker model described in Section 2.1, this supposes that the attacker gets first access to the trusted cloud infrastructure, bypasses the trusted system software protection mechanisms, and then injects code to observe memory transactions, interferes with the bus access and implants side channel or timing attacks.
- **Secure storage:** the sensitive data is stored in a secure storage that is protected by the hardware-based solution. This criterion may be part of page swapping from protected to unprotected memory that is implemented by the untrusted system software according to the attacker model. As the protected memory region maybe insufficient, the hardware-based solution may offer such a feature to ensure data confidentiality and integrity when they are copied outside the secure memory regions.
- **Secure boot:** secure boot verifies the integrity of each stage of the boot process by computing a hash and comparing the result with a cryptographic signature. This is the first step in the chain of trust in a cloud environment as defined in Section 2 because software or hardware components outside the TCB including the BIOS, the SMM, etc. may interfere with the boot sequence. So, hardware-based solutions may offer mechanisms to prevent data tampering and preserve their integrity.
- **Cache memory protection:** cache-timing attacks are a serious threat to security-critical software. Side-channel attacks are based on timing information leaked from the CPU cache memory. They can be used to recover critical algorithm state such as key material. This type of protection may be considered in every execution environment as caches become a fundamental part of modern processors. However, in a specific cloud setup as described in Section 2, this requires that the attacker gets access to the cloud infrastructure, runs arbitrary software at any privilege level and interferes with the memory subsystem. The attacker model we consider permits all these capabilities.
- **Memory access protection:** the hardware-based solution that protects the memory must provide a mechanism of protection against unauthorized access to the memory. It must permit only the authorized entities to access secret data in the memory. As VMs from different (an maybe antagonist) cloud tenants are co-located in the same physical memory space, enforcing this protection may help preventing lot of threats that try to bypass such a control access.

4.1.2. *Functional criteria*

These criteria measure the physical protection of the data through the entire usage cycle: receipt, input, processing, output, and validation. They give the requirements of a tamper-resistant TEE for data in all their lifecycle.

- **TCB size:** the TCB comprises the system components that must work correctly to ensure security. Security solutions aim to reduce the TCB size and keep it as small as possible to reduce the susceptibility of compromise. This criterion is common to every computing environment, but it is much more in demand in open ones such as the cloud setup defined in Section 2.
- **Debugging:** it is the ability to examine and inspect a program and step through its code. A hardware-based solution may disable the debugging of the programs in order to ensure the security of the code. In a hardware-based solution for trust in the cloud, this feature is highly required especially to debug remote code within a TEE.
- **Attestation:** hardware-based attestation provides evidence that the right application is executing on an authentic platform. This is particularly useful in the cloud environment we consider because clients need a third party to establish trust with other cloud tenants via remote attestation. Therefore, hardware-based solutions may offer such a feature to enable inter-domain (secure and non-secure) communications and share VMs securely in untrusted cloud.
- **Sealing:** sealing is the ability to encrypt the secure data and store it on a non-volatile memory such as a hard disk drive, so the data can be used later. This criterion offers protection to data outside the DRAM as the VMs may be suspended and secure memory space becomes insufficient. Sealing is interestingly useful as the hardware-based solutions do not trust the hypervisor/host OS/guest OS file systems according to our cloud setup model.
- **Execution privilege level:** a privilege level controls the access of the software currently running on the processor to the resources such as memory regions, I/O ports, etc. The user applications are performed with ring 3 privilege level (least privileged) and the OS on the ring 0 (most privileged). In the cloud environment assumed in Section 2, a hardware-based solution usually accepts any privilege level of code running inside the protected memory. However, there may be some restrictions about this criterion as the solutions do not guarantee code/data protection if its execution privilege is below a certain level. The attacker model we consider accepts the first alternative, but does not reject the second.

4.1.3. *Deployability criteria*

These criteria measure the barriers to adopt a security solution.

- **TEE features**: TEE is a secure area that resides in an electronic device and ensures that sensitive data is stored, processed and protected in a trusted environment. In our comparison, we show the TEE proprieties provided by each hardware-based security solution in the light of cloud setup described in Section 2, including verified launch of VMs, inter-VM sharing between multiple cloud tenants, remote attestation for VM image transferring and installing, etc.

- **Application modification:** it reflects whether the deployment of legacy applications requires a modification in the code, the design or the implementation of the application to run on a specified hardware-based security solution. Although, cloud-native applications do not require this feature as they have their own design and implementation models, it may be of great interest to have it in hardware-based solutions for migrating legacy applications to a trusted cloud because a non-trivial complete redesign of these applications maybe required. As less as a solution requires application modification, its adoption in IT industry becomes faster.
- **Performance:** a hardware-based solution gives more security performances but drives generally to an overhead (control mechanisms, cryptographic primitives, interruptions, etc.) that can affect the other performances such as the execution time, resource consumption, etc. This performance degradation should be justified by a hardware-based solution in regards with the attacker model it deals with and the trust guarantees offered to the users, customers and CSPs compared to software-based solutions.
- **VM migration:** live migration is the movement of a running VM from a source platform to a destination platform. This process should take place without any noticeable effect and without losing any state such as network conditions or CPU status (Fan et al., 2015; Yamada, 2016). This feature induces a challenge when secret data/code state in VMs are protected by the hardware-based solutions because they are neither readable nor writable by any hypervisor/kernel or cloud management software stack. This challenge is worsen when a service or an application is deployed in a hybrid cloud because, it is not trivial to determine whether the source and the target platforms support the same hardware solution and software requirements for migration at runtime.
- **License agreement:** license agreement is used to indicate if a contract between the two parties (client and TEE vendor) is required to use the hardware-based security solution. This is an additional separate agreement added to the standard SLA (Service Level Agreement) established by every CSP.
- **Ecosystem:** The ecosystem represents the extend of the hardware-based solutions and its adoption by IT hardware and software vendors. Actually, there are two kinds of ecosystems: non-cloud (standalone) ecosystem and cloud ecosystem. The standalone ecosystem includes the support by low-level software tools, compilers, and SDKs, and can be measured by the number and spread of supporting OSs and hypervisors; it is ready for all these solutions and users can exploit them. For the cloud ecosystem, it depends on each technology.

4.2. *Comparison with respect to security criteria*

4.2.1. *Isolation level*

Intel TXT provides an isolation on the system for data, applications, and VMs by protecting them from unauthorized direct memory access when used in a virtualized environment with Intel Virtualization Technology (Intel VT-x) (Greene, 2010-2012). AMD SEV isolation level is at the memory where it encrypts and protects the system memory (Mofrad et al., 2018) of each VM. With ARM TrustZone, each of the physical processor cores provides two virtual cores, one considered non-secure and the other secure. The Secure World and Normal World have independent memory address spaces and different privileges. Thus, ARM TrustZone allows strong isolation of the physical memory as well as the peripherals (ARM, 2009; Santos et al., 2014). Intel SGX provides an isolation level at the CPU (the enclave mode) and the memory, i.e., enclaves in the PRM (Processor Reserved Memory).

Regarding the cloud setup defined in Section 2, isolation is mainly ensured in memory as per VM/process/thread basis and enforced by the processor.

4.2.2. *Memory confidentiality and integrity protection*

Both AMD SEV and Intel SGX ensure the memory confidentiality but Intel TXT and ARM TrustZone do not enforce memory encryption (ARM, 2009; Intel Corporation, 2013; Mofrad et al., 2018; Sobchuk et al., 2018). Nonetheless, except AMD Memory Encryption Technology that does not provide integrity protection for the encrypted memory spaces of the guest VMs (SEV or SME) (ARM, 2009; Greene, 2013b; Intel Corporation, 2013; Mofrad et al., 2018; Sobchuk et al., 2018), all the other technologies, i.e., Intel TXT, ARM TrustZone and Intel SGX provide such a protection. Intel SGX is particularly interesting in the cloud environment defined in Section 2 where it is paramount to ensure both confidentiality and integrity of cloud tenants' data and code at the very low-level against known and unknown system software vulnerabilities and its potentially malicious behavior.[4] The other technologies guarantee either confidentiality or integrity but not both.

4.2.3. *Protection against compromised OS/VMM/BIOS/SMM*

Contrary to AMD SEV that provides a protection against the VMM (Jang et al., 2018; Kaplan et al., 2016) but not the guest OS (Sabt et al., 2015; Santos et al., 2014), all the other solutions provide a protection against both the VMM and the guest OS (Sabt et al., 2015; Santos et al., 2014; Sobchuk et al., 2018), especially Intel TXT which ensures a secure load of the OS or the VMM even in a potentially compromised machine (Greene, 2010-2012; 2013b; Intel Corporation, 2013; Wendt and Guise, 2011; Wojtczuk and Rutkowska, 2009).

Moreover, Intel SGX provides a protection against a compromised BIOS or even a compromised SMM (Costan and Devadas, 2016), but Intel TXT does not validate the SMM memory while the software is being loaded. Any malware that is hidden in the SMM can compromise the software that has just been launched (Costan and Devadas, 2016; Sasy, 2017; Wojtczuk and Rutkowska, 2009). Similarly, AMD SEV does not provide protection against a malicious BIOS or SMM.

The definition of such a protection for each solution helps assessing the risk of having such a software in the TCB, and thus evaluating the security threats on the whole cloud infrastructure.

[4] That are very difficult to discover and analyze because of the complexity of such a software.

4.2.4. *Protection against physical attacks*

In one hand, both Intel TXT and ARM TrustZone do not offer protection against such a type of attacks because the former does not implement DRAM encryption or HMACs (Hashed Message Authentication Code) (Sasy, 2017), and the latter does not have any countermeasures for physical attacks (Costan and Devadas, 2016). Therefore, both are vulnerable to the physical attacks. In the other hand, VMs' encryption with AMD SEV can help protect them from physical threats (Kaplan et al., 2016), and Intel SGX does not protect from the attackers that are able to unpack the CPU and run attacks such as invasive fault injection. However, it resists against physical memory access attacks (Mofrad et al., 2018). In a public cloud infrastructure with serious physical security threats, none of the four technologies protect against such a type of attacks. Nonetheless, it is preferable to adopt Intel SGX especially for relatively controlled private clouds with IaaS, PaaS or even SaaS as it offers the hardest defense in this context.

4.2.5. *Protection against memory snooping*

All the four instances provide a protection against memory snooping: for instance, Intel TXT provides hardware-assisted methods that remove residual data at an improper MLE shutdown, thus protecting data from memory snooping software and reset attacks (Greene, 2010-2012). For ARM TrustZone, on-SoC memory is the main reason for preventing snooping attacks (ARM, 2009). For AMD SEV, using a random key provides a strong protection against DRAM interface snooping and similar types of attacks (Kaplan et al., 2016) thanks to full memory encryption. Similarly, Intel SGX encrypts the content of enclaves using silicon-stored keys every time they leave the CPU (Costan and Devadas, 2016; Sobchuk et al., 2018).

4.2.6. *Secure storage*

Intel TXT uses TPM that can provide a secure storage for cryptographic keys. In the same way, AMD SEV keys' management operations such as generation, storage and delivery are carried out by the AMD Secure Processor and the encryption keys are kept hidden from untrusted parts of the platform (Mofrad et al., 2018). Intel SGX uses the enclaves i.e., implemented in the PRM as a secure and tamper-proof for providing a secure storage. At the opposite of the three above-described solutions, current ARM platform specifications do not include a root of trust for long-term secure storage. Platform hardware vendors are free to choose and implement a proprietary mechanism if desired. The Secure Element (SE) is one of those proprietary solutions (Vasudevan et al., 2012). This may give flexibility to the CSP but complicate license agreement for end-users because multiple parties could be concerned.

4.2.7. *Secure boot*

Intel TXT provides secure boot by measuring each component of the boot process to ensure that the system was initialized securely and has not been altered from a known trusted configuration (Sobchuk et al., 2018). ARM TrustZone is often completed with additional features such as secure boot and root of trust (RoT) hardware module, which allow TrustZone to satisfy all the requirements of a tamper-resistant environment (Sabt et al., 2015; Sobchuk et al., 2018). Since 2006, AMD has incorporated a silicon-level feature called SKINIT in all its CPU products to support secure boot. However, Intel SGX does not provide such a feature (Sobchuk et al., 2018). This could be unnecessary because Intel SGX, as AMD SEV though, does not build its chain of trust upon a sequence of booted system software, but considers a direct far-distance relationship between the trusted hardware and the user level pieces of software. And that is what it makes its particularity among the other solutions.

4.2.8. *Cache memory protection*

Cache memory attacks are the most difficult to prevent. All the four instances do not provide a protection from such an attack. Intel TXT may be subject to cache attacks as it allows trusted execution using processor cache lines to minimize interaction with external resources when cache coherency is not needed (Cheruvu et al., 2020). In an ARM TrustZone system, both partitions (normal and secure) share the same CPU, so there are strong indications that a cache-based side-channel vulnerabilities exist (Lesjak et al., 2015). AMD TEE is vulnerable to memory side-channel attacks because data inside the SoC appears in clear text. Thus, the information in the cache is in plaintext and the cache access time can be measured by a malicious high-privileged entity (Mofrad et al., 2018). Side channel attacks against SGX-enabled programs are also possible (Van Bulck et al., 2018) and Intel SGX specifications put the responsibility on the user to write a safe code (Sobchuk et al., 2018).

4.2.9. *Memory access protection*

All the four instances provide protection against unauthorized memory access. Intel TXT implements two chipset mechanisms that can be used to protect regions of memory from DMA (Direct Memory Access) by bus master devices (Intel Corporation, 2017), namely: DPR (DMA Protected Range) and PMRs (Protected Memory Regions). ARM TrustZone partitions the DRAM into several memory regions, each of which can be configured to be used in a normal world or a secure world. These execution worlds have independent memory address spaces and different privileges (Santos et al., 2014). AMD extensions enable either encryption of large parts of the RAM or specific VMs to remove trust from the hypervisor (Horsch et al., 2017). They introduce an AES 128 encryption engine inside the SoC that transparently encrypts and decrypts data when they leave or enter the SoC, respectively. Intel SGX uses PRM (Processor Reserved Memory) to protect EPC (Enclave Page Cache) from unauthorized access (Costan and Devadas, 2016; Harnik et al., 2018; Sobchuk et al., 2018).

4.3. *Comparison with respect to functional criteria*

4.3.1. *TCB size*

Intel TXT has a large TCB comprising the TPM, the CPU, the motherboard and the system buses as shown in Fig. 1 (Shepherd et al., 2016). ARM TrustZone provides a TEE in which sensitive applications execute securely, thus offering a TCB smaller than rich environments several orders of magnitude (Pinto and Santos, 2019). Similarly, AMD SEV puts the underlying OS and the hypervisor in the TCB, thus induces

a big TCB size and enlarges the attack surface (Mofrad et al., 2018). In counterpart, Intel SGX considers only the CPU package and enclaves' code in the TCB; the other parts of the system are considered untrusted. Thus, it has the minimum TCB size (including both hardware and software).

4.3.2. *Debugging*

With Intel TXT, PCR 16 (Platform Configuration Register) is reserved for debugging but not used by Intel TXT as stated in the specifications (Greene, 2013b). The debugging solutions provided by ARM split into two parts: the processor debug components and the system debug components (ARM, 2009). In AMD SEV, the debugging interface allows a hypervisor to dump protected memory of a guest OS in plaintext (Buhren et al., 2019). Finally, Intel SGX enclaves can be built in debug mode or in release mode. A debug mode enclave is inspectable, i.e., we can attach to it with the Intel SGX debugger tool, examine its state, and step through its code just as debugging any other application. A release mode cannot be debugged under any circumstances and this restriction is enforced by the CPU (Hetzelt and Buhren, 2017).

4.3.3. *Attestation*

Attestation is provided by Intel TXT and Intel SGX (Jang et al., 2018; Kaplan et al., 2016; Sabt et al., 2015; Shepherd et al., 2016; Sobchuk et al., 2018). The latter provides both local and remote attestation. ARM TrustZone's documentation does not describe any software attestation implementation (Costan and Devadas, 2016; Maene et al., 2018; Sabt et al., 2015). In contrast, AMD SEV provides indeed an attestation of the guest launch that proves to the guest owners that their VMs are securely launched with SEV enabled (Pinto and Santos, 2019). It is done through the AMD Secure Processor (Mofrad et al., 2018).

4.3.4. *Sealing*

Intel TXT provides secret data sealing (Greene, 2013b), but, ARM TrustZone and AMD SEV do not (Pinto and Santos, 2019). Intel SGX, in counterpart, provides such an ability to seal data of a specific enclave for long-term storage in non-volatile memory. The data is cryptographically sealed using a key derived via SGX hardware before being written to the disk (Sobchuk et al., 2018).

4.3.5. *Execution privilege level*

Intel TXT uses TPM where the execution level is the ring 3 (Pinto and Santos, 2019) (execution ring in some architectures is called execution level and we use both terms interchangeably). In ARM TrustZone, the execution level is ring 2, whereas AMD SEV protected VMs provide ring 0 and high-privileged access. With Intel SGX, the enclave code can only execute in ring 3 (Mofrad et al., 2018).

4.3.6. *Comparison between TCB sizes and hardware-software interactions*

Fig. 6 shows a comparison between (a) Intel TXT, (b) ARM TrustZone, (c) AMD SEV, and (d) Intel SGX in terms of the TCB parts and sizes, the trusted hardware and software and the interactions between them.

Intel TXT hardware verifies the firmware of the platform via measurement checking prior to startup, and blocks the machine boot if the measures do not match. After that, it measures the hypervisor and compares it with a prior value. Again, if the measures do not match, it blocks the hypervisor, and the system startup is blocked. If the measures are identical, the hypervisor is trusted in this case, and it is allowed to load up the VMs (guest OS + applications). From subfigure (a), we can see that the hypervisor is the only part that is trusted when deploying Intel TXT in a cloud environment. However, other parts such as the guest OS and the user applications may also be measured and verified by the hypervisor, thus creating a chain of trust from the hardware up to the applications.

ARM TrustZone defines another mode in the ARM processors that controls the world where the VM is executed. Explicit SMC is required to switch between the Secure World and the Normal World. The TCB in ARM TrustZone comprises the hardware of the Monitor Mode and the trusted applications running in the secure world. As we can see in subfigure (b), the TCB in ARM TrustZone is not restricted to the user applications, but extended to one or more guest OSs (Kwon et al., 2020).

Similarly, AMD SEV trusted software should comprise the whole VMs, and can be extended to multiple VMs. Indeed, AMD SEV does not include the hypervisor or the host kernel, but from subfigure (c), the hardware part of its TCB is smaller compared to subfigure (d), i.e., including only the microcode and the Secure Processor. The VMs bypass the hypervisor for trust enforcing that is providing a strong and direct interaction between the different parts of the TCB.

Subfigure (d) shows a scheme of Intel SGX in contrast to AMD SEV. Contrary to AMD SEV, Intel SGX hardware part of the TCB is larger as it includes the microcode and the Memory Encryption Engine among others but a reduced software part which is the enclaves. In the whole, TCB size of SGX is smaller because there are less software in the enclaves. This maybe a good design choice because the vulnerabilities/threats in/on the hardware are fewer compared to the software, especially in a cloud setup where the end-users are the only responsible party of the outsourced secret data/code. However, this alternative can turn out to an obstacle or induces some performance penalties if the clients require more use cases of Intel SGX because of the lack of protected memory (Gu et al., 2020).

Overall, the four trust technologies are comparable as they all offer a hardware TCB part more or less reduced, and a strict hardware-software interactions ruled by ISA interfaces that do not trust intermediate system software stacks.

4.4. *Comparison with respect to deployment criteria*

4.4.1. *TEE features*

Intel TXT is a TEE that provides protected execution and memory space where sensitive data can be securely processed. It provides also verified launch, secret protection, and attestation (Intel Corporation, 2013). With ARM TrustZone, the TEE security level and trustworthiness is maintained by adopting restrictions where only strictly verified applications can be deployed in the TEE (Jang et al., 2018). AMD SEV is appropriate for a broader range of applications, particularly for those requiring many system calls (Mofrad et al., 2018) such as VMs in a cloud environment. Intel SGX trusted code executes in ring 3, thus it is not a suitable TEE for applications that require many

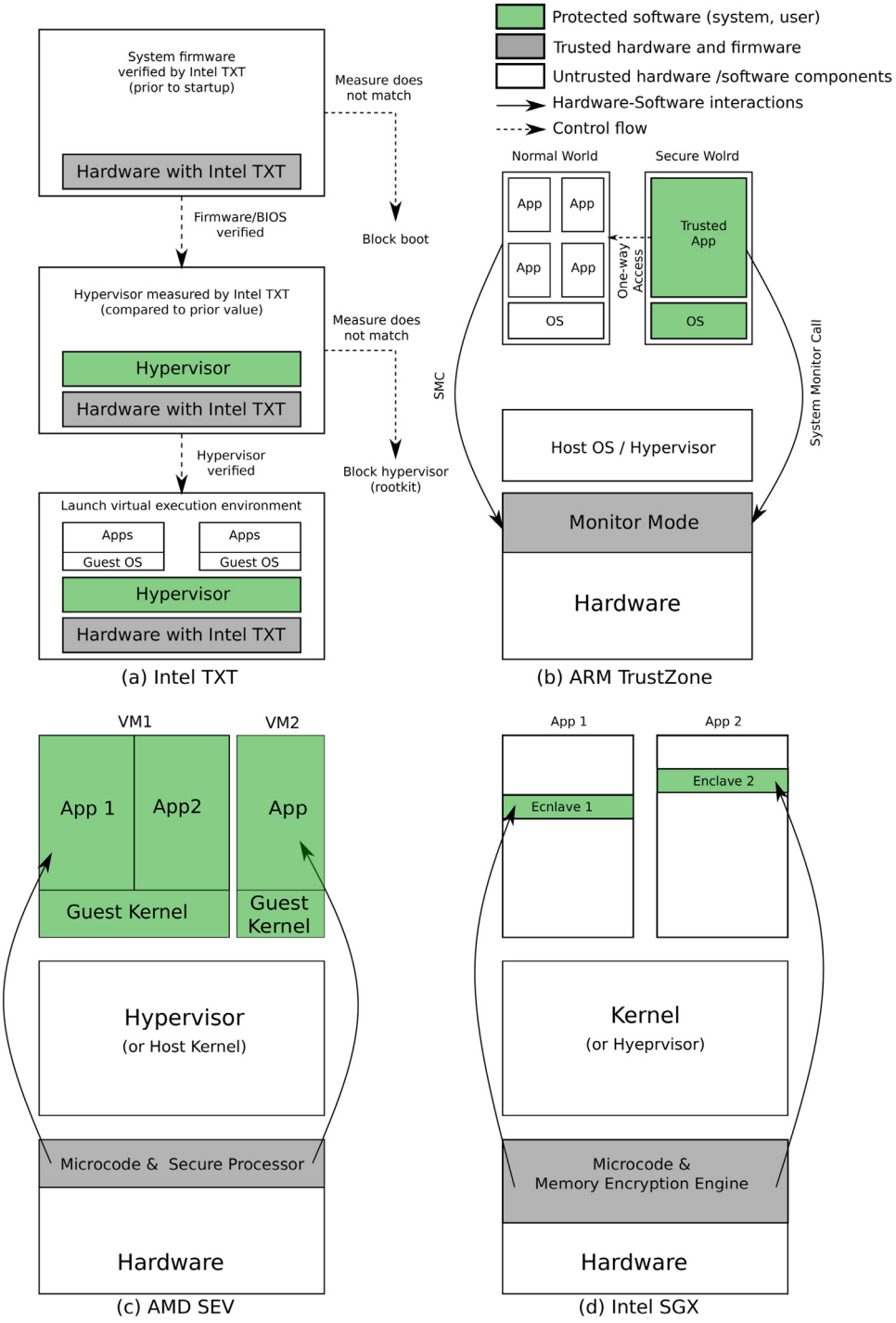


Fig. 6 – Comparison between the TCB and hardware-software interaction of Intel TXT, ARM TrustZone, AMD SEV and Intel SGX (Gu et al., 2020).

system calls (Mofrad et al., 2018). It is rather appropriate for applications that require an enhanced-degree of security protection (Mofrad et al., 2018) against higher privilege level system software. It offers besides mechanisms for shrinking TCB and reducing the attack surface with attestation and sealing options. From a cloud setup perspective, AMD SEV is the best that offers as many as TEE features (except for memory integrity protection) followed by Intel SGX.

4.4.2. *Application modification*

Intel TXT does not explicitly require application modification, but must include a piece of software called ACM signed by the manufacturer of the chipset[5] (Futral and Greene, 2013). ARM TrustZone can run legacy systems without modifications,

[5] That is generally Intel as most Intel-based platforms use Intel chipsets.

since each world has its own privileged modes, and thereby removing the necessity of instruction emulation (Sabt et al., 2015). Similarly, AMD SEV does not require code modification for securing legacy applications. Therefore, using SEV by end-users in cloud environments is almost with no effort (Mofrad et al., 2018). At the counter-part, Intel SGX applications often require a new software design model sectioning code into at least two sections: trusted section and non-trusted section. Legacy applications may require complete redesign thus making their migration to Intel SGX-enabled cloud execution environments non-trivial (Mofrad et al., 2018).

4.4.3. *Performance*

Since Intel TXT does not interfere with the OS/application execution, it has no effect on the performance. Additionally, except for secure launch, it does not consume memory bandwidth or other resources (Greene, 2013b).

The performance decrease in ARM TrustZone is negligible. It is overall estimated by Ning et al. (2018) around 0.13%. Nonetheless, the context switch between the secure world and the normal world is the key operation that creates an overhead and may lead to performance degradation (Pinto et al., 2014). It is stated in Amacher and Schiavoni (2019) that the switch from REE (Rich Execution Environment) to TEE (110 µs) is more time-consuming than from TEE to REE (47 µs).

Thanks to MEE (Memory Encryption Engine), AMD Memory Encryption Technology outperforms Intel SGX for securing large regions of memory resources (Mofrad et al., 2018). AMD SEV can run at native speed (Göttel et al., 2018); the average SEV performance slowdown indicates 1×, or identical performance to the AMD unprotected VM (Mofrad et al., 2018). Besides, because of the limited size of PRM memory (EPC first 128 MB in SGX v1 and then 256 MB in SGX v2), Intel SGX can degrade the performance of applications especially when larger trusted memory space is required for the enclaves (Mofrad et al., 2018). The latency of enclave creation is non-negligible and the interactions between the enclaves involve important overhead: it is estimated that a function call that crosses trusted/untrusted boundary may consume more than 7000 processor cycles (Gu et al., 2020). In addition, protecting memory integrity may slowdown read/write operations by a factor of 10 compared to normal random operations (Göttel et al., 2018).

The memory protection offered by Intel SGX comes with some time and storage penalties, because of the overhead induced by encryption/decryption operations enforced by the MEE. For instance, a factor of 19.31 of performance slowdown on average is incurred when an enclave executes an intensive floating point operations, and a factor of 9.45 of performance overhead in comparison with its unprotected workload (Mofrad et al., 2018). Read/Write operations from/to main memory may experience an additional penalty compared to usual cache misses (Sobchuk et al., 2018). SGX is slower than SEV because the usable SGX EPC memory is up to 96 MB and for every buffer greater than 96 MB, the memory pages should be safely swapped between the untrusted system memory and the enclave, thus resulting in a significant performance overhead (Mofrad et al., 2018).

4.4.4. *VM migration*

Intel TXT creates a pool of trusted hosts to which VMs can migrate with the assurance that Intel TXT is enabled on those hosts and platform launch integrity has been verified (Greene, 2010-2012). ARM TrustZone support for live migration of VMs is not specified, and may require software emulation to access secure world states (Smirnov et al., 2013). Besides, to protect the confidentiality of guest VMs during migration, AMD SEV encrypts their memory images with a key that can be recovered by the destination host platform. This protection is supported by the same set of interfaces provided by the firmware of the different hosts (Advanced Micro Devices, 2018b). Unlike Intel TXT and AMD SEV, VMs with Intel SGX enclaves loses the capability of live migration, because the SGX hardware prevents privileged system software running outside the enclaves from accessing their states, thus making copies of the memory images impossible without additional emulation/simulation mechanisms (Gu et al., 2017). A clause in the SLA should specify the capability of migration of live VMs in multi-clouds prior to any such operation for the four solutions.

4.4.5. *License agreement*

Existing projects on TrustZone span across various application domains, notably mobile, industry, automotive, aerospace, etc. and are released under different licensing policies (open-source and proprietary) (Pinto and Santos, 2019). Intel SGX enclaves launched in Release mode and the production version of the Attestation Service require a signed Commercial Use License Agreement.

4.4.6. *Ecosystem*

Intel TXT Intel TXT is included in Xeon processors with support for Intel VT-x (VMX and SMX). Also, A TPM must be integrated within Intel chipset to provide isolation capabilities for the MLE. These features were introduced with Xeon 5600 series processors (Mulnix, 2015).

Intel TXT ecosystem is growing over time thanks to the spreading support of the technology by different platforms, software products and service providers (Greene, 2013a). For the OSs/hypervisors, Intel has been maintaining "*tboot*" (Trusted Boot) project i.e., the most widely used mechanism as a foundation for software vendors to enable their OSs/hypervisors (Greene, 2013b). Tboot is an open source pre-kernel/VMM module, included in multiple open source OSs/hypervisors, from Linux to Xen/KVM (Kernel-based Virtual Machine), as well as a number of commercial products such as Red Hat, Citrix XenServer, etc. It uses Intel TXT to perform a measured and verified launch of an OS/hypervisor (Kernel.org, 2017). To enable Intel TXT for their software, some other vendors have implemented their own tboot-like functions. In the Cloud, Intel TXT is already supported by IBM Softlayer cloud service and Amazone EC2 (Greene, 2013b).

ARM TrustZone For ARM TrustZone ecosystem, TrustZone technology is included in Cortex-A and Cortex-M processors, plus a range of ARM TrustZone-enabled CPUs.

Solutions with TrustZone-assisted TEE have been developed primarily to secure data and applications on mobile platforms and designed to operate on a standalone basis. Later,

other systems have been conceived to be tightly coupled with a cloud backend (Pinto and Santos, 2019).

Virtualization in the secure world is supported with ARMv8.4 architecture that introduced "Secure EL2" extension (ARM, 2018) and with the new generation of Cortex-M microcontrollers. Two lightweight RTOS (Real-Time OS) can run side by side, one as secure VM and the other as non-secure VM (Pinto et al., 2017). For hypervisors support, LTZVisor is a lightweight TrustZone-assisted hypervisor that allows the consolidation of two VMs, running each of them in an independent virtual world (secure and non-secure).

ARM applications can be developed with a MDK (Microcontroller Development Kit); a comprehensive software development solution for creating and debugging secure and non-secure applications for ARM8-M based devices. It includes an IDE, a C/C++ compiler, a debugger and a software pack management.

Qualcomm 48-core Centriq 2400 server is the all-new chip that could compete with Xeon processors. This tendency shows the intention of ARM to introduce an ARM server chip in the datacenter infrastructure (Pinto and Santos, 2019). It supports ARM TrustZone secure operating environment and hypervisors for virtualization. Microsoft has been testing its Windows Server OS on ARM-based server machines made by Qualcomm and Cavium, and it will begin using ARM-based servers in its Azure cloud.

AMD SEV AMD SEV has been integrated into the core architecture of x86-based EPYC processor platform, and was further enhanced with the 2nd generation EPYC platform launched in August 2019. These processors can cryptographically isolate up to 509 VMs using AMD SEV without application modification (Ashish Nadkarni, 2019).

SEV requires enablement in the guest OS and the hypervisor. The hypervisor changes use hardware virtualization instructions and communication with the AMD Secure Processor to manage the appropriate keys in the memory controller. AMD SEV is ready to take part of Linux 4.16, i.e., it will be part of the mainline kernel and KVM. AMD and Google Cloud announced the beta availability of Confidential VMs for Google Compute Engine powered by 2nd generation of AMD EPYC processors (Aaron Grabein, 2020; Wu et al., 2018).

Intel SGX In the case of Intel SGX ecosystem, SGX is supported by the sixth, seventh, eighth and ninth generations of Intel Core processors, Celeron J4105 and J4005 models that include BIOS-enabled SGX, Intel Xeon processors E3-1500 version 5 and 6 and Intel Xeon E family 2100. Also, Intel SGX must be supported and enabled by the system BIOS; else, Intel SGX applications can be built and debugged in simulation mode only.

Software developers can create, debug and deploy Intel SGX-enabled applications using C/C++ and Intel SGX SDK, which is a collection of APIs, libraries, documentation, sample source code and tools and it is suitable with many production implementations. Intel SGX is supported by Window and Linux OSs. It is virtualized using the KVM virtualization module in the Linux kernel with QEMU VMM (Andrew Baumann and Hunt, 2014; Christopherson, 2017). With Intel SGX Data Center Attestation Primitives (DCAP), it is allowed for the enterprises, datacenters and CSPs to build and deliver an attestation service themselves that allow all verifications to remain on the local network rather than using the remote attestation from 3rd party provider (Muhammad Usama Sardar, 2021; Schuster et al., 2015). Among the Intel SGX ecosystem partners we find: Alibaba Cloud ECS Bare Metal Instance, Baidu (FaaS) Function as a Service, Fortanix Enclave Development Platform, IBM Cloud Bare Metal and Microsoft Azure Confidential Computing (Intel, 2020).

5. Summary and discussion

Tables 1 and 2 summarize our comparison between the four industrial solutions studied above. The checkmark "√" means that the criterion is supported or provided by the hardware-based solution, and the crossmark "✗" means the opposite, i.e., it is not supported or provided. When the criterion is not applicable or not specified, the ringmark "●" is used accordingly.

As we can see on the tables, the four security solutions provide a hardware-assisted TEE. Intel TXT can detect unauthorized changes to the BIOS, the boot sequence and the OS. More importantly, it provides an attestation mechanism that can be used by any entity that wishes to make a trust decision which is a desired feature in the cloud (Greene, 2013b). However, the problem with SMM attacks against Intel TXT is not solved yet even with some attempts to harden SMM protections. Also, all physical attacks that succeed for a standalone TPM, e.g., LPC (Low Pin Count) bus tapping, also succeed for Intel TXT (Maene et al., 2018).

ARM TrustZone provides a strong hardware-enforced isolation for trusted software, but it does not provide attestation and sealing mechanisms. Its design cannot prevent physical memory disclosure attacks such as cold boot attack from getting the sensitive contents in the memory (Zhang et al., 2016). The current availability of this technology today is on mobile devices. The datacenter and server market is dominated by Intel and AMD and this can be a barrier for the deployment of TrustZone-assisted TEE in the cloud (Pinto and Santos, 2019).

AMD SEV is suited for securing complex and legacy applications but it does not provide memory integrity protection which is a key element for security that may weaken its protection capabilities (Mofrad et al., 2018). AMD SEV suffers from some attacks such as Replay Code Attacks (Hetzelt and Buhren, 2017). Despite the discovered serious design issues in AMD SEV (Intel, 2019), it is still considered as a promising technology particularly because it is the only solution that supports clouding environment by design.

Intel SGX TCB comprises only the processor, thus any other entity is considered potentially risky. It can protect the data/code from any other software (OS, VMM, BIOS, SMM, Applications). Therefore, it is suitable for applications that require a high level of security protection of sensitive data. Moreover, Intel SGX prevents any attempt of data snooping and establishes attestations to ensure that the code is executing on a secure Intel CPU, i.e., wanted feature in a cloud setup. Legacy applications with Intel SGX require a modification in design and implementation, and it is the responsibility of the programmer to write a safe code against side channel attacks. Even with its strong model for securing applications against software or hardware attacks, Intel SGX remains vulnerable to cache-timing attacks. This puts the responsibility on the de-

Table 1 – Summary of the comparison between hardware-based solutions: Intel TXT, ARM TrustZone, AMD SEV, and Intel SGX (Security criteria).

	Propriety	Intel TXT	ARM TrustZone	AMD SEV	Intel SGX
Security Criteria	- Isolation Level	Memory. Isolation is provided to applications, data, and VMs by protecting from unauthorized DMA	Processor. Secure World and Normal World modes have independent memory address spaces and different privileges. Strong isolation of physical memory and peripherals.	Memory. VMs memory space is encrypted and protected with specific encryption keys	Processor and Memory (EPC). Enclave (trusted) and non-enclave (untrusted)
	- Confidentiality & Integrity Protection	✗Confidentiality √Integrity	✗Confidentiality √Integrity	√Confidentiality ✗Integrity	√
	- Protection Against OS/VMM/BIOS/SMM	√Host OS ✗Guest OS √Load Trusted VMM √BIOS ✗SMM	√Trusted OS √VMM •BIOS •SMM	√OS √VMM ✗BIOS ✗SMM	√
	- Protection Against Physical Attacks	✗	✗	√(for memory)	√(except CPU)
	- Protection Against Memory Snooping	√	√	√	√
	- Secure Storage	√	✗proprietary mechanism can be implemented if desired	√	√
	- Secure Boot	√	✗TrustZone itself does not provide hardware roots of trust but can be completed with additional features	√	✗
	- Cache Memory Protection	✗	✗	✗	✗
	- Memory Access Protection	√Implements two chipset mechanisms: DPR and PMRs	√DRAM is partitioned into several memory regions to be used be in normal world or secure world	√Provides strong protection against DRAM interface unauthorized access.	√The PRM content is protected from unauthorized access

velopers to protect their applications themselves against such attacks (Gotzfried et al., 2017).

6. Related work

There are a few works in the literature that compare hardware-based solutions offered by major vendors supporting trusted cloud computing. In this section, we review them and show their differences compared to our approach.

Maene et al. (2018) have made a chronological-style comparison between hardware-based security solutions including both academic and industrial instances. They have compared twelve instances according to seven security criteria with a special focus on architectures that provide isolation and attestation. The compared solutions range from lightweight embedded devices to high-performance server processors, but with much focus on embedded systems rather than server systems as in our work. As CSPs rely only on industrial-scale hardware-based solutions, we proceed differently in our comparison by considering only these solutions. We include AMD SEV security solution that is not addressed in Maene et al. (2018). Moreover, differently from authors of Maene et al. (2018) who have compared academic and industrial solutions with respect to seven security criteria, we adopt more comparison criteria in our work (thirteen security criteria) and compare only hardware-based security solutions available for end-users.

Shepherd et al. (2016) have compared trusted hardware and software technologies from a security perspective with focus on their application to the emerging domains of the Internet of Things (IoT) and Cyber-Physical Systems (CPS) without including AMD technologies.

Mofrad et al. (2018) have done an enhanced comparison study between Intel SGX and AMD Memory Encryption Technology without considering the other technologies such as Intel TXT and ARM TrustZone. Similarly, Ngabonziza et al. (2016) have compared ARM TrustZone with other hardware-based solutions including Intel TXT and Intel SGX.

Table 2 – Summary of the comparison between hardware-based solutions: Intel TXT, ARM TrustZone, AMD SEV, and Intel SGX (Functional and deployment criteria).

	Propriety	Intel TXT	ARM TrustZone	AMD SEV	Intel SGX
Functional Criteria	- TCB Size	Large	Small	Large	Small
	- Debugging	√PCR16 reserved to debug, not used by Intel TXT	√	√	√in debug mode but ✗in release mode
	- Attestation	√	✗	√	√
	- Sealing	√	✗	✗	√
	- Execution Privilege Level	Ring −3 for TPM	Ring −2	Ring 0	Ring 3 for Enclaves
Deployment Criteria	- TEE Features	Verified Launch, Secret Protection and Attestation	TEE security level and trustworthiness is maintained by adopting restrictions where only strictly verified applications can be deployed in the TEE	Appropriate for a broader range of applications, particularly for those that require many system calls	Suitable for applications that require an enhanced-degree of security protection
	- Legacy Application Modification	Not required	Not required	Not required	Required
	- Performance	Does not degrade performance	0.13%, Context switch between secure world and normal world creates an overhead (110μs for REE→TEE and 47μs for TEE→REE)	Performs faster when a protected application requires a large amount of secure memory resources, slowdown of 1× or identical	Performance slowdown: 9.45× up to 19.31×. Integrity mechanism overhead: 10×, cross-boundary function call: 7000+ CPU cycles, 96 MB EPC memory limit
	- VM Migration	√	•	√	✗
	- License Agreement	✗	√	✗	√
	- Ecosystem	Growing	Small and under tests	Small and under tests	Large

Pinto et al have made a comprehensive survey in Pinto and Santos (2019) about ARM TrustZone technology where they compared it with other solutions from academia and industry and presented existing systems in two main areas, namely: TEE and hardware-assisted virtualization. In another way, our comparison includes a deep analysis with more criteria driven by IT managers' requirements.

Finally, Zhang and Zhang (2016) have made a study of hardware-assisted isolation security use cases in execution environments including Intel ME (Management Engine) and SMM that are particularly intended for hardware vendors.

7. Conclusion and future work

Hardware-based security solutions are a promising way toward a trusted cloud computing environment where user data and code are secured from any malicious software. They have been rapidly adopted by CSPs to give more guarantees to customers who worry about their sensitive data.

In this paper, we have presented four industrial-scale hardware-based trust solutions, namely: Intel TXT, ARM TrusZone, AMD SEV, and Intel SGX, that can be used by CSPs in datacenters. According to our comparison with respect to three criteria categories, we have found that all the four solutions can offer security guarantees in some particular settings, and each technology can offer certain security services that the others do not provide to achieve trusted and secure computing in that particular settings.

Our comparison can help cloud customers and IT platforms security managers deciding about which the best suitable hardware-based security solution to adopt for their security requirements and future cloud migrations. It also presents a comprehensive overview and a useful guide for future research in this area.

From our point of view, we believe that Intel SGX is a better choice but still not the best. It provides a strong and robust protection for the client applications on cloud computing platforms, but requires code modification of legacy applications, and puts the responsibility on the application developers to write code that resists to side channel attacks. Moreover, it does not offer sufficient protected memory in its TCB for some memory-demanding use cases.

As a future work, we will make a practical comparison of the four studied solutions by designing and implementing a typical cloud application, and comparing them according to some other criteria such as: time of prototyping, resistance to security threats due to compromised privileged software, costs, etc.

We will also consider open source hardware platforms such as RISC-V and its trust solution called MultiZone for further investigation and comparison with proprietary solutions.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the Network Engineering Laboratory of the Polytechnic Military School (EMP), Algiers. We thank M. Abedennour Amamra, Ph.D. for his precious review for the English usage, grammatical and typographic revision of this paper.

REFERENCES

Aaron Grabein, L. G., 2020. Advanced security features of AMD EPYC processors enable new Google cloud confidential computing portfolio. Web Site.

Advanced Micro Devices. In: White Paper. Enhance your Cloud Security with AMD EPYC™ Hardware Memory Encryption. AMD, Inc; 2018. https://www.amd.com/system/files/documents/cloud-security-epyc-hardware-memory-encryption.pdf. Last accessed August 22, 2019

Advanced Micro Devices, 2018b. Secure Encrypted Virtualization API Version 0.17. Technical Preview, Last accessed August 22, 2019.

Amacher J, Schiavoni V. On the performance of ARM TrustZone. In: Pereira J, Ricci L, editors. In: Distributed Applications and Interoperable Systems. Cham: Springer International Publishing; 2019. p. 133–51.

Andrew Baumann MP, Hunt G. Shielding applications from an untrusted cloud with haven. In: 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI'14). USENIX Association; 2014. p. 267–83.

ARM, 2009. ARM Security Technology Building a Secure System using TrustZone™ Technology. http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf, Last accessed August 22, 2019.

ARM, 2018. Isolation using virtualization in the Secure world. White Paper, secure world software architecture on Armv8.4.

Ashish Nadkarni, F. D., Shane, R., 2019. Processor Security Underpins Secure Datacenter Infrastructure. White Paper, US45487819, IDC.

Asvija B, Eswari R, Bijoy M. Security in hardware assisted virtualization for cloud computing—state of the art issues and challenges. Comput. Netw. 2019;151:68–92.

Atamli-Reineh, A., Martin, A. P., 2017. Securing application with software partitioning: acase study using SGX. Comput. Res. Repos.CoRR abs/1706.03006.

Buhren R, Werling C, Seifert J-P. Insecure until proven updated. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019.

Chakrabarti S, Leslie-Hurd R, Vij M, McKeen F, Rozas C, Caspi D, Alexandrovich I, Anati I. Intel software guard extensions (Intel SGX) architecture for oversubscription of secure memory in a virtualized environment. Proceedings of the Hardware and Architectural Support for Security and Privacy. New York, NY, USA: Association for Computing Machinery, 2017.

Christopherson, S., 2017. Web Site, Last Access: 09/19/2020, https://github.com/intel/kvm-sgx.

Cheruvu S, Kumar A, Smith N, Wheeler DM. Demystifying Internet of Things Security Successful IoT Device/Edge and Platform Security Deployment. Berkeley, CA: Apress; 2020.

Costan, V., Devadas, S., 2016. Intel SGX explained. Cryptology ePrint Archive, Report 2016/086.

Costan V, Lebedev I, Devadas S. Secure processors part I: Background, taxonomy for secure enclaves and Intel SGX architecture. Found. Trends® Electron. Des. Autom. 2017;11(1-2):1–248.

Fan P, Zhao B, Shi Y, Chen Z, Ni M. An improved vTPM-VM live migration protocol. Wuhan Univ. J. Nat. Sci. 2015;20(6):512–20.

Futral W, Greene J. Fundamental Principles of Intel TXT. Berkeley, CA: Apress; 2013. p. 15–36.

Gonzalez J. Operating system support for run-time security with a trusted execution environment -usage control and trusted storage for linux-based systems-. IT University of Copenhagen; 2015. Ph.D Thesis. Advisor: Philippe Bonnet. Submitted: January 31, 2015. Last Revision: May 30, 2015.

Gotzfried J, Eckert M, Schinzel S, Muller T. Cache attacks on Intel SGX. Proceedings of the 10th European Workshop on Systems Security - EuroSec 17. ACM Press, 2017.

Göttel C, Pires R, Rocha I, Vaucher S, Felber P, Pasin M, Schiavoni V. Security, performance and energy trade-offs of hardware-assisted memory protection mechanisms (practical experience report). In: IEEE 37th International Symposium on Reliable Distributed Systems. IEEE Computer Society: IEEE; 2018. p. 133–42.

Greene, J., 2010-2012. Intel Trusted Execution Technology Hardware-based Technology for Enhancing Server Platform Security. White Paper, Last accessed August 22, 2019.

Greene, J., 2013a. Intel Trusted Execution Technology Hardware-based Technology for Enhancing Server Platform Security. White Paper, Intel.

Greene WF. Intel Trusted Execution Technology for Server Platforms. Apress, Berkeley, CA; 2013. p. 89–103. Trusted Computing: Opportunities in Software

Gu J, Hua Z, Xia Y, Chen H, Zang B, Guan H, Li J. Secure live migration of SGX enclaves on untrusted cloud. In: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN); 2017. p. 225–36.

Harnik, D., Tsfadia, E., Chen, D., Kat, R. I., 2018. Securing the storage data path with SGX enclaves. Comput. Res. Repos.CoRR abs/1806.10883.

Gu J, Wu X, Zhu B, Xia Y, Zang B, Guan H, Chen H. Enclavisor: a hardware-software co-design for enclaves on untrusted cloud. IEEE Trans. Comput. 2020:1-1 In press. doi:10.1109/TC.2020.3019704.

Hetzelt F, Buhren R. Security Analysis of Encrypted Virtual Machines. In: 13th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments. ACM; 2017.

Horsch J, Huber M, Wessel S. TransCrypt: transparent main memory encryption using a minimal ARM hypervisor. In: 2017 IEEE Trustcom/BigDataSE/ICESS. IEEE; 2017. p. 152–61.

Intel Corporation, Intel Trusted Execution Technology (Intel TXT): Soft- 984 ware development guide Software development guide Measured Launched 985 Environment Developper's Guide, 2017 Nov.

Intel, 2019. Website, last Access March 28, https://software.intel.com/en-us/documentation/intel-sgx-web-based-training/debugging-enclaves.

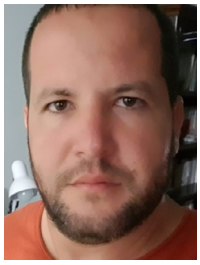
Intel, 2020. Add enhanced security to your applications. Web Site, Last Access: 09/22/2020.

Intel Corporation, 2013. Deploying intel trusted execution technology to enable a trusted private high performance cloud. Web Site, last accessed August 22, 2019.

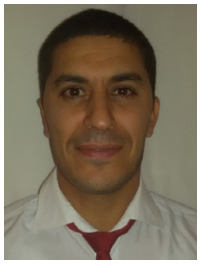
Jabir RM, Khanji SIR, Ahmad LA, Alfandi O, Said H. Analysis of cloud computing attacks and countermeasures. 2016 18th International Conference on Advanced Communication Technology (ICACT). IEEE, 2016.

Jang J, Choi C, Lee J, Kwak N, Lee S, Choi Y, Kang BB. PrivateZone: providing a private execution environment using ARM TrustZone. IEEE Trans. Dependable Secure Comput. 2018;15(5):797–810.
Kaplan, D., Powell, J., Woller, T., 2016. AMD Memory Encryption. White Paper, last accessed August 22, 2019.
Kernel.org, 2017. Last access 09/19/2020, Web Site, https://www.kernel.org/doc/documentation/intel_txt.txt.
Kocher P, Horn J, Fogh A, Genkin D, Gruss D, Haas W, Hamburg M, Lipp M, Mangard S, Prescher T, Schwarz M, Yarom Y. In: 40th IEEE Symposium on Security and Privacy (S&P'19). Spectre attacks: Exploiting speculative execution; 2019.
Kwon D, Seo J, Cho Y, Lee B, Paek Y. PrOS: light-weight privatized se cure OSes in ARM TrustZone. IEEE Trans. Mob. Comput. 2020;19(6):1434–47.
Lesjak C, Hein D, Winter J. Hardware-security technologies for industrial IoT: TrustZone and security controller. IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2015.
Li W, Chen H, Chen H. Research on ARM TrustZone, GetMobile. Mob. Comput. Commun. 2019;22(3):17–22.
Lie D, Maniatis P. Glimmers. Proceedings of the 16th Workshop on Hot Topics in Operating Systems - HotOS 17. ACM Press, 2017.
Lipp M, Schwarz M, Gruss D, Prescher T, Haas W, Fogh A, Horn J, Mangard S, Kocher P, Genkin D, Yarom Y, Hamburg M. In: 27th USENIX Security Symposium (USENIX Security 18). Meltdown: Reading kernel memory from user space; 2018.
Maene P, Gotzfried J, de Clercq R, Muller T, Freiling F, Verbauwhede I. Hardware-based trusted computing architectures for isolation and attestation. IEEE Trans. Comput. 2018;67(3):361–74.
Modi C, Patel D, Borisaniya B, Patel A, Rajarajan M. A survey on security issues and solutions at different layers of cloud computing. J. Supercomput. 2012;63(2):561–92.
Mofrad S, Zhang F, Lu S, Shi W. A comparison study of Intel SGX and AMD memory encryption technology. Proceedings of the 7th International Workshop on Hardware and Architectural Support for Security and Privacy - HASP 18. ACM Press, 2018.
Muhammad Usama Sardar CF, Rasha F. Formal foundations for intel SGX data center attestation primitives. 22nd International Conference on Formal Engineering Methods (ICFEM 2020). Singapore: Springer LNCS and IEEE Press, 2021. Conference delayed because of the COVID-19 pandemic
Mulnix DL. Intel Trusted Execution Technology (Intel TXT) Enabling Guide. second ed. Intel Copr; 2015.
Mutlu, O., Kim, J. S., 2019. Rowhammer: a retrospective. Comput. Sci. arXiv:1904.09724.
Ngabonziza B, Martin D, Bailey A, Cho H, Martin S. TrustZone explained: architectural features and use cases. 2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC). IEEE, 2016.
Ning Z, Liao J, Zhang F, Shi W. Preliminary study of trusted execution environments on heterogeneous edge platforms. In: 2018 IEEE/ACM Symposium on Edge Computing (SEC); 2018. p. 421–6.
Pinto S, Oliveira D, Pereira J, Cardoso N, Ekpanyapong M, Cabral J, Tavares A. Towards a lightweight embedded virtualization architecture exploiting ARM TrustZone. Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA). IEEE, 2014.
Pinto S, Santos N. Demystifying arm TrustZone. ACM Computing Surveys 2019;51(6):1–36.
Pinto S, Pereira J, Gomes T, Tavares A, Cabral J. LTZVisor: TrustZone is the key. Euromicro Conference on Real-Time Systems (ECRTS), 2017.
Sabt M, Achemlal M, Bouabdallah A. The dual-execution-environment approach: analysis and comparative evaluation. In: ICT Systems Security and Privacy Protection. Springer International Publishing; 2015. p. 557–70.
Santos N, Raj H, Saroiu S, Wolman A. Using ARM TrustZone to build a trusted language runtime for mobile applications. In: Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS 14. ACM Press; 2014. p. 67–80.
Sasy, S., 2017. Securing cloud computations with oblivious primitives from Intel sgx. UWSpace.
Schwarz M, Weiser S, Gruss D. Practical enclave malware with Intel SGX. In: Detection of Intrusions and Malware, and Vulnerability Assessment. Springer International Publishing; 2019. p. 177–96.
Schuster F, Costa M, Fournet C, Gkantsidis C, Peinado M, Mainar-Ruiz G, Russinovich M. VC3: Trustworthy data analytics in the cloud using SGX. In: IEEE Symposium on Security and Privacy. IEEE Computer Society; 2015. p. 38–54.
Shepherd C, Arfaoui G, Gurulian I, Lee RP, Markantonakis K, Akram RN, Sauveron D, Conchon E. In: 2016 IEEE Trustcom/BigDataSE/ISPA. Secure and trusted execution: past, present, and future - a critical review in the context of the internet of things and cyber-physical systems. IEEE; 2016.
Singh J. Cyber-attacks in cloud computing: a case study. Int. J. Electron. Inf. Eng. 2014;1(2):78–87.
Smirnov A, Zhidko M, Pan Y, Tsao P, Liu K, Chiueh T. Evaluation of a server-grade software-only ARM hypervisor. In: 2013 IEEE Sixth International Conference on Cloud Computing; 2013. p. 855–62.
Sobchuk J, O'Melia S, Utin D, Khazan R. In: 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA). Leveraging Intel SGX technology to protect security-sensitive applications. IEEE; 2018.
Szefer J, Lee RB. Architectural support for hypervisor-secure virtualization. In: Proceedings of the Seventeenth International Conference on Architectural Support for Programming Languages and Operating Systems. New York, NY, USA: Association for Computing Machinery; 2012. p. 437–50.
Van Bulck J, Minkin M, Weisse O, Genkin D, Kasikci B, Piessens F, Silberstein M, Wenisch TF, Yarom Y, Strackx R. Foreshadow: extracting the keys to the Intel SGX kingdom with transient out-of-order execution. Proceedings of the 27th USENIX Security Symposium. USENIX Association, 2018.
Vasudevan A, Owusu E, Zhou Z, Newsome J, McCune JM. Trustworthy execution on mobile devices: what security properties can my mobile platform give me?. In: Trust and Trustworthy Computing. Springer Berlin Heidelberg; 2012. p. 159–78.
Wang J, Hong Z, Zhang Y, Jin Y. Enabling security-enhanced attestation with Intel SGX for remote terminal and IoT. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 2018;37(1):88–96.
Wendt JD, Guise MJ. In: Tech Report. Trusted computing technologies, intel trusted execution technology; 2011. Last accessed August 22, 2019
Wojtczuk, R., Rutkowska, J., 2009. Attacking intel trusted execution technologyLast accessed August 22, 2019.
Yamada H. Survey on mechanisms for live virtual machine migration and its improvements. Inf. Media Technol. 2016;33:101–15.
Wu Y, Liu Y, Liu R, Chen H, Zang B, Guan H. Comprehensive VM protection against untrusted hypervisor through retrofitted AMD memory encryption. In: IEEE International Symposium on High Performance Computer Architecture. IEEE Computer Society; 2018. p. 441–53.
Zhang F, Zhang H. SoK: a study of using hardware-assisted isolated execution environments for security. Proceedings of the Hardware and Architectural Support for Security and Privacy 2016 on - HASP 2016. ACM Press, 2016.

Zhang N, Sun K, Lou W, Hou YT. In: 2016 IEEE Symposium on Security and Privacy (SP). CaSE: cache-assisted secure execution on ARM processors. IEEE; 2016.
Zoltn dm Mann AM. Optimized cloud deployment of multi-tenant software considering data protection concerns. In: 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE Computer Society; 2017. p. 609–18.



Oualid Demigha is an assistant professor of computer science. He received his Ph.D. degree from Ecole Nationale Superieure d'Informatique at Algiers, Algeria in 2015 with the partnership of Université de Bordeaux, France. He is working as a lecturer-researcher at Ecole Militaire Polytechnique (EMP), Algiers. His research interests include: energy-efficiency in wireless sensor networks, routing protocols in mobile adhoc networks, fault-tolerance in distributed systems, dependability in operating systems, protocol formal verification and operating system security. He participated as a TPC member in CSA (Computing Systems and Applications) conference in 2016 and in 2018 at EMP, Algiers. and in many international conferences: IEEE WCNC 2015, ICC 2015, GC AHSN 2013, IEEE WCNC 2014, MSWiM 2013. He co-edited the proceeding of CSA conference in 2018 entitled "Advances in Computing Systems and Applications" in Lecture Notes in Networks and Systems Series, Volume 50, Springer. He is serving as a regular reviewer in many journals: IEEE Sensors, IEEE Communication Surveys and Tutorials, IEEE Communication Letters, Journal of Information Science, etc. in some of them he published some of his research results.



Ramzi Larguet is an engineer and senior developer. He received his Masters degree in Computer Science in 2019. His main research intrests include: Wireless Sensor Networks, Formal Verification of Security Protocols, Cloud Security and Computer Architectures.