IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 26, 2021, accepted May 23, 2021, date of publication June 2, 2021, date of current version June 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3085524

Systematic Literature Review on the Use of Trusted Execution Environments to Protect Cloud/Fog-Based Internet of Things Applications

DALTON CÉZANE GOMES VALADARES[1,2], NEWTON CARLOS WILL[3], (Member, IEEE), JEAN CAMINHA[4], MIRKO BARBOSA PERKUSICH[5], ANGELO PERKUSICH[6], (Member, IEEE), AND KYLLER COSTA GORGÔNIO[1]

[1]Computer Science Department, Federal University of Campina Grande, Campina Grande 58428-830, Brazil
[2]Mechanical Engineering Department, Federal Institute of Pernambuco, Caruaru 50740-545, Brazil
[3]Computer Science Department, Federal University of Technology - Paraná, Dois Vizinhos 80230-901, Brazil
[4]Computer Science Department, Federal University of Mato Grosso, Cuiabá 78060-900, Brazil
[5]VIRTUS, Campina Grande 58428-830, Brazil
[6]Electrical Engineering Department, Federal University of Campina Grande, Campina Grande 58428-830, Brazil

Corresponding author: Dalton Cézane Gomes Valadares (dalton.valadares@embedded.ufcg.edu.br)

ABSTRACT Trusted Execution Environments have been applied to improve data security in many distinct application scenarios since they enable data processing in a separate and protected region of memory. To investigate how this technology has been applied to the different IoT scenarios, which commonly deal with specific characteristics such as device resource constraints, we carried out a systematic literature review. For this, we selected and analyzed 58 papers from different conferences and journals, identifying the main IoT solutions and scenarios in which TEE has been employed. We also gathered the mentioned TEE advantages and disadvantages as well as the suggestions for future works. This study gives a general overview of the use of TEEs for cloud/fog-based IoT applications, bringing some challenges and directions.

INDEX TERMS Trusted computing, Internet of Things, trusted execution environments, data security, Intel SGX, ARM TrustZone.

I. INTRODUCTION

The Internet of Things (IoT) term was used for the first time in 1999 [1], [2], by Kevin Ashton, in a presentation in which he described the network connecting physical devices to the Internet. Since then, academics and industries have helped in the IoT evolution, mainly related to device technologies, communication protocols, and applications. Some of the IoT advantages are automation, low cost, and remote management [3]. According to McKinsey&Company, the economic impact of IoT can rise from 2.7 to 6.2 trillion dollars by the year 2025, while Gartner predicts the deployment of 20.8 billion IoT devices by the year 2020 [4] (in 2017, this number was equivalent to 8.4 billions of devices in the world, 31% more than in 2016 [5]).

Although the IoT term was born with RFID (Radio-Frequency Identification) projects, which used RFID tags, nowadays, there are many different types of devices (the things) with many different processing capabilities. IoT devices have very constrained resources, but those containing more computational power are called smart objects [6], [7]. The smart objects and their interconnection enable many IoT applications in many domains, such as logistics, transportation, industry, and healthcare.

In many situations, due to resource constraints (battery, memory, and processing), the smart objects' capabilities are not enough to perform more complex tasks. To support the execution of such complex tasks, more robust devices become necessary. These devices are commonly called *gateways* and are deployed near the IoT devices to assist in the processing of tasks that demand more computational power. This is the basis for edge computing, a decentralized computing infrastructure that enables IoT devices to process data on more powerful devices, generally near the network edge. Commonly, edge devices are geographically closer to IoT devices, becoming a good alternative when the operations are time-constrained or there is a poor network connectivity [8]. The advantages of edge and cloud computing, such as lower communication latency and larger processing

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaolong Li.

capacity, empower the Internet of Things, enabling cloud/edge-based IoT.

Given that IoT devices can collect information such as location, time, and context, enabling inferring personal habits, behavior, and individual preferences, data protection is commonly needed to acquire, manage, transit, and use/process. Sensitive data generated by IoT devices are attractive to unauthorized third parties, which results in a significant concern to end-users and companies as they can lose control over their data, especially when these data are stored in cloud servers. In general, risks and threats must be considered at the beginning of the application development, i.e., during the application design. Privacy by Design[1] [9] and Privacy by Evidence[2] [10] techniques recommend. Since it is difficult to address all the privacy aspects at the development process, like the Privacy by Design approach proposes, the privacy settings are usually left to the user [6], [11]. The privacy sphere is then defined as the network and all IoT devices that a user owns and trusts to preserve sensitive data [6].

In general, the main security/privacy concern is related to managing the high quantity of generated data to protect the storage and communication [12]. IoT devices, such as smart cameras and health monitoring devices, can reveal private information about users in distinct application scenarios. Therefore, end-to-end security critical in such scenarios, becoming a mandatory requirement to protect data against unauthorized accesses and attacks [13]. Thus, the applications should consider a protected communication protocol (data in transit) and a protected storage/processing (data in rest).

According to Ayoade *et al.* [14], most IoT devices have not enough processing power to implement many of the security schemes. In many cases, it is not recommended to manage and store secrets in these devices. Often they send data for processing in external servers, needing to trust that these servers will guarantee users' data protection and privacy [14]. When an acceptable level of privacy is not reached, the consequences vary from the non-acceptance of the cloud services to severe, costly lawsuits. A current technical challenge is implementing the ''right to be forgotten'', a data protection regulation proposed by European Union stating that information about someone has to be automatically removed after some time [6].

Due to the inherently distributed nature of IoT applications, adversaries are exploring their vulnerabilities to generate DDoS (Distributed Denial of Service) attacks using the IoT devices [15]–[17]. For instance, the number of DDoS attacks increased by 91% in 2017 due to IoT devices/applications. This leads the users to adopt extra effort to ensure security/privacy in IoT applications since many times, they need to trust the external providers (servers). The lack of proper security mechanisms in IoT applications is dangerous in sensitive domains, like healthcare. Unfortunately, there is still no guide for applying the right security level in this kind of application [3].

To minimize these security and privacy concerns intrinsic to the IoT application scenarios and avoid or mitigate such attacks, Trusted Execution Environments (TEE) had been adopted. TEE is a tamper-resistant environment that runs on a separated kernel, according to an approach that considers two execution environments (''trusted world'' and ''normal world''), guaranteeing the code authenticity, the runtime states integrity (memory and CPU registers), and the stored code and data confidentiality [18]. TEE also has to provide a remote attestation mechanism that can prove its trustworthiness for third parties.

The TEE concept arose with the need to protect data processing in increasingly complex systems. This technology became a new approach for Trusted Computing (TC), which was developed by the Trusted Computing Group (TCG)[3] in the earlier 2000s to provide better levels of secure computation, privacy, and data protection. The TC was initially implemented using Trusted Platform Modules (TPM), which are separate tamper-evident hardware modules for platform security that allow cryptographic keys protection and data integrity. Since TPMs do not allow third parties to run code inside them, TEE was proposed to provide an isolated and protected execution environment for third-parties applications. TEE secure characteristics include isolated code execution and integrity of the runtime code, execution files, and control flow [19]. It maintains the confidentiality of data and code even when under a machine physical control attack. This way, for instance, users can use cloud services without worrying about data security and privacy [13].

Since 2010, Global Platform[4] has standardized the TEE specifications, which include the system architecture and APIs, such as TEE Client API, TEE Internal Core API, TEE Secure Element API.[5] The two main TEE technologies currently available in the market are ARM TrustZone[6] and Intel SGX.[7] Unlike ARM TrustZone, Intel SGX is not compliant with the Global Platform specifications. For instance, instead of the specified ''trusted world'', that must be implemented on a ''trusted operating system'', Intel SGX creates isolated memory zones, called enclaves, that run on the same operating system.

Given the current adoption of TEEs for data protection in IoT applications, we decided to carry out a Systematic Literature Review (SLR) to investigate what kind of solutions

[1] Privacy by Design is an approach that considers privacy concerns during all the development process and demands the adoption of privacy techniques during all the software development cycle.

[2] Since the Privacy by Design adoption sometimes is considered difficult to enforce, the Privacy by Evidence methodology was proposed, providing a guide that helps developers to apply privacy techniques according to a well-defined process that contains a checklist and generates evidence that the development considered and dealt with privacy concerns.

[3] https://trustedcomputinggroup.org/

[4] http://globalplatform.org/

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

[6] https://developer.arm.com/ip-products/security-ip/trustzone

[7] https://software.intel.com/pt-br/sgx

have been proposed. For this SLR, we considered the research questions listed below:

1) What kind of IoT solutions has TEE been used for?
2) What kind of IoT scenarios has TEE been used in?
3) What are the advantages and disadvantages of TEE usage?
4) What have the authors proposed for future work?

The initial search in the main scientific repositories of Computer Science resulted in 541 papers. After removing duplicates and considering exclusion and inclusion criteria, we selected 58 relevant papers for the SLR. Then, we performed the data extraction and quality assessment for each selected paper.

The main contributions of this study are:

- An overview on state of the art regarding the use of TEEs to protect IoT applications, considering relevant papers;
- An SLR protocol that can be followed to replicate, verify and update this study in the future;
- Challenges and directions regarding the use of TEEs for IoT scenarios.

The remainder of this paper is organized as follows. In Section II we present the two main TEE technologies available in the market: ARM TrustZone and Intel Software Guard eXtensions (SGX). We present the defined research protocol in Section III. In Section IV, we exhibit the general results regarding the quantitative collected data. In Section V, we present the results for the four research questions, as well as the threats to the validity and challenges and directions related to the use of TEEs. We mention some related works in Section VI and, finally, we present the conclusions in Section VII.

II. BACKGROUND

A. ARM TRUSTZONE

TrustZone is the Arm TEE technology, which provides system-wide hardware isolation for trusted applications [20]. It is more suitable for IoT devices since it is embedded in Arm processors architectures that are among the most used for embedded systems and devices in general, such as micro-controllers, mobile phones, and tablets. The TrustZone applications are divided into two worlds, a secure and an insecure one. The secure world runs a trusted OS, responsible for isolating and running trusted applications, providing confidentiality and integrity to the system. The insecure world runs an untrusted OS, commonly called Rich OS, a common OS such as any Linux distribution. We show the TrustZone basic application architecture in Figure 1.

An untrusted software cannot access data and resources from the secure world. As seen in Figure 1, both untrusted and trusted applications use the GlobalPlatform TEE client API and TEE Internal Core API specifications. TrustZone is used in billions of device applications to protect code and sensitive data in processes such as authentication, payment, and content protection [20].

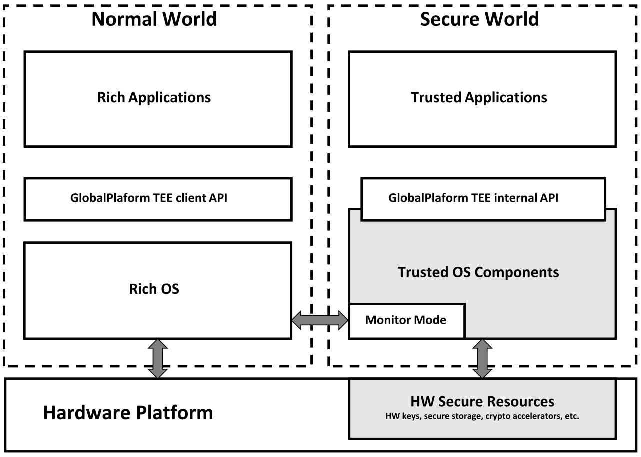


FIGURE 1. TrustZone basic application architecture [20].

B. INTEL SOFTWARE GUARD eXtensions (SGX)

Intel SGX is a hardware-assisted Trusted Execution Environment (TEE) technology that protects code and data from disclosure or modification [21]. Applications intended to be safe are executed on specially protected memory regions. Their code and data are isolated from other software running in the system, even with higher privilege, like Operating System (OS). These specialized regions in the memory are called *enclaves*, which are created and manipulated through a distinct set of processor instructions, with the help of a software development kit (SDK) provided by Intel. Intel promises that code and data remain protected with these hardware-based capabilities even when drivers, OSs, or BIOS are compromised.

The execution of an SGX application is shown in Figure 2, according to the following flow:

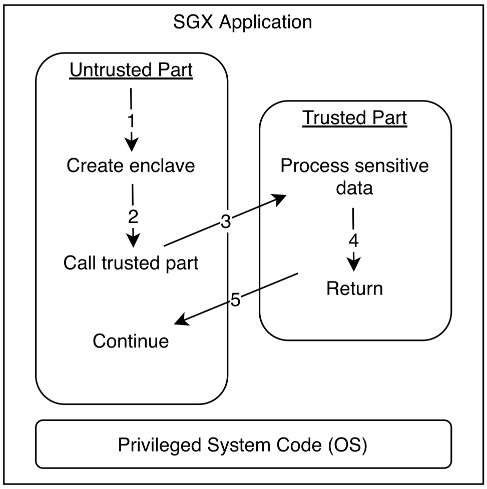


FIGURE 2. SGX application execution [22].

1) once the application is instantiated, an enclave is created;
2) the trusted portion of the code, which runs in a protected region of memory, is called;

TABLE 1. First search results.

Research repository	First paper in the result
ACM Digital Library (5)	A Secure, Privacy-Preserving IoT Middleware using Intel SGX
IEEE Xplore Digital Library (6)	IIoTEED: An Enhanced, Trusted Execution Environment for Industrial IoT Edge Devices
Scopus (61)	Fog Orchestration for the Internet of Everything: State-of-the-Art and Research Challenges
Springer Link (40)	MIPE: a Practical Memory Integrity Protection Method in a Trusted Execution Environment
ScienceDirect (23)	A Security Authorization Scheme for Smart Home Internet of Things Devices
Google Scholar (703)	A Survey of Fog Computing: Concepts, Applications and Issues

3) the application performs any protected processing inside the trusted part of the application;
4) the result of the trusted processing is returned to the untrusted part of the application;
5) the untrusted part of the application proceeds the execution.

SGX works with a tiny attack surface formed by the processor boundaries, preventing direct attacks on executing code or sensitive data in memory. The enclaves work in these boundaries, shielding the data and code inside them and encrypting data when they need to leave the enclave. Once data return, the enclaves perform integrity checking.

Intel SGX also provides a way to enable remote parties to check if an application executes in a valid enclave of a real TEE processor. This mechanism ensures an enclave's authenticity, validating, for the remote party, the application enclave's identity. This is possible due to a remote attestation process that follows a specific protocol. In this process, both parts exchange information enabling the remote application to verify the supposed SGX application's authenticity by accessing the Intel Attestation Service (IAS) and checking the information received from the enclave. At the end of the remote attestation, both sides will have a symmetric shared key, which they can use to encrypt and exchange sensitive information. The authenticity of the code can also be checked, but the description of this process is out of scope for this work.

III. RESEARCH PROTOCOL

This Section presents the protocol defined to carry out this systematic literature review.

A. SEARCH STRATEGY

We used common keywords related to this study and the best synonyms of each keyword to perform the search. The Population, Intervention, Context, and Outcome (PICO) approach was also applied to delimit the related work range better. High-quality documents, searched at Computer Science conferences and journals, were considered. We achieved that using the main research repositories available. Below, we list the keywords and PICO terms.

- Keywords: Trusted Execution Environment, Internet of Things, Security
- Population: Internet of Things, Web of Things, Edge of Things, Internet of Everything
- Intervention: Trusted Execution Environment, Trusted Execution Technology
- Context: Security, Privacy, Confidentiality, Integrity, Trustworthiness, Protection
- Outcome: Solution, Technique, Tool, Approach, Method, Mechanism, Advantage, Benefit, Positive Point, Disadvantage, Drawback, Negative Point

We analyzed some papers regarding title, abstract, and keywords to improve the number of keywords used in the source selection criteria. We selected these papers by collecting the first result of a search in the main research repositories: ACM Digital Library, IEEE Xplore Digital Library, Scopus, Springer Link, ScienceDirect, and Google Scholar. Thus, we used the described keywords as search string: "Internet of Things" AND "Trusted Execution Environment" AND "Security". The returned papers were then ordered by default according to their relevance, with the most relevant first. Table 1 shows the first paper at the resulted list for each search composing a total of 6 papers.

Since the first result in the ACM returned list is not a scientific paper (Hardware-Assisted Security: Promises, Pitfalls, and Opportunities), the second result was chosen (as listed in the Table 1). After this phase, we considered some other terms to compose the search string, as shown in Table 2.

B. SEARCH STRING

The search defined string defined is composed of keywords, alternative terms, and some of the PICO terms. We used the boolean operators AND/OR to link these terms, resulting in a representative search string. Based on this, the defined search string is as follows:

> ("Internet of Thing" OR "Edge Computing" OR "Edge of Things" OR "Embedded Systems" OR "EoT" OR "Internet of Everything" OR "IoE" OR "IoT" OR "Web of Things" OR "WoT") AND ("Trusted Execution Environment" OR "Intel SGX" OR "Keystone Enclave" OR "Sanctum" OR "SGX" OR "Software Guard Extensions" OR "TEE" OR "Trusted Execution Technology" OR "Trustzone") AND ("Advantage" OR "Benefit" OR "Positive Point" OR "Disadvantage" OR "Drawback" OR "Negative Point" OR "Security" OR "Confidentiality" OR "Integrity" OR "Privacy" OR "Protection" OR "Trustworthiness" OR "Solution" OR

TABLE 2. Terms considered for the search string.

Keyterms	Alternative terms
Trusted Execution Environment	Trusted Execution Technology, Trustzone, Intel SGX, SGX, Software Guard Extensions, TEE, Keystone Enclave, Sanctum
Internet of Things	Edge Computing, Edge of Things, Embedded Systems, EoT, Internet of Everything, IoE, IoT, Web of Things, WoT
Security	Confidentiality, Integrity, Trustworthiness, Privacy, Protection
Solution	Approach, Framework, Mechanism, Method, Technique, Tool, Advantage, Benefit, Positive Point, Disadvantage, Drawback, Negative Point

TABLE 3. Validation of the search string.

Papers	Result
T2Droid: A Trustzone-based Dynamic Analyser for Android Applications	OK
TrustShadow: Secure Execution of Unmodified Applications with ARM TrustZone	OK
IIoTEED: An Enhanced, Trusted Execution Environment for Industrial IoT Edge Devices	OK
A Security and Trust Framework for Virtualized Networks and Software-defined Networking	OK
Secure and Trusted Application Execution on Embedded Devices	OK
MIPE: a Practical Memory Integrity Protection Method in a Trusted Execution Environment	OK

"Approach" OR "Framework" OR "Mechanism"
OR "Method" OR "Technique" OR "Tool")

After determining the search string, we searched to validate its application by analyzing if some known papers were found. Table 3 shows that all the validation papers were found using the defined search string.

C. SEARCH REPOSITORIES

To expand the possibilities of finding good works, covering a high quantity of research sources, we decided to use the following scientific repositories (Table 4) that gather publications from many essential conferences and journals:

TABLE 4. Selected scientific repositories.

Scientific Repository	URL
ACM Digital Library	http://portal.acm.org
El Compendex	http://www.engineeringvillage.com
IEEE Digital Library	http://ieeexplore.ieee.org
Wiley Online Library	http://onlinelibrary.wiley.com
Scopus	http://www.scopus.com
Springer Link	http://link.springer.com

As ScienceDirect and Scopus are both from Elsevier and the first has strict limitations regarding the search string (characters and boolean operators quantity), we decided to consider only the Scopus base. For the same reason, Google Scholar was excluded from our previous list of research indexers since it also limits the search string's size, making it difficult to use the one we defined.

D. SELECTION CRITERIA

We defined selection criteria for exclusion and inclusion of papers to refine the search, avoid results that are not relevant to answer the research questions, and improve the possibility of obtaining relevant results, as shown as follows.

- Exclusion Criteria:
  - Posters, short papers and extended abstracts (papers with less than 3 pages);
  - Book chapters, PhD and Master theses;
  - Papers not written in English;
  - Papers that do not focus on TEE usage for IoT security;
  - Duplicate results;
  - Papers published before 2000 (trusted computing was defined in earlies 2000);
  - Secondary studies.
- Inclusion Criteria
  - Journal and conference articles;
  - Results that answer the research questions.

E. SELECTION PROCEDURE

To select the interesting documents for this SLR, we used the selection criteria and followed these steps:

1) Exclude duplicate documents;
2) Exclude documents published before 2000 or documents not written in English;
3) Exclude documents that were not published in conferences or journals;
4) Exclude book chapters, Ph.D. and Master theses, extended abstracts, short papers, posters, secondary studies (e.g., surveys and reviews);
5) Exclude irrelevant documents, i.e., documents that are not related to TEE usage for IoT security.

After we performed the exclusions described in the four initial points, two reviewers analyzed each entry of the resulting list to exclude irrelevant documents, according to the following criteria:

- Each reviewer classifies the document as relevant, undefined, or irrelevant;
- Documents classified as relevant by two reviewers are maintained;

- Documents classified as irrelevant by two reviewers are excluded;
- Documents classified as undefined by two reviewers are better analyzed, through a fast reading of the entire document, and then reclassified as relevant or irrelevant;
- Documents classified as relevant or undefined by one reviewer and as irrelevant by another reviewer are discussed between both until they agree regarding one of the previous classifications.

F. QUALITY ASSESMENT

To assess the quality of the papers, we elaborated the following list of questions:

1) Is the text well organized and clear (easy to understand)?
2) Is the motivation and objective well described?
3) Does the paper present an application scenario or case study?
4) Is the methodology clear (easy to understand and replicate)?
5) Does the paper have many references and good related works?
6) Does the study present some implementation and practical results?
7) Does the paper clearly present any advantages/ disadvantages regarding the technology used?
8) Do the authors present good validation?
9) Are the results clear enough (explicit and well discussed/evaluated)?
10) Do the authors suggest future works?

For each selected paper, the reviewers answered each question with three possible answers: yes, partially, or no. We assessed the paper quality according to the score attributed for each answer:

- Yes - 1;
- Partially - 0.5;
- No - 0.

Thus, since there are ten questions, the maximum score can be 10, whereas the minimum score can be 0, indicating the highest and the lowest quality for a paper, respectively. Depending on the papers score, we classified them as:

- High quality, if scored above 6;
- Medium quality, if scored between 4 and 6;
- Low quality, if scored below 4.

To answer each of the quality assessment questions, we elaborated the following guideline, enumerated according to the question number:

1) Only receives 1 if the overall text is well written, organized, and clear;
2) Only receives 1 if the motivation and the objectives are clear and well explained;
3) Only receives 1 if the paper clearly mentions an application scenario (e.g., smart home application, payment system, and so on);
4) Only receives 1 if the methodology is clear, involving the experimental design, experiment execution, and results' definition;
5) Only receives 1 if the paper is well-grounded, with good related references;
6) Only receives 1 if there is an implementation for the proposal and practical results well described;
7) Only receives 1 if there are advantages and disadvantages about the use of TEE;
8) Only receives 1 if the authors present a good validation for the proposal, practical or formal, well-described and grounded;
9) Only receives 1 if the results are clear, well discussed, and well-evaluated;
10) Only receives 1 if the authors suggest future works related to their proposals.

G. DATA EXTRACTION

The data to be extracted are given as follows:

- Title;
- Authors;
- Abstract;
- Year;
- Type of article;
- Conference/Journal name;
- Country/countries where the research was carried out;
- Number of pages;
- Number of citations;
- Quality;
- IoT solution;
- IoT scenario;
- Advantages and disadvantages of TEE usage;
- Future work suggestions.

IV. GENERAL RESULTS

After performing the search in all the selected scientific repositories, in August 2018, we got a list of 541 results. Each entry of this list was judged and classified, by four reviewers, as relevant, irrelevant, or undefined. After this process, we obtained 134 undefined papers, 43 relevant papers, and 364 irrelevant papers.

We then started a new review process, dividing the 134 undefined papers into two groups of 67 papers for each reviewer pair. The reviewers performed a better analysis of each undefined paper's relevance and reclassified them as irrelevant or relevant. In the end, we selected a total of 58 relevant papers for the SLR. We list the selected papers and a summary for each presented work in Table 5.

As we can notice in Fig. 3, the number of papers has grown over the years. In 2018 we have just 14 papers because the search was performed at the end of August. We have to consider also delays at the publishing phase regarding conference and journal papers already accepted during the first semester, but that only becomes available during the second semester. Among these papers, 47 were published in conference, symposium, or workshop proceedings, while 11 were published

TABLE 5. Selected papers.

Title	Year	Summary
Design and Implementation of Secure Embedded Systems Based on Trustzone [23]	2008	Access control between trusted and untrusted applications
Securing Peer-to-peer Distributions for Mobile Devices [24]	2008	Secure P2P distribution for mobile devices with owner control
Lightweight Anonymous Authentication with TLS and DAA for Embedded Mobile Devices [25]	2011	Anonymous authentication method for mobile devices
Secure device access for automotive software [26]	2013	Secure device access to restrict direct access for the extension software
TEEM: A User-Oriented Trusted Mobile Device for Multi-platform Security Applications [27]	2013	System to address security requirements in mobile and embedded platforms
Hardware Security for Device Authentication in the Smart Grid [28]	2013	Private keys protection without user interaction and device authentication
Trust-E: A Trusted Embedded Operating System Based on the ARM Trustzone [29]	2014	Trusted embedded OS architecture
TrustDump: Reliable Memory Acquisition on Smartphones [30]	2014	Tool to obtain RAM and CPU registers from a compromised mobile OS
Hardware Intrinsic Security to Protect Value in the Mobile Market [31]	2014	Implementation of an electronic fingerprint derived from device physical characteristics
Hardware-security technologies for industrial IoT: TrustZone and security controller [32]	2015	Device snapshot authentication system
Secure and Trusted Application Execution on Embedded Devices [33]	2015	Holistic approach to the security and trust of embedded devices
A Novel Method of APK-Based Automated Execution and Traversal with a Trusted Execution Environment [34]	2016	Tool for lightweight applications using TEE
CacheKit: Evading Memory Introspection Using Cache Incoherence [35]	2016	Rootkit to include malicious code in TrustZone
IoTEE-An integrated framework for rapid trusted IOT application development [36]	2016	Framework for development of trusted IoT applications
C-FLAT: Control-Flow Attestation for Embedded Systems Software [37]	2016	Attestation for the application execution path without requiring the source code
Remote Attestation for Embedded Systems [38]	2016	Implementation of Trusted Platform Module (TPM) with ARM processor
CaSE: Cache-Assisted Secure Execution on ARM Processors [39]	2016	Cache-assisted protection against multi-vector attacks and memory disclosure
A trust enclave-based architecture for ensuring run-time security in embedded terminals [19]	2017	Architecture to secure embedded terminals
TruApp: A TrustZone-based authenticity detection service for mobile apps [40]	2017	Authenticity and integrity verifier for mobile apps
OPTZ: a Hardware Isolation Architecture of Multi-Tasks Based on TrustZone Support [41]	2017	Multitask hardware isolation between normal and secure worlds
TM-Coin: Trustworthy management of TCB measurements in IoT [42]	2017	Trustworthy management system for TCB measurements in IoT
IIoTEED: An Enhanced, Trusted Execution Environment for Industrial IoT Edge Devices [43]	2017	Real Time Operating System inside the secure world of ARM TrustZone
Establishing Mutually Trusted Channels for Remote Sensing Devices with Trusted Execution Environments [44]	2017	Remote attestation in a single run for small-scale devices
MIPE: A Practical Memory Integrity Protection Method in a Trusted Execution Environment [45]	2017	Protection against kernel data and direct memory access attacks
Towards the Security of Motion Detection-based Video Surveillance on IoT Devices [46]	2017	Framework to secure the motion detection procedures
Ditio: Trustworthy Auditing of Sensor Activities in Mobile & IoT Devices [47]	2017	Sensor activities auditing and compliance checking
TrustShadow: Secure Execution of Unmodified Applications with ARM TrustZone [48]	2017	Protection for legacy applications running in untrusted OS
T2Droid: A trustzone-based dynamic analyser for android applications [49]	2017	Dynamic API function calls and kernel syscalls analyser for Android
A private user data protection mechanism in trustzone architecture based on identity authentication [50]	2017	Identity certification of Client Applications to avoid data leakage
TZ-KPM:Kernel Protection Mechanism on Embedded Devices on Hardware-Assisted Isolated Environment [51]	2017	Kernel protection mechanism to detect malicious processes
TICS: Trusted Industry Control System Based on Hardware Security Module [52]	2017	Trustworthiness defense based on attestation and whitelist mechanisms
Boot Attestation: Secure Remote Reporting with Off-The-Shelf IoT Sensors [53]	2017	Light scheme to measure software integrity during the boot phase
Secure mobile device structure for trust IoT [54]	2017	Secure software domain separation to protect IoT data
Cryptographic key protection against FROST for mobile devices [55]	2017	Cryptographic key protection scheme for mobile devices
Encrypting data to pervasive contexts [56]	2017	Communication security between building blocks
LTZVisor: TrustZone is the key [57]	2017	Hypervisor architecture to assist virtualization
An Effective Authentication for Client Application Using ARM TrustZone [58]	2017	Client authentication scheme using ARM TrustZone
ARMHEx: A hardware extension for DIFT on ARM-based SoCs [59]	2017	Protection for DRIFT (Dynamic Information Flow Tracking)
Privacy-Preserving Location-Based Services by Using Intel SGX [60]	2017	User privacy enforcement with anonymity and indistinguishability
Secure and Privacy-Aware Data Dissemination for Cloud-Based Applications [61]	2017	Secure data dissemination for cloud-based IoT applications

TABLE 5. ***(Continued.)*** **Selected papers.**

On the performance of a trustworthy remote entity in comparison to secure multi-party computation [62]	2017	Performance comparison of SGX-based TRE and secure MPC frameworks
Secure edge computing with ARM trust zone [63]	2017	Platform to connect IoT devices and proprietary cloud solutions
LogSafe: Secure and Scalable Data Logger for IoT Devices [13]	2018	SGX-based trusted logger for IoT devices data
Decentralized IoT Data Management Using BlockChain and Trusted Execution Environment [14]	2018	Decentralized system for IoT data management using blockchain and TEE
Program-flow attestation of IoT systems software [64]	2018	Remote attestation process for IoT devices
Enabling Security-Enhanced Attestation With Intel SGX for Remote Terminal and IoT [65]	2018	Enhance security for remote terminals (bring your own device)
Embedded policing and policy enforcement approach for future secure IoT technologies [66]	2018	Activities monitor on the SoC communication bus
TEE Based Session Key Establishment Protocol for Secure Infotainment Systems [67]	2018	Secure communication between a user device and a vehicle telematics
Feasibility of societal model for securing Internet of Things [68]	2018	Societal model for IoT security, where the "stronger" takes care of the "weaker"
Centralized duplicate removal video storage system with privacy preservation in IoT [69]	2018	Privacy-preserving deduplicate video on storage systems
CryptMe: Data Leakage Prevention for Unmodified Programs on ARM Devices [70]	2018	Integration of memory encryption and TrustZone-based memory access controls
PrivacyGuard: Enforcing Private Data Usage with Blockchain and Attested Execution [71]	2018	Framework that integrates blockchain and trusted execution environment
Development of an Embedded Platform for Secure CPS Services [72]	2018	TrustZone-based platform to secure cyber-physical devices and gateways
Towards Decentralized Accountability and Self-sovereignty in Healthcare Systems [73]	2018	Data Management system to protect personal health data
BASTION-SGX: Bluetooth and Architectural Support for Trusted I/O on SGX [74]	2018	Architectural support for bluetooth trusted I/O
Security and privacy aware data aggregation on cloud computing [75]	2018	Architecture for secure data aggregation in cloud-based IoT
EmLog: Tamper-Resistant System Logging for Constrained Devices with TEEs [76]	2018	Tamper-resistant logging system for constrained devices
Building a Trustworthy Execution Environment to Defeat Exploits from both Cyber Space and Physical Space for ARM [8]	2018	System to shield legacy applications on IoT devices

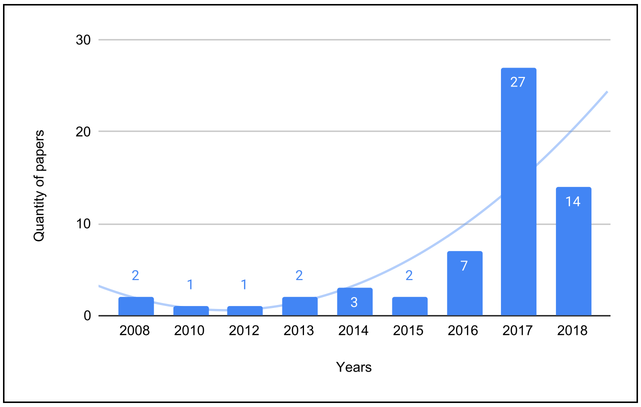


FIGURE 3. Quantity of papers by year.

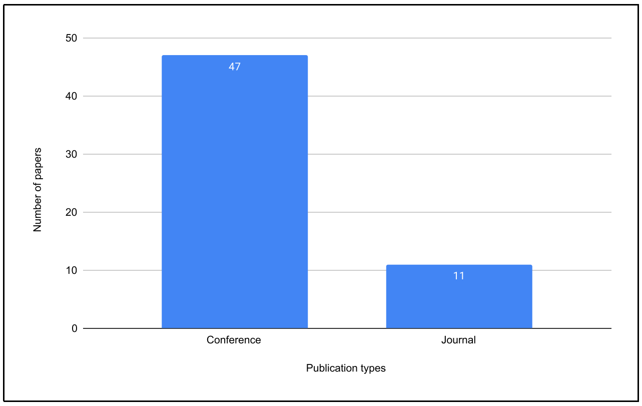


FIGURE 4. Number of papers by type.

in journals, as seen in Fig. 4. The conference names are listed in Table 6 and journal names are listed in Table 7. The names with an asterisk * mean that the conference or journal had two papers published.

As seen in Fig. 5, most of the papers involve the collaboration between 3, 4, 5, or 6 authors. A lower quantity has just 2 or more than 6 authors. Except for 4 works, all the papers present 6 or more pages, as seen in Fig. 6. Fig. 7 presents the number of papers according to the number of citations. As many papers were still recently published when we searched, they have no citations or just a few (1 to 6). Only a few articles have more than 6 citations.

We present in Fig. 8 the quality assessment grades and the respective quantity of papers for each grade. According to the quality assessment criteria, we can see that most of the papers got a grade between 5.5 and 8.5. In Fig. 9, we can see the number of papers regarding the quality classification, which was attributed considering the papers' quality score. The percentage distribution of papers by answers (yes, partially, and no) for each quality assessment question can be seen in the Table 8. More than 50 % of the papers present well-organized texts, with motivation and objective well described. Also, more than 50 % of the papers present many references and good related works and some implementation with practical results. The methodology explanation, the validation, and results discussion are points that can be improved. Lastly, less than 50 % of the papers presented an application scenario

TABLE 6. Conference names.

Intl. Conf. on Information Reuse and Integration for Data Science
Intl. Conf. on Embedded Software and Systems
Intl. Conf. on Information Security
intl. Conf. on Connected Vehicles and Expo
Intl. Conf. on Autonomic & Trusted Computing
Intl. Conf. on High Performance Computing and Communications
Intl. Conf. for Information Technology and Communications
Intl. Conf. on Wireless and Mobile Computing, Networking and Communications
Intl. Conf. on Trust, Security and Privacy in Computing and Communications*
Intl. Conf. on Mobile Systems, Applications, and Services
Intl. Conf. on Field Programmable Logic and Applications
Intl. Conf. on Information and Communications Security
Intl. Wksp on Data Privacy Management
Intl. Wksp on Hardware and Architectural Support for Security and Privacy
Intl. Wksp on the Security of Industrial Control Systems and Cyber-Physical Systems*
Euromicro Conf. on Real-Time Systems
European Symposium on Research in Computer Security*
European Symposium on Security and Privacy
Intl. Symposium on Cyberspace Safety and Security
Intl. Symposium on Research in Attacks, Intrusions, and Defenses
Symposium on Security and Privacy
Learning and Technology Conference
Intl. Conf. on Information Security Theory and Practice
Intl. Conf. on Information Security Practice and Experience*
Intl. Conf. on Recent Trends in Electronics, Information & Communication Technology
Intl. Conf. on Trust and Trustworthy Computing
Intl. Conf. on Computational Intelligence and Security
Intl. Conf. on Ubiquitous Computing and Communications
Intl. Conf. on Pervasive Computing and Communications
Intl. Conf. on Availability, Reliability and Security
Intl. Conf. on Utility and Cloud Computing
Intl. Conf. on the Internet of Things
Intl. Conf. on Internet of Things, Big Data and Security
Intl. Conf. on Internet-of-Things Design and Implementation
Intl. Workshop on Smart Grid Security
Intl. Wksp on Human-centered Sensing, Networking, and Systems
PerCom Wksp On Security, Privacy And Trust In The Internet of Things
Conf. on Embedded Network Sensor Systems
Conf. of the IEEE Industrial Electronics Society
Conf. on Computer and Communications Security
Intl. Wireless Communications and Mobile Computing Conf.
Information Security Solutions Europe Conf.
Thematic Workshops of ACM Multimedia
Living in the Internet of Things: Cybersecurity of the IoT

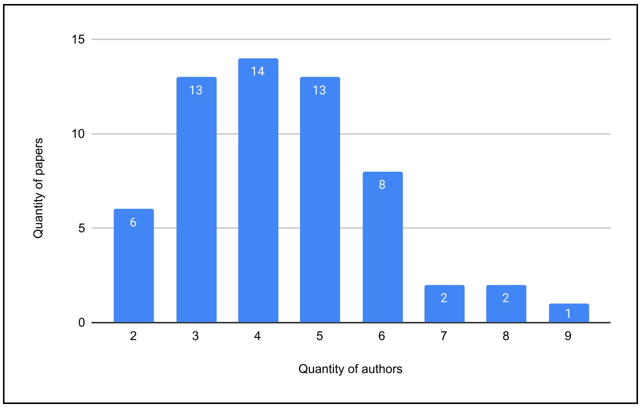


FIGURE 5. Quantity of papers by quantity of authors.

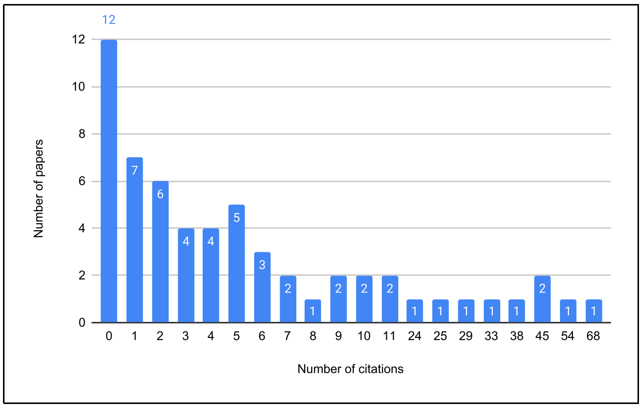


FIGURE 7. Quantity of papers by quantity of citations.

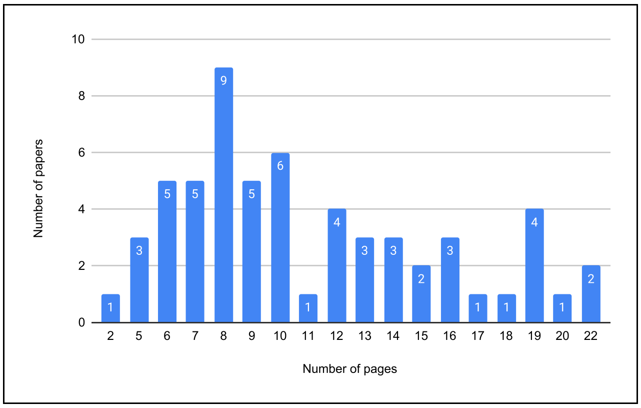


FIGURE 6. Quantity of papers by quantity of pages.

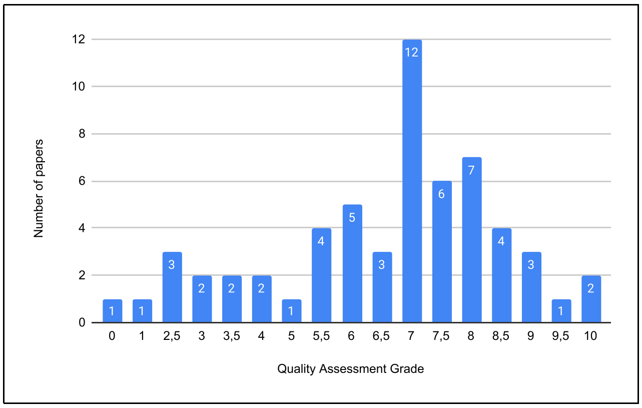


FIGURE 8. Quality assessment grades.

or case study, clear advantages and disadvantages of the technologies, and suggestions of future works.

Considering the TEE technologies applied in the proposed works, Fig. 10 presents the proportion of papers applying ARM TrustZone, Intel SGX, or none. "None" was considered when the authors do not mention a specific TEE solution. As seen, most of the papers applied TrustZone technology, which we could expect since it is more suitable for IoT devices due to the ARM architecture. We could also expect that most of the works have applied TrustZone or SGX, as both are the two most known TEE technologies available in the market.

TABLE 7. Journal names.

Tsinghua Science and Technology*
Trans. on Dependable and Secure Computing
The Journal of Supercomputing
Sensors
Design Automation for Embedded Systems
Cluster Computing*
Internet Computing - ICT for Smart Industries
Journal of Internet Services and Applications
Trans. on Computer-Aided Design of Integrated Circuits and Systems

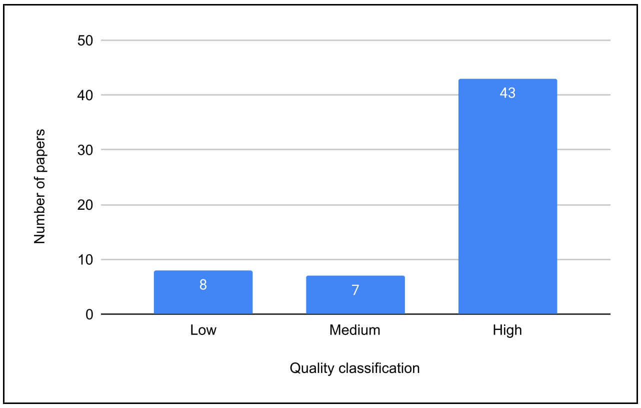


FIGURE 9. Quality classification.

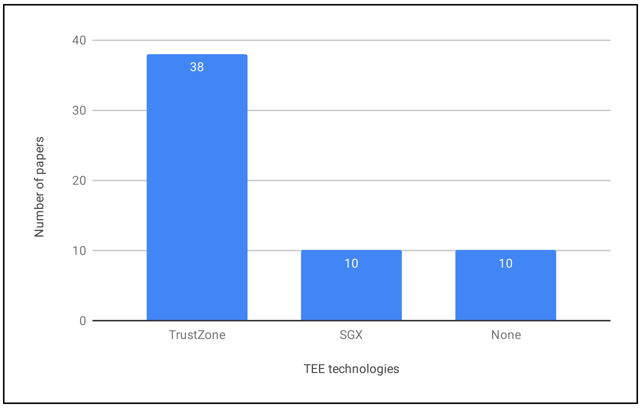


FIGURE 10. TEE technologies.

V. RESEARCH RESULTS

This section shows the collected information regarding each of the four research questions, i.e., IoT solutions, IoT scenarios, advantages/disadvantages, and future works.

A. IOT SOLUTIONS

All the IoT solutions approached in the selected papers can be classified in one of the following solution types:

- Application - considering all the works that propose security solutions to specific applications, such as protection for video applications or framework to develop secure solutions;
- OS (Operating System) - considering all the works that propose security solutions related to OS, such as protection for application execution or memory;
- Security - considering all the works that propose security mechanisms as solutions, such as authentication or attestation methods.

In Fig. 11, we can see the proportion of the number of works classified according to the solution types. As observed, the number of works addressing the three types of solutions is well distributed, presenting similar proportions. We can see a difference of just six papers between the greater bar (OS solutions) and the smaller one (Security solutions). The number of works addressing OS solutions is slightly higher than the number of works addressing Application solutions. In contrast, the latter is slightly higher than the number of works addressing Security solutions.

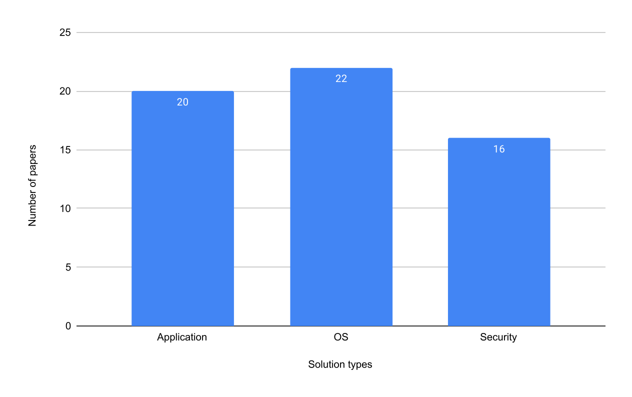


FIGURE 11. Solution types.

Among the works classified as OS, we found five types of solutions: hypervisor, application execution, communication, memory dump, and access control. As we can see in Fig. 12, the majority of OS solutions propose protection for the execution of the application. We list all the OS solution types in Table 9.

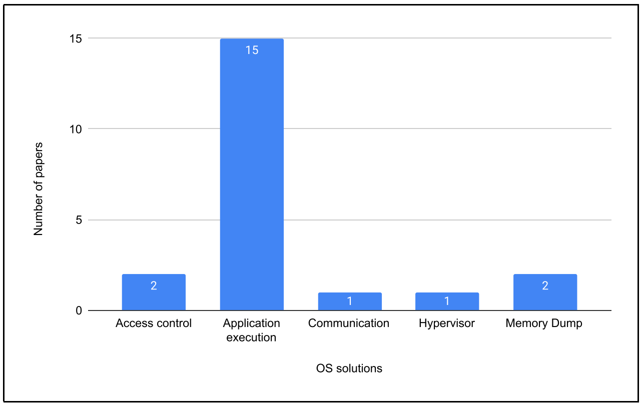


FIGURE 12. OS solutions.

We found six types of solutions for the works classified as Application: system, framework, architecture, platform, communications checker, and comparison with other solutions (multi-party computation). In Fig. 13, we can see that most

TABLE 8. Percentage of papers according to quality assessment questions.

Questions/Answers	Yes	Partially	No
Is the text well organized and clear (easy to understand)?	55,93 %	40,68 %	3,39 %
Is the motivation and objective well described?	76,27 %	20,34 %	3,39 %
Does the paper present an application scenario or case study?	42,37 %	47,46 %	10,17 %
Is the methodology clear (easy to understand and replicate)?	32,20 %	55,93 %	11,86 %
Does the paper have many references and good related works?	52,54 %	42,37 %	5,08 %
Does the study present some implementation and practical results?	71,19 %	18,64 %	10,17 %
Does the paper clearly present any advantages/disadvantages regarding the technology used?	30,51 %	54,24 %	15,25 %
Do the authors present good validation?	25,42 %	52,54 %	22,03 %
Are the results clear enough (explicit and well discussed/evaluated)?	37,29 %	47,46 %	15,25 %
Do the authors suggest future works?	28,81 %	15,25 %	55,93 %

TABLE 9. OS solution proposals.

Protection for legacy applications [8]	Access control between trusted and untrusted worlds [23]
Protection for embedded devices [33]	Hardware isolation architecture [41]
Protection for applications execution [34]	General-purpose trusted computing platform [27]
Protection for real-time OS [43]	Trusted OS [29]
Protection against data leakage [70]	Memory dump mechanism [30]
Protection for memory integrity [45]	Cache-assisted secure execution [39]
Protection for legacy applications [48]	Runtime security architecture [19]
Protection for services [72]	Kernel protection mechanism [51]
Protection for SoC communication bus [66]	TCB measurements management [42]
Trusted I/O [74]	Hypervisor [57]
Secure device access method [26]	Secure mobile device framework [54]

TABLE 10. Application solution proposals.

Protection for flow tracking application [59]	Trusted framework to develop IoT applications [36]
Protection for health data [73]	Secure architecture for P2P scenarios [24]
Protection for edge computing [63]	Comparison between TEE and secure multi-party computation application [62]
Protection for video application [46], [69]	Secure logger [13], [76]
Protection for system analyser [49]	Societal model for IoT security [68]
Protection for location-based services [60]	Trusted auditor [47]
Protection for data dissemination [61]	Data encryption mechanism [56]
Protection for data management [14]	Data protection (app) [71]
Protection for data aggregation (app) [75]	Checker for Industrial gateway communications [52]

TABLE 11. Security solution proposals.

Lightweithg anonymous authentication [25]	Remote attestation mechanism [38]
Device snapshot authentication system [32]	Control-flow attestation [37]
Authentication scheme [58]	Remote attestation and channel protection [44]
Secure authentication and key distribution [67]	Boot attestation [53]
Protection for data through authentication [50]	Protection and attestation for remote terminal [65]
Device private keys protection architecture [28]	Remote attestation [64]
Keys derivation from device characteristics [31]	Authenticity detection service [40]
Keys protection against cold boot attacks [55]	Cache rootkit exploiting TrustZone [35]

of the Application solutions propose some system to protect data. Table 10 shows all the application solution types.

Lastly, we found six types of solutions for the works classified as Security: authentication, attestation, authenticity, keys derivation, keys distribution, and rootkit. We can see in Fig. 14 that authentication and attestation proposals are prevailing in the Security solutions. We show all the security solution types in Table 11.

B. IOT SCENARIOS

Among all the selected papers, approximately only 24% presented IoT scenarios. All the identified scenarios are listed below:

- Automotive - device access for automotive software [26] and secure communication between vehicle infotainment system and user devices [67];
- Healthcare - secure heartbeat sensor application [36], critical medical services [72] and healthcare data monitoring system [73];
- Smart metering - data dissemination [61] and data aggregation [75];
- Video application - video surveillance devices [46] and duplicate removal operations [69];
- Edge computing - industrial IoT edge devices [43] and edge-cloud communications [63];
- Industrial maintenance - secure smart service for industrial maintenance scenarios [32];

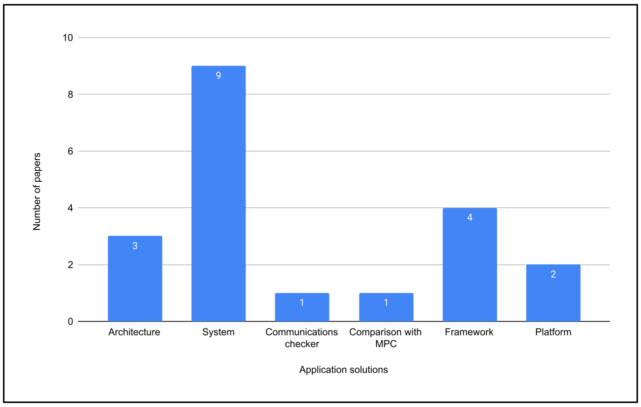


FIGURE 13. Application solutions.

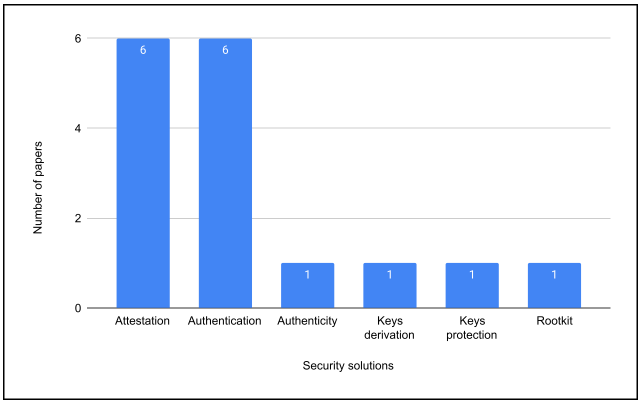


FIGURE 14. Security solutions.

- Payment system - trusted OS for mobile payment system [29];
- Smart home - device authentication for smart home/ smart grid scenario [28].

The rest of the papers (about 76%) did not present any IoT scenario. These works are related to embedded systems or mobile devices/smartphones solutions.

C. TEE ADVANTAGES AND DISADVANTAGES

Only 41 % of the selected papers presented any advantage or disadvantage regarding the adoption of TEEs. The TEE vendors, e.g., ARM and Intel, already present many of the general advantages, such as hardware isolation (normal world and the secure world) and memory protection [8], [13], [14], [19], [29], [32], [42], [74]–[76]. This helps to prevent malicious applications and to isolate sensitive data [19], and enables to protect code running in the secure world [8], [40] as well as to protect against software attacks [76]. Another considered advantages are low performance overhead [46]–[48], [50], [54], [59]–[62], [65], low power consumption overhead [46] and low latency overhead [54].

One mentioned disadvantage is the well-known fact that TEE is not resistant to side-channel attacks [32], [54]. According to Zhang *et al.* [35], the incoherence between cache in the normal and secure worlds is a disadvantage related to TrustZone since someone can explore it as a vulnerability. Although some works presented low overheads, some authors consider that some operations with TEEs present high performance overhead [44], [49], e.g., when using the SGX monotonic counter [13], or high power consumption overhead [47]. For others, the need to use specific hardware is also considered a disadvantage [75].

D. SUGGESTED FUTURE WORKS

Only 48 % of the selected papers presented suggestions for future work. In general, all the suggestions relate to improvements or evaluations of the proposals, integration with other mechanisms, or comparisons with other solutions.

Osterhues *et al.* [24] suggested integrating the secure architecture for P2P scenarios with the OpenMoko project. For the general-purpose trusted computing platform, Feng *et al.* [27] suggested implementing it in a specific board, improving the prototype, adding a TPM 2.0 module, and implementing Trusted Applications. Yang *et al.* proposed to improve and evaluate their Trusted OS [29]. Lesjak *et al.* [32] recommended comparing other solutions with their device snapshot authentication system.

Kylanpaa and Rantala [38] suggested implementing their remote attestation mechanism using Qemu and a real TrustZone to modify the TEE initialization and change the TA loading to the bootloader phase, to extend the measurement mechanism and to improve the keys storage or add a Public Key Infrastructure. Zhang *et al.* [35] proposed to test their cache rootkit in architectures different from TrustZone and to improve the defense mechanisms.

Pinto *et al.* [57] suggested, for their proposed hypervisor (LTZVisor), evaluating real-time aspects with short-term and long-term tests, investigating timing interferences and sources of non-determinism, and extending the solution for new platforms. Zhang *et al.* [55] recommended extending the keys' protection against cold boot attacks to store multiple keys and ensure a parallel encryption process. Park and Kim [42] suggested extending the TCB measurements management system and applying SGX to the miners.

Shepherd *et al.* [76] proposed group logging schemes for multiple devices and comparing the TEE performance for the secure logger solution. Pinto *et al.* [43] proposed implementing the protection for real-time OS in edge devices and integrating it with other hardware anchors. For memory integrity protection, Chang *et al.* [45] suggested improving its algorithms and process, implementing it with the B method, and formalizing it. Liang *et al.* [73] proposed integrating health data protection with a blockchain-based access control scheme.

Guan *et al.* [48] proposed introducing another level of virtual memory to their solution, protection for legacy applications, and integrating it with existing shielding mechanisms. Yalew *et al.* [40] indicated to optimize their authenticity detection service. Zhao *et al.* [50] suggested providing secure communication for their authentication-based data protection solution. Qin *et al.* [52] recommended adding security enforcement policy for their industrial gateway communications checker. Kulkarni *et al.* [60] proposed to evaluate their solution, a protection system for location-based services, and compare it with other approaches.

Schulz *et al.* [53] suggested supporting additional device types, establishing trusted channels, and extending and refining the protocol for the boot attestation process. For the remote attestation and channel protection solution, Shepherd *et al.* [44] proposed establishing a trusted channel shared between multiple devices, extending the performance evaluation, and reducing critical sizes and computational overhead with ECDH cryptography. Ayoade *et al.* [14] suggesting adding private blockchains for their data management protection system. Cao *et al.* [70], for their mechanism against data leakage, proposed to adjust the size of the sliding window for individual processes and to find a smarter page replacement algorithm.

Silva *et al.* [75] recommended combining different approaches to strengthen their data aggregation system's security and privacy. Siddiqui *et al.* [66] proposed securing configuration and providing hardware resources segregation in their SoC communication bus. Raes *et al.* [72] suggested exploring use cases opportunities for their services protection solution. Peters *et al.* [74] recommended extending their trusted Bluetooth I/O to address other I/O paths (e.g., WiFi and NFC) and evaluating its performance cost. Lee and Lee [67] suggested adding a new authentication method using blockchain to secure authentication and key distribution system. Guan *et al.* [8] proposed improving the signing mechanism of their solution, protecting legacy applications, and verifying a manifest's authenticity.

E. THREATS TO VALIDITY

To deal with the internal validity, i.e., get confidence for the whole performed process, we adopted a peer review process, minimizing each reviewer's subjective bias since pairs of reviewers carried out the overall selection process. Although each reviewer's knowledge is distinct from the others, each pair of reviewers had a reviewer with more knowledge on the study topic. Whenever doubts arose during the selection process, both reviewers achieved a consensus regarding including or excluding a paper. We think the defined search string is comprehensive enough to comprise the central primary studies to extract the answers for the research questions.

To minimize possible problems with the external validity, i.e., the generalization of the found results, we decided not to limit the search for specific journals and conferences. This way, we included papers from any conferences and journals, avoiding removing some primary studies just because of the publication venue. Furthermore, we also avoided excluding papers with a low score since the score could be skewed according to the reviewers' subjective bias, even following the quality assessment guidelines.

Related to conclusion validity and to avoid bias, a reviewer checked the data collected by another one for each selected paper. This way, if the proposed protocol is applied, following these considerations to replicate this study, we are confident that the same results can be achieved.

Lastly, since this study's focus is the Internet of Things applications, we could restrict the search string to involve only the TrustZone technology since it is the only TEE available in the market that comes in some IoT devices due to the ARM processor. If we did this, we would be excluding all the solutions that consider the Intel SGX to improve security in edge/fog/cloud-based IoT scenarios. This way, to avoid problems with the construct validity, we elaborated the search string to include all TEE solutions applied to IoT applications in distinct scenarios (edge, fog, and cloud).

F. CHALLENGES AND DIRECTIONS

Next, we present a discussion with respect to challenges and directions regarding the use of TEEs considering the following topics: *vulnerabilities*, *development complexity*, *remote attestation*, *communication channel* and *solutions integration*.

Concerning *vulnerabilities*, one of the main challenges regarding the adoption of TEEs, regardless of the application scenario, is the known vulnerability to side-channel attacks. Although this kind of attack is considered a bit complex to execute, Demanding a specialized level of technical knowledge, we should always consider it as a possible threat. Given this problem, technical artifices can be proposed to mitigate or avoid this threat. Besides, concerns remain with code running inside a trusted application since it may contain vulnerabilities that unauthorized parties can explore to compromise the TEE system. Thus, developers must continue following the recommendations to secure code, avoiding, for instance, buffer overflows, race conditions, and uninitialized variables. This can also be a research topic to be explored.

In the context of the *development complexity*, another known challenge is the learning curve needed to develop trusted applications with either TrustZone or SGX. Efforts, in this sense, can also be applied to ease the development of trusted applications. For example, Scone [77] facilitates the deployment of SGX applications through containers, Python SGX [78] provided means to develop SGX applications using Python (it is not maintained anymore) and Rust OP-TEE TrustZone SDK [79] enables the development of TrustZone applications using Rust.

For *remote attestation*, applications running inside a TEE should provide a means that attests their trustworthiness to any interested third-party, i.e., that proves they are running inside real TEE hardware. For this, the SGX technology provides a remote attestation protocol in which the third-party application challenges the trusted application, supposedly running inside an SGX enclave. The trusted application replies with specific information from the enclave, allowing the third-party application to verify this content with the Intel Attestation Service (IAS). After a successful verification with the IAS, both parties can establish a secure communication channel once the trustworthiness is proved since they generate an ECDH (Elliptic-Curve Diffie-Helmann) shared key during the process. As TrustZone does not provide a specific and standardized development kit, like SGX, each Trusted OS should provide its remote attestation process. Thus, remote attestation is also a good topic for future researches since some Trusted OSes, such as OP-TEE [80], still do not provide such a mechanism.

The *communication* between the untrusted applications and the trusted applications, i.e., between the normal world running a Rich OS (e.g., Linux or Android) and the trusted world running a Trusted OS (e.g., OP-TEE or Kinibi [81]), can also be a target of attacks. To mitigate this, the platforms should implement secure communication giving special attention to the shared memory between both worlds. This extends the possibilities for new research works. In this sense, the platforms must avoid unauthorized processes running inside the normal world to access information running inside the trusted world.

Considering the *solutions integration*, we can notice that TEE is being applied together with other security solutions and mechanisms. For instance, some works that apply TEE to protect data also apply blockchain [14], [42], [71], [73]. In general, the data are encrypted to be processed only inside a TEE application, being protected in rest or transit. Thus, the data hashes can be stored in the blockchain for auditing operations, verifying the data integrity once the trusted applications process them.

VI. RELATED WORK

Manifavas *et al.* [82] presented a survey considering many EU-funded research projects that approach embedded systems security. Some presented works consider hardware-related security modules, such as TPM, Secure Element, JavaCard, TrustZone, and PUFs. Other presented works consider the security aspects of virtualization solutions, with some exploiting TPM or TrustZone. Lightweight cryptographic mechanisms are also approached, referencing works that present solutions in this sense. The authors still present some challenges regarding the security in network technologies, including node attestation and authentication, privacy and anonymity, access control, secure routing, secure aggregation, and intrusion detection. Lastly, some proposed middleware and architectures are presented.

Yi *et al.* [83] discussed security and privacy issues related to fog computing. The authors briefly mentioned challenges the following concerns that solutions should address in fog scenarios: authentication, trust models (Secure Element, Trusted Execution Environments and Trusted Platform Modules), rogue nodes, network, and storage security, secure processing, data privacy, access control mechanisms, and intrusion detection techniques.

Zhang *et al.* [84] presented a survey related to privacy leakage and protection in mobile applications, giving some research directions. They presented works that deal with identifying suspicious behaviors in mobile apps, which can indicate privacy leakage. Some privacy leakage detection techniques (static data propagation analysis, dynamic taint tracking, and a combination of both) and privacy protection approaches (permission and access control, sandboxing, and isolation) are discussed. The authors present TEE to protect data privacy, considering the TrustZone technology as the most popular TEE solution in the market.

Shepherd *et al.* [85] presented the concepts related to the evolution of secure and trusted computing in the IoT context. They described TPM and Secure Element as hardware solutions, Java Card as an example of virtualization, and Intel Trusted Execution Technology as an example that uses a hypervisor. TEE and some software components are also presented as a way to enable the development and deployment of trusted applications. Many of these components are based on TrustZone (e.g., Trustonic, OP-TEE, and Trusty OS). The authors conclude that developers can combine some of the described technologies to improve the security of the applications.

Sau *et al.* [86] presented a survey regarding crypto processors and their applications, including the definition of trusted execution environments, trusted platform modules, and hardware security model. The authors present a brief description of TrustZone and SGX and describe some methods for providing security to the boot process (secure, trusted, certified, measured, verified, and authenticated boot). Some concerns with side-channel (timing analysis, power analysis, and template), fault analysis, and memory attacks are also described. According to the authors, the use of TEEs is among the main features required to secure hardware and software.

To the best of our knowledge, this is the first systematic literature review investigating the use of TEEs for cloud/fog-based IoT applications. Although the previous related work presented surveys regarding security topics, they are general surveys, not a systematic review, and they do not approach TEEs for IoT.

VII. CONCLUSION

Trusted Execution Environments have been applied to improve data security in different application scenarios, considering cloud, fog, and edge computing. We carried out a systematic literature review to analyze how this technology is applied to protect data in IoT applications. For this, we defined a protocol to ease the replication of this study, and, according to it, we selected 58 works from the main scientific repositories, considering journal and conference papers.

We presented the overall results related to the collected data, including quantitative information and the research questions' research results. Answering each of the four defined questions, based on the papers' information, we discussed the IoT solutions and scenarios applications of TEEs. We then identified the advantages and disadvantages of the use of TEEs and the suggested future works. We grouped the IoT solutions into three main types: OS, Application, and Security. The OS solutions propose protection related to some components directly related to the OSes; the Application solutions propose data protection in typical applications; and the Security solutions propose protection to security mechanisms, such as authentication and attestation. Besides, we discussed some challenges and directions for new researches on the adoption of TEEs.

As future work for this systematic literature review, we envisage its extension considering the snowballing technique to add even more works. Also, as we recently noticed a growing number of publications, an update with more papers can be performed shortly. Another possibility is replicating this study for other application scenarios since we focused on cloud/fog-based IoT scenarios.

REFERENCES

[1] K. Ashton. (2009). *That 'Internet of Things' Thing*. Accessed: Jan. 7, 2019. [Online]. Available: https://www.rfidjournal.com/articles/view?4986

[2] A. Gabbai. (2015). *Kevin Ashton Describes the Internet of Things*. Accessed: Jan. 7, 2019. [Online]. Available: https://bit.ly/2PvshSn

[3] F. Alsubaei, A. Abuhussein, and S. Shiva, "Quantifying security and privacy in Internet of Things solutions," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Taipei, Taiwan, Apr. 2018, pp. 1–6.

[4] E. Bertino, "Data security and privacy: Concepts, approaches, and research directions," in *Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Atlanta, GA, USA, Jun. 2016, pp. 400–407.

[5] R. van der Meulen. *Gartner Says 8.4 Billion Connected 'Things' Will Be in Use in 2017, Up 31 Percent From 2016*. Accessed: Jan. 13, 2019. [Online]. Available: https://gtnr.it/3snbJL9

[6] M. Henze, L. Hermerschmidt, D. Kerpen, R. Häußling, B. Rumpe, and K. Wehrle, "A comprehensive approach to privacy in the cloud-based Internet of Things," *Future Gener. Comput. Syst.*, vol. 56, pp. 701–718, Mar. 2016.

[7] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the Internet of Things," *IEEE Internet Comput.*, vol. 14, no. 1, pp. 44–51, Jan. 2010.

[8] L. Guan, C. Cao, P. Liu, X. Xing, X. Ge, S. Zhang, M. Yu, and T. Jaeger, "Building a trustworthy execution environment to defeat exploits from both cyber space and physical space for ARM," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 3, pp. 438–453, May 2019.

[9] D. Le Métayer, "Privacy by design: A formal framework for the analysis of architectural choices," in *Proc. Conf. Data Appl. Secur. Privacy*, San Antonio, TX, USA, 2013, pp. 95–104.

[10] P. Y. S. Barbosa, "Privacy by evidence: A software development methodology to provide privacy assurance," Ph.D. dissertation, Univ. Federal Campina Grande, Campina Grande, Brazil, 2018.

[11] J. Werner, C. M. Westphall, and C. B. Westphall, "Cloud identity management: A survey on privacy strategies," *Comput. Netw.*, vol. 122, pp. 29–42, Jul. 2017.

[12] R. H. Weber, "Internet of Things: Privacy issues revisited," *Comput. Law Secur. Rev.*, vol. 31, no. 5, pp. 618–627, Oct. 2015.

[13] H. Nguyen, R. Ivanov, L. T. X. Phan, O. Sokolsky, J. Weimer, and I. Lee, "LogSafe: Secure and scalable data logger for IoT devices," in *Proc. IEEE/ACM 3rd Int. Conf. Internet-Things Design Implement. (IoTDI)*, Orlando, FL, USA, Apr. 2018, pp. 141–152.

[14] G. Ayoade, V. Karande, L. Khan, and K. Hamlen, "Decentralized IoT data management using BlockChain and trusted execution environment," in *Proc. IEEE Int. Conf. Inf. Reuse Integr. (IRI)*, Salt Lake City, UT, USA, Jul. 2018, pp. 15–22.

[15] A. D. Rayome. *DDoS Attacks Increased 91% in 2017 Thanks to IoT*. Accessed: Jan. 13, 2019. [Online]. Available: https://tek.io/2PoPk1i

[16] S. Weagle. *The Rise of IoT Botnet Threats and DDoS Attacks*. Accessed: Jan. 13, 2019. [Online]. Available: https://bit.ly/2Qs4bIL

[17] J. Fruhlinger. *The Mirai Botnet Explained: How Teen Scammers and CCTV Cameras Almost Brought Down the Internet*. Accessed: Jan. 13, 2019. [Online]. Available: https://bit.ly/3snxJ8v

[18] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," in *Proc. Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Helsinki, Finland, vol. 1, 2015, pp. 57–64.

[19] R. Chang, L. Jiang, W. Chen, Y. Xie, and Z. Lu, "A trustenclave-based architecture for ensuring run-time security in embedded terminals," *Tsinghua Sci. Technol.*, vol. 22, no. 5, pp. 447–457, Sep. 2017.

[20] ARM. *TruztZone—ARM Developer*. Accessed: Jan. 13, 2021. [Online]. Available: https://developer.arm.com/technologies/trustzone

[21] Intel. *Intel Software Guard Extensions (Intel SGX)*. Accessed: Jan. 13, 2021. [Online]. Available: https://software.intel.com/en-us/sgx

[22] Intel. *Intel Software Guard Extensions (Intel SGX)*. Accessed: Jan. 13, 2021. [Online]. Available: https://software.intel.com/sgx/details

[23] X. Yan-Ling, P. Wei, and Z. Xin-Guo, "Design and implementation of secure embedded systems based on trustzone," in *Proc. Int. Conf. Embedded Softw. Syst.*, Chengdu, China, 2008, pp. 136–141.

[24] A. Osterhues, A.-R. Sadeghi, M. Wolf, C. Stüble, and N. Asokan, "Securing peer-to-peer distributions for mobile devices," in *Proc. Int. Conf. Inf. Secur. Pract. Exper.* Sydney, NSW, Australia: Springer, 2008, pp. 161–175.

[25] C. Wachsmann, L. Chen, K. Dietrich, H. Löhr, A.-R. Sadeghi, and J. Winter, "Lightweight anonymous authentication with TLS and DAA for embedded mobile devices," in *Proc. Int. Conf. Inf. Secur.* Boca Raton, FL, USA: Springer, 2011, pp. 84–98.

[26] S. W. Kim, C. Lee, M. Jeon, H. Y. Kwon, H. W. Lee, and C. Yoo, "Secure device access for automotive software," in *Proc. Int. Conf. Connected Vehicles Expo (ICCVE)*, Las Vegas, NV, USA, Dec. 2013, pp. 177–181.

[27] W. Feng, D. Feng, G. Wei, Y. Qin, Q. Zhang, and D. Chang, "TEEM: A user-oriented trusted mobile device for multi-platform security applications," in *Proc. Int. Conf. Trust Trustworthy Comput.* London, U.K.: Springer, 2013, pp. 133–141.

[28] A. J. Paverd and A. P. Martin, "Hardware security for device authentication in the smart grid," in *Proc. Int. Workshop Smart Grid Secur.* Berlin, Germany: Springer, 2013, pp. 72–84.

[29] X. Yang, P. Shi, B. Tian, B. Zeng, and W. Xiao, "Trust-E: A trusted embedded operating system based on the ARM trustzone," in *Proc. Int. Conf. Autonomic Trusted Comput.*, Bali, Indonesia, 2014, pp. 495–501.

[30] H. Sun, K. Sun, Y. Wang, J. Jing, and S. Jajodia, "TrustDump: Reliable memory acquisition on smartphones," in *Proc. Eur. Symp. Res. Comput. Secur.*, vol. 8712. Vienna, Austria: Springer, 2014, pp. 202–218.

[31] V. Leest, R. Maes, G.-J. Schrijen, and P. Tuyls, "Hardware intrinsic security to protect value in the mobile market," in *ISSE 2014 Securing Electronic Business Processes*. Cham, Switzerland: Springer, Jan. 2014, pp. 188–198.

[32] C. Lesjak, D. Hein, and J. Winter, "Hardware-security technologies for industrial IoT: TrustZone and security controller," in *Proc. 41st Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Yokohama, Japan, Nov. 2015, pp. 2589–2595.

[33] K. Markantonakis, R. N. Akram, and M. G. Msgna, "Secure and trusted application execution on embedded devices," in *Proc. Int. Conf. Inf. Technol. Commun.* Bucharest, Romania: Springer, 2015, pp. 3–24.

[34] R. Chang, L. Jiang, Q. Yin, W. Liu, and S. Zhang, "A novel method of APK-based automated execution and traversal with a trusted execution environment," in *Proc. 12th Int. Conf. Comput. Intell. Secur. (CIS)*, Wuxi, China, Dec. 2016, pp. 254–258.

[35] N. Zhang, H. Sun, K. Sun, W. Lou, and Y. T. Hou, "CacheKit: Evading memory introspection using cache incoherence," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Saarbruecken, Germany, Mar. 2016, pp. 337–352.

[36] A. Yadav, N. Rakesh, S. Pandey, and R. K. Singh, "IoTEE—An integrated framework for rapid trusted IoT application development," in *Proc. IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, Bangalore, India, May 2016, pp. 1829–1834.

[37] T. Abera, N. Asokan, L. Davi, J.-E. Ekberg, T. Nyman, A. Paverd, A.-R. Sadeghi, and G. Tsudik, "C-FLAT: Control-flow attestation for embedded systems software," in *Proc. Conf. Comput. Commun. Secur.*, Vienna, Austria, 2016, pp. 743–754.

[38] M. Kylanpaa and A. Rantala, "Remote attestation for embedded systems," in *Proc. Int. Workshop Secur. Ind. Control Syst. Cyber-Phys. Syst.*, 2016, pp. 79–92.

[39] N. Zhang, K. Sun, W. Lou, and Y. T. Hou, "CaSE: Cache-assisted secure execution on ARM processors," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Jose, CA, USA, May 2016, pp. 72–90.

[40] S. D. Yalew, P. Mendonca, G. Q. Maguire, S. Haridi, and M. Correia, "TruApp: A TrustZone-based authenticity detection service for mobile apps," in *Proc. IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Rome, Italy, Oct. 2017, pp. 1–9.

[41] H. Dai and K. Chen, "OPTZ: A hardware isolation architecture of multitasks based on TrustZone support," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. With Appl., IEEE Int. Conf. Ubiquitous Comput. Commun. (ISPA/IUCC)*, Guangzhou, China, Dec. 2017, pp. 391–395.

[42] J. Park and K. Kim, ''TM-Coin: Trustworthy management of TCB measurements in IoT,'' in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Kona, HI, USA, Mar. 2017, pp. 654–659.

[43] S. Pinto, T. Gomes, J. Pereira, J. Cabral, and A. Tavares, ''IIoTEED: An enhanced, trusted execution environment for industrial IoT edge devices,'' *IEEE Internet Comput.*, vol. 21, no. 1, pp. 40–47, Jan. 2017.

[44] C. Shepherd, R. N. Akram, and K. Markantonakis, ''Establishing mutually trusted channels for remote sensing devices with trusted execution environments,'' in *Proc. 12th Int. Conf. Availability, Rel. Secur.*, Reggio Calabria, Italy, Aug. 2017, pp. 1–10.

[45] R. Chang, L. Jiang, W. Chen, Y. Xiang, Y. Cheng, and A. Alelaiwi, ''MIPE: A practical memory integrity protection method in a trusted execution environment,'' *Cluster Comput.*, vol. 20, no. 2, pp. 1075–1087, Jun. 2017.

[46] X. Feng, M. Ye, V. Swaminathan, and S. Wei, ''Towards the security of motion detection-based video surveillance on IoT devices,'' in *Proc. Thematic Workshops ACM Multimedia*, Mountain View, CA, USA, 2017, pp. 228–235.

[47] S. Mirzamohammadi, J. A. Chen, A. A. Sani, S. Mehrotra, and G. Tsudik, ''Ditio: Trustworthy auditing of sensor activities in mobile & IoT devices,'' in *Proc. Conf. Embedded Netw. Sensor Syst.*, Delft, The Netherlands, 2017, pp. 1–14.

[48] L. Guan, P. Liu, X. Xing, X. Ge, S. Zhang, M. Yu, and T. Jaeger, ''TrustShadow: Secure execution of unmodified applications with ARM TrustZone,'' in *Proc. 15th Annu. Int. Conf. Mobile Syst., Appl., Services*, Niagara Falls, NY, USA, Jun. 2017, pp. 488–501.

[49] S. D. Yalew, G. Q. Maguire, S. Haridi, and M. Correia, ''T2Droid: A TrustZone-based dynamic analyser for Android applications,'' in *Proc. Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Sydney, NSW, Australia, Aug. 2017, pp. 240–247.

[50] B. Zhao, Y. Xiao, Y. Huang, and X. Cui, ''A private user data protection mechanism in trustzone architecture based on identity authentication,'' *Tsinghua Sci. Technol.*, vol. 22, no. 2, pp. 218–225, Apr. 2017.

[51] X. Zheng, Y. He, J. Ma, G. Shi, and D. Meng, ''TZ-KPM: Kernel protection mechanism on embedded devices on hardware-assisted isolated environment,'' in *Proc. Int. Conf. High Perform. Comput. Commun.*, Sydney, NSW, Australia, 2016, pp. 663–670.

[52] Y. Qin, Y. Zhang, and W. Feng, ''TICS: Trusted industry control system based on hardware security module,'' in *Proc. Int. Symp. Cyberspace Saf. Secur.* Xi'an, China: Springer, Oct. 2017, pp. 485–493.

[53] S. Schulz, A. Schaller, F. Kohnhäuser, and S. Katzenbeisser, ''Boot attestation: Secure remote reporting with off-the-shelf IoT sensors,'' in *Proc. Eur. Symp. Res. Comput. Secur.* Oslo, Norway: Springer, 2017, pp. 437–455.

[54] Y.-K. Lee, J.-N. Kim, K.-S. Lim, and H. Yoon, ''Secure mobile device structure for trust IoT,'' *J. Supercomput.*, vol. 74, no. 12, pp. 6646–6664, Dec. 2018.

[55] X. Zhang, Y.-A. Tan, Y. Xue, Q. Zhang, Y. Li, C. Zhang, and J. Zheng, ''Cryptographic key protection against FROST for mobile devices,'' *Cluster Comput.*, vol. 20, no. 3, pp. 2393–2402, Sep. 2017.

[56] H. Wirtz, T. Zimmermann, M. Ceriotti, and K. Wehrle, ''Encrypting data to pervasive contexts,'' in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Kona, HI, USA, Mar. 2017, pp. 309–315.

[57] S. Pinto, J. Pereira, T. Gomes, A. Tavares, and J. Cabral, ''LTZVisor: TrustZone is the key,'' in *Proc. Euromicro Conf. Real-Time Syst.* Dagstuhl, Germany: Schloss Dagstuhl, 2017, pp. 1–22.

[58] H. Jiang, R. Chang, L. Ren, W. Dong, L. Jiang, and S. Yang, ''An effective authentication for client application using ARM trustzone,'' in *Proc. Int. Conf. Inf. Secur. Pract. Exper.* Melbourne, VIC, Australia: Springer, Dec. 2017, pp. 802–813.

[59] M. A. Wahab, P. Cotret, M. N. Allah, G. Hiet, V. Lapotre, and G. Gogniat, ''ARMHEx: A hardware extension for DIFT on ARM-based SoCs,'' in *Proc. 27th Int. Conf. Field Program. Log. Appl. (FPL)*, Ghent, Belgium, Sep. 2017, pp. 1–7.

[60] V. Kulkarni, B. Chapuis, and B. Garbinato, ''Privacy-preserving location-based services by using Intel SGX,'' in *Proc. Int. Workshop Hum.-Centered Sens., Netw., Syst.*, Delft, The Netherlands, 2017, pp. 13–18.

[61] L. Sampaio, F. Silva, A. Souza, A. Brito, and P. Felber, ''Secure and privacy-aware data dissemination for cloud-based applications,'' in *Proc. 10th Int. Conf. Utility Cloud Comput.*, Austin, TX, USA, Dec. 2017, pp. 47–56.

[62] R. Ankele and A. Simpson, ''On the performance of a trustworthy remote entity in comparison to secure multi-party computation,'' in *Proc. Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Sydney, NSW, Australia, 2017, pp. 1115–1122.

[63] R. Pettersen, H. D. Johansen, and D. Johansen, ''Secure edge computing with ARM trustzone,'' in *Proc. Int. Conf. Internet Things, Big Data Secur.* Porto, Portugal: SciTePress, 2017, pp. 102–109.

[64] N. Ahmed, M. A. Talib, and Q. Nasir, ''Program-flow attestation of IoT systems software,'' in *Proc. 15th Learn. Technol. Conf. (L&T)*, Jeddah, Saudi Arabia, Feb. 2018, pp. 67–73.

[65] J. Wang, Z. Hong, Y. Zhang, and Y. Jin, ''Enabling security-enhanced attestation with intel SGX for remote terminal and IoT,'' *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 1, pp. 88–96, Jan. 2018.

[66] F. Siddiqui, M. Hagan, and S. Sezer, ''Embedded policing and policy enforcement approach for future secure IoT technologies,'' in *Proc. Living Internet Things, Cybersecur. IoT*, London, U.K., 2018, p. 10.

[67] S. Lee and J.-H. Lee, ''TEE based session key establishment protocol for secure infotainment systems,'' *Design Autom. Embedded Syst.*, vol. 22, no. 3, pp. 215–224, Sep. 2018.

[68] H. Tsunoda and G. M. Keeni, ''Feasibility of societal model for securing Internet of Things,'' in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Valencia, Spain, Jun. 2017, pp. 541–546.

[69] H. Yan, X. Li, Y. Wang, and C. Jia, ''Centralized duplicate removal video storage system with privacy preservation in IoT,'' *Sensors*, vol. 18, no. 6, p. 1814, Jun. 2018.

[70] C. Cao, L. Guan, N. Zhang, N. Gao, J. Lin, B. Luo, P. Liu, J. Xiang, and W. Lou, ''CryptMe: Data leakage prevention for unmodified programs on ARM devices,'' in *Proc. Int. Symp. Res. Attacks, Intrusions, Defenses*. Heraklion, Greece: Springer, 2018, pp. 380–400.

[71] N. Zhang, J. Li, W. Lou, and Y. Hou, ''Privacyguard: Enforcing private data usage with blockchain and attested execution,'' in *Proc. Eur. Symp. Res. Comput. Secur.* Guildford, U.K.: Springer, 2018, pp. 345–353.

[72] V. Raes, J. Vossaert, and V. Naessens, ''Development of an embedded platform for secure CPS services,'' in *Proc. Int. Workshop Secur. Privacy Requirements Eng.* Oslo, Norway: Springer, 2018, pp. 19–34.

[73] X. Liang, S. Shetty, J. Zhao, D. Bowden, D. Li, and J. Liu, ''Towards decentralized accountability and self-sovereignty in healthcare systems,'' in *Proc. Int. Conf. Inf. Commun. Secur.* Beijing, China: Springer, 2017, pp. 387–398.

[74] T. Peters, R. Lal, S. Varadarajan, P. Pappachan, and D. Kotz, ''BASTION-SGX: Bluetooth and architectural support for trusted I/O on SGX,'' in *Proc. Int. Workshop Hardw. Archit. Support Secur. Privacy*, Los Angeles, CA, USA, 2018, pp. 1–9.

[75] L. V. Silva, P. Barbosa, R. Marinho, and A. Brito, ''Security and privacy aware data aggregation on cloud computing,'' *J. Internet Services Appl.*, vol. 9, no. 1, pp. 1–13, Dec. 2018.

[76] C. Shepherd, R. N. Akram, and K. Markantonakis, ''EmLog: Tamper-resistant system logging for constrained devices with TEEs,'' in *Proc. Int. Conf. Inf. Secur. Theory Pract.* Heraklion, Greece: Springer, Dec. 2017, pp. 75–92.

[77] Scone Team. *Scone*. Accessed: Jan. 13, 2021. [Online]. Available: https://scontain.com/index.html?lang=en

[78] Adombeck. *Python SGX*. Accessed: Jan. 13, 2021. [Online]. Available: https://github.com/adombeck/python-sgx

[79] Mesalock Team. *Rust OP-TEE TrustZone SDK*. Accessed: Jan. 13, 2021. [Online]. Available: https://github.com/mesalock-linux/rust-optee-trustzone-sdk

[80] OP-TEE Team. *OP-TEE Trusted OS*. Accessed: Jan. 13, 2021. [Online]. Available: https://github.com/OP-TEE/optee_os

[81] Azeria Team. *Trustonic's Kinibi TEE Implementation*. Accessed: Jan. 13, 2020. [Online]. Available: https://azeria-labs.com/trustonics-kinibi-tee-implementation/

[82] C. Manifavas, K. Fysarakis, A. Papanikolaou, and I. Papaefstathiou, ''Embedded systems security: A survey of EU research efforts,'' *Secur. Commun. Netw.*, vol. 8, no. 11, pp. 2016–2036, Jul. 2015.

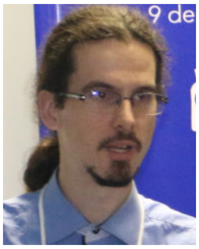
[83] S. Yi, Z. Qin, and Q. Li, ''Security and privacy issues of fog computing: A survey,'' in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.* Qufu, China: Springer, Aug. 2015, pp. 685–695.

[84] L. Zhang, D. Zhu, Z. Yang, L. Sun, and M. Yang, ''A survey of privacy protection techniques for mobile devices,'' *J. Commun. Inf. Netw.*, vol. 1, no. 4, pp. 86–92, Dec. 2016.

[85] C. Shepherd, G. Arfaoui, I. Gurulian, R. P. Lee, K. Markantonakis, R. N. Akram, D. Sauveron, and E. Conchon, ''Secure and trusted execution: Past, present, and future—A critical review in the context of the Internet of Things and cyber-physical systems,'' in *Proc. Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Tianjin, China, 2016, pp. 168–177.

[86] S. Sau, J. Haj-Yahya, M. M. Wong, K. Y. Lam, and A. Chattopadhyay, "Survey of secure processors," in *Proc. Int. Conf. Embedded Comput. Syst., Archit., Modeling, Simulation (SAMOS)*, Pythagorion, Greece, Jul. 2017, pp. 253–260.

DALTON CÉZANE GOMES VALADARES received the bachelor's, master's and D.Sc. degrees in computer science, and the M.B.A degree in project management. He did a technical course in informatics. He is currently a Professor with the Federal Institute of Pernambuco (IFPE) and a Researcher with the Embedded Laboratory, Federal University of Campina Grande (UFCG). He has over 15 years of experience in IT, having worked on several research and development projects assuming different roles, including a systems analyst, an embedded systems/a web developer, a quality/testing analyst, and a project manager. He currently develops research collaboration at ISE Group, Embedded Laboratory, UFCG, and at GPRSC, UTFPR.



NEWTON CARLOS WILL (Member, IEEE) graduated in information systems from the Federal University of Technology – Paraná (UTFPR), in 2007. He received the master's degree in electrical engineering from the UTFPR, in 2012, and the Ph.D. degree in computer science from the Federal University of Paraná State (UFPR), in 2020. He has been a Professor with the Computer Science Department, UTFPR, Dois Vizinhos, since 2015. He is currently the Head of the Computer Networks and Security Research Group (GPRSC), UTFPR. His main research interest includes computational security.

JEAN CAMINHA received the M.Sc. and Ph.D. degrees in electrical engineering from the Federal University of Campina Grande, Brazil, in 2008 and 2018, respectively. He also holds a specialization in innovation management at the University of New Mexico. He has been a Professor with the Federal University of Mato Grosso, Brazil, since 2010. He is the Head of the Information Technology Secretary, Federal University of Mato Grosso. His current research interests include artificial intelligence applied to the security of the Internet of Things and self-healing systems.

MIRKO BARBOSA PERKUSICH received the Ph.D. degree in computer science. He is currently a Research Manager at the VIRTUS Innovation Center. His current research interests include applying intelligent techniques, including recommender systems, to solve complex software engineering problems, with over 60 published articles.

ANGELO PERKUSICH (Member, IEEE) received the master's and Ph.D. degrees in electrical engineering from the Federal University of Paraiba, in 1987 and 1994, respectively. He was a Visiting Researcher with the Department of Computer Science, University of Pittsburgh, PA, USA, from 1992 to 1993. He has been a Professor with the Electrical Engineering Department (DEE), Federal University of Campina Grande (UFCG), since 2002. He is currently the Founder and the Director of the VIRTUS Innovation Center and the Embedded and Pervasive Computing Laboratory. He has over 400 articles published and advised 85 master thesis and 25 Ph.D. dissertations. His main research interests include embedded systems, software engineering, and cyber-physical systems.

KYLLER COSTA GORGÔNIO graduated in computer science from the Universidade Federal da Paraíba, in 1999. He received the master's degree in computer science from the Universidade Federal da Paraíba, in 2001, and the Ph.D. degree in software from the Universitat Politècnica de Catalunya, in 2010. He is currently a Professor with the Universidade Federal de Campina Grande. He has experience in computer science focusing on software engineering. His research interests include Petri nets, protocols, asynchronous communication mechanisms, coloured Petri nets, and model checking.