

# Trusted Execution Environments for Cloud/Fog-based Internet of Things Applications

Dalton Cézane Gomes Valadares<sup>1,2</sup><sup>a</sup>, Newton Carlos Will<sup>3</sup><sup>b</sup>, Marco Aurélio Spohn<sup>4</sup><sup>c</sup>, Danilo Freire de Souza Santos<sup>2</sup>, Angelo Perkusich<sup>2</sup><sup>d</sup> and Kyller Costa Gorgonio<sup>2</sup><sup>e</sup>

<sup>1</sup>Federal Institute of Pernambuco, Mechanical Engineering Department, Caruaru, PE, Brazil

<sup>2</sup>Federal University of Campina Grande, Informatics and Electrical Engineering Center, Computer Science, Campina Grande, PB, Brazil

<sup>3</sup>Federal University of Technology - Paraná, Dois Vizinhos, PR, Brazil

<sup>4</sup>Federal University of Fronteira Sul, Chapecó, SC, Brazil

**Keywords:** Trusted Execution Environments, Internet of Things, Intel SGX, ARM TrustZone, Fog Computing, Security.

**Abstract:** Cloud services and fog-based solutions can improve the communication and processing efficiency of the Internet of Things (IoT). Cloud and fog servers offer more processing power to IoT solutions, enabling more complex tasks within reduced time frames, which could not be possible when relying solely on IoT devices. Cloud and fog computing benefits are even better when considering sensitive data processing once IoT devices can hardly perform complex security tasks. To improve data security in cloud/fog-based IoT solutions, Trusted Execution Environments (TEEs) allow the processing of sensitive data and code inside protected and isolated regions of memory. This paper presents a brief survey regarding TEEs' adoption to protect data in cloud/fog-based IoT applications. We focus on solutions based on the two leading TEE technologies currently available in the market (Intel SGX and ARM Trustzone), pointing out some research challenges and directions.

## 1 INTRODUCTION

The Internet of Things (IoT) paradigm has gained power when combined with the cloud or fog computing paradigm. Such a combination alleviates the concerns regarding the IoT devices' typical constraints, such as limited memory and processing capabilities, by allowing applications demanding complex and fast processing to run in fog/cloud servers. Despite these benefits of fog and cloud computing, concerns remain when cloud/fog-based IoT applications deal with sensitive data.

Among the commonly employed solutions to protect data generated by IoT applications, we find the use of Trusted Execution Environments (TEEs). TEE can be defined as a tamper-resistant processing environment, running in a separate kernel, which pro-

vides an adequate level of authenticity, integrity, and confidentiality for the executed code (Sabt et al., 2015). A TEE should supply a remote attestation process, enabling third-parties to prove its trustworthiness. Nonetheless, TEE is not a bullet-proof solution for systems security: an adversary can still explore Side-Channel attacks<sup>1</sup>. Since 2010, Global Platform<sup>2</sup> is responsible for TEE standardization through its TEE System Architecture and API specifications, which comprise TEE Client API, TEE Internal Core API, TEE Secure Element API, among others<sup>3</sup>.


To investigate how TEE intends to protect data in cloud/fog-based IoT applications, we defined the following research questions:


1. What are the current proposals regarding the use of TEE in IoT applications?


<sup>1</sup>Attacks based on particular hardware characteristics, such as timing information, power consumption, electromagnetic leaks, and sound, requiring a good technical knowledge about the internal operation of the system.


<sup>2</sup><http://globalplatform.org/>


<sup>3</sup><https://globalplatform.org/specs-library/?filter-committee=tee>

<sup>a</sup> <https://orcid.org/0000-0003-1709-0404>

<sup>b</sup> <https://orcid.org/0000-0003-2976-4533>

<sup>c</sup> <https://orcid.org/0000-0002-9265-9421>

<sup>d</sup> <https://orcid.org/0000-0002-7377-1258>

<sup>e</sup> <https://orcid.org/0000-0001-9796-1382>

## 2. What sort of IoT solutions is TEE currently in use?

To answer these questions, we performed a basic literature review, searching for related papers in some of the main Computer Science scientific repositories (e.g., Scopus<sup>4</sup>, IEEE Digital Library<sup>5</sup> and ACM Digital Library<sup>6</sup>). For this search, we used the following keywords as search topics: “Internet of Things” AND “Trusted Execution Environment” AND “Security”. We have taken into account only the main TEE technologies available in the market: ARM TrustZone and Intel SGX. We decided to focus on ten selected papers for each TEE technology, which is enough to get an overview of the developed research and get insights/directions to guide future works. Our main contributions are listed as follows:

- A survey on TEE, applied for protecting cloud/fog-based IoT applications, presenting relevant related papers;
- A discussion regarding the challenges in the adoption of TEE for IoT applications and key research directions;
- A starting point to carry out a Systematic Literature Review regarding the research questions.

This work is the first review regarding the use of TEE in IoT applications to the best of our knowledge. The remainder of this paper has the following organization. Section 2 presents the fundamentals of ARM TrustZone and Intel SGX, meanwhile in Section 3 we present works employing Intel SGX in their proposals, while Section 4 focuses on works related to the context of ARM TrustZone. Section 5 presents the main vulnerabilities of the two TEE technologies under consideration. In Section 6, we present relevant challenges and directions to the application of TEE in IoT, and we discuss related works in Section 7. Conclusions are then laid out in Section 8.

## 2 BACKGROUND

This Section presents a brief description of the two leading TEE technologies currently available in the market, Intel Software Guard Extensions and the ARM TrustZone, some differences between both, and common scenarios for IoT applications.

<sup>4</sup><http://www.scopus.com>

<sup>5</sup><http://ieeexplore.ieee.org>

<sup>6</sup><http://portal.acm.org>

## 2.1 Intel Software Guard Extensions (SGX)

Intel Software Guard Extensions (Intel SGX) is an extension to the x86 architecture instruction set that allows applications to run in a protected memory area, called *enclave*, which contains the application code and data. An enclave is a protected area in the application’s address space that guarantees the confidentiality and integrity of the data, preventing this data from being accessed by malware and even other software with high execution privileges, such as VM monitors, BIOS, and the operating system (McKeen et al., 2013; Jain et al., 2016; da Rocha. et al., 2020). We describe an SGX enclave’s attack surface, as shown in Fig. 1.

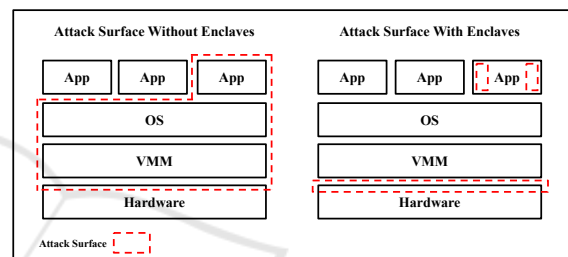


Figure 1: Attack surface of a security-sensitive application without SGX enclaves (left) and with SGX enclaves (right).

Memory encryption employs standard algorithms containing protections against replay attacks. The encryption key is stored in registers inside the CPU, not accessible to external components, and is changed randomly at each hibernation or system restart event (Intel, 2016b). Each enclave has a certificate signed by the enclave author containing information that allows the Intel SGX architecture to detect whether any part of the enclave has been tampered with. However, the hardware only checks the enclave’s measurement when loaded (Intel, 2016a).

Applications can also request a specific key (*sealing key*) to the enclave to protect their keys and data when they want to save them outside the protection of the enclave, such as on disk. Enclaves can also attest to each other, enabling establishing a secure communication channel for sharing sensitive information (Anati et al., 2013).

The main goal of SGX is to reduce the Trusted Computing Base (TCB), allowing only sensitive parts of the application to be within enclaves. Splitting the application into two components brings some advantages, with fewer failure points in the trusted part of the application, resulting in safer software.

## 2.2 ARM TrustZone

ARM TrustZone is a hardware architecture that extends the security aspect to the entire system design, allowing any part of the system to be protected. TrustZone technology provides a basic infrastructure that allows SoC designers to choose a range of components that can assist with specific functions within a secure environment. The architecture's main goal is to enable the construction of a programmable environment that allows the confidentiality and integrity of almost all assets to be protected from specific attacks and can be used to build a set of security solutions that are not possible with traditional methods. (ARM, 2009).

With the ARM TrustZone architecture, the system can be isolated in two logical states: a *secure* world and a *normal* world (Fig. 2). These states are also signaled to all peripheral devices via the system bus, allowing them to make access control decisions based on the system's current state. The mechanism responsible for exchanging context between the two states is called *monitor*.

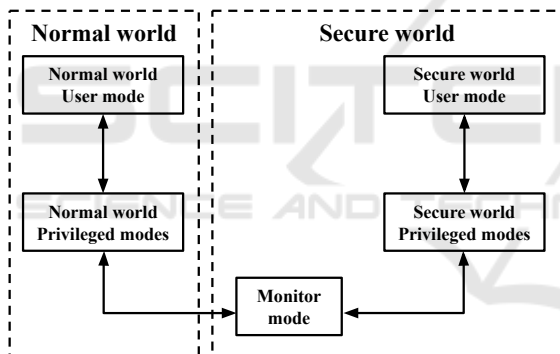


Figure 2: Execution modes in the ARM TrustZone architecture.

When an application runs in a secure world, it can isolate parts of the memory for its use, preventing applications running in an ordinary world from accessing these locations. The memory controller using the TrustZone architecture premises guarantees such isolation, providing access control for memory regions based on the current state. This memory partitioning can be static or programmable at runtime.

Secure world applications can also force specific interrupts or exceptions to be caught only in a secure world, and this control is also in charge of the interrupt controller. The system can also block access to particular devices for applications that are not running in a secure world, ensuring these devices' exclusivity only to secure applications (Lesjak et al., 2015).

## 2.3 Differences between ARM TrustZone and Intel SGX

Table 1 presents a comparison between ARM TrustZone and Intel SGX TEE technologies' main characteristics. We can notice that Intel SGX technology covers the main characteristics necessary for the development of secure applications without the need to trust the operating system or other high privileged components. At the same time, ARM TrustZone can provide a trusted communication path to compatible devices.

Table 1: Comparison between the ARM TrustZone and Intel SGX TEE technologies.

	ARM TrustZone	Intel SGX
Architecture	ARM	x86-64
Secure Storage		✓
Attestation		✓
Memory Isolation	✓	✓
Cryptographic Accelerator	✓	✓
Trusted I/O	✓	

While Intel SGX technology aims to offer a complete solution in terms of CPU and memory components' communication, ARM TrustZone lacks a component capable of offering a trusted code measurement. A device-unique key is the basis of the secure storage and attestation mechanisms. In conjunction with a TPM, or another module capable of providing a unique key and code measurement, it is possible to offer such features.

On the other hand, the Intel SGX technology is focused only on the CPU and the communication with the memory, offering no native feature to enable secure communication with I/O devices, unlike ARM TrustZone. It is necessary to combine Intel SGX with other solutions to enable the before-mentioned secure communication, such as hypervisor-based trusted path architectures (Weiser and Werner, 2017).

## 2.4 Common IoT Scenarios

In general, IoT applications consist of distributed systems involving various devices, systems, and servers. As already mentioned, to deal with the constraints of the known devices, the different IoT scenarios demand cloud, fog, and edge computing paradigms. We show the common IoT scenarios considering these paradigms in Fig. 3.

The main entities are the IoT devices that collect data from the environment. These data can be processed locally on the devices or sent to an edge gateway, a fog server, or a cloud server. Whenever us-

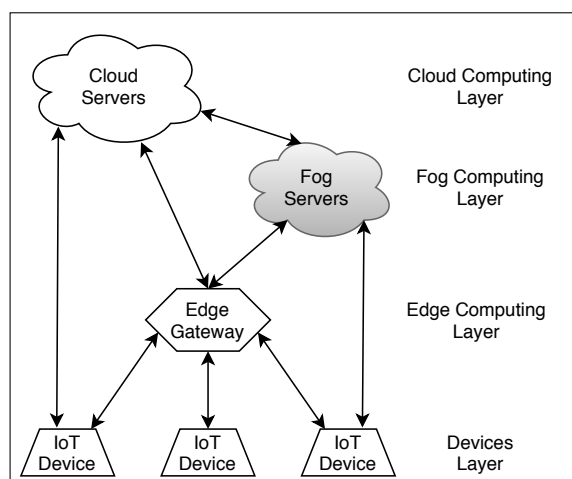


Figure 3: Common Scenarios for IoT Applications.

ing an edge gateway, it can also communicate with a fog or cloud server when it demands more processing, memory, and storage. The fog servers can also exchange data with the cloud servers. The arrows in Fig. 3 represent the communication possibilities between the devices and the edge, fog, and cloud layers.

Knowing these common scenarios, we can think of applying TEE to protect sensitive data in any of the possible combinations. The TrustZone is more suitable to IoT since its architecture is available in many devices based on ARM processors, including micro-controllers. Unlike TrustZone, SGX only is supported by computers running Intel processors. This way, the solutions employing trusted applications to protect data in these IoT scenarios commonly apply TEE to edge gateways or fog/cloud servers.

IoT solutions can combine ARM TrustZone and Intel SGX technologies. The first one is used to ensure trusted operations in IoT and edge devices, and the second one in fog and cloud servers. Cryptography techniques and secure protocols (e.g., Transport Layer Security) are employed to protect data in transit after leaving the devices. If the devices do not have enough capabilities to process such security tasks, they can run in a closer edge gateway. Intel SGX provides mechanisms to perform a remote attestation procedure with third parties, enabling a fog/cloud server to attest IoT/edge devices and creates a trusted communication channel between them.

### 3 INTEL SGX SOLUTIONS

Milutinovic et al. (Milutinovic et al., 2016) presented a blockchain that uses a proof of lucky consensus protocol. Low-latency in the transaction validation, low

energy consumption, and deterministic confirmation time is reached through a random number generation based on a Trusted Execution Environment. The authors applied the Intel SGX capabilities for this and explained the protection provided by this solution.

Liang et al. (Liang et al., 2017) proposed using Intel SGX and blockchain to protect sensitive health data, achieving accountability for data access. A Personal Health Data Management system is proposed, with a user-centric approach, allowing patients to collect and manage their health data. According to the authors, the proposal achieves self-sovereign data ownership, permanent data record with integrity, scalable processing, decentralized and distributed privacy, access control, and trusted accountability.

Sampaio et al. proposed a data dissemination platform. (Sampaio et al., 2017), providing data security and privacy levels. The proposed solution gives full control to data producers over consumers' access; i.e., producers allow or deny the consumers access to sensitive data and set the granularity level. Thus, sensitive data can be produced and repeatedly anonymized or aggregated by trusted entities, using Intel SGX, and then consumed by untrusted applications, ensuring original data privacy. The use of Intel SGX makes the solution more feasible than homomorphic encryption. The authors evaluated the solution, which achieved a lower overhead and can be useful for medium-scale systems with small data dissemination volumes.

SGX is also used to enforce user privacy in location-based services since sharing location traces with untrusted service providers may have privacy implications. Kulkarni et al. (Kulkarni et al., 2017) described an architecture where the user device (IoT, mobile phone, or a GPS enabled device) initiates an attestation process with an enclave that runs in an untrusted provider, establishing a secure channel between the device and the enclave and allowing a secure information exchange. All user data are processed within an enclave, avoiding data leakage to an untrusted cloud. The authors evaluated the solution with a marginal overhead while providing near-to-the-perfect results, describing it as a better solution than currently available, as spatial-cloaking with k-anonymity.

Nguyen et al. (Nguyen et al., 2018) proposed LogSafe, a distributed, scalable, fault-tolerant, and trusted logger for IoT devices data. Using Intel SGX, the proposed logger satisfies confidentiality, integrity, and availability and provides tamper detection, protecting against replay, injection, and eavesdropping attacks. Experiments demonstrated that LogSafe has high scalability, which allows it to work with a high number of IoT devices and a high data transmission



Table 2: Summary of Intel SGX solutions.

Title	Year	Solution
Proof of Luck: an Efficient Blockchain Consensus Protocol (Milutinovic et al., 2016)	2016	Blockchain Consensus Protocol
Towards Decentralized Accountability and Self-Sovereignty in Healthcare Systems (Liang et al., 2017)	2017	Personal Health Data Management system
Secure and Privacy-Aware Data Dissemination for Cloud-based Applications (Sampaio et al., 2017)	2017	Data Aggregation and Dissemination Platform
Privacy-Preserving Location-Based Services by Using Intel SGX (Kulkarni et al., 2017)	2017	Secure Architecture for Location-based Services
LogSafe: Secure and Scalable Data Logger for IoT Devices (Nguyen et al., 2018)	2018	Trusted logger
Decentralized IoT Data Management Using Blockchain and Trusted Execution Environment (Ayoade et al., 2018)	2018	Decentralized Data Management system
BASTION-SGX: Bluetooth and Architectural Support for Trusted I/O on SGX (Peters et al., 2018)	2018	Bluetooth Trusted I/O
Security and privacy aware data aggregation on cloud computing (Silva et al., 2018)	2018	Smart Metering Data Aggregation
Achieving Data Dissemination with Security using FIWARE and Intel Software Guard Extensions (Valadares et al., 2018)	2018	Trusted Architecture
Enabling Security-Enhanced Attestation With Intel SGX for Remote Terminal and IoT (Wang et al., 2018)	2018	Security-enhanced Attestation

rate.

Ayoade et al. (Ayoade et al., 2018) proposed a decentralized system for data management in IoT applications using blockchain and TEE technologies. The idea is to enforce access control by using blockchain smart contracts and storing only data hashes in the blockchain while keeping raw data in a TEE application. The authors implemented the proposal using Ethereum blockchain and Intel SGX, including experiments regarding the processing costs at blockchain and SGX application (gas usage, throughput, and CPU time considering the primary operations). The obtained results demonstrate that the solution has an acceptable efficiency.

Peters et al. (Peters et al., 2018) proposed BASTION-SGX, architectural support for Bluetooth trusted I/O using Intel SGX. This work has the goal of protecting I/O data even when considering adversaries with high-level privileges. Authors described challenges regarding the design and implementation of “trusted I/O”, presenting a possible solution and describing the implementation of a proof-of-concept, which extends the existing Bluetooth security to an SGX enclave, securing the data between it and the Bluetooth controller. The solution includes a secure tunnel between an SGX enclave and the Bluetooth hardware.

Silva et al. (Silva et al., 2018) presented an architecture for data aggregation in cloud computing, considering two approaches for data security and privacy. The proposed architecture comprises four main components: message bus, producers, aggregators, and consumers. Proofs of the concept were implemented and evaluated regarding the response time to process routine operations in smart metering, such as instant energy consumption and monthly bill calculations. Two different aggregators were implemented: one considering the Intel SGX technology and one considering a homomorphic encryption technique. Tests were performed considering the host machine, virtual machines, and containers. The achieved results demonstrate that Intel SGX enables lower response times than the homomorphic encryption technique. The authors also presented advantages and disadvantages for each approach and presented a security anal-

ysis for both.

A trusted architecture using Intel SGX to protect sensitive data in IoT applications was proposed by Valadares et al. (Valadares et al., 2018). The proposal adds security components to a common publish/subscribe architecture, including authentication, authorization, cryptography and trusted processing with a TEE. When compared to a solution without any security mechanism, the authors implemented a prototype and performed experiments to verify the proposal’s time overhead. The results presented low overhead regarding all the communication flow with the security processes and indicated an excellent scalability level.

Wang et al. (Wang et al., 2018) proposed a security-enhanced attestation for remote terminals and IoT devices, suitable to use the “bring your own device” policy in enterprise networks. The solution achieves shielded execution for measurements and attestation, with a small trusted computing base and dynamic attestation based on multiple enclaves. It also provides a policy-based measurement mechanism that enables administrators to collect and monitor the runtime status in a trusted way, ensured by SGX. The evaluated prototype shows a little overhead in the attestation procedure.

In Table 2, we present all these ten papers that used Intel SGX to provide some data security solutions in fog/cloud-based IoT applications.

## 4 ARM TRUSTZONE SOLUTIONS

Yang et al. (Yang et al., 2014) presented Trust-E, a trusted embedded operating system architecture, compliant with Global Platform TEE specifications. The authors designed and implemented the Trust-E solution and implemented a mobile payment application as a demo to test their solution regarding correctness and effectiveness. According to the authors, the results demonstrated that the proposed solution could effectively meet the security requirements.

Lesjak et al. (Lesjak et al., 2015) proposed a security solution for industrial maintenance scenarios,

Table 3: Summary of ARM Trustzone solutions.

Title	Year	Solution
Trust-E: A Trusted Embedded Operating System Based on the ARM Trustzone (Yang et al., 2014)	2014	Trusted Embedded Operating System Architecture
Hardware-security technologies for industrial IoT: TrustZone and security controller (Lesjak et al., 2015)	2015	Device Snapshot Authentication System
CacheKit: Evading Memory Introspection Using Cache Incoherence (Zhang et al., 2016)	2016	Processor Cache Exploitation through a Rootkit
OPTZ: a Hardware Isolation Architecture of Multi-Tasks Based on TrustZone Support (Dai and Chen, 2017)	2017	Multitask Hardware Isolation Architecture
TM-Coin: Trustworthy Management of TCB Measurements in IoT (Park and Kwangjo Kim, 2017)	2017	Trustworthy TCB Measurements Management System
LTZVisor: TrustZone is the Key (Pinto et al., 2017)	2017	Hypervisor to Assist Virtualization
Secure Edge Computing with ARM TrustZone (Pettersen. et al., 2017)	2017	Trusted Edge Computing Platform
A TrustEnclave-Based Architecture for Ensuring Runtime Security in Embedded Terminals (Chang et al., 2017)	2017	Runtime Security for Embedded Terminals
TruApp: A TrustZone-based authenticity detection service for mobile apps (Demesie Yalew et al., 2017)	2017	Mobile App Authenticity and Integrity Checker
Building a Trustworthy Execution Environment to Defeat Exploits from both Cyber Space and Physical Space for ARM (Guan et al., 2019)	2019	Shield System for Legacy Applications

designing and implementing a device snapshot authentication system. This solution was implemented with ARM TrustZone and Security Controller, comparing both technologies. The results indicated that the TrustZone solution presents greater flexibility and performance, while the Security Controller solution presents better protection against physical attacks. The authors concluded that the chosen technology depends on the use case. They proposed a hybrid approach, using both technologies, which maximizes performance and security: employing TrustZone with software components that demand more processing power and Security Controller within software components demanding more security.

Zhang et al. (Zhang et al., 2016) presented a systematic study about a cache incoherence behavior between regular and secure worlds in the ARM TrustZone, and proposed a rootkit called Cachekit to show the feasibility of including malicious code in the processor cache, keeping it hidden. Due to the incoherent state between regular and secure worlds, allowing the rootkit to evade introspection from detection tools was possible. The authors compared the Cachekit with other rootkits regarding detection methods. It proved to be the best, without any detection, since the malicious code completely hides inside the cache.

Dai and Chen (Dai and Chen, 2017) proposed the design and implementation of OPTZ (Open TrustZone), a multitask hardware isolation between regular and secure worlds, with an architecture composed of a secure OS (for the secure world), a standard OS (for the ordinary world), secure services and a communication mechanism. The authors focused on the communication between regular and secure worlds through the secure monitor, considering a system interrupt design and a multitask hardware isolation. They implemented the architecture with the TrustZone technology and carried out experiments to verify its correctness, testing the physical memory access. The results demonstrated the effectiveness of the proposed hardware isolation.

Park and Kim (Park and Kwangjo Kim, 2017) proposed a trustworthy management system for TCB (Trusted Computing Base) measurements from IoT

applications. Authors called the solution TM-Coin, which uses TrustZone and blockchain. They presented the protocol and transactions flow to distribute the TCB measurements in the blockchain securely. TrustZone was used to generate and protect the TM-Coin transactions containing the TCB measurements. The solution applied a remote attestation mechanism and evaluated its performance overhead through experiments carried out with an implemented prototype.

Pinto et al. (Pinto et al., 2017) proposed LTZVisor, a hypervisor that uses TrustZone to assist virtualization. The authors presented the hypervisor architecture and details of its implementation, which was experimentally evaluated considering three metrics: memory footprint, performance overhead, and interrupt latency. The experimental results demonstrated a low-performance overhead, satisfying strict requirements for real-time environment virtualization when running unmodified rich operating systems.

Pettersen et al. (Pettersen. et al., 2017) used both ARM TrustZone and Intel SGX to create a generic platform that enables IoT, mobile, and cloud systems from different vendors to seamlessly connect and integrate into a privacy-preserving and secure manner. The proposed architecture has three vertically stacked layers. The top-most layer has an ARM TrustZone enabled client device. The middle layer is also a client device, in the same administrative domain, with ARM TrustZone or Intel SGX capabilities, such as a fog device or an enterprise cloud server. The third layer is the public cloud, with Intel SGX enabled. The solution achieves a secure design to integrate IoT edge devices to back-end cloud servers with a small overhead.

Chang et al. (Chang et al., 2017) proposed the use of TEE to address runtime security problems efficiently since it uses hardware isolation technology. They presented the TrustZone architecture, explaining its basic working, divided into regular and secure worlds (untrusted and trusted, respectively). They tested a TrustZone-enabled hardware device, evaluating the proposal, which achieved experimental results demonstrating its effectiveness and feasibility.

Yalew et al. (Demesie Yalew et al., 2017) pre-

sented TruApp, which validates the authenticity and integrity of a mobile app by checking some measurements and static/dynamic watermarks, and a verification key issued by the TruApp provider or the app vendor. TruApp is protected since it executes mostly in a TrustZone secure world and verifies the part running in the insecure world. The authors implemented the proposal and carried out experiments, concluding that the measurements are more effective in detecting not authentic apps but incur higher overhead costs. They proposed to analyze means of optimizing the TruApp as future works.

Guan et al. (Guan et al., 2019) proposed TrustShadow, a new system to shield legacy applications running on multiprogramming IoT devices from untrusted OSes. It uses ARM TrustZone technology, securing critical applications through a lightweight runtime system responsible for the communication between the applications and the OS running in the ordinary world. This runtime system does not provide system services. However, it forwards the untrusted ordinary world's requests, verifying the responses and employing a page-based encryption mechanism to protect all the data segments from a security-critical application. Whenever an encrypted page is accessed, the page is decrypted in the internal RAM, immune to physical exploits. It is not necessary any modifications to legacy applications. The authors tested the proposed solution with microbenchmarks and real applications, which presented negligible and, in a few cases, moderate overhead when running real applications.

We list all these ten published papers and their solutions in Table 3, which considered ARM Trustzone to provide or improve data security in IoT applications.

## 5 SGX AND TRUSTZONE VULNERABILITIES

Intel SGX and TrustZone do not consider side-channel or reverse-engineering attacks in their threat model. The Intel Software Guard Extensions Developer Guide (Intel, 2016a) points out that it is up to developers to build enclaves resistant to these types of attacks. Side-channel attacks could generate enough information for the attacker to infer the sequence of running instructions or passive address translation attacks that could give information from the attacker to memory access patterns with page granularity. These threats are classified into four attack vectors by (Wang et al., 2017): power statistics, cache miss statistics, branch timing, and page accesses via page tables. Al-

though the enclave is cryptographically protected in the EPC, in SGX applications, the data inside cache memory are in plain-text. The same occurs for TrustZone applications, even considering that the secure cache lines are not accessible by the untrusted world (ordinary world) (Lesjak et al., 2015; Cerdeira et al., 2020).

Several works in the literature were able to extract sensitive information from enclaves by side-channel, such as a key from RSA processing (Schwarz et al., 2017; Brassier et al., 2017) and AES keys (Moghimi et al., 2017). Spectre vulnerabilities can also be used to infer secrets contained in an SGX enclave (Chen et al., 2019a) or TrustZone trusted applications (Guan et al., 2019). Seeking to mitigate the effects caused by side-channel attacks in applications that use SGX, Shih et al. (Shih et al., 2017) propose T-SGX. The resources provided by Transactional Synchronization Extensions (TSX) are employed to isolate attempts to unauthorized access to the enclave data, eradicating the effects of known side-channel attack techniques.

Weichbrodt et al. (Weichbrodt et al., 2016) also addresses thread synchronization issues in enclaves, using techniques such as use-after-free and time-of-check-to-time-of-use, allowing the attacker to hijack the flow of control or bypass enclave access controls, interrupting threads and forcing segmentation failures in enclaves.

## 6 CHALLENGES AND DIRECTIONS

The use of TEE brings some challenges that one must take into account when developing robust and efficient solutions. However, the Intel SGX architecture provides efficient mechanisms to ensure the security of an application's data. Side-channel attacks or reverse-engineering attacks are not in the architecture's threat model. Therefore, the SGX architecture is vulnerable to Spectre attacks: applications outside the enclave can influence an enclave's code execution prediction. The control flow of the enclave can be manipulated to execute instructions leading to observable cache state changes, which an adversary can use to learn secrets from the memory of the enclave or its registers (Chen et al., 2019b).

Thread synchronization problems in enclaves are also addressed by using techniques such as use-after-free and time-of-check-to-time-of-use, allowing the attacker to hijack the control flow or bypass enclave access controls, interrupting threads, and forcing segmentation failures in enclaves (Weichbrodt et al., 2016). Some challenges to enforcing remote attesta-

tion protocol to build secure and scalable applications with SGX are discussed by (Beekman and Porter, 2017).

The ARM TrustZone architecture also suffers from side-channel attacks, as demonstrated by Lipp et al. (Lipp et al., 2016), where the authors use the ordinary world to monitor activities performed in TrustZone secure world. Besides, there are several security bulletins related to TrustZone, such as bugs in kernel and drivers, and also hardware-related vulnerabilities, which affect different hardware parts of the platform (Pinto and Santos, 2019).

Software developers may assume that TEE is 100% secure, but it is not, and they must consider bugs and vulnerabilities in hardware and software components. Also, they must consider performance issues in both Intel SGX and ARM TrustZone. A broad range of views of the challenges about TEEs is discussed by Ning et al. (Ning et al., 2017).

Regarding the directions for new researches, we could group the critical solutions found in the selected papers for both Intel SGX and ARM TrustZone, as seen in Fig. 4. We noticed that both discussed TEE technologies had been employed to provide security for data aggregation and management systems among the proposed solutions. Considering only the Intel SGX solutions, we identified proposals related to trusted architectures, attestation, and trusted logging and I/O operations systems. When considering only ARM TrustZone solutions, we identified proposals related to authentication and authenticity systems, trusted OS and hypervisor, rootkit, trusted platform for edge computing, and protection for embedded terminals and legacy systems.

Securing the firmware update process of IoT devices is paramount for any IoT ecosystem. One should employ a reliable and secure mechanism for updating the firmware of any IoT device while allowing it to roll back to a previous working instance in case of update failure (resulting from an accidental or intentional error). In this context, Surdu (Surdu, 2018) presents a method based on TEE that isolates the main entities involved in the firmware update process. The new firmware is properly instantiated in a staged TEE region, interfacing with emulated drivers during the upgrade process to avoid any interference with the current working system. Once the new firmware is fully functional, the system transitions to the new configuration; otherwise, it continues to operate with the previous working firmware.

As seen, many topics can be explored, and many solutions can benefit from using TEE. Blockchain is a related topic that was identified among the solutions with both SGX and TrustZone, emphasizing the fol-

lowing aspects:

1. Data are encrypted and stored securely locally or in a protected server (*e.g.*, fog or cloud);
2. The data hashes are stored in the blockchain infrastructure.

To this end, TEE is an option to keep cryptographic keys secure and perform operations on sensitive data (encryption, decryption, and processing). We suggest applying TEE for the most critical portions of the application, which can be explored as a target by interested adversaries without access permission.

## 7 RELATED WORK

Works have been done exploring trust and security issues in the Internet of Things, proposing solutions, and exploring different technologies for these solutions. Based on this potential and broad research area, surveys and reviews were executed to map these trust and security works as expected.

In terms of surveys, Aly et al. (Aly et al., 2019) focused on listing and discussing works related to security threats and challenges in general terms. Other surveys, such as Kouicem et al. (Kouicem et al., 2018), and Di Martino et al. (Di Martino et al., 2018), present more general research and discussion about different aspects of IoT, such as interoperability and architecture, making a parallel about how each cited work relates to trust and security issues. However, none of these surveys present works related to Trusted Execution Environment and its application on Edge/Fog solutions.

Coppolino et al. (Coppolino et al., 2019) presented a survey of hardware-assisted security solutions, focusing on edge computing scenarios. Different types of hardware-assisted security technologies are presented in their work, including the potential use of Trusted Execution Environments. However, it is not presented a specific survey of works that explore TEE technologies in their solutions, including its use in Edge and Internet of Things scenarios.

This way, to the best of our knowledge, currently, there is no published survey, mapping, or review regarding the different uses of TEE for IoT applications.

## 8 CONCLUSION

In this work, we carried out a literature review to gather relevant papers related to the use of TEE in the cloud and fog-based solutions to improve security



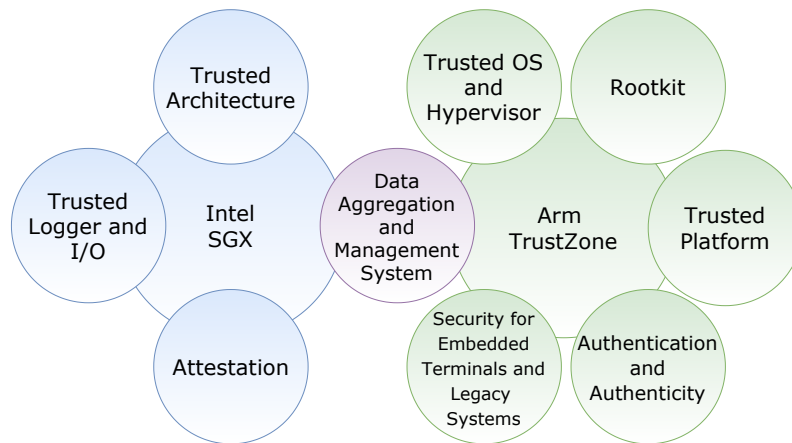


Figure 4: Key solutions found in the selected papers.

for IoT data applications. We summarized the solutions and analyzed possible challenges and directions for future work. For this study, we focused on 20 published papers: 10 TrustZone based and 10 SGX based solutions.

We presented a summary for each selected paper and a discussion about the main challenges related to the use of TEEs. Besides, we also carried on a concise discussion regarding the main research topics addressed by TEEs usage and their improvements: secure and private data processing, secure storage, authentication, virtualization, among others. As future work, we plan a Systematic Literature Review focusing on all the relevant papers published in the top conferences and journals.

## REFERENCES

- Aly, M., Khomh, F., Haoues, M., Quintero, A., and Yacout, S. (2019). Enforcing security in internet of things frameworks: A systematic literature review. *Internet of Things*, page 100050.
- Anati, I., Gueron, S., Johnson, S. P., and Scarlata, V. R. (2013). Innovative technology for CPU based attestation and sealing. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, Tel-Aviv, Israel. ACM.
- ARM (2009). Security technology: Building a secure system using TrustZone technology (white paper). *ARM Limited*.
- Ayoade, G., Karande, V., Khan, L., and Hamlen, K. (2018). Decentralized iot data management using blockchain and trusted execution environment. In *IEEE Intl. Conf. on Information Reuse and Integration*.
- Beekman, J. G. and Porter, D. E. (2017). Challenges for scaling applications across enclaves. In *Proceedings of the 2nd Workshop on System Software for Trusted Execution*, New York, NY, USA. ACM.
- Brasser, F., Müller, U., Dmitrienko, A., Kostianen, K., Capkun, S., and Sadeghi, A.-R. (2017). Software Grand Exposure: SGX cache attacks are practical. In *Proceedings of the 11th USENIX Workshop on Offensive Technologies*, Vancouver, BC, Canada. USENIX Association.
- Cerdeira, D., Santos, N., Fonseca, P., and Pinto, S. (2020). SoK: Understanding the Prevailing Security Vulnerabilities in TrustZone-Assisted TEE Systems. In *Proceedings of IEEE Symposium on Security and Privacy*.
- Chang, R., Jiang, L., Chen, W., Xie, Y., and Lu, Z. (2017). A TrustEnclave-Based Architecture for Ensuring Runtime Security in Embedded Terminals. *Tsinghua Science and Technology*, 22(5).
- Chen, G., Chen, S., Xiao, Y., Zhang, Y., Lin, Z., and Lai, T.-H. (2019a). SgxPectre: Stealing intel secrets from SGX enclaves via speculative execution. In *Euro Sym on Security and Privacy*, Stockholm, Sweden. IEEE.
- Chen, G., Chen, S., Xiao, Y., Zhang, Y., Lin, Z., and Lai, T. H. (2019b). Stealing Intel secrets from SGX enclaves via speculative execution. In *Proc. of the 4th IEEE European Symp. on Security and Privacy*. IEEE.
- Coppolino, L., D'Antonio, S., Mazzeo, G., and Romano, L. (2019). A comprehensive survey of hardware-assisted security: From the edge to the cloud. *Internet of Things*, 6(100,055).
- da Rocha, M., Valadares, D. C. G., Perkusich, A., Gorgonio, K. C., Pagno, R. T., and Will, N. C. (2020). Secure cloud storage with client-side encryption using a trusted execution environment. In *Proceedings of the 10th International Conference on Cloud Computing and Services Science - Volume 1: CLOSER*, pages 31–43. INSTICC, SciTePress.
- Dai, H. and Chen, K. (2017). OPTZ: a Hardware Isolation Architecture of Multi-Tasks Based on TrustZone Support. In *IEEE International Symposium on Parallel and Distributed Processing with Applications and IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, pages 391–395.
- Demesie Yalew, S., Mendonca, P., Maguire, G. Q., Haridi, S., and Correia, M. (2017). Truapp: A trustzone-based authenticity detection service for mobile apps. In *IEEE 13th International Conference on Wireless*

- and Mobile Computing, Networking and Communications.
- Di Martino, B., Rak, M., Ficco, M., Esposito, A., Maisto, S., and Nacchia, S. (2018). Internet of things reference architectures, security and interoperability: A survey. *Internet of Things*, 1:99–112.
- Guan, L., Cao, C., Liu, P., Xing, X., Ge, X., Zhang, S., Yu, M., and Jaeger, T. (2019). Building a trustworthy execution environment to defeat exploits from both cyber space and physical space for arm. *IEEE Transactions on Dependable and Secure Computing*, 16(3).
- Intel (2016a). *Intel Software Guard Extensions Developer Guide*. Intel Corporation.
- Intel (2016b). *Intel Software Guard Extensions SDK for Linux OS Developer Reference*. Intel Corporation.
- Jain, P., Desai, S., Kim, S., Shih, M.-W., Lee, J., Choi, C., Shin, Y., Kim, T., Kang, B. B., and Han, D. (2016). OpenSGX: An open platform for SGX research. In *Proc. of the Network and Distributed System Security Symposium*, San Diego, CA, USA. Internet Society.
- Kouicem, D. E., Bouabdallah, A., and Lakhlef, H. (2018). Internet of things security: A top-down survey. *Computer Networks*, 141.
- Kulkarni, V., Chapuis, B., and Garbinato, B. (2017). Privacy-preserving location-based services by using intel sgx. In *Proceedings of the First International Workshop on Human-Centered Sensing, Networking, and Systems*, New York, NY, USA. ACM.
- Lesjak, C., Hein, D., and Winter, J. (2015). Hardware-security technologies for industrial iot: Trustzone and security controller. In *41st Annual Conference of the IEEE Industrial Electronics Society*.
- Liang, X., Shetty, S., Zhao, J., Bowden, D., Li, D., and Liu, J. (2017). Towards decentralized accountability and self-sovereignty in healthcare systems. In *Proceedings of the International Conference on Information and Communications Security*.
- Lipp, M., Gruss, D., Spreitzer, R., Maurice, C., and Mangard, S. (2016). Armageddon: Cache attacks on mobile devices. In *25th USENIX Security Symposium*, Austin, TX. USENIX Association.
- McKeen, F., Alexandrovich, I., Berenzon, A., Rozas, C. V., Shafi, H., Shanbhogue, V., and Savagaonkar, U. R. (2013). Innovative instructions and software model for isolated execution. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, Tel-Aviv, Israel. ACM.
- Milutinovic, M., He, W., Wu, H., and Kanwal, M. (2016). Proof of luck: An efficient blockchain consensus protocol. In *Proceedings of the 1st Workshop on System Software for Trusted Execution*, SysTEX '16, New York, NY, USA. Association for Computing Machinery.
- Moghimi, A., Irazoqui, G., and Eisenbarth, T. (2017). CacheZoom: How SGX amplifies the power of cache attacks. In *Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems*, pages 69–90, Taipei, Taiwan. Springer.
- Nguyen, H., Ivanov, R., Phan, L. T. X., Sokolsky, O., Weimer, J., and Lee, I. (2018). LogSafe: Secure and Scalable Data Logger for IoT Devices. In *IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 141–152.
- Ning, Z., Zhang, F., Shi, W., and Shi, W. (2017). Position paper: Challenges towards securing hardware-assisted execution environments. In *Proceedings of the Hardware and Architectural Support for Security and Privacy*, New York, NY, USA. ACM.
- Park, J. and Kwangjo Kim (2017). Tm-coin: Trustworthy management of tcb measurements in iot. In *IEEE International Conference on Pervasive Computing and Communications Workshops*.
- Peters, T., Lal, R., Varadarajan, S., Pappachan, P., and Kotz, D. (2018). BASTION-SGX: Bluetooth and Architectural Support for Trusted I/O on SGX. In *Proc. of the Intl Workshop on Hardware and Architectural Support for Security and Privacy*, New York, NY, USA. ACM.
- Pettersen, R., Johansen, H. D., and Johansen, D. (2017). Secure edge computing with arm trustzone. In *Proc. of the 2nd Intl Conference on Internet of Things, Big Data and Security*. INSTICC, SciTePress.
- Pinto, S., Pereira, J., Gomes, T., Tavares, A., and Cabral, J. (2017). Ltzvisor: Trustzone is the key. In *Proceedings of the 29th Euromicro Conference on Real-Time Systems (ECRTS)*.
- Pinto, S. and Santos, N. (2019). Demystifying arm trustzone: A comprehensive survey. *ACM Comput. Surv.*, 51(6).
- Sabt, M., Achemlal, M., and Bouabdallah, A. (2015). Trusted execution environment: What it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 57–64.
- Sampaio, L., Silva, F., Souza, A., Brito, A., and Felber, P. (2017). Secure and privacy-aware data dissemination for cloud-based applications. In *Proceedings of the 10th International Conference on Utility and Cloud Computing*, page 47–56, New York, NY, USA. ACM.
- Schwarz, M., Weiser, S., Gruss, D., Maurice, C., and Mangard, S. (2017). Malware Guard Extension: Using SGX to conceal cache attacks. In *Proceedings of the 14th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Bonn, Alemanha. Springer.
- Shih, M.-W., Lee, S., Kim, T., and Peinado, M. (2017). T-SGX: Eradicating controlled-channel attacks against enclave programs. In *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, USA. Internet Society.
- Silva, L., Barbosa, P., Silva, R., and Brito, A. (2018). Security and privacy aware data aggregation on cloud computing. *Journal of Internet Services and Applications*, 9.
- Surdu, O. (2018). Reliable and secure firmware update for internet of things (iot) devices. US Patent US20180081666A1.
- Valadares, D. C. G., da Silva, M. S. L., Brito, A. E. M., and Salvador, E. M. (2018). Achieving data dissemination with security using FIWARE and Intel Software Guard Extensions (SGX). In *Proc. of the IEEE Symposium on Computers and Communications*, Natal, RN, Brazil.

- Wang, J., Hong, Z., Zhang, Y., and Jin, Y. (2018). Enabling security-enhanced attestation with intel sgx for remote terminal and iot. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(1):88–96.
- Wang, W., Chen, G., Pan, X., Zhang, Y., Wang, X., Bind-schaedler, V., Tang, H., and Gunter, C. A. (2017). Leaky cauldron on the dark land: Understanding memory side-channel hazards in SGX. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 2421–2434, Dallas, Texas, USA. ACM.
- Weichbrodt, N., Kurmus, A., Pietzuch, P., and Kapitza, R. (2016). AsyncShock: Exploiting synchronisation bugs in Intel SGX enclaves. In *Proceedings of the European Symposium on Research in Computer Security*, pages 440–457, Heraklion, Greece. Springer.
- Weiser, S. and Werner, M. (2017). SGXIO: Generic trusted I/O path for Intel SGX. In *Proceedings of the 7th Conference on Data and Application Security and Privacy, CODASPY '17*, page 261–268, Scottsdale, AZ, USA. ACM.
- Yang, X., Shi, P., Tian, B., Zeng, B., and Xiao, W. (2014). Trust-e: A trusted embedded operating system based on the arm trustzone. In *IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing, and 11th Intl Conf on Autonomic and Trusted Computing, and 14th Intl Conf on Scalable Computing and Communications*, pages 495–501.
- Zhang, N., Sun, H., Sun, K., Lou, W., and Hou, Y. T. (2016). Cachekit: Evading memory introspection using cache incoherence. In *IEEE European Symposium on Security and Privacy (EuroSP)*.

SCIENCE AND TECHNOLOGY PUBLICATIONS