Internet of Things 19 (2022) 100543

Contents lists available at ScienceDirect

Internet of Things

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

Open Software and Data

Confidential computing in cloud/fog-based Internet of Things scenarios

Dalton Cézane Gomes Valadares [a,b,*], Newton Carlos Will [c], Marco Aurélio Spohn [d], Danilo Freire de Souza Santos [b], Angelo Perkusich [b], Kyller Costa Gorgônio [b]

[a] *Federal Institute of Pernambuco, Caruaru, PE, Brazil*
[b] *Embedded Systems and Pervasive Computing Laboratory, Federal University of Campina Grande, Campina Grande, PB, Brazil*
[c] *Federal University of Technology - Paraná, Dois Vizinhos, PR, Brazil*
[d] *Federal University of Fronteira Sul, Chapecó, SC, Brazil*

ARTICLE INFO

Keywords:
Confidential computing
Trusted execution environments
Intel SGX
ARM trustZone

ABSTRACT

Internet of Things (IoT) devices are increasingly present in people's daily lives, collecting different types of data about the environment, user behavior, medical data, and others. Due to limited processing power, such devices share the collected data with cloud/fog environments, which raises concerns about users' privacy. To ensure privacy and confidentiality guarantees, many cloud/fog-enhanced IoT applications use Trusted Execution Environments, such as ARM TrustZone and Intel SGX, which are the basis for Confidential Computing. Confidential Computing aims at protecting data during processing, besides transit and rest. This paper presents a review regarding TEEs' adoption to protect data in cloud/fog-based IoT applications, focusing on the two aforementioned technologies. We highlight the challenges in adopting these technologies and discuss the vulnerabilities present in both Intel SGX and ARM TrustZone.

1. Introduction

The Internet of Things (IoT) model has augmented power when combined with the cloud or fog computing paradigm. Such a combination mitigates the IoT devices' typical constraints, such as limited memory and processing capabilities, by enabling applications requiring complex and fast processing to run in fog/cloud servers. Despite these benefits of fog and cloud computing, attention remains when cloud/fog-based IoT applications deal with sensitive data.

Among the commonly applied solutions to protect data produced by IoT applications, we find the use of Trusted Execution Environments (TEEs). TEE can be described as a tamper-resistant processing environment, running in a separate kernel, which affords an adequate level of authenticity, integrity, and confidentiality for the executed code [1]. A TEE should provide a remote attestation process, enabling third parties to prove its trustworthiness. Nevertheless, TEE is not a bullet-proof solution for systems security: an adversary can yet explore Side-Channel attacks.[1] Since 2010, Global Platform[2] has been responsible for TEE standardization by its TEE System Architecture and API specifications, which involve TEE Client API, TEE Internal Core API, TEE Secure Element API, among others.[3]

* Corresponding author at: Federal Institute of Pernambuco, Caruaru, PE, Brazil.
E-mail addresses: dalton.valadares@embedded.ufcg.edu.br (D.C.G. Valadares), will@utfpr.edu.br (N.C. Will), marco.spohn@uffs.edu.br (M.A. Spohn), danilo.santos@virtus.ufcg.edu.br (D.F.d.S. Santos), perkusic@dee.ufcg.edu.br (A. Perkusich), kyller@computacao.ufcg.edu.br (K.C. Gorgônio).

[1] 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.

[2] http://globalplatform.org/.

[3] https://globalplatform.org/specs-library/?filter-committee=tee.

https://doi.org/10.1016/j.iot.2022.100543
Received 3 March 2022; Received in revised form 3 May 2022; Accepted 9 May 2022
Available online 19 May 2022

According to the Confidential Computing Consortium,[4] TEE is the basis for Confidential Computing, which considers protecting data during computation. To investigate how TEE has been applied to protect data in cloud/fog-based IoT applications, we established the following research questions:

1. What are the prevailing proposals concerning the use of TEE in IoT applications?
2. What classes of IoT solutions are currently using TEE?

To elucidate these questions, we conducted a primary literature review, searching for related papers in some of the principal Computer Science scientific repositories (*e.g.*, Scopus,[5] IEEE Digital Library[6] and ACM Digital Library[7]). We used the following keywords for this search: "Internet of Things" AND "Trusted Execution Environment" AND "Security". We have considered 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 acceptable to overview the developed research and get insights/directions to guide future works.

1.1. Contributions

Our main contributions are listed as follows:

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

This paper is an extended version of our previous work entitled *Trusted Execution Environments for Cloud/Fog-based Internet of Things Applications* [2], which has been presented at the *11th International Conference on Cloud Computing and Services Science (CLOSER)*.

1.2. Outline

The remainder of this paper has the following organization. Section 2 presents the fundamentals of Intel SGX and ARM TrustZone. In Section 3, we discuss related works. In Section 4 we present works employing Intel SGX in their proposals, while in Section 5 we focus on works related to the context of ARM TrustZone. Section 6 presents relevant challenges and directions to the application of TEE in IoT and the main vulnerabilities of the two TEE technologies under consideration. We finish this work in Section 7.

2. Background

This Section offers a slight description of the two leading TEE technologies currently available in the market, Intel Software Guard Extensions and the ARM TrustZone, some variations between both, four provider solutions with TEE, and typical scenarios for IoT applications.

2.1. Intel Software Guard Extensions (SGX)

Intel Software Guard Extensions (Intel SGX) is an extension to the x86 architecture instruction set that enables applications to run in a protected memory area, called *enclave*, which includes the application code and data. An enclave is a protected area in the application's address space that ensures the confidentiality and integrity of the data, restricting this data from being accessed by malware and even other software with high execution privileges, such as VM monitors, BIOS, and the operating system [3–5]. We describe an SGX enclave's attack surface, as shown in Fig. 1.

Memory encryption applies standard algorithms carrying protection against replay attacks. The encryption key is stored in registers inside the CPU, not available to external components, and is changed randomly at each hibernation or system restart event [7]. Each enclave has a certificate signed by its author containing information that lets the Intel SGX architecture detect whether any part of the enclave has been tampered with. However, the hardware only checks the enclave's measurement when loaded [8].

Applications can further demand a specific key (*sealing key*) to the enclave to protect their keys and data when they require to save them outside the protection of the enclave, such as on disk. Enclaves can additionally attest to each other, enabling ascertaining a secure communication channel for sharing sensitive information [9].

The main goal of SGX is to lessen the Trusted Computing Base (TCB), allowing barely sensitive parts of the application to be within enclaves. Splitting the application into two elements brings some benefits, with fewer failure points in the trusted part of the application, producing safer software.

[4] https://confidentialcomputing.io/.
[5] http://www.scopus.com.
[6] http://ieeexplore.ieee.org.
[7] http://portal.acm.org.

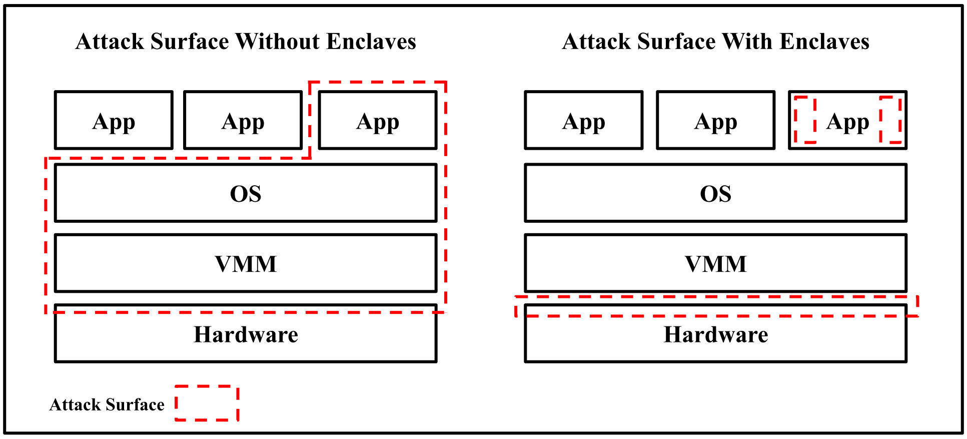

Fig. 1. Attack surface of a security-sensitive application without SGX enclaves (left) and with SGX enclaves (right) [6].

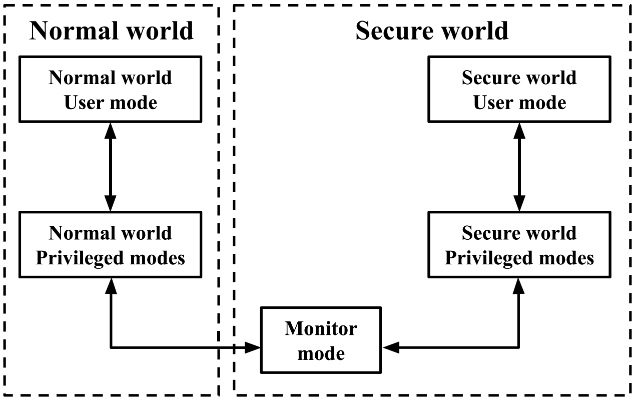

Fig. 2. Execution modes in the ARM TrustZone architecture [10].

2.2. ARM TrustZone

ARM TrustZone is a hardware architecture that extends the security aspect to the entire system design. TrustZone technology provides a basic infrastructure that empowers SoC designers to pick a range of components to assist with specific functions within a secure environment. The architecture's primary purpose is to construct a programmable environment that enables the confidentiality and integrity of almost all assets to be protected from particular attacks, providing a set of security solutions that are not possible with traditional methods [10].

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

When an application runs in a secure world, it can confine parts of the memory for its use, restricting applications running in an ordinary world from reaching these locations. Using the TrustZone architecture premises, the memory controller guarantees isolation, providing access control for memory regions based on the current situation. This memory partitioning can be static or programmable at runtime.

Secure world applications can further require specific interrupts or exceptions to be caught only in a secure world state, and this control is also in charge of the interrupt controller. The system can additionally block access to particular devices for applications that are not running in a secure world, warranting these devices' exclusivity only to secure applications [11].

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 remark that Intel SGX technology covers the main characteristics necessary for developing secure applications without trusting the operating system or other high privileged components. At the same time, ARM TrustZone can render a trusted communication path to compatible devices.

Table 1
Comparison among ARM TrustZone and Intel SGX TEE technologies [2].

	ARM TrustZone	Intel SGX
Architecture	ARM	x86-64
Secure storage		✓
Attestation		✓
Memory isolation	✓	✓
Cryptographic accelerator	✓	✓
Trusted I/O	✓	

While Intel SGX technology proposes a complete solution for communication between CPU and memory components, ARM TrustZone lacks a trusted code measurement capability. A device-unique key is the foundation 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 traits.

On the other hand, the Intel SGX technology focuses on the CPU and the memory, offering no native feature to let secure communication with I/O devices, unlike ARM TrustZone. It is crucial to combine Intel SGX with other solutions to allow the before-mentioned secure communication, such as hypervisor-based trusted path architectures [12].

2.4. Provider solutions

Many cloud providers are already offering confidential computing solutions for their clients. The AWS Nitro Enclaves[8] provide means to create isolated compute environments, which securely process sensitive data by using the Nitro Hypervisor technology. This hypervisor provides virtual environments with CPU and memory isolation. The Nitro Enclaves also includes the attestation mechanism, which can be used to ensure that only authorized code is running. This technology is provided with no additional charges for AWS clients.

The Microsoft cloud services also contain its confidential computing solution: Azure Confidential Computing.[9] The Azure CC also provides TEE capabilities to protect data processing inside the created virtual machines or containers. Its services allow measuring the confidential VM's integrity. One of these services is the Azure Attestation, responsible for verifying the security of a VM. Google's product for confidential computing is named Confidential VM.[10] Besides the security benefits, this solution promises that a client can run existing workloads in a Confidential VM by clicking in one checkbox without making any code change in the applications. The IBM has the Cloud Hyper Protect Virtual Servers,[11] offering a confidential computing environment to protect sensitive data during computation and providing encryption for data in transit and rest.

2.5. Common IoT scenarios

In general, IoT applications consist of distributed systems involving multiple devices and servers. As already mentioned, to deal with the devices' limitations, the diverse IoT scenarios require cloud, fog, and edge computing support. We show the typical IoT scenarios considering those models in Fig. 3.

The main objects are the IoT devices that collect data from the environment. Those data can be processed locally on the devices or sent to an edge gateway, a fog server, or a cloud server. An edge gateway can similarly communicate with a fog or cloud server when the gateway demands more processing, memory, and storage. The fog servers can likewise exchange data with the cloud servers. The arrows in Fig. 3 represent the communication opportunities between the devices and the edge, fog, and cloud layers.

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

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

[8] https://aws.amazon.com/ec2/nitro/nitro-enclaves/.

[9] https://azure.microsoft.com/en-us/solutions/confidential-compute.

[10] https://cloud.google.com/confidential-computing.

[11] https://www.ibm.com/cloud/hyper-protect-virtual-servers.

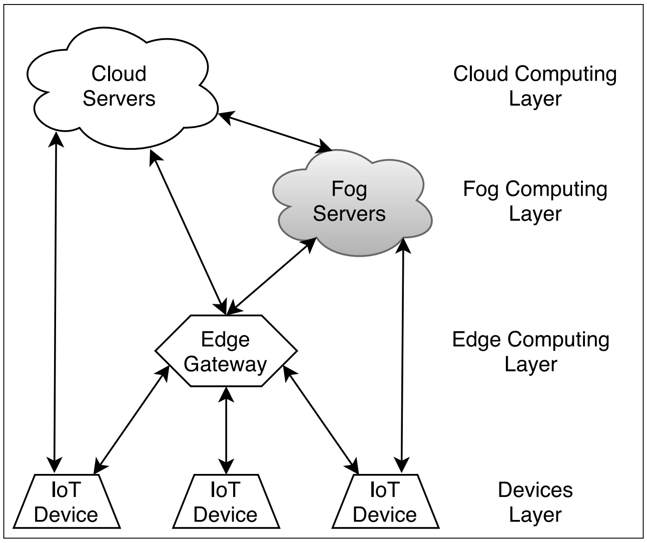

Fig. 3. Common scenarios for IoT applications [2].

3. Related work

There are works exploring trust and security issues in the Internet of Things, offering solutions, and investigating various technologies for these solutions. Surveys and reviews map these trust and security works as expected based on this potential and broad research area.

In terms of surveys, Aly et al. [13] centered on listing and discussing works correlated to security threats and challenges in usual terms. Other surveys, such as Kouicem et al. [14], and Di Martino et al. [15], offer more broad research and discussion about distinct aspects of IoT, such as interoperability and architecture, making a correspondence about how each cited work compares to trust and security issues. However, none of these surveys present works associated with Trusted Execution Environment and its employment for Cloud/Fog-based IoT solutions.

Coppolino et al. [16] surveyed hardware-assisted security solutions, concentrating on edge computing scenarios. Their work presents various types of hardware-assisted security technologies, including the potential use of Trusted Execution Environments. Nevertheless, the survey does not explicitly explore works targeting TEE technologies in their solutions, including their use in Edge and Internet of Things scenarios.

Currently, to the best of our knowledge, Valadares et al. [17] published the only work reviewing the various uses of TEE for IoT applications, which is a systematic literature review. The authors analyzed 58 papers that apply TEE to protect data in cloud/fog-based IoT applications, discussing the main IoT scenarios and solutions, the TEE advantages and disadvantages, and challenges and directions regarding the TEE adoption. A similar work, but only considering the use of Intel SGX for IoT scenarios, is presented by Will et al. [18]. The authors performed a systematic mapping study regarding the Intel SGX employment to protect data in IoT applications. They considered 35 papers for the study, providing an overview of the application scenarios and solutions. Zheng et al. [19] published a comprehensive survey on the use of Intel SGX, but they did not focus on cloud/fog-based IoT applications. The authors analyzed 128 papers, describing the applications and distinct scenarios considered in each one. Besides, they also presented some attack methods and summarized the advantages and disadvantages of using SGX.

4. Intel Software Guard Extensions - SGX

We reviewed thirteen papers that employed Intel SGX to provide some data security solutions in fog/cloud-based IoT applications. Table 2 summarizes these papers, and details of these works are presented in this section.

Milutinovic et al. [20] introduced a blockchain that applies a proof of lucky consensus protocol. It reaches low latency in the transaction validation, reduced energy consumption, and deterministic confirmation time through a random number generation based on a Trusted Execution Environment. The authors used the Intel SGX capabilities to implement the protection provided by the solution.

Liang et al. [21] proposed using Intel SGX and blockchain to protect sensitive health data, attaining accountability for data access. A Personal Health Data Management system is presented, with a user-centric approach, letting patients collect and manage their health data. According to the authors, the proposal delivers self-sovereign data ownership, permanent data record with integrity, scalable processing, decentralized and distributed privacy, access control, and trusted accountability.

Sampaio et al. [22] introduced a data dissemination platform, implementing data security and privacy levels. The intended solution gives complete control to data producers over consumers' access; i.e., producers allow or deny the consumers access to

Table 2
List of selected Intel SGX solutions.

Title	Year	Solution
Proof of luck: an Efficient Blockchain Consensus Protocol [20]	2016	Blockchain consensus protocol
Towards Decentralized Accountability and Self-Sovereignty in Healthcare Systems [21]	2017	Personal health data management system
Secure and Privacy-Aware Data Dissemination for Cloud-Based Applications [22]	2017	Data aggregation and dissemination platform
Privacy-Preserving Location-Based Services by Using Intel SGX [23]	2017	Secure architecture for location-based services
LogSafe: Secure and Scalable Data Logger for IoT Devices [24]	2017	Trusted logger
Decentralized IoT Data Management Using Blockchain and Trusted Execution Environment [25]	2018	Descentralized data management system
BASTION-SGX: Bluetooth and Architectural Support for Trusted I/O on SGX [26]	2018	Bluetooth trusted I/O
Security and Privacy Aware Data Aggregation on Cloud Computing [27]	2018	Smart metering data aggregation
Achieving Data Dissemination with Security using FIWARE and Intel Software Guard Extensions [28]	2018	Trusted architecture
Enabling Security-Enhanced Attestation With Intel SGX for Remote Terminal and IoT [29]	2018	Security-enhanced attestation
Secure Data Processing for IoT Middleware Systems [30]	2019	Secure framework
SecGrid: A Secure and Efficient SGX-Enabled Smart Grid System With Rich Functionalities [31]	2020	Secure smart grid system
Practical Hybrid Confidentiality-Based Analytics Framework with Intel SGX [32]	2021	Secure data analytics framework

sensitive data and establish the granularity level. Therefore, sensitive data can be produced and regularly anonymized or aggregated by trusted entities, using Intel SGX, and then consumed by untrusted applications, guaranteeing original data privacy. The use of Intel SGX makes the solution more fit than homomorphic encryption. The authors evaluated their solution, which achieved a lower overhead, valuable for medium-scale systems with small data dissemination volumes.

SGX also enforces user privacy in location-based services since sharing location traces with untrusted service providers may have privacy implications. Kulkarni et al. [23] specified an architecture where the user device (IoT, mobile phone, or a GPS enabled device) launches an attestation process within an enclave that runs in an untrusted provider, setting a secure channel between the device and the enclave and enabling a secure information exchange. All user data are processed within an enclave, dodging data leakage to an untrusted cloud. The authors evaluated the solution with a marginal overhead while providing near-to-the-perfect results, reporting it as a better solution than currently feasible, as spatial-cloaking with k-anonymity.

Nguyen et al. [24] proposed LogSafe, a distributed, scalable, fault-tolerant, and trusted logger for IoT devices' data. The proposed logger uses Intel SGX to satisfy confidentiality, integrity, and availability and provide tamper detection, protecting against replay, injection, and eavesdropping attacks. Experiments demonstrated that LogSafe has high scalability, allowing it to work with many IoT devices and a high data transmission rate.

Ayoade et al. [25] introduced a decentralized system for data management in IoT applications using blockchain and TEE technologies. The idea is to impose 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 concerning the processing costs at blockchain and SGX application (gas usage, throughput, and CPU time considering the primary operations). The achieved results illustrate that the solution has a satisfactory efficiency.

Peters et al. [26] proposed BASTION-SGX, architectural support for Bluetooth trusted I/O using Intel SGX. This work has the purpose of protecting I/O data even when acknowledging adversaries with high-level privileges. The authors described challenges concerning the design of "trusted I/O", offering a possible solution and clarifying 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 Bluetooth hardware.

Silva et al. [27] introduced an architecture for data aggregation in cloud computing regarding two approaches for data security and privacy. The presented architecture contains four main components: message bus, producers, aggregators, and consumers. Proofs of the concept were implemented and assessed concerning the response time to routine operations in smart metering, such as instant energy consumption and monthly bill estimations. Two different aggregators are available: one employs the Intel SGX technology, while the other uses a homomorphic encryption technique. The authors ran the performance tests in the host machine, virtual machines, and containers. The obtained results show that Intel SGX allows lower response times than the homomorphic encryption technique. The authors also presented advantages and disadvantages for each approach, including a security analysis for both.

A trusted architecture using Intel SGX to protect sensitive data in IoT applications was introduced by Valadares et al. [28]. Their solution includes security components, such as authorization, cryptography, and trusted processing within a TEE in a publish/subscribe architecture. The authors implemented a prototype and performed experiments to verify the proposal's time overhead. The results showed moderate overhead regarding all the communication flow with the security processes, denoting an excellent scalability level.

Table 3
Comparison between the Intel SGX solutions.

Paper	Context	SGX features used		Guarantees provided by the solution			
		Attestation	Sealing	Confidentiality	Integrity	Privacy	Access control
Proof of Luck: an Efficient Blockchain Consensus Protocol [20]	Blockchain	✓			✓		
Towards Decentralized Accountability and Self-Sovereignty in Healthcare Systems [21]	Healthcare	✓	✓	✓	✓	✓	✓
Secure and Privacy-Aware Data Dissemination for Cloud-Based Applications [22]	Smart metering	✓		✓		✓	✓
Privacy-Preserving Location-Based Services by Using Intel SGX [23]	Location-based services	✓	✓	✓	✓	✓	
LogSafe: Secure and Scalable Data Logger for IoT Devices [24]	Data management	✓	✓	✓	✓	✓	
Decentralized IoT Data Management Using Blockchain and Trusted Execution Environment [25]	Data management	✓	✓	✓	✓	✓	✓
BASTION-SGX: Bluetooth and Architectural Support for Trusted I/O on SGX [26]	Trusted I/O	✓		✓	✓		
Security and Privacy Aware Data Aggregation on Cloud Computing [27]	Smart metering	✓		✓	✓	✓	
Achieving Data Dissemination with Security using FIWARE and Intel Software Guard Extensions [28]	Smart metering	✓		✓	✓	✓	✓
Enabling Security-Enhanced Attestation With Intel SGX for Remote Terminal and IoT [29]	Trusted I/O	✓	✓	✓	✓		✓
Secure Data Processing for IoT Middleware Systems [30]	Data management	✓	✓	✓	✓		✓
SecGrid: A Secure and Efficient SGX-Enabled Smart Grid System With Rich Functionalities [31]	Smart metering	✓	✓	✓	✓	✓	
Practical Hybrid Confidentiality-Based Analytics Framework with Intel SGX [32]	Data management	✓		✓			

Wang et al. [29] introduced a security-enhanced attestation for remote terminals and IoT devices, proper 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 allows administrators to gather and monitor the runtime status in a trusted way, ensured by SGX. The evaluated prototype shows a slight overhead in the attestation procedure.

Ayoade et al. [30] employed SGX in the IoT gateway and the cloud service to protect data during processing, transit, and rest, enforcing message integrity with key hashing (key-hashed message authentication code). The proposed framework, SgxIoTGuard, also implements data access policies and separates data processing into trusted and untrusted modules. The authors performed experiments to evaluate the SgxIoTGuard, considering five IoT devices, a mobile phone, and an IoT gateway. The results showed that the SGX does not apply significant overhead to the system.

Li et al. [31] designed, implemented, and evaluated a smart grid system named SecGrid, which uses SGX to protect data processing during operations such as data aggregation and load forecasting. SecGrid considers that the smart meters perform AES encryption. The authors discussed three case studies and presented a security analysis of the system. Besides, they carried out experiments, concluding that SecGrid is much faster than other privacy-preserving schemes for smart grids.

Alabdulatif [32] proposed a confidentiality-based data analytics framework for IoT applications through the use of SGX. The framework combines AES cryptography with an SGX application to ensure end-to-end privacy protection during data communication, processing, and storage. The author performed experiments to analyze performance and accuracy. The results demonstrated a high level of accuracy compared to the insecure version of the application.

Finally, Table 3 presents a comparison of all Intel SGX solutions described in this section regarding the SGX features used and the security guarantees provided by these solutions.

5. ARM TrustZone

We also reviewed thirteen published papers that considered ARM TrustZone to implement or improve data security in IoT applications, analyzing the solution provided. Table 4 presents these papers, which are discussed in this section.

Yang et al. [33] introduced Trust-E, a trusted embedded operating system architecture compliant with Global Platform TEE specifications. The authors devised and implemented the Trust-E solution and a mobile payment application as a demo to test their correctness and effectiveness. According to the authors, the results demonstrate that the proposed solution could adequately meet the security specifications.

Lesjak et al. [11] offered a security solution for industrial maintenance scenarios, designing and implementing a device snapshot authentication system. This solution was implemented with ARM TrustZone and Security Controller, analyzing both technologies. The conclusions indicate that the TrustZone solution grants greater flexibility and performance, while the Security Controller solution

Table 4
List of ARM TrustZone solutions.

Title	Year	Solution
Trust-E: A Trusted Embedded Operating System Based on the ARM TrustZone [33]	2014	Trusted embedded operating system architecture
Hardware-Security Technologies for Industrial IoT: TrustZone and Security Controller [11]	2015	Device snapshot authentication system
CacheKit: Evading Memory Introspection Using Cache Incoherence [34]	2016	Processor cache exploitation through a rootkit
OPTZ: A Hardware Isolation Architecture of Multi-Tasks Based on TrustZone Support [35]	2017	Multitask hardware isolation architecture
TM-Coin: Trustworthy Management of TCB Measurements in IoT [36]	2017	Trustworthy TCB measurements management system
LTZVisor: TrustZone is the Key [37]	2017	Hypervisor to assist virtualization
Secure Edge Computing with ARM TrustZone [38]	2017	Trusted edge computing platform
A TrustEnclave-Based Architecture for Ensuring Runtime Security in Embedded Terminals [39]	2017	Runtime security for embedded terminals
TruApp: A TrustZone-Based Authenticity Detection Service for Mobile Apps [40]	2017	Mobile app authenticity and integrity checker
An Effective Authentication for Client Application Using ARM TrustZone [41]	2017	Authentication system
Building a Trustworthy Execution Environment to Defeat Exploits from both Cyber Space and Physical Space for ARM [42]	2019	Shield system for legacy applications
StreamBox-TZ: Secure Stream Analytics at the Edge with TrustZone [43]	2019	Secure stream analytics
MQT-TZ: Hardening IoT Brokers Using ARM TrustZone [44]	2020	Secure MQTT broker

offers better protection against physical attacks. The authors conclude that the adopted technology depends on the use case. They proposed a hybrid strategy, employing both technologies, which maximizes performance and security: TrustZone with software components that demand more processing power and Security Controller within software components demanding more security.

Zhang et al. [34] produced a systematic study about a cache incoherence behavior between regular and secure worlds in the ARM TrustZone. They propose a rootkit called CacheKit to show the feasibility of including malicious code in the processor cache and keeping it hidden. The incoherent state between regular and secure worlds enables the rootkit to evade introspection from detection tools. The authors compare the CacheKit with other rootkits concerning detection methods. It proved to be the best since the malicious code completely hides inside the cache without any detection.

Dai and Chen [35] introduced the design and implementation of OPTZ (Open TrustZone), a multitask hardware isolation between regular and secure worlds, with an architecture comprised of a secure OS (for the secure world), a standard OS (for the ordinary world), secure services and a communication mechanism. The authors centered on the communication between regular and secure worlds through the secure monitor, viewing a system interrupt design, and multitasking hardware isolation. They implemented the architecture with the TrustZone technology and conducted experiments to verify its correctness, testing the physical memory access. The results displayed the effectiveness of the proposed hardware isolation.

Park and Kim [36] introduced a trustworthy management system for TCB (Trusted Computing Base) measurements from IoT applications. Authors called their solution TM-Coin, which uses TrustZone and blockchain. They showed the protocol and transactions flow to distribute the TCB measurements in the blockchain securely. TrustZone was used to create and protect the TM-Coin transactions holding the TCB measurements. The solution utilized a remote attestation mechanism and assessed its performance overhead through experiments with an implemented prototype.

Pinto et al. [37] offered LTZVisor, a hypervisor that uses TrustZone to assist virtualization. The authors presented the hypervisor architecture and its implementation details, evaluating them using three metrics: memory footprint, performance overhead, and interrupt latency. The experimental results confirmed a low-performance overhead, providing strict requirements for real-time environment virtualization when running unmodified rich operating systems.

Pettersen et al. [38] applied both ARM TrustZone and Intel SGX to create a generic platform that enables IoT, mobile, and cloud systems from diverse vendors to seamlessly connect and integrate into a privacy-preserving and secure method. The introduced architecture has three vertically stacked layers. The top-most layer has an ARM TrustZone enabled client device. In the same administrative domain, the middle layer is also a client device 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 delivers a secure design to integrate IoT edge devices to back-end cloud servers with a small overhead.

Chang et al. [39] introduced the use of TEE to approach runtime security problems efficiently since it uses hardware isolation technology. They presented the TrustZone architecture, emphasizing its basic working, divided into regular and secure worlds (untrusted and trusted, respectively). They examined a TrustZone-enabled hardware device, assessing the proposal, which achieved experimental results displaying its effectiveness and feasibility.

Yalew et al. [40] introduced TruApp, which verifies the authenticity and integrity of a mobile app by examining 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 primarily in a TrustZone secure world and verifies the part running in the insecure world. The authors implemented the proposal and carried out experiments, reasoning that the measurements are more effective in detecting not authentic apps while incurring higher overhead costs. They proposed to investigate means of optimizing the TruApp as future works.

Jiang et al. [41] presented a client application authentication scheme using TrustZone. To implement the proposal, the authors used QEMU and OP-TEE. They implemented a trusted application responsible for protecting the user credentials and performing the authentication process. The authors carried out experiments with the implemented system, concluding that it can detect changes in the application, avoid data leakage, and prevent DoS attacks.

Guan et al. [42] proposed TrustShadow, a new system to shield legacy applications running on multiprogramming IoT devices from untrusted OSes. It uses ARM TrustZone technology, guarding critical applications by a lightweight runtime system accountable for the communication between the applications and the OS running in the ordinary world. This runtime system does not implement system services. However, it forwards the untrusted ordinary world's requests, checking the responses and applying a page-based encryption mechanism to protect all the data segments from a security-critical application. The page is decrypted in the internal RAM whenever an encrypted page is accessed, resistant to physical exploits. It is not required any modifications to legacy applications. The authors examined the proposed solution with microbenchmarks and real applications, which showed negligible or moderate overhead when running real applications.

Park et al. [43] proposed StreamBox-TZ, a secure stream analytics system that uses TrustZone to protect data processing in the edge. The StreamBox-TZ supports remote attestation and presents strong data security with good performance. According to the experiments performed, the overhead added by the security mechanisms is lower than 25%. The proposal adds less than 50KB of executable code to the trusted computing base (TCB), only 16% of the entire TCB.

MQTT [45] is likely the most employed communication protocol in IoT application development. Clients communicate through a broker that relays published topic messages to their corresponding subscribers following the publish/subscribe (P/S) approach. MQTT supports synchronous as well as asynchronous communication, complying with most IoT application requirements. Even though MQTT can secure communication through Transport Layer Security (TLS) and Secure Sockets Layer (SSL), MQTT servers process data in the clear, leaving plenty of room for security concerns. Segarra et al. [44] propose Mqt-Tz as a secure MQTT *mosquitto* broker based on TrustZone acting as a trusted proxy with mutual TLS established handshake and a two-layer end-to-end encryption scheme. The results show that the TEE broker performance can improve as new TrustZone-specific cryptographic hardware accelerators are available (e.g., Arm CryptoCell family). Since *mosquitto* has a small CPU footprint and it follows a single core running pattern, it is well suited for TrustZone single-threaded model. However, it raises a scalability concern.

Finally, Table 5 presents a comparison of all ARM TrustZone solutions described in this section regarding the TrustZone features used and the security guarantees provided by these solutions.

6. Challenges and vulnerabilities

This section discusses challenges in adopting Intel SGX and ARM TrustZone in cloud/fog-based IoT applications and presents the vulnerabilities to which these technologies are susceptible.

6.1. Challenges

The use of TEE brings some challenges that one must consider when developing robust and efficient solutions. Nonetheless, the Intel SGX architecture implements efficient mechanisms to guarantee the security of an application's data. Side-channel attacks or reverse-engineering attacks are not in the architecture's threat model. Consequently, the SGX architecture is vulnerable to Spectre attacks: applications outside the enclave can influence an enclave's code execution prediction. One could manipulate the control flow of the enclave to execute instructions leading to observable cache state changes, which an adversary can use to uncover secrets from the memory of the enclave or its registers [46].

Techniques such as use-after-free and time-of-check-to-time-of-use can be employed to address thread synchronization problems in enclaves, allowing the attacker to hijack the control flow or bypass enclave access controls, interrupting threads, and forcing segmentation failures in enclaves [47]. Beekman and Porter [48] present some difficulties in needing remote attestation protocol to build secure and scalable applications with SGX.

The ARM TrustZone architecture likewise suffers from side-channel attacks, as demonstrated by Lipp et al. [49], where the authors use the ordinary world to monitor activities performed in TrustZone secure world. Additionally, there are several security bulletins associated with TrustZone, such as bugs in kernel and drivers and also hardware-related vulnerabilities, which concern different hardware parts of the platform [50].

Software developers may suppose TEE is 100% secure, but it is not, and they must acknowledge possible bugs and vulnerabilities in hardware and software components. Also, they must recognize 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. [51].

Concerning 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 saw that both discussed TEE technologies are employed to provide security for data aggregation and management systems among the proposed solutions. Regarding only the Intel SGX solutions, we identified proposals associated with trusted architectures, attestation, and trusted logging and I/O operations systems. When examining only ARM TrustZone solutions, we distinguished proposals associated with authentication and authenticity systems, trusted OS and hypervisor, rootkit, trusted platform for edge computing, and protection for embedded terminals and legacy systems.

Table 5
Comparison between the ARM TrustZone solutions.

Paper	Context	TrustZone features used		Guarantees Provided by the solution			
		Trusted I/O	Memory isolation	Confidentiality	Integrity	Privacy	Authentication
Trust-E: A Trusted Embedded Operating System Based on the ARM TrustZone [33]	Trusted OS	✓	✓	✓	✓		
Hardware-Security Technologies for Industrial IoT: TrustZone and Security Controller [11]	Industrial IoT	✓	✓				✓
CacheKit: Evading Memory Introspection Using Cache Incoherence [34]	Rootkit		✓				
OPTZ: A Hardware Isolation Architecture of Multi-Tasks Based on TrustZone Support [35]	Hardware isolation	✓	✓		✓		
TM-Coin: Trustworthy Management of TCB Measurements in IoT [36]	Trusted Measurements		✓	✓	✓		✓
LTZVisor: TrustZone is the Key [37]	Hypervisor		✓		✓		
Secure Edge Computing with ARM TrustZone [38]	Cloud integration		✓	✓		✓	
A TrustEnclave-Based Architecture for Ensuring Runtime Security in Embedded Terminals [39]	Secure runtime		✓	✓	✓		
TruApp: A TrustZone-Based Authenticity Detection Service for Mobile Apps [40]	Trusted measurements		✓		✓		✓
An Effective Authentication for Client Application Using ARM TrustZone [41]	Authentication system		✓		✓		✓
Building a Trustworthy Execution Environment to Defeat Exploits from both Cyber Space and Physical Space for ARM [42]	Secure legacy apps		✓	✓	✓		
StreamBox-TZ: Secure Stream Analytics at the Edge with TrustZone [43]	Data management		✓	✓	✓		
MQT-TZ: Hardening IoT Brokers Using ARM TrustZone [44]	Industrial IoT		✓	✓			✓

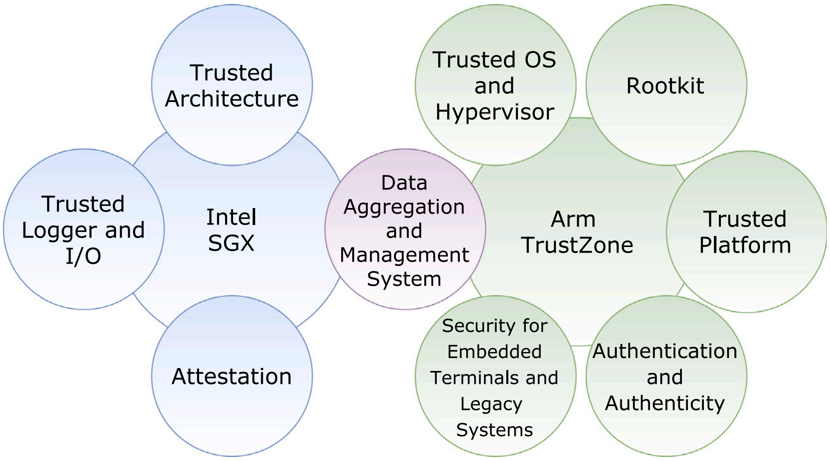
Trusted Architecture
Trusted Logger and I/O
Intel SGX
Attestation
Data Aggregation and Management System
Trusted OS and Hypervisor
Rootkit
Arm TrustZone
Trusted Platform
Security for Embedded Terminals and Legacy Systems
Authentication and Authenticity

Fig. 4. Key solutions found in the selected papers [2].

Securing the firmware update process of IoT devices is paramount for any IoT ecosystem. One should apply a reliable and secure mechanism for updating the firmware of any IoT device while supporting it to roll back to a previous working instance in case of update failure (rising from an accidental or intentional error). In this context, Surdu [52] displays a method based on TEE that confines the main entities involved in the firmware update process. The new firmware is fitly instantiated in a staged TEE region, interfacing with emulated drivers through the upgrade process to avoid any interference with the current working system. Once the new firmware is thoroughly functional, the system shifts to the new configuration; oppositely, it continues to operate with the last working firmware.

As seen, many topics can be explored, and many solutions can benefit from using TEE. Blockchain is a relevant topic that was identified among the solutions with both SGX and TrustZone, highlighting the following 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 opportunity to keep cryptographic keys secure and execute operations on sensitive data (encryption, decryption, and processing). We advise applying TEE for the most critical portions of the application, which can be searched as a target by interested foes without access permission.

6.2. Vulnerabilities

Intel SGX and TrustZone do not acknowledge side-channel or reverse-engineering attacks in their threat model. The Intel Software Guard Extensions Developer Guide [8] points out that it is up to developers to build enclaves immune to these types of attacks. Side-channel attacks could generate adequate 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 categorized into four attack vectors by [53]: power statistics, cache miss statistics, branch timing, and page accesses via page tables. Although 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 recognizing that the secure cache lines are not accessible by the untrusted world (ordinary world) [11,54].

Various works in the literature were capable of extracting sensitive information from enclaves by side-channel, such as a key from RSA processing [55,56], and AES keys [57]. Spectre vulnerabilities can also be used to infer secrets enclosed in an SGX enclave [46] or TrustZone trusted applications [42]. Attempting to relieve the effects caused by side-channel attacks in applications that use SGX, Shih et al. [58] propose T-SGX. The resources provided by Transactional Synchronization Extensions (TSX) are applied to isolate efforts to unauthorized access to the enclave data, annihilating the effects of known side-channel attack techniques.

Weichbrodt et al. [47] address thread synchronization issues in enclaves. They employ techniques such as use-after-free and time-of-check-to-time-of-use, which allows the attacker to hijack the control flow or bypass enclave access controls, interrupting threads and forcing segmentation failures in enclaves.

7. Conclusion

In this work, we carried out a literature review to gather relevant papers related to confidential computing in IoT scenarios. For this, we considered the use of TEE in the cloud and fog-based solutions to improve security for IoT data applications. We summarized the solutions and analyzed possible challenges and directions for future work. We focused on 26 published papers for this study: 13 TrustZone-based and 13 SGX-based proposals.

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 suggest performing a Systematic Literature Review focusing on all the relevant papers published in the top conferences and journals considering confidential computing in fog/edge scenarios, not only considering IoT applications.

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.

References

[1] M. Sabt, M. Achemlal, A. Bouabdallah, Trusted execution environment: What it is, and what it is not, in: Proceedings of the 14th International Conference on Trust, Security and Privacy in Computing and Communications, IEEE, Helsinki, Finland, 2015, pp. 57–64, http://dx.doi.org/10.1109/Trustcom.2015.357.

[2] D.C.G. Valadares, N.C. Will, M.A. Spohn, D.F.d.S. Santos, A. Perkusich, K.C. Gorgonio, Trusted execution environments for cloud/fog-based Internet of Things applications, in: Proceedings of the 11th International Conference on Cloud Computing and Services Science, SciTePress, Prague, Czech Republic, 2021, pp. 111–121, http://dx.doi.org/10.5220/0010480701110121.

[3] F. McKeen, I. Alexandrovich, A. Berenzon, C.V. Rozas, H. Shafi, V. Shanbhogue, U.R. Savagaonkar, Innovative instructions and software model for isolated execution, in: Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy, ACM, Tel-Aviv, Israel, 2013, pp. 1–8, http://dx.doi.org/10.1145/2487726.2488368.

[4] P. Jain, S. Desai, S. Kim, M.-W. Shih, J. Lee, C. Choi, Y. Shin, T. Kim, B.B. Kang, D. Han, Opensgx: an open platform for sgx research, in: Proceedings of the Network and Distributed System Security Symposium, Internet Society, San Diego, CA, USA, 2016, pp. 1–16, http://dx.doi.org/10.14722/ndss.2016.23011.

[5] M. da Rocha, D.C.G. Valadares, A. Perkusich, K.C. Gorgonio, R.T. Pagno, N.C. Will, 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, SciTePress, Prague, Czech Republic, 2020, pp. 31–43, http://dx.doi.org/10.5220/0009130600310043.

[6] J. Sobchuk, S. O'Melia, D. Utin, R. Khazan, Leveraging Intel SGX technology to protect security-sensitive applications, in: Proceedings of the 17th International Symposium on Network Computing and Applications, IEEE, Cambridge, MA, USA, 2018, pp. 1–5, http://dx.doi.org/10.1109/NCA.2018.8548184.

[7] Intel, Intel Software Guard Extensions SDK for Linux OS Developer Reference, Intel Corporation, 2016, https://download.01.org/intel-sgx/sgx-linux/2.8/docs/Intel_SGX_Developer_Reference_Linux_2.8_Open_Source.pdf.

[8] Intel, Intel Software Guard Extensions Developer Guide, Intel Corporation, 2016, https://download.01.org/intel-sgx/sgx-linux/2.8/docs/Intel_SGX_Developer_Guide.pdf.

[9] I. Anati, S. Gueron, S.P. Johnson, V.R. Scarlata, Innovative technology for CPU based attestation and sealing, in: Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy, ACM, Tel-Aviv, Israel, 2013, pp. 1–7, https://www.intel.com/content/dam/develop/external/us/en/documents/hasp-2013-innovative-technology-for-attestation-and-sealing-413939.pdf.

[10] ARM, ARM security technology: building a secure system using TrustZone technology (white paper), ARM Limited (2009) https://documentation-service.arm.com/static/5f212796500e883ab8e74531.

[11] C. Lesjak, D. Hein, J. Winter, Hardware-security technologies for industrial IoT: TrustZone and security controller, in: Proceedings of the 41st Annual Conference of the IEEE Industrial Electronics Society, IEEE, Yokohama, Japan, 2015, pp. 2589–2595, http://dx.doi.org/10.1109/IECON.2015.7392493.

[12] S. Weiser, M. Werner, SGXIO: generic trusted I/O path for Intel SGX, in: Proceedings of the 7th Conference on Data and Application Security and Privacy, ACM, Scottsdale, AZ, USA, 2017, pp. 261–268, http://dx.doi.org/10.1145/3029806.3029822.

[13] M. Aly, F. Khomh, M. Haoues, A. Quintero, S. Yacout, Enforcing security in Internet of Things frameworks: A systematic literature review, Internet Things 6 (100050) (2019) 1–24, http://dx.doi.org/10.1016/j.iot.2019.100050.

[14] D.E. Kouicem, A. Bouabdallah, H. Lakhlef, Internet of Things security: A top-down survey, Comput. Netw. 141 (2018) 199–221, http://dx.doi.org/10.1016/j.comnet.2018.03.012.

[15] B. Di Martino, M. Rak, M. Ficco, A. Esposito, S. Maisto, S. Nacchia, Internet of Things reference architectures, security and interoperability: A survey, Internet Things 1 (2018) 99–112, http://dx.doi.org/10.1016/j.iot.2018.08.008.

[16] L. Coppolino, S. D'Antonio, G. Mazzeo, L. Romano, A comprehensive survey of hardware-assisted security: From the edge to the cloud, Internet Things 6 (100055) (2019) 1–17, http://dx.doi.org/10.1016/j.iot.2019.100055.

[17] D.C.G. Valadares, N.C. Will, J. Caminha, M.B. Perkusich, A. Perkusich, K.C. Gorgônio, Systematic literature review on the use of trusted execution environments to protect cloud/fog-based Internet of Things applications, IEEE Access 9 (2021) 80953–80969, http://dx.doi.org/10.1109/ACCESS.2021.3085524.

[18] N.C. Will, D.C. Gomes Valadares, D.F. De Souza Santos, A. Perkusich, Intel software guard extensions in Internet of Things scenarios: A systematic mapping study, in: Proceedings of the 8th International Conference on Future Internet of Things and Cloud, IEEE, Rome, Italy, 2021, pp. 342–349, http://dx.doi.org/10.1109/FiCloud49777.2021.00056.

[19] W. Zheng, Y. Wu, X. Wu, C. Feng, Y. Sui, X. Luo, Y. Zhou, A survey of Intel SGX and its applications, Front. Comput. Sci. 15 (2021) http://dx.doi.org/10.1007/s11704-019-9096-y.

[20] M. Milutinovic, W. He, H. Wu, M. Kanwal, Proof of luck: An efficient blockchain consensus protocol, in: Proceedings of the 1st Workshop on System Software for Trusted Execution, ACM, Trento, Italy, 2016, pp. 1–6, http://dx.doi.org/10.1145/3007788.3007790.

[21] X. Liang, S. Shetty, J. Zhao, D. Bowden, D. Li, J. Liu, Towards decentralized accountability and self-sovereignty in healthcare systems, in: Proceedings of the 19th International Conference on Information and Communications Security, Springer, Beijing, China, 2017, pp. 387–398, http://dx.doi.org/10.1007/978-3-319-89500-0_34.

[22] L. Sampaio, F. Silva, A. Souza, A. Brito, P. Felber, Secure and privacy-aware data dissemination for cloud-based applications, in: Proceedings of the 10th International Conference on Utility and Cloud Computing, ACM, Austin, TX, USA, 2017, pp. 47–56, http://dx.doi.org/10.1145/3147213.3147230.

[23] V. Kulkarni, B. Chapuis, B. Garbinato, Privacy-preserving location-based services by using Intel SGX, in: Proceedings of the 1st International Workshop on Human-Centered Sensing, Networking, and Systems, ACM, Delft, Netherlands, 2017, pp. 13–18, http://dx.doi.org/10.1145/3144730.3144739.

[24] H. Nguyen, R. Ivanov, L.T.X. Phan, O. Sokolsky, J. Weimer, I. Lee, LogSafe: secure and scalable data logger for iot devices, in: Proceedings of the 3rd International Conference on Internet-of-Things Design and Implementation, IEEE, Orlando, FL, USA, 2018, pp. 141–152, http://dx.doi.org/10.1109/IoTDI.2018.00023.

[25] G. Ayoade, V. Karande, L. Khan, K. Hamlen, Decentralized IoT data management using blockchain and trusted execution environment, in: Proceedings of the 19th International Conference on Information Reuse and Integration, IEEE, Salt Lake City, UT, USA, 2018, pp. 15–22, http://dx.doi.org/10.1109/IRI.2018.00011.

[26] T. Peters, R. Lal, S. Varadarajan, P. Pappachan, D. Kotz, BASTION-SGX: bluetooth and architectural support for trusted I/O on SGX, in: Proceedings of the 7th International Workshop on Hardware and Architectural Support for Security and Privacy, ACM, Los Angeles, CA, USA, 2018, pp. 1–9, http://dx.doi.org/10.1145/3214292.3214295.

[27] L. Silva, P. Barbosa, R. Silva, A. Brito, Security and privacy aware data aggregation on cloud computing, J. Internet Serv. Appl. 9 (1) (2018) 1–13, http://dx.doi.org/10.1186/s13174-018-0078-3.

[28] D.C.G. Valadares, M.S.L. da Silva, A.E.M. Brito, E.M. Salvador, Achieving data dissemination with security using FIWARE and intel software guard extensions (SGX), in: Proceedings of the 23rd Symposium on Computers and Communications, IEEE, Natal, RN, Brazil, 2018, pp. 1–7, http://dx.doi.org/10.1109/ISCC.2018.8538590.

[29] J. Wang, Z. Hong, Y. Zhang, Y. Jin, 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) (2018) 88–96, http://dx.doi.org/10.1109/TCAD.2017.2750067.

[30] G. Ayoade, A. El-Ghamry, V. Karande, L. Khan, M. AlRahmawy, M.Z. Rashad, Secure data processing for IoT middleware systems, J. Supercomput. 75 (8) (2019) 4684–4709, http://dx.doi.org/10.1007/s11227-018-2686-x.

[31] S. Li, K. Xue, D.S.L. Wei, H. Yue, N. Yu, P. Hong, SecGrid: A secure and efficient SGX-enabled smart grid system with rich functionalities, IEEE Trans. Inform. Forens. Secur. 15 (2020) 1318–1330, http://dx.doi.org/10.1109/TIFS.2019.2938875.

[32] A. Alabdulatif, Practical hybrid confidentiality-based analytics framework with Intel SGX, J. Syst. Softw. 181 (2021) 1–8, http://dx.doi.org/10.1016/j.jss.2021.111045.

[33] X. Yang, P. Shi, B. Tian, B. Zeng, W. Xiao, Trust-E: A trusted embedded operating system based on the ARM TrustZone, in: Proceedings of the 11th International Conference on Autonomic and Trusted Computing, IEEE, Bali, Indonesia, 2014, pp. 495–501, http://dx.doi.org/10.1109/UIC-ATC-ScalCom.2014.15.

[34] N. Zhang, H. Sun, K. Sun, W. Lou, Y.T. Hou, CacheKit: Evading memory introspection using cache incoherence, in: Proceedings of the 1st European Symposium on Security and Privacy, IEEE, Saarbruecken, Germany, 2016, pp. 337–352, http://dx.doi.org/10.1109/EuroSP.2016.34.

[35] H. Dai, K. Chen, OPTZ: A hardware isolation architecture of multi-tasks based on TrustZone support, in: Proceedings of the 16th International Conference on Ubiquitous Computing and Communications, IEEE, Guangzhou, China, 2017, pp. 391–395, http://dx.doi.org/10.1109/ISPA/IUCC.2017.00062.

[36] J. Park, Kwangjo Kim, TM-Coin: trustworthy management of TCB measurements in IoT, in: Proceedings of the 15th International Conference on Pervasive Computing and Communications Workshops, IEEE, Kona, HI, USA, 2017, pp. 654–659, http://dx.doi.org/10.1109/PERCOMW.2017.7917640.

[37] S. Pinto, J. Pereira, T. Gomes, A. Tavares, J. Cabral, LTZVisor: TrustZone is the key, in: Proceedings of the 29th Euromicro Conference on Real-Time Systems, Schloss Dagstuhl, Dubrovnik, Croatia, 2017, pp. 1–22, http://dx.doi.org/10.4230/LIPIcs.ECRTS.2017.4.

[38] R. Pettersen., H.D. Johansen., D. Johansen., Secure edge computing with ARM TrustZone, in: Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security, SciTePress, Porto, Portugal, 2017, pp. 102–109, http://dx.doi.org/10.5220/0006308601020109.

[39] R. Chang, L. Jiang, W. Chen, Y. Xie, Z. Lu, A TrustEnclave-based architecture for ensuring runtime security in embedded terminals, Tsinghua Sci. Technol. 22 (5) (2017) 447–457, http://dx.doi.org/10.23919/TST.2017.8030534.

[40] S. Demesie Yalew, P. Mendonca, G.Q. Maguire, S. Haridi, M. Correia, TruApp: A TrustZone-based authenticity detection service for mobile apps, in: Proceedings of the 13th International Conference on Wireless and Mobile Computing, Networking and Communications, IEEE, Rome, Italy, 2017, pp. 1–9, http://dx.doi.org/10.1109/WiMOB.2017.8115820.

[41] H. Jiang, R. Chang, L. Ren, W. Dong, L. Jiang, S. Yang, An effective authentication for client application using ARM TrustZone, in: Proceedings of the 13th International Conference on Information Security Practice and Experience, Springer, Melbourne, Australia, 2017, pp. 802–813, http://dx.doi.org/10.1007/978-3-319-72359-4_50.

[42] L. Guan, C. Cao, P. Liu, X. Xing, X. Ge, S. Zhang, M. Yu, T. Jaeger, Building a trustworthy execution environment to defeat exploits from both cyber space and physical space for ARM, IEEE Trans. Dependable Secure Comput. 16 (3) (2019) 438–453, http://dx.doi.org/10.1109/TDSC.2018.2861756.

[43] H. Park, S. Zhai, L. Lu, F.X. Lin, StreamBox-TZ: Secure stream analytics at the edge with TrustZone, in: Proceedings of the USENIX Annual Technical Conference, USENIX Association, Renton, WA, USA, 2019, pp. 537–554, https://www.usenix.org/conference/atc19/presentation/park-heejin.

[44] C. Segarra, R. Delgado-Gonzalo, V. Schiavoni, MQT-TZ: hardening IoT brokers using ARM TrustZone, in: Proceedings of the 39th Symposium on Reliable Distributed Systems, IEEE Computer Society, Shanghai, China, 2020, pp. 256–265, http://dx.doi.org/10.1109/SRDS51746.2020.00033.

[45] OASIS Standard, MQTT version 5.0, 2020, http://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html.

[46] G. Chen, S. Chen, Y. Xiao, Y. Zhang, Z. Lin, T.H. Lai, SgxPectre: Stealing Intel secrets from SGX enclaves via speculative execution, in: Proceedings of the 4th European Symposium on Security and Privacy, IEEE, Stockholm, Sweden, 2019, pp. 142–157, http://dx.doi.org/10.1109/EuroSP.2019.00020.

[47] N. Weichbrodt, A. Kurmus, P. Pietzuch, R. Kapitza, AsyncShock: Exploiting synchronisation bugs in Intel SGX enclaves, in: Proceedings of the 21st European Symposium on Research in Computer Security, Springer, Heraklion, Greece, 2016, pp. 440–457, http://dx.doi.org/10.1007/978-3-319-45744-4_22.

[48] J.G. Beekman, D.E. Porter, Challenges for scaling applications across enclaves, in: Proceedings of the 2nd Workshop on System Software for Trusted Execution, ACM, Shanghai, China, 2017, pp. 1–2, http://dx.doi.org/10.1145/3152701.3152710.

[49] M. Lipp, D. Gruss, R. Spreitzer, C. Maurice, S. Mangard, ARMageddon: Cache attacks on mobile devices, in: Proceedings of the 25th USENIX Security Symposium), USENIX Association, Austin, TX, USA, 2016, pp. 549–564, https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/lipp.

[50] S. Pinto, N. Santos, Demystifying ARM TrustZone: A comprehensive survey, ACM Computing Surveys 51 (6) (2019) 1–36, http://dx.doi.org/10.1145/3291047.

[51] Z. Ning, F. Zhang, W. Shi, W. Shi, Position paper: Challenges towards securing hardware-assisted execution environments, in: Proceedings of the 6th International Workshop on Hardware and Architectural Support for Security and Privacy, ACM, Toronto, ON, Canada, 2017, pp. 1–8, http://dx.doi.org/10.1145/3092627.3092633.

[52] O. Surdu, Reliable and Secure Firmware Update for Internet of Things (IoT) Devices, Google Patents, 2018, https://patents.google.com/patent/US20180081666A1/en, US Patent US20180081666A1.

[53] W. Wang, G. Chen, X. Pan, Y. Zhang, X. Wang, V. Bindschaedler, H. Tang, C.A. Gunter, Leaky cauldron on the dark land: Understanding memory side-channel hazards in SGX, in: Proceedings of the 24th Conference on Computer and Communications Security, ACM, Dallas, TX, USA, 2017, pp. 2421–2434, http://dx.doi.org/10.1145/3133956.3134038.

[54] D. Cerdeira, N. Santos, P. Fonseca, S. Pinto, SoK: understanding the prevailing security vulnerabilities in TrustZone-assisted TEE systems, in: Proceedings of the 41st Symposium on Security and Privacy, IEEE, San Francisco, CA, USA, 2020, pp. 1416–1432, http://dx.doi.org/10.1109/SP40000.2020.00061.

[55] M. Schwarz, S. Weiser, D. Gruss, C. Maurice, S. Mangard, 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, Springer, Bonn, Germany, 2017, pp. 3–24, http://dx.doi.org/10.1007/978-3-319-60876-1_1.

[56] F. Brasser, U. Müller, A. Dmitrienko, K. Kostiainen, S. Capkun, A.-R. Sadeghi, Software Grand Exposure: SGX cache attacks are practical, in: Proceedings of the 11th USENIX Workshop on Offensive Technologies, USENIX Association, Vancouver, BC, Canada, 2017, pp. 1–12, https://www.usenix.org/conference/woot17/workshop-program/presentation/brasser.

[57] A. Moghimi, G. Irazoqui, T. Eisenbarth, CacheZoom: How SGX amplifies the power of cache attacks, in: Proceedings of the 19th International Conference on Cryptographic Hardware and Embedded Systems, Springer, Taipei, Taiwan, 2017, pp. 69–90, http://dx.doi.org/10.1007/978-3-319-66787-4_4.

[58] M.-W. Shih, S. Lee, T. Kim, M. Peinado, T-SGX: Eradicating controlled-channel attacks against enclave programs, in: Proceedings of the Network and Distributed System Security Symposium, Internet Society, San Diego, CA, USA, 2017, pp. 1–15, http://dx.doi.org/10.14722/ndss.2017.23193.