Received: 20 June 2019 | Revised: 24 July 2019 | Accepted: 21 August 2019 | Published on: 28 November 2019

DOI: 10.1002/spy2.93

SURVEY PAPER

WILEY

Recent trends in applying TPM to cloud computing

Shohreh Hosseinzadeh[1] | Bernardo Sequeiros[2] | Pedro R. M. Inácio[2] | Ville Leppänen[1]

[1]Department of Future Technologies, University of Turku, Turku, Finland

[2]Universidade da Beira Interior and Instituto de Telecomunicações, Covilhã, Portugal

Correspondence
Shohreh Hosseinzadeh, Department of Future Technologies, University of Turku, Vesilinnantie 5, Turku 20500, Finland.
Email: shohos@utu.fi

Funding information
Cyber Trust research program; DIMECC Oy; FCT/COMPETE/FEDER, POCI-01-0145-FEDER-030657, UID/EEA/50008/2019; FCT-Fundação para a Ciência e Tecnologia, SFRH/BD/133838/2017; Nokia Foundation; Tekes Finnish Funding Agency for Innovation; Tekniikan Edistämissäätiö

Abstract

Trusted platform modules (TPM) have become important safe-guards against variety of software-based attacks. By providing a limited set of cryptographic services through a well-defined interface, separated from the software itself, TPM can serve as a root of trust and as a building block for higher-level security measures. This article surveys the literature for applications of TPM in the cloud-computing environment, with publication dates comprised between 2013 and 2018. It identifies the current trends and objectives of this technology in the cloud, and the type of threats that it mitigates. Toward the end, the main research gaps are pinpointed and discussed. Since integrity measurement is one of the main usages of TPM, special attention is paid to the assessment of run time phases and software layers it is applied to.

KEYWORDS

cloud computing, security, trusted computing, trusted platform modules

1 | INTRODUCTION

The cloud computing paradigm has facilitated access to enterprises to various computing resources including storage, server, and application. Utilizing cloud services brings along competitive advantages, such as lower cost, higher performance, accessibility, and scalability. These advantages have motivated enterprises to increasingly use clouds to deliver services to their customers and it is critical to secure the cloud infrastructure and protect its data and computation from both insider and outsider attacks. There is a large body of research on the security threats endangering cloud computing, and the security solutions to overcome or minimize them.

Many of the recent attacks are performed by exploiting the vulnerabilities of the software-based security solutions. This indicates that security solutions that are merely software based do not purvey full-proof security anymore. Therefore, in order to provide more robust security, researchers and developers are leaning toward hardware-backed solutions such as trusted computing. Trusted computing technology is elaborated by the group known as trusted computing group (TCG)[1] to tackle the computer security problems via hardware security enhancements. This happens through utilizing a physical chip, called trusted platform module (TPM) that protects the software and applications from tampering running on systems where the chip is integrated, further ensuring that the system is functioning as expected.

1.1 Trusted platform module

Trusted computing (TC) refers to the technologies that use hardware-based roots of trust to improve computer security through hardware enhancements and modification of the associated software. TC establishes a secure environment known as trusted computing base (TCB) and provides trust and privacy. It guarantees that the system is secure and it behaves as expected. Various major hardware manufacturers have developed and promoted specifications for protecting computer resources from the malicious entities.

The TPM, which was conceived by the TCG consortium, is a tamper resilient coprocessor chip that provides various security solutions to the hosting platform, such as trusted boot, remote attestation, integrity checking, and cryptographic functionalities. Nowadays, a vast number of personal computers (PCs) and computing devices are shipped with TPM integrated and benefit from its security solutions. TPM specification version 1.2 was published in 2011 and, in 2016, TPM 2.0 came to the market with better support for algorithms and higher cryptographic capabilities.

TPM provides root of trust for storage, integrity protection, measurement, and reporting. This makes the act of authentication, identification, integrity verification, and encryption of a device feasible. Remote attestation is the primary feature provided by the trusted computing technology that extends the trust step-by-step from the lower levels up to higher levels and applications. This is referred to as the chain of trust. In the chain of trust scenario, all the components that are going to be loaded are considered untrusted, and therefore, need to be measured before being loaded. TPM encompasses protected memory locations called platform configuration registers (PCRs), that store sensitive security information such as measurement information.

In addition to the strong isolated storage, TPM holds a unique endorsement key (EK), used for cryptography operations. This key is generated at time of TPM manufacturing, and the private part of the keys never leaves the TPM. At the time of remote attestation, for preserving the privacy of the identity of the platform, attestation identity keys (AIKs) are utilized. When a platform (the *attester*) receives a request for remote attestation (from a *verifier*), it sends an integrity report composed of PCR values and their digital signature that are computed with an AIK. Due to the fact that private part of the AIK has never left the TPM, the integrity and authenticity of the report is guaranteed.[2]

1.2 Aims and research questions

The successful outcome of using TPM for security platforms and applications and providing trust to the systems has motivated various execution environments to take advantage of the application of TPM. Mobile devices, IoT devices and cloud computing are examples of such environments.

TCG has developed the mobile trusted module (MTM) to provide hardware root of trust for mobile devices. MTM supports secure transaction, integrity protection, and secure storage of keys and certificates.

TPM equipped IoT devices can measure themselves and each other before establishing a telecommunications session. This hardware root of trust supports secure boot, remote attestation, and device authentication.

Trusted computing has also led its way in cloud computing to address its security and establish trust between the cloud service provider and its client. This motivated us to investigate in what way trusted computing has boosted the security in cloud, what threats has it mitigated, and how the integrity is measured.

In this paper, we study the current trends of TPM application in cloud computing, in recently published literature, with the aim of answering the following questions:

- RQ1: For what *aims/purposes* is TPM used in the cloud (eg, access right management, integrity measurement? This is further discussed in Section 4.1.
- RQ2: What types of *threats* have been reportedly mitigated via the utilization of TPM in cloud (eg, man-in-the-middle attack, tampering)? This is further discussed in Section 4.2.
- RQ3: If the paper is addressing *integrity measurement*, what is the exact component of the assessment? In other words, what interpretations are given for integrity in cloud? This subject is discussed in Section 4.3.
- RQ4: At what *level* (eg, hardware, virtual machine [VM], application) are TPM primitives invoked? This question is discussed in Section 4.4.
- RQ5: At which *phase* (eg, boot time, run time) are the TPM features used? Further discussed in Section 4.5.

This section presents the background on the terms and concepts related to this work, and the motivation behind this survey and study. The remainder of this article is organized as follows: In Section 2 we discuss the method of study, the search criteria and procedure, selection and the data extraction process. Section 3 presents the status of the field of research and Section 4

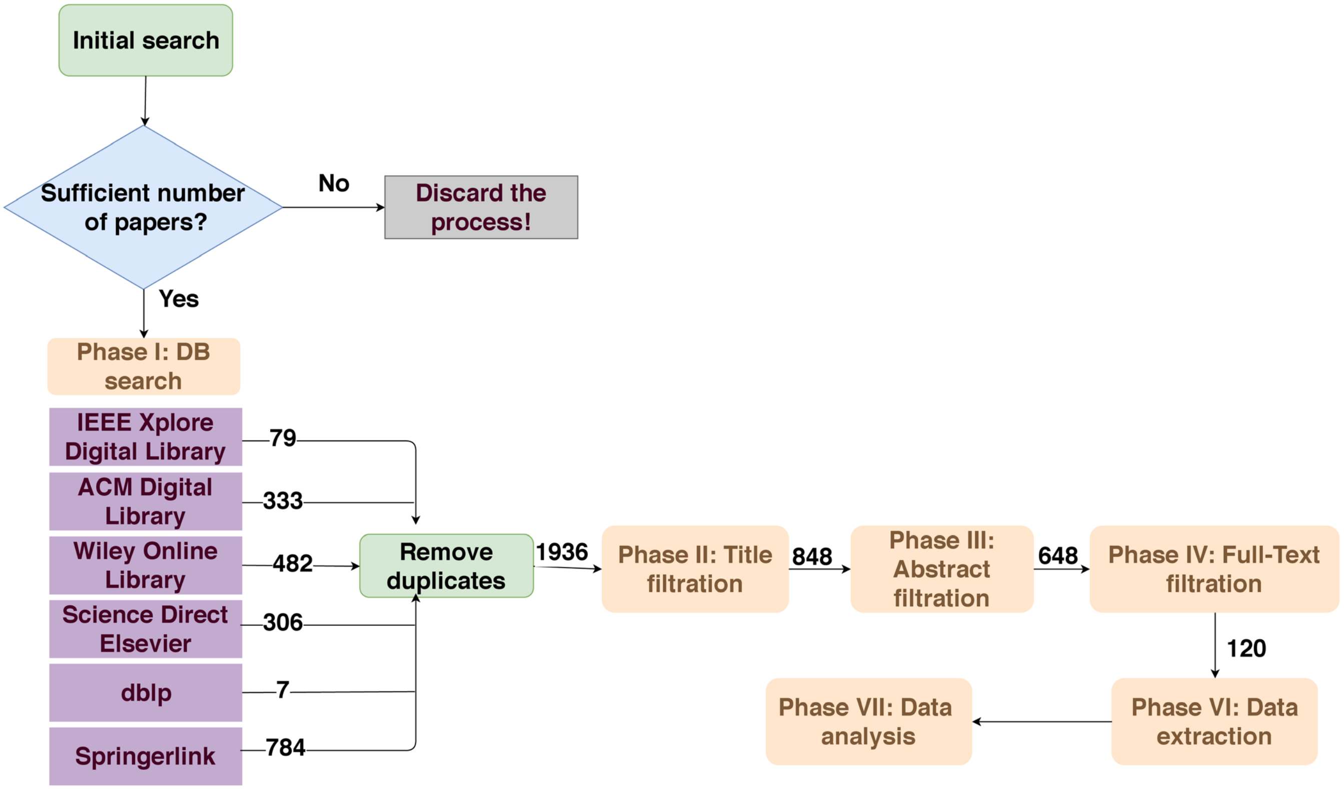

FIGURE 1 The process of search and selection. Six online databases and the number of papers collected from each is shown on the left side, and the various phases of selection process with the number of papers after each filtration is shown on the right

presents the results of analyzing the collected data. Section 5 discusses the challenges and open issues of this field of research, and sheds light on the future direction. Concluding remarks come in Section 6.

2 | METHOD OF STUDY

The method of study we used for this work is systematic literature review (SLR), which is a method to identify, evaluate, and interpret the well quality studies associated with an area of research or a specific research question Kitchenham, 2013. A SLR classifies and maps the scattered research studies and helps identifying the research gaps which directs baselines in future research. We conducted an SLR to identify the studies that apply TPM in cloud computing environment, in order to point out how cloud computing security could benefit from functionalities provided by TPM, and also what are the gaps in this domain. To carry out this SLR, we followed the protocol designed by Kitchenham et al.[3] In accordance to this protocol, before conducting the search, we designed the search string to collect the publications from the online databases. We then defined the inclusion and exclusion criteria, the research questions and planned the way to extract data from the collected publications, and synthesize the results.

We perform the search and selection process in six different phases (Figure 1). We started the search process by first certifying that there is a sufficiently large number of papers published in this domain to make a survey paper. Using the designed search string, we collected the papers from the five online databases that are most commonly used in Computer Science, including ACM Digital Library, IEEEXplore Digital Library, ScienceDirect, Wiley online library, SpringerLink, and dblp. Figure 1 shows the process of data collection and the number of publications we had at each phase.

After collecting the papers, the duplicates were removed and we proceeded to the selection phases. We filtered the papers according to the inclusion and exclusion criteria we had defined, first based on their title, then based on their abstracts and finally based on their full-text. We only included the papers that were written in English language and peer-reviewed and they were applying TPM in cloud architecture/environment aiming to improving the security in cloud. We excluded the papers that were studying other means of hardware security (eg, SGX), other execution environments different from cloud computing, survey papers that were not proposing/applying a novel technique, and the publications that were not available online and we could not access them in any ways.

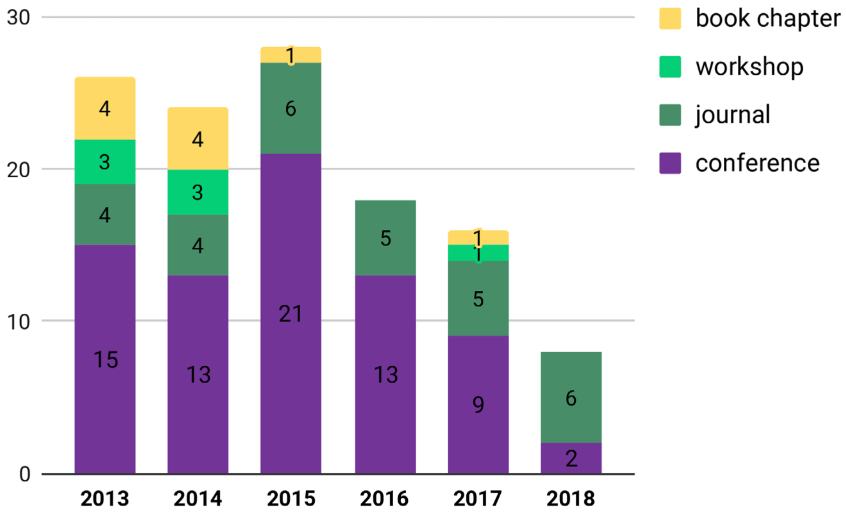

FIGURE 2 Publication forum types for the studied publications in the considered time-span

After finalizing the screening phase, we ended up with 120 papers, from which we extracted data by answering the designed research questions (RQ1-RQ5 in Section 1). We analyzed the collected data and classified them. The result of this analysis is presented in Sections 3 and 4.

3 | STATUS OF THE FIELD

As a result of the search and selection step, we collected 120 publications, from which we extracted data. These publications were published in four different forms, including journal article, conference paper, book chapter, and workshop paper. Figure 2 presents the total number of published studies in the field in different years, and the number of publications in each forum type. As is seen in the figure, the majority of the considered set of studies were published in conferences.

Identifying the annual increase/decline in the number of studies published in a studied field provides an idea on the interest rate to the field. An upward trend could be an indication of rising interest, and a descending trend could be a sign that the researchers have lost interest in this field. Figure 2 presents the annual distribution of the papers, the types of forums the studies were published in, and the number of publications in each type. According to the numbers, in the past 6 years, there has been a growth of interest to this field of study before 2015, and a decline afterward. One explanation to this decline could be the appearance of other hardware means of trust and trusted execution environments, such as Intel SGX. However, we believe that this fall in the number of studies does not mean that TPM and its useful functionalities are being replaced by other technologies. Despite of the fact that SGX has been widely used for solving many security problems, it still suffers from vulnerabilities that could be exploited and disclose the privacy of its users. To name some, we can refer to the side channel attacks that have been successful in leaking the secrets out of the SGX enclaves. However, combining these technologies, TPM and Intel SGX or other trusted computing technologies, and studying the possibility of their application in environments such as cloud computing has become a significant research direction.[4]

Analyzing the affiliation data of the publications designates that the majority of studies in this field, in the considered time-span, were the result of research done in academic sector (61%). Thirteen percent were made by industrial organizations and 36% were resulting from the collaboration of academia and industry (26%). Figure 3 illustrates the associated sector of the organizations as a function of the publication year, for the set of publications under consideration. From the 120 studied publications, 73 of them are published by academic bodies. While this distribution is expected for theoretical research, on the other hand, raises the concerns about the correspondence and applicability of this field of research.

We analyzed the affiliations of the authors of the collected publications, in order to connect the papers to their originating countries and organizations. Figure 4 illustrates the top most affiliated countries in the studied publications. United States has eminently the largest share (24.5% of the studies), followed by China (18.7% of the studies). United Kingdom, South Korea, and Germany, respectively have 6%, 5.7%, and 5.3% of the publications.

Figure 5 illustrates the top most affiliated organizations in the studied publications. From this plot, it is notable that Rutgers University, Shandong University, and University of Oxford are the most active academic parties, and IBM research, SICS Swedish ICT are the top industrial organizations in this field. It is worth noting that in this figure, we only plot the top organizations that corresponds to three affiliations or more (35% of the total affiliations for these publications). This, arguably, imply that large number of publications were produced by relatively limited set of organizations and the other 65% of the publications could be traced to a single organization.

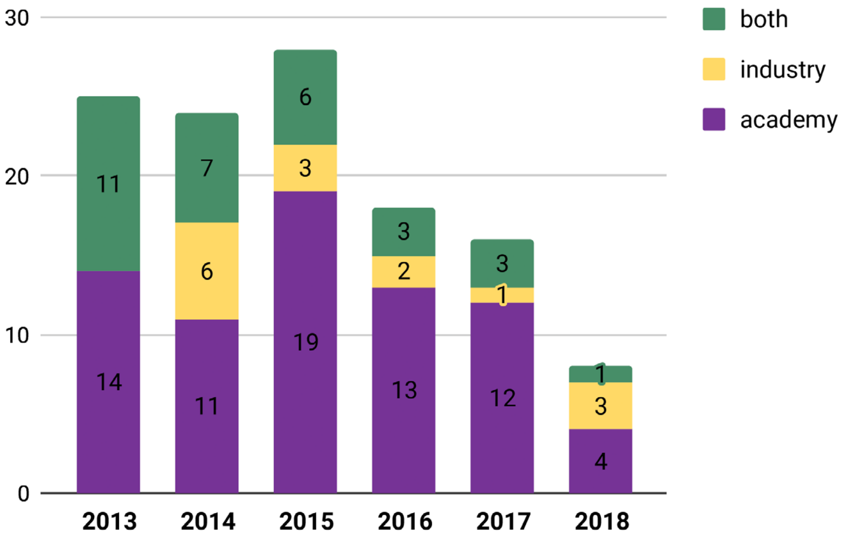

FIGURE 3 Organizations sector of the studied publications in the considered time-span

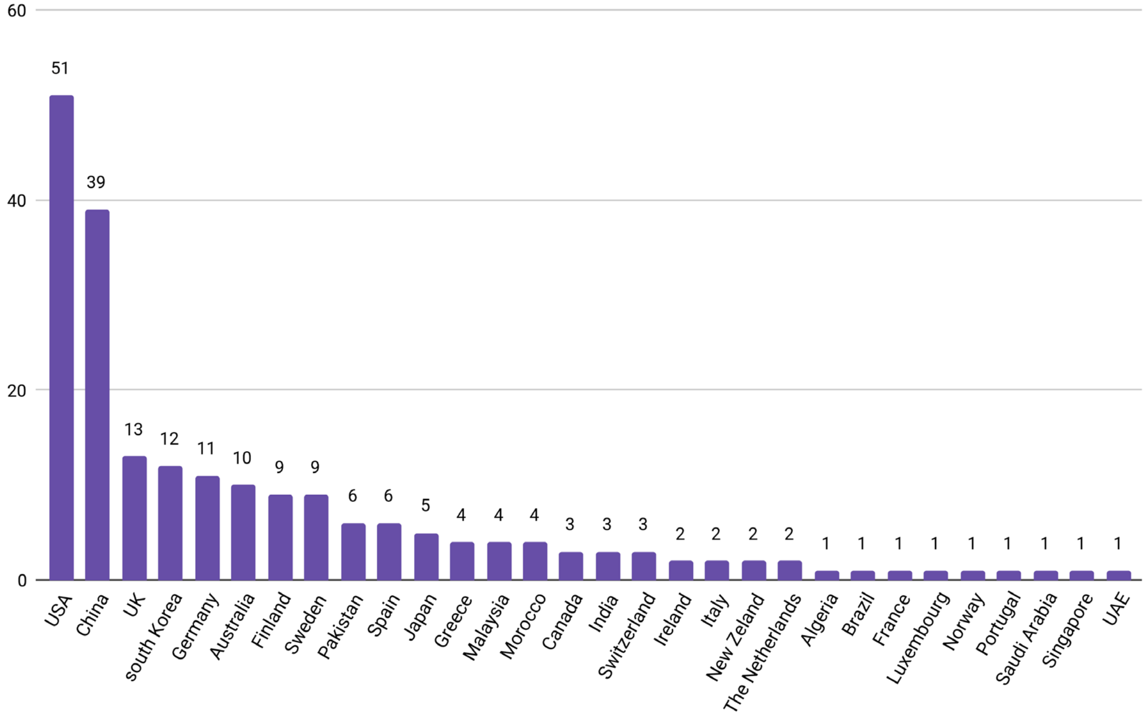

FIGURE 4 Top most affiliated countries in the studied publications

4 | ANALYSIS OF THE RESULT

This section provides answers or analyses the identified research questions. Each research question is approached in its specific subsection.

4.1 | RQ1: Aims of using TPM in cloud computing

The usage of TPM in a cloud-computing environment can be applied to several different purposes, as a way of providing an extra layer of security for operations, storage, communications, monitoring, and so on. The first research question analyses and discusses the different applications that have been surveyed in the literature, and the different types of use cases for which TPM was used for each of them (Figure 6). The following purposes were then identified:

1 *Remote attestation*: A TPM can be used to provide this method, which allows for an entity to authenticate itself, in terms of software, hardware, or both, with another, remote, entity. This has several practical applications in cloud computing.

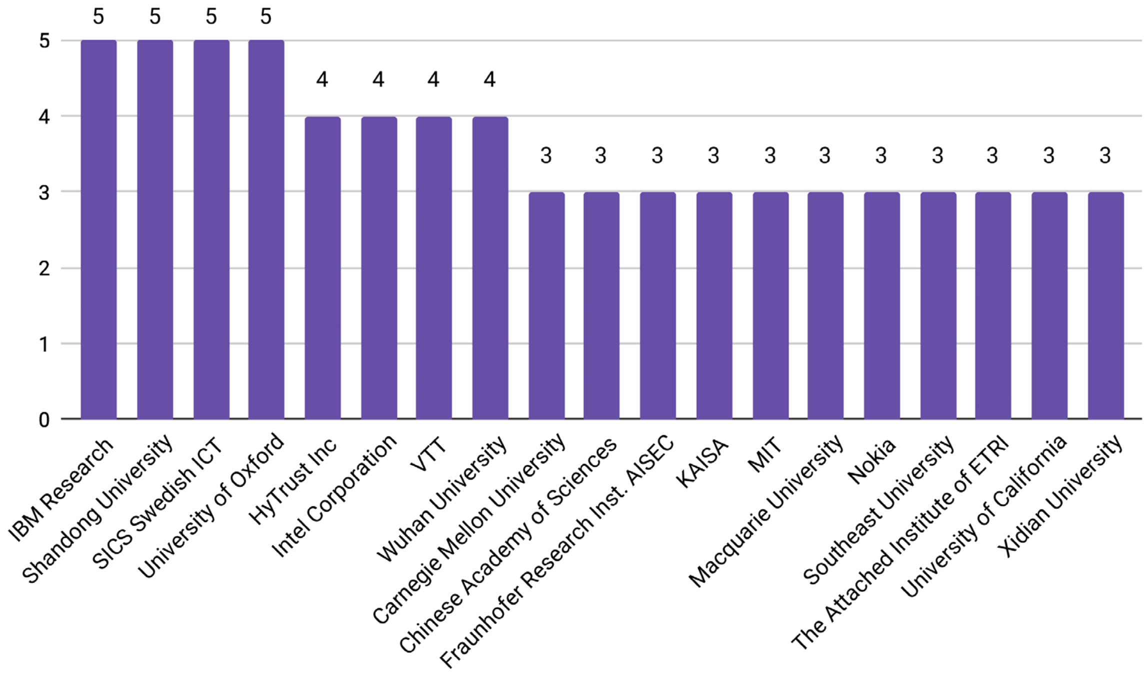

FIGURE 5 Top most affiliated organizations for the studied publications

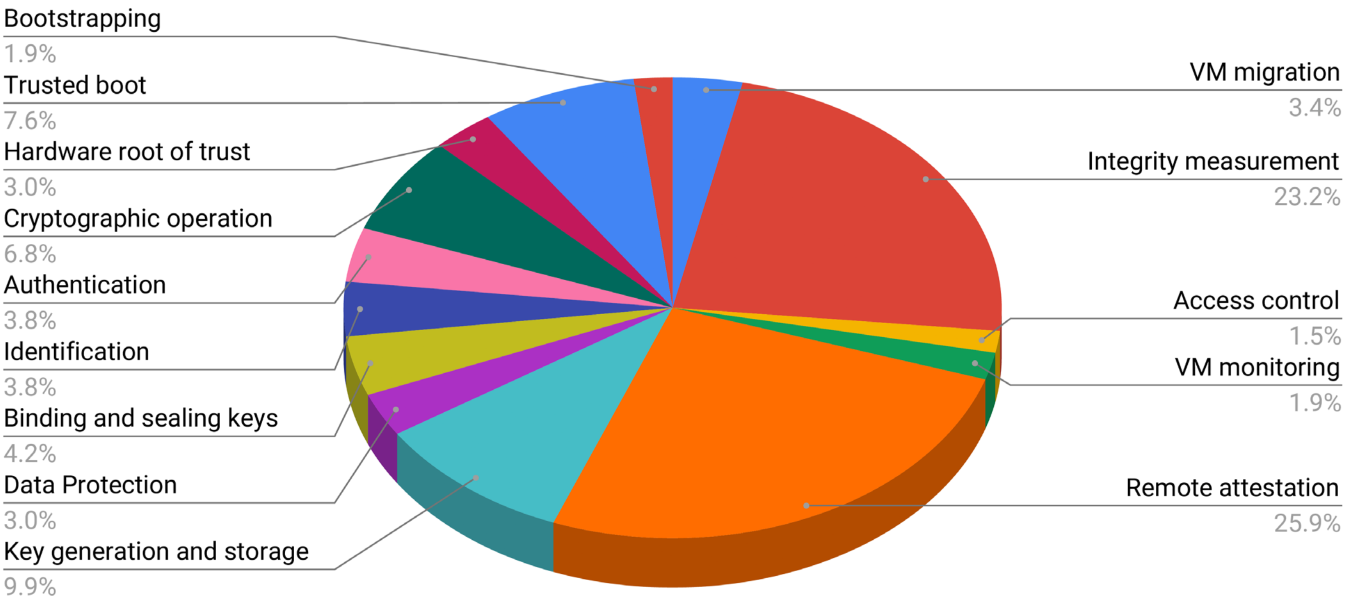

FIGURE 6 Aims of using trusted platform module (TPM) in cloud computing

In a way, the entity proves the requester that it is trustworthy and that its systems have not been tampered with. The basic functioning of this operation is similar to how integrity measurement works, as it serves as the basis of the attestation process, with it then being transmitted to the requester. The process works as follows: the requesting entity sends a request to the remote entity, where a *nonce* is included. The receiving platform will then forward the request to the TPM, that will perform an integrity measurement, and also sign the *nonce* sent to it. This is then sent to the requester, that will verify the response with a trust attestation authority (TAA), verifying that (a) the TPM is genuine and (b) that the received values match the known values. If everything is correctly verified, the requester will consider the entity to be trustworthy.[5] Javanmard et al,[6] use the TPM to remotely attest VMs for their cytometry analysis system, which relies on cloud VMs to process the cytometry data. Before sending the data, the trustworthiness of the VMs is remotely attested to ensure that no tampering happened, through the virtualization of a physical TPM on the VMs (vTPM). In Reference 7, the authors use remote attestation to verify the geolocation of cloud users through the GPS module of their mobile devices.

They create a hypervisor that creates a secure channel, through the TPM, with the mobile device, and verifies the GPS location data sent to ensure, that is, that the user is located in a region from which he is allowed to access the cloud service.

2 *Integrity measurement*: The TPM is able to calculate and store in its registers hash values regarding different components of a system. These different hashes are transformed in a hash chain, whose value that is then used to compare with the stored value. This can be done both for hardware and software, to detect any unauthorized changes.[8] It can also be used for verifying data integrity. There are two main perspectives on types of integrity measurement: static, where the measurement is made on binaries or firmware (eg, BIOS), before they are run, to ensure that no tampering has happened, and dynamic, where changes are monitored and detected during run time, to ensure that no tampering or unauthorized changes were done.[9] In Reference 10, the authors present a model for integrity on data possession in a multiuser setting, where data can be owned and shared across multiple users, with provable data integrity and freshness. Du et al[11] propose a dynamic integrity measurement model for the Xen virtualization platform, capable of both measuring statically, and dynamically during run time, to ensure the VMs are not compromised.

3 *Generation and secure storage of keys*: Cryptographic keys can be both generated and stored in a TPM, to be then securely used by, that is, a VM. The TPM can generate random secure keys, using entropy both within and outside of the TPM (from where entropy is gathered is a user choice). Keys can then be generated through three different methods: from a seed, from a random number generator, or directly imported to the TPM. The keys are then securely stored inside of the TPM. External keys can also be stored inside of the TPM, where they are encrypted with the public part of the storage root key (SRK), a key pair generated inside of the TPM.[8] The storage process can either be a binding or a sealing of the key, which will be explained in greater detail below. In Reference 12, the authors propose a Linux block device mapper capable of guaranteeing data integrity during run time on a remote storage, where the keys are stored on the TPM. In Reference 13, cryptographic keys are generated and stored with the TPM that are used to prevent man-in-the-middle (MitM) attacks between the user and a security agent, for securing encryption and decryption operations in a VM.

4 *Trusted boot*: The TPM can use integrity measurement to provide this functionality. When a system starts its booting up process, the TPM checks for the different hardware, software and/or firmware components and runs an integrity measurement check on them, to ensure that the system was not tampered or changed before the startup process. In Reference 14, the authors propose a hypervisor for PCs that utilizes trusted booting to ensure no malware or rootkit is running on the guest OS of the device. Zou et al[15] present a framework for monitoring the cloud platforms, where trusted boot is used to ensure the integrity of the monitored environment, both for cloud tenants and provider.

5 *Cryptographic operations*: Included in the TPM is a coprocessor that handles cryptographic operations, such as hashing, random number generation, asymmetric encryption, and key generation, that can be used directly or for several other purposes (such as some of those referenced above and below, for example, integrity measurement or binding and sealing of keys).[8] All cryptographic operations on the TPM are hardware based, and use the SHA1, HMAC, and RSA algorithms. In Reference 16, a distributed cloud storage is proposed, where TPM is used to authenticate clients and securely encrypt their data. Chen et al[17] present a verifiable resource accounting system for clients to verify that the reported consumption on cloud resources are trustworthy. Reporting is signed through the TPM. Thilakanathan et al[18] propose introduce a secure data sharing protocol in which the data owner stores the data securely on an untrusted clouds. In this approach, an external TPM-based device is used to address key management and to enable secure data transfer. The decryption keys are stored on this external device, and therefore, only the device issuer can only decrypt the data.

6 *Secure authentication and communication*: A TPM can be used to authenticate a system and to secure communications between two entities. The TPM, being a root of trust, can authenticate a system when other entities request it, as the hardware of the TPM is certified through a certification authority.[19] In Reference 20, TPM is used to authenticate remote users on hybrid clouds when using the MapReduce framework. For communications, the TPM, through its AIK pairs, can help establishing a secure channel so that information can be exchanged between two entities.[8] In Reference 21, a secure logger for medical devices, based on the cloud, is capable of creating a secure channel between a medical device and the cloud logger, through the use of a TPM.

7 *Binding and sealing of keys*: TPM provides data protection through securely preserving the cryptographic keys and through binding and sealing functions that it exposes. According to Reference 22, a message that has been encrypted/bound using the public key of a particular TPM, can only be decrypted through using the private key of that TPM. Sealing functionality is a special case of binding, in which an encrypted messages that has been produced through binding could be decrypted when the platform is in a specific state to which the message is sealed. This state is defined by the PCR values. In this way, it is assured that the message is decrypted by a platform that is found in a specific prescribed state. In Reference 23, a trusted store cloud mechanism is presented, where data is sealed to a specific trusted host configuration.

8 *Identification*: Through the TPM, it is possible to uniquely identify a user. A TPM has a single unique key, called the EK, and embedded key pair unique to each TPM. This key pair also has a certificate associated with it, of which the public key part of the EK comprises. This certificate ensures that the TPM conforms to specification. However, it is rarely used to verify the TPM. This falls on the AIKs that are generated from the EK. The AIKs can then be used to verify the identity of the user of that specific TPM.[8] In Reference 24, the authors propose the use of TPM for ensuring the policies of an academic organization in cloud services that can be enforced on the cloud service provider, including identity management. Krauß et al[25] present a method to detect the geographical location of a virtual resource, where the identity of a physical machine is verified through the TPM to ensure that a resource is not moved by the cloud provider to a location undesired by the user.

9 *VM migration*: In cloud services, VMs are often moved between physical systems. This act is named migration, and it can be a point of exploitation if not defended properly, as the VM may not be isolated during the process, and is physically changing hosts. The trustworthiness of the new host needs to be attested, and the data in the VM protected.[26] The TPM can provide for these requirements, for example, through its cryptographic methods or its remote attestation properties. Syed et al,[27] propose a module that uses the TPM to secure data when migrating within the cloud. Openstack is used as a case study for secure migration of instances. In Reference 28, the authors propose an architecture capable of providing secure migrations through integrity measurement, capable of detecting when a hypervisor is compromised, in a way that it migrate the VMs securely from a malicious hypervisor to a trusted one.

10 *Data protection*: In the scheme proposed in Reference 29 TPM offers remote attestation and integrity measurement of the streaming data. This assures the secure collection and transmission of forensic data from cloud services, and prevents manipulation and falsification of the forensic data. The monitoring data is collected from the VM instances and stored in the log files. This data could be accessed and forged by a malicious insider or an external intruder. In the framework proposed by Kanstrén et al[30] TPM is used to support the trusted monitoring for conducting security measurements and to guarantee the availability and integrity of the monitoring data in the cloud.

In several works encryption feature that the TPM offers, through its securely stored keys and binding feature, is leveraged to protect.[31,32]

11 *Hardware root of trust*: A root of trust is a source that could be trusted in a cryptographic system. Cryptographic security relies on the keys to perform functions such as encryption, decryption, and generating/verifying digital signatures. Therefore, in such system a hardened hardware module is included that is not easily broken and provides a level of trust that is guaranteed to be genuine. TPM has been a successful hardware root of trust to support secure boot,[17,33,34] integrity measurements of the components before being loaded and at run time cite,[25] and to protect the measurement values.[35]

12 *Bootstrapping*: Prior to entrusting a computer with a secret, the user should get some assurance about the trustworthiness of the computer. Using secure hardware mechanisms for monitoring and reporting the platform's software state is one way of bootstrapping trust in a computer system.[36] To this end, TPM has become a popular means to achieve this goal.[37–39] With proper software support, TPM can measure and record the loaded and executed pieces of software, and can convey this information securely to a remote party. Hence, TPM could establish trust in the platform's software.

Keylime scheme[40] provides secure bootstrapping that enables the tenants of a system to securely install a root secret into cloud nodes which becomes the long term cryptographic identity of the nodes. The tenant will chain other secrets to it for enabling secure services. The authors present a bootstrap key derivation protocol, which integrates integrity measurement and the tenant intent for installing secrets into the cloud nodes. In the P-Cop design,[41] bootstrapping and attestation of compute nodes check the SW configuration of a nodes and verify whether they are configured properly with a trusted container run time.

13 *VM monitoring*: One of the services offered by TPM to the cloud infrastructure to increase the confidence in cloud is secure monitoring of VMs.[42–46] In Reference 43 the authors propose an architecture that uses TPM for trusted monitoring of the host infrastructure and guest VMs on these infrastructures. TPM verifies the integrity of the probes and the measurement results that they have provided. This ensures that the system is running in the expected way, the monitoring probes are not altered and their integrity is preserved.

14 *Access rights management*: TPM could be used to control the access of a user to a cloud provider,[47] and detecting an unauthorized access.[48] In the architecture proposed by Jayarathna et al,[49] TPM is emulated to provide trust management with hypervisor level policy based access control. This prevents the access of unauthorized users to the resources.

4.2 | RQ2: Security threats mitigated in cloud computing using TPM

Vulnerabilities in computer systems make them prone to security threats, and consequently, security attacks. A component/resource is vulnerable, if it is: *corrupted* (due to some unintended changes, the resources do not function as

TABLE 1 Top threats mitigated in cloud computing environment through the use of trusted platform module (TPM)

Threats mitigated by TPM	No. of papers
Tampering of data	15
Unauthorized access	13
Data leakage	11
Malicious cloud provider	10
Run time attack to software stack	10
VM tampering	9
Malicious insider	7
Stealing cryptographic keys and nonces	6
Tampering of audit log records	6
Malware	5
Reverse engineering of applications on the cloud	5
Repaly attacks	5
MitM	5
Malicious coresident VM	4
Measurement data tampering	4
Attacks involving privilege escalation	3
Leakage of geolocation	3
Identity privacy	3
Leakage of data from compromised platform	3

they should), *leaky* (if there is an unauthorized access to the resources), or *unavailable* (if the system is disabled or slow to render services).[50] Attackers can violate the system and harm the assets by taking advantage of the vulnerabilities. In the cloud-computing environment, there are several different points that are prone to security attacks. Compromising of any of the vulnerable parts will result in unavailability of resources, malfunctioning of the system, and disclosure of information to unauthorized parties. In the literature, there exist wide spectrum of security measures to impede the risk of these attacks. In the following, we present how trusted computing technology has been used to mitigate the most common security attacks. Through answering this research question (RQ2) we identify the types of threats that are mitigated with the help of trusted computing. Table 1 presents the top threats that were mitigated using TPM and the number of papers that were addressing that threat using TPM. We also discuss these threats in more detail and present a classification for them. Depending on the point in cloud computing environment that the attack could potentially occur and the target of the attack, we have classified the security threats into six main categories of (1) network attack, (2) application attack, (3) malicious/untrusted cloud service provider, (4) attacks on cloud back-end, (5) data tampering, and (6) data leakage.

1. *Network attack*:

   - *MitM* attack could occur on the remote attestation protocol. It comes between the verifier and the prover to intercept and manipulate the communicated data, that is, the attestation information. The MitM attacker should be prevented from altering/manipulating/forging this information. In Reference 51, the hypervisor, after processing the integrity measurements, stores the result of the measurements in TPM registers in a concealed manner.[51] Each measurement is concealed through using a pseudorandom value to avoid the extraction of the plain measurement values. Each TPM register holds measurement values of a different VM. In Reference 13, TPM has been used to generate and hold a public-private key pair in order to thwart MitM attacks between a security agent and the user. An eavesdropping attack occurs when the adversary monitors the communication channels and takes advantage of the disclosure of resources to obtain unauthorized access to confidential data. This type of attack does not disrupt the normal functionality of the system, so might not be detected immediately. Encryption of measurement data before storing them on TPM protects them from eavesdropping attacks.[21]

- The *Replay attack* can be regarded as the combination of two attacks; eavesdropping and injection attacks. In this scenario, first the intruder intercepts the communication to capture the data on transit, and then she retransmits this data to the receiving party.[52] On that account, the attacker acquires access to confidential data. The attacker tries to looks legitimate while carrying out the malicious intent. Nguye et al[21] have designed a secure tamper-evident logging system which leverages SGX and TPM to hamper the tampering of stored logs. Upon receiving the logs, the logger encrypts the content, generates and signs the hash value of this new entry. The signed hash value is stored inside TPM. Later, at the time of audit, the whole hash chain (from the first signed hash and the log entries) are recomputed and compared with the last value stored on TPM to verify the signature and also to detect if any of the verification steps fails. Any changes in the log information stored results in a dissimilar hash value and therefore dissimilar hash chain, which makes the attacker being detected. Moreover, the logger detects replay attacks on the storage, in which the adversary clones some version of logger storage and restores it later on. The data, although authentic, is an old version. The logger can detect the attack by comparing the version of data with the latest hash value stored on the TPM.[21]
- *Denial of service and distributed denial of service (DDoS)* attacks typically take advantage of the vulnerabilities in the network to overwhelm and exhaust the services and resources in order to make them unavailable to legitimate users. This risk could be mitigated by using a fine-grained access control method, guaranteeing the integrity, confidentiality, and freshness of the data.[20] In the scheme proposed in Reference 20, TPM has the role of ensuring the integrity and the authenticity of the computation and providing communication security through binding keys to exchange the session keys in a secure manner between the communicating parties. Wang et al[25] persent a trust certificate hypervisor module that provides the functionalities such as software authorization, read, and write operations in the authorized environment with TPM based encryption, and counting the number of trust certificates issued for the guest OS and guest service. Then they demonstrate a form of DDoS attack as an instance of internal security attacks that take advantage of the split device driver and multitenancy characteristic of the cloud servers to launch the attacks. This experiment demonstrated that the TPM based module restricted the scale of malicious services and mitigated the risk of internal DDoS attacks.
- *Attacks to steal cryptographic keys and secrets* stored on TPM or vTPM allows the attacker to gain access to the encrypted files and perform its malicious activity.[13,53] For protecting against a local adversaries, TPM provides a secure storage for the keys on an untrusted device.[12] Also in order to avoid this type of attack, the security scheme Kernel-based virtual machines tries to secure the vTPM, and its secrets are encrypted and stored on the host. This approach also supports the migration of keys at the time of VM migration. In another work,[54] the author presents an architecture that separates the keys and cryptographic primitives from the VM clients, and stores them on a secure domain that is managed by the hypervisor. FADE-TPM[55] considers the key management issue from a different point of view. It assures that the files and keys are not recovered by an unauthorized party after being deleted from the cloud. To achieve this, they have designed a TPM-equipped key manager that resides on the client side and stores the encryption keys. The key manager encrypts the files and data before storing them on the cloud using a control key, and also decrypts the data received from the cloud for the client. Key manager removes this control key when a predefined period expires.
- *Unauthorized access* is the first step for an adversary to view, steal, or modify confidential data, or to make damage. One of the consequences of the unauthorized access is a successful injection attack that occurs when the attacker gains knowledge on the system and attempts to tamper with the communication channels with the aim of injecting false data or wiping evidence from the logger. The adversary could inject new messages, or change/reorder the message content.[21] In this scenario, TPM could provide secure authentication in order to prevent the unwanted accesses.[16] Paladi et al.[23] propose a trusted storage protection scheme that provides access control per-VM instance using storage management policies. This allows the client to control the read/write access rights from/to the storage at launch time.
- *Identification problems or identity spoofing*; While the great opportunities the cloud computing brings with its open, dynamic and large scale characteristics, the same characteristics has brought along security risks as well. One of the security threat as consequence, is the authenticity of the user's identity. Therefore, one of the security measures that need to be in place is the identification of the users, VMs and the platforms. Among all security measures, TPM has also had a significant role to ensure the authenticity of the identification.[56–58] The scheme proposed in Reference 52 provides identity authentication of the users accessing cloud services with the help of trusted platform. TPM is used for identifying the user and mutual authentication, as well as for generating keys, establishing communication channels between a remote user and the cloud server in a secure manner, and performing encryption and decryption operations. In another framework,[57] TPM performs mutual attestation and renders platform verification checks in order to provide trustworthiness and authentication in the cloud.

2 *Application attack*

- Malware is a piece of software that is designed to cause an intentional damage to the system or conduct malicious actions. In cloud infrastructures, TPM has been used to reduce the threats and costs of certain types of malware and installation of malicious rootkits, for instance, by starting the system with a root of trust and measuring the launch environment, and by verifying the integrity of the platform.[59] Malware could also be residing in the OS, however, it cannot compromise the BIOS, firmware, hypervisor and hardware devices, since they are protected by the trusted boot, and hardware-assisted virtualization techniques. In the proposed scheme by Wang et al[60] through the access control mechanism that is based on domain-type enhancement, also the static and dynamic measurement of host and VMs, the system introduces in depth security for the virtual machine monitor and the VMs. This defends against the Trojan horses and viruses that aim at tampering the running environment of the system. In Reference 61, trusted system boot offered by TPM from lower levels to upper levels introduces protection against the rootkits, bootkits, and the malware that could be initiates during the system boot or after the launch of a VM.
- In order to prevent the malware from obtaining keys and making copies from them at the time of migration, and also prevent the malware from leaking the keys to a malicious device, there needs to be a secure key exchange protocol. TPM-equipped devices establish a secure communication channel and TPMs take specific steps to exchange the keys without leaking them, even if the hosts are compromised and infected by malware.[62]
- *(Run time) attacks to software stack*; the modern cloud infrastructure offers new SW deployment model, which is integrated in the cloud environment, its policies and configuration. This indicates that the cloud is not only facing new challenges, but also suffers from the traditional challenges existing in the ecosystem of cloud software. Integrity of the SW stack could be compromised either by an attacker, intentionally, or through human errors, unintentionally. Presence of a vulnerable software could potentially affect the entire software stack and consequently the whole system. Some examples of threats that could compromise the vulnerable software stack are presence of malware, absence of proper security software (eg, anti-virus, firewall), missing security patches to fix vulnerabilities, and also the misconfiguration of the software which relates to the behavior of the software in unintended behavior due to the effect of malware/virus. Aslam et al[63] have proposed a platform security monitoring and verification model that utilizes TCG-SCAP synergy to shift the source of user trust to a trusted third party instead of the platform owner. Software stack integrity is verified through remote attestation. The SW stack presents a report to the remote attester/verifier that is created by the root of trust for measurement mechanism of TCG.
- Furthermore, many of the recent attacks have proved that software based solutions could themselves be vulnerable to attacks and therefore cannot provide full-proof security. On that account, hardware based solutions such as the use of TPM has become helpful. One of the use cases of TPM is to provide hardware root of trust to alleviate the attacks on software security solutions,[27] for instance by securing the hypervisor and VMs. SIMM module[27] enables the binding function of trusted computing to improve the existing migration of the instances. Another use case of TPM is integrity measurement (see Section 4.3). Measuring and monitoring the run time system integrity prevents the attacks on software stack that compromise the data or code of the software running on the cloud such as buffer overflow attack.[64] Zou and Zhang[42] propose a method in which run time system integrity is measured through a monitoring tool that resides in a privileged VM, dom0.
- -*Reverse engineering of applications on the cloud and tampering them* is the result of the exploitation of the run time software vulnerabilities that impacts the trustworthiness of the cloud infrastructure, and therefore, needs to be protected. CAFE[65] is presented to assure the secure execution of the sensitive software logic and protect it from piracy and reverse engineering, in a VM, even if the guest OS and its kernel is compromised. To do that, the developer delivers the program binary in two different sets of public and private binaries. While the first is comprised of files associated with the application, such as configuration files, the later is related to the confidential logic that should be protected. When an application is executing in the VM of the user, secret binaries are fetched at run time upon demand. The hypervisor loads the secret binaries securely through a cryptographically protected communication channel after authenticating the VM. In this scenario, the role of TPM is to attests the integrity of the hypervisor and ensuring its genuineness. Moreover, TPM creates a secure channel by generating the RSA key pair for the hypervisor and other secrets used for encryption. This provides higher level of security compared to the standard transport layer security (TLS). The RSA key pair is wrapped with the SRK of the TPM, which is unique and nonmigratable. This feature makes the key to be only usable on that specific machine, and the attacker cannot upwrap it without that particular TPM.
- Trusted cloud root broker[66] is proposed to robustly guarantee the trustworthiness of the JVM based applications. The broker which is the application-root of the trust evaluates the run time trustworthiness, and supports dynamic attestation of

the integrity of an application with the help of a Java VM. In order to prevent the compromising of the code and software running in the cloud, Wei et al[64] have developed a dynamic analysis tool to detect scoped invariants. As a case study to scoped invariants they have chosen Xen hypervisor, which is the foundation software for a vast number of cloud providers. The developed tool identifies the scoped invariants that are important to the run time integrity of the Xen hypervisor. One of the properties of such invariant is isolation of the guests, the violation of which shows that the adversary, to achieve her goal, only requires modifying a single byte in the global descriptor table.

- *Piracy, illegal reproduction of copyright-protected*; one of the possible attacks on the software is the copying and piracy of the software. One of the usages of the approach CAFE[65] is to protect the software from illegal copying and reproduction. Also, in the digital right management (DRM) approach proposed by Lee et al,[58] some of the TPM functionalities are employed to impede the attacks on DRM and protect the digital contents. To this end, TPM is incorporated in the compliant devices to offer the functionalities including attestation, memory containing, sealing, and isolation with the aim of protecting illegal and unauthorized access to the content by modifying the device operations. More specifically, such attacks could be thwarted by using the TPM's remote attestation protocol, and avoiding the use of system-wide secret keys.

3 *Malicious/untrusted cloud service provider*: One of the very significant threat types in cloud security are the risks associated with the threats coming from the inside of the cloud service provider. This class of threats could appear in the form of cloud mismanagement, malicious insider, adversarial administrator, and malicious cloud. Due to the fact that cloud administrators hold high level of privileged access to the cloud nodes on which the customer data resides and their computation take place, misuse of such privileges could introduce major harm. Hence, as like other security threats, it is very important to mitigate this security threat as well. P-Cop[41] is a scheme proposed to protect the Docker containerized PaaS services against threats related to cloud mismanagement. It employs TPM to offer remote attestation capability by external clients and prevent the access of untrusted cloud administrators to the guest containers. All the severs installed in the cluster are supplied with TPM chips, that each has a unique AIK key pair. The private part of the key is bound to the chips and the corresponding public key is certified by cloud service provider at the time hardware being deployed to ascertain the property of that server by the cloud provider. P-cop keeps a list of verified nodes with trusted run time software, which could accommodate guest containers. This verification is attained when the auditor validates and signs the software, and also the leveraging TPM attests that this software is executing on cloud nodes.

- Sometimes the cloud infrastructure hosts sensitive data, such as medical data of patients. Therefore, it is essential to make sure that the patient data is not placed on a malicious cloud server and is not tampered with or accessed by unauthorized parties. Javanmard et al[6] have presented a cloud-based framework for cytometry data analysis. In this framework, TPM verifies the integrity of cloud VMs. A vTPM instance provides the TPM functionalities on each VM. Remote attestation technique is used for platform authentication (ie, authentication of hardware and also software stack running on the cloud) using the RSA key unique to the TPM, to prove the integrity of the cloud to the remote authorized users, and to identify if any unauthorized alteration occurs to the execution environment. In Reference 29, the authors present a scheme for collecting and transmitting forensic data from cloud services in a secure way. For instance, a course of snapshots of the suspect VM are taken by cloud provider and sent to the forensic investigator. TPM is used to guarantee the authenticity of custodies, that is, who owns, transmits, and receives these forensic data. To be precise, TPM is used for attesting the trustworthiness of the nodes that the forensic data is collected from or destined to, conducting key migration between the receiver and sender of the forensic data, and auditing the write and read activities of the custodies. In Reference 67, TPM is used for location assurance (ie, identification of the physical machines and the trusted authority that verifies the location), and for platform integrity attestation (ie, verification of the trustworthiness of the machine of the cloud operator). Using TPM helps the users to locate their virtual resources and assure that the service provider is genuine. Moreover, this mechanism prevents the cloud provider from moving the resources to an unwanted location. In a different work,[68] again the cloud operator (and not the cloud provider) is considered untrusted. Meaning that it could be malicious or the dom0 is compromised. This could result in compromising the VM of the costumer and the data on it. To hamper such threat, TPM is used to verify the integrity of the VM management dashboard of the costumer and assure that the VM created by the client is not tampered with. For assessing the trust level of the VM service, Wang et al[25] proposed a limiting trust capacity model. In this model, trust tokens are used to define the amount of resources a guest service could use. In the case that a malicious service attempts to consume a lot of resources, trust tokens limit the capacity of the resources that party could use and in this way, control the scale of the attack. TPM checks the integrity of the SW that is loaded on the system for execution, and provides a chain of trust from BIOS as the root of trust to the upper levels and services.
- *Malicious coresident VM* could influence other VMs in the system. Therefore, the cloud provider needs to guarantee the security of its VMs and the applications running on top of them. The risk of having a malicious VM could be alleviated

by detecting the infected node through periodic attestation and isolation of invalid VMs from the environment.[35] TPM provides this attestation, node identification, integrity measurement, and acts as hardware root of trust for the nodes. CloudMonatt[69] is a trusted cloud-monitoring scheme that relies on property based attestation, which is implemented on top of OpenStack. In this architecture, the VM health is monitored over its life cycle, and the overall goal is to avoid coresident VMs from maliciously affecting other VMs and also mitigate the threats coming from applications within VM. TPM is used to carry out remote attestation between the clients and their VMs through vTPM instances.

- *Attacks involving privilege escalation* is another consequence of a vulnerable cloud infrastructure. In this type of attack, an adversary exploits the vulnerabilities in order to gain higher privilege level and then uses the compromised machines to launch further attacks.[70] One example of this class of attack is root take-over attack,[71] in which an attacker disguises himself as the root user or root system and takes advantages of this role for further attacks. In the proposed work in Reference 56, a switch is unable to join a network without TPM. This, first prevents the compromise of the switch BIOS, second ensures the credibility of the spanning tree protocol and prevents an illegal switch from participating in the protocol and launching root take-over attack. Another example of attack in this threat category is when remote administrators of the IaaS provider attains root access and replaces a VM with a malicious one.[72] To mitigate this attack, the authors present a protocol to launch the VM instances of the IaaS cloud in a trusted way. To do this, TPM verifies the integrity of the VM images and also the compute host. A security profile is given to each of the compute hosts, based on which the scheduler chooses the host on which to launch the VM. This decision is made according to the trust level of the host and the trust lower bound demanded by the client when he requests launching a VM. Another proposed security architecture (see[70]) tries to counter this type of attack by combining intrusion detection, access control, and trust management. TPM is embedded on the physical machine to measure the state of the hypervisor and VMs at boot time and confirms their trustworthy behavior.

4 *Attacks on cloud backend* concerns a category that classifies the attacks based on what part of the cloud architecture they target. The attack may target any of the layers in the architecture from the low-level hardware (via BIOS manipulation or conducting replay attacks on TPM messages[73]) to upper level applications. In Reference 59, an example of a potential attack on *BIOS and firmware attacks* is given, and such attack could be mitigated by utilizing a root of trust and a trusted launch provided by trusted computing technology. Nanivati et al[74] also discuss that attacks on BIOS could be mitigated through remote attestation using TPM. From the architectural point of view, hypervisor bases the foundation of the virtualization platform, and is responsible for memory management and isolation, CPU multiplexing, and provides the required functionalities for hosting the VMs. Therefore, the *compromise of the hypervisor* could have a severe effect on the security of the upper layers, and it needs to be protected.[73,74] Trusted boot technology offered by TPM verifies the identity of the booted underlying platform and ensures the trustworthiness of the loaded virtualization platform.[74] In another layer, the attack could come from a *malicious underlying OS*.[75] Masti et al[75] propose an architecture capable of securely creating and managing multiple, concurrent execution environments. Virtualized TPM in this architecture supports multiple dynamic concurrent requests for root of trust from various VMs.

- *Forking attack or roll back attack* is one of the attacks that happens when the service provider is malicious. When ones retrieves a root hash that is stored on a remote server, there is no way to know whether the server is returning the most recent value or not. This scenario could happen when the server makes a snapshot of the recent data, for example, the root hash, and then it accepts an alteration from the system of the user. When the user (or another authorized) user accesses this data from a separate device, cloud provides the old copy of the snapshot. Therefore, these two versions could be evolved separately from two separate devices. This results in forking the contents of the stored data into multiple histories of modifications.[10] Such could be resolved by the direct communication of the devices, but to minimize the level of trust required, Tate et al[10] have proposed to use TPM and its functionalities in a virtual monotonic counters that never get repeated values.

5 *Data tampering* is the act of deliberately and maliciously altering/manipulating the data through unauthorized access. Data may be at rest or in transit. One of the use cases of TPM is minimizing the risk of such threat. OB-IMA[76] is an approach proposed for measuring the integrity of the guest VMs and the processes running on them, through reading the syscall parameters of these processes and using TPM for hashing the files. The hash of these new files are stored and the files that had been already measured will be compared to the ones stored to assure that the files (primarily configuration files or script) have not been modified. CUMULUS[77] is another framework proposed for verifying the security properties and certificates validity in all three architectural models, SaaS, PaaS, and IaaS. The framework works with issuing and revoking the security

certificates based on monitoring, tests, and trusted computing monitoring. The evidences of the security properties which are the results of tests and monitoring are store in a database. Security experts design security models based on which the system is tested against. These models are also stored in the database. In this framework, TPM confirms the integrity of the components that run the monitoring events, that is, collect and analyze the monitoring evidences, and as well the components of the framework itself. In this framework, various threats could be defined as security properties to which the systems should conform to. For instance, threats such as unauthorized access or alteration of e-Health data needs to be protected against.

- *Tampering of audit log records*: Monitoring the security of the cloud-based systems is, in some ways, similar to the procedure in-house networks, but different in the aspect that the hardware is hosted externally and out of our control. Kanstrén et al[30] discuss the opportunities of using TPM to increase the confidence and trust in security monitoring of the cloud from the viewpoint of a customer. The authors of Reference 78 propose a framework for cloud provenance, which centralizes the log records from various sources into a single database, separate from cloud resources generating the log data. In this framework, TPM is used for measuring the integrity of all the physical layer devices, for protecting the provenance data, and controlling access to the logs. In this way, it is ensured that the log records of the activities happened in cloud environments is not tampered with, and access is only given to the trusted parties in case if incident investigation required.
- *Tampering of measurement data*: In the proposed architecture in Reference 43, TPM verifies the integrity of the monitoring probes and the measurement data they provide. This ascertains that the system is executing in the expected way, the integrity of the measurement data is being protected and, the monitoring probes are not tampered with. In another work,[51] measurement data is being protected by TPM offering secure storage. In the proposed system, the hypervisor by implementing a multiplexing agent processes the measurements values and stores them securely on the TPM registers. Each VM could also run a measurement agent which is in charge of calculating integrity measurement, monitoring file execution, and communicating with multiplexing agent. The TPM registers not only hold the integrity measurement values, but also a unique identifier for each VM in the hash chain. Each measurement value is concealed via the usage of a pseudorandom value that works like a salt, to protect plain measurement values from extraction. Each of the TPM register holds measurements value of different VMs.
- *Tampering of VMs and its data at the time of transmission*: Migration of VMs has many advantages for the cloud infrastructure, such as load balancing. However, there are many risks involved in this transmission including the risk of moving the VMs to an unauthorized and untrusted destination platform and alteration and tampering of data during such transmission. TPM has been utilized to hinder these attacks by offering attestation service and encryption. Attestation establishes trusted connection between the migration nodes, and helps the hypervisor to authenticate itself to a remote hypervisor. Also, using the TPM keys, the (attestation) data is encrypted to be protected while being transmitted.[31] Zhang et al[26] discuss the several risks threatening the VM migration, including data being snooped or tampered, presence of MitM on the channel established between the migration source and destination, and replay attacks. To hamper such threats, they propose an approach to preserve the privacy and protect the integrity during and also after the live migration of the VMs. TPM is used to perform remote attestation and secure the migration process. TASMR[20] is an architecture proposed to secure MapReduce computation in hybrid clouds. Through the use of TPM, it provides authentication and authorization, integrity, confidentiality, and data auditing. It also prevents the tampering of target VMs. In another work by Wang and Liu,[79] a trusted measurement model relying on dynamic policies to protect the privacy of the VMs of users in an IaaS is proposed. TPM is used to source secure random numbers and security keys and for integrity measurements in such proposal.

6 *Data leakage* occurs when the data is maliciously exfiltrated electronically or physically to an unauthorized destination/party. To address data leakage threats in IaaS clouds, Paladi et al[23] propose a scheme for protecting the integrity and data confidentiality. In this scheme, a TPM module is used for attestation, encryption/decryption, integrity protection, and storing cryptographic keys. Hu and Ji[80] discuss that Cuckoo attack could be practical on a computer with TPM, to gain control over the communication between the TPM and the verifier. To solve this issue, they propose a protocol that makes a trust channel client and server. In this protocol, TPM generates random numbers and keys, hashes the messages, and signs AIK. Sometimes the data leakage threat is within the cloud infrastructure,[15] for instance the data could leak from one tenant to another. To solve this issue, a trust-monitoring framework is proposed that eliminates the privileged domain from the chain of trust and also partitions the different cloud tenants. In this implementation, TPM is used for trusted boot and remote attestation of the monitoring framework and virtual TPM is utilized to establish independent TCB for each tenant.

- *leakage of geolocation data* occurs when an attacker obtains the physical location of the devices (eg, mobile devices). Park et al[81] propose a trusted geolocation framework that delivers a trusted channel between the mobile client with tiny hypervisor and geolocation server and protects the cloud user device. TPM is used as for dynamic root of trust measurement, generating and storing the RSA keys, and remote attestation of geolocations of the devices in order to detect if the hypervisor is compromised. This also prevents the attacker to attain geolocation of mobile devices. Addressing the data location problem, Noman and Adams[82] present a cryptographic protocol through which the customers could verify the accurate geolocation of their data within the cloud service provider infrastructure, for instance, in which data-center their data resides. To do this, TPM is used for data possession proofs. Each of the data-center storage have built in GPS enabled TPMs. After correctly identifying themselves, clients could establish a secure communication channel with these TPMs. In this method, the TPM features such as remote attestation, integrity checking, and cryptographic functionality support are used.

4.3 | RQ3: Target of integrity measurement

This research question (RQ3) identifies the exact component of integrity assessment in the studied set of publications. Below, we discuss these targets of assessments and present some examples from the studied set of publications. In some of the works, the target of integrity measurement was a component of the cloud back-end, and in some other works the target of integrity measurements were components not specific to the cloud back-end. We divide the discussion in two parts according to this aspect:

- *Components on cloud back-end*: The studies that fall into this category aim at protecting the cloud back-end using trusted computing technology:
  - Cloud computing platform/hardware: In Reference 63, the authors propose a platform for security auditing and continuous monitoring of the remote platforms. In this approach, TPMs are used for hash-based binary measurements to assert the integrity of binaries and if the software is patched and up to date in order to avoid misconfiguration in the running software. The work proposed in Reference 74 is an online verifiable resource scheme for memory and CPU allocation in cloud computing. In this work, TPM assures the integrity of BIOS through remote attestation.
  - Hypervisor: Cafe[65] is a system developed to help developers to deploy software via a cloud marketplace that is protected from illegal piracy and copying and the reverse engineering of the applications. In this work, TPMs are utilized to attest the integrity of the hypervisors that host the VMs. The applications are installed and executed on these VMs. TPM RSA key generation is also used for secret sharing purposes between the server and hypervisors in order to transmit binaries securely.
  - VM: The distributed trust protocol[83] is proposed to provide trust between the cloud provider and customers in IaaS clouds. In this approach, TPM is used to support the trusted boot and verify the integrity of the VMs in order to protect the VMs from tampering.
  - Operating system and software stack: On the approach proposed in Reference 84, the authors address the scalability limitations that exist in the current trusted boot and integrity measurement approaches. Their middleware system closes the gap between the applications and their integrity with the hardware infrastructure, and transforms the servers into trusted IaaS platforms via the use of TPM features such as integrity measurement and monitoring. In another work,[85] an auditing and certification scheme that utilizes TPM for integrity measurement and integrity measurement to protect the complete software stack is proposed.
- *Other components*:
  - Data: In order to protect the confidentiality of data, TPM has been used to preserve data freshness and integrity,[10,86] provide integrity attestation, verify client signature authenticity, and prevent the unauthorized access to the stored and processed data.[23] The medical logs could be protected, through ensuring the secure boot of the medical devices, verifying the authentication, and detecting data tampering.[21]
  - IoT device hardware and software: As a part of the architecture proposed in Reference 87, TPM is used to provide attestation between the connecting nodes acting as root of trust for software stack. Secure storage of the TPM is used to hold keys, and sealing functionality of TPM is used to bind the container's storage protection key to TPM and making sure it is only unsealed if the container initiates in an expected way. In the authentication model[88] proposed for IoT clouds, TPM ensures the integrity of the software and hardware.

- Code: In the solution presented in Reference 89, an XML-based language framework is proposed to for managing access rights. The binding and sealing operations of TPM are used to encrypt data. This data can only be decrypted if the machine is in the expected state and provides attestation. For providing this attestation, integrity of some pieces of code needs to be assured.
- Files: The OB-IMA Scheme[76] is proposed to measure the integrity of processes that run on VMs through using TPM for hashing the file that are accessed and executed by a process. The main goal of this work is to prevent tampering of these files. A similar approach has been used in another work[90] to protect health information platforms through integrity measurements. In this work, the files hashes are protected through static attestation, and running processes are protected through dynamic attestation.

4.4 | RQ4: Level of integrity measurement

Compared to conventional computing systems, in a virtual environment, there are additional layers to support the multitenancy feature of the cloud. The hypervisor residing between the hardware and the OS, and the OS is now running on VMs on top of the hypervisor. These added layers expand the attack surface and introduce new attacks in addition to the ones in traditional computer systems. Hence, it is important that all these layers are being protected and trust is measured. To this end, trusted computing technology has been widely used to offer security and trust in various layers. This research question (RQ4) is designed to identify the different layers of cloud computing that TPM features have been leveraged in to present. In the following, we present a categorization of these layers and some examples of the studied works that fall in these categories:

1. *Hardware, platform, server*: In distributed systems models such as IoT and cloud computing the security of the host platform is highly critical as it is hosting the storage or the process of the (great amount of) data. TPM is one robust solution for authentication of the platform as well as verifying its integrity. In the proposed solution in Reference 90, the HCloud (a Private cloud for health-care data) is extended and TPM is embedded on the host platform to authenticate the clients, and to measure the static and dynamic integrity. It is discussed that a trusted cloud could provide external certified services and conduct internal trusted scheduling. Each service that is provided to external has a unique image/process running in memory. This process needs to be verified when being loaded to the memory. Static measurement could be performed using TPM, to verify the files and packages that are stored in the physical repository. Also, the integrity of the database is protected using TPM signatures to provide hardware level security. For dynamic integrity measurement of the processes, TPM PCRs are utilized to store and present the step-by-step measured values. Yang et al[86] propose a design that provides guarantees to the clients about the freshness, validity, and integrity of their data stored on the cloud. To achieve this goal, they use a trusted hardware device on an untrusted server that enables the clients to validate the freshness and integrity of their data.

   Before the system runs, it needs to be verified whether the platform has booted in the trusted way. Reference 91 discusses the TPM as the root of trust for measuring the boot process and attestation of the critical steps for asserting the platform boot integrity. In a virtual environment, TPM provides a chain of trust rooted in hardware that will be extended to the hypervisor. Platform boot integrity is ensured if the key components (ie, BIOS, firmware, and hypervisors) have demonstrated their integrity. To ensure the integrity of the launch components, two steps are required: first, the boot process is measured, and second, attestation, that is, assurance and that the executed components are trusted components.
2. *Hypervisor*: A trusted geolocation framework which creates a trusted channel between the server and small hypervisors on the mobile clients is presented in Reference 81. A TPM is embedded on these hypervisors to act as a root of trust and provides dynamic trust measurements to attest the locations of the cloud devices. An attestation protocol verifies whether the hypervisor is compromised. Wu et al[28] discuss that at the time of VM migration there are two issues that need to be addressed: first is the proper time to start the migration and second, the trustworthiness of the source hypervisor. To address the later issue, they propose a secure migration framework based on the Xen hypervisor. This design uses trusted computing and adjacent integrity measurement to dynamically monitor the integrity of the adjacent hypervisor. More specifically, to ensure that the host is being monitored at all times, when a host initializes, TPM module verifies the hypervisor and sends an update message to the integrity validation table (which holds the list of trusted hypervisors) and marks that hypervisor as trusted. This means that if such message is not received, that particular hypervisor is considered untrusted. Moreover, the running state of the hypervisor is dynamically monitored and, in case the hypervisor is compromised, a corresponding update message is sent to report the compromised situation. Based on the result of this integrity table, the decision is made whether the source hypervisor is trusted and the VM migration process should be initiated. In another work,[92] an hypervisor-based remote attestation scheme is proposed for virtualizing network functions and private clouds. This work supports a great amount of VM attestations using a physical module.

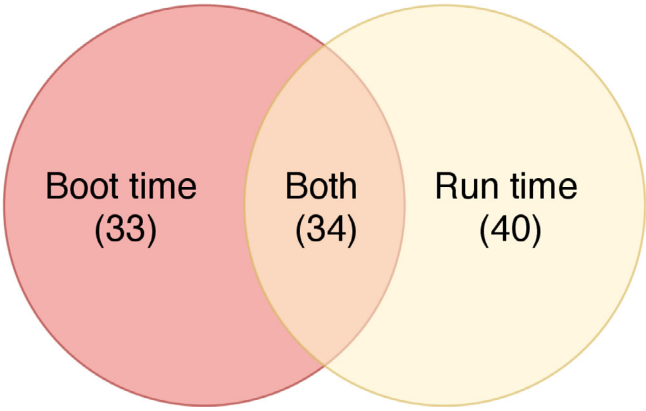

FIGURE 7 Distribution of studies classified based on the phase that trusted platform module (TPM) features are used

3. *VM*: Through the usage of trusted computing technology, the integrity of the VMs are verified at the time of launching and booting them as well as later at the time of run. In the implementation of the monitoring framework in Reference 15, the remote attestation functionality of TPM and its support for trusted boot are utilized for certifying the integrity of the monitoring environment. To prevent and detect the attacks that misuse the cloud infrastructure to conduct their malicious intent, Varadharajan et al[93] propose an approach that, with the help of trusted attestation techniques certifies the security of the cloud provider, its VMs and the services running on top of the VMs. If there is a discrepancy between the behavior of a VM and the defined security properties, that VM will be isolated and terminated. To do this, using TPM, binary and property attestation is performed between the tenant VM and the customer. Binary attestation certifies the state of the components at boot time and property attestation indicate the run time state. Periodic attestation of the integrity provided by keylime Scheme[40] ensures the run time integrity of the system.

   The use of TPM to support VM migration has been extensively studied. The research works in this area focus on proposing a secure protocol[71,80,83] to support the live migration with the use or propose approaches that leverage TPM to perform integrity measurement and address the problem of untrusted source in VM migration,[28] verify the integrity of the destination platform and to ensure that only the destination node receives the migrated VM data,[94] present attestation Services to establish a trusted connection between the nodes of migration, provide keys used for encrypting data for attestation,[31] and to provide node attestation and generating EK.[95]
4. *Application, software*: Security and integrity of the SW stack is a crucial factor in the overall security status of the platform. This means that the security of the platform improves through a secure SW stack, or could be degraded through SW exposure. Although TPM is a physical chip residing on the hardware of the platform, its functionalities could be leveraged to secure the application layer and the SW stack. Shield[35] presents a hardware root of trust on the network nodes and protects the SW stack from boot to the application layer, by measuring each component, securing the measurements with digital signatures, and unique identification of the nodes. Jayaram et al[96] propose using TPM as a hardware-rooted integrity verification means to validate the integrity of the SW stack on IaaS. TPM measures the integrity of hypervisor, OS, the SW packages, and also validates these measurements by interacting with the attestation services. Pasquier et al[97] propose a hardware-backed decentralized information flow control for managing data in a cloud-supported IoT. In this framework, trusted computing offers remote attestation to elevate the trust level in the infrastructure. Remote attestation verifies the integrity of information flows and validates the SW configurations. More specifically, TPM performs binary attestation and SW integrity verification. The proposed scheme in Reference 27 uses hardware root of trust to support the VM migration and to mitigate the attacks on SW security solutions. In the P-Cop approach,[41] TPMs installed on the servers perform bootstrapping and attestation of the computing nodes to verify the SW configuration of these nodes and check if they have been set up properly with a trusted container.
5. *Various layers*: In a big portion of the studied papers, TPM features have been used to offer trust and integrity in various layers in the virtual environment. For instance, integrity measurement was done in different layers, to present the hardware-rooted chain of trust from lower layers to upper ones.[25,98,99]

4.5 | RQ5: Phase of integrity measurement

Figure 7 illustrates the distribution of studies when categorized based on the phase in which the TPM features are used. These phases are boot time and run time. In some of the studied works (ie, 34 papers) TPM primitives are invoked at both phases, during the boot and at run time. It is worth noting that among the 120 studied publications, there were 13 cases in which the phase was not clearly discussed or was not possible to extract this data. In the following discussion, we discuss how TPM features have been used in each of the phases.

- *Boot time*: One of the principles pursued in trusted platform technologies is verifying the trust and integrity of the critical component before they are loaded and executed. Therefore, before the run time integrity checking, it needs to be ensured that the system is correctly booted and an appropriate operating system is running. Trusted platform achieves this trust through creating a chain of trust that starts with the core root of trust for measurement (CRTM) that is a trusted code in BIOS boot block. CRTM measures the integrity value of the other entities, and remains unaltered throughout the lifetime of the platform. CRTM is a supplement to the normal BIOS, and it first runs to measure the BIOS block and the hardware, and then passes the control over to the bootloader. Then measures the OS kernel image before passing the control to the OS. At each step in this boot process, the measurement value is taken and according to that the relevant TPM PCR value is extended. This measurement process attests the system integrity, and ensures that the boot and OS software is the version intended by the manufacturer and has not been tampered with by malicious third parties or a malware.[100] Nanavati et al[74] argue that in virtual environments, trusted boot technology allows the users to attest the identity of the underlying booted platform, and also ensure that the loaded virtualization platform is trustworthy. This gives a concrete guarantee to the users about the virtualization platform to which they could commit critical data. TPM, by using cryptographic primitives, provides this trusted boot for the virtualization platform and establishes root of trust for them. In the ALIBI trusted monitoring framework,[17] TPM is used to offer root of trust on the service provider platform. The TPM PCRs record the state of software that is executing on the platform, and since the PCR values are append-only, the previous records are eliminated only via rebooting. TPM also holds a public-private key-pair. The private key remains inside the secure environment of the TPM only, and is used to compute the signature of the attestation value (the measurements accumulated from authenticated boot), that the TPM has generated. Using the public key of the TPM, the external verifier could check the validity of the signature and deduce that the PCR values present the state of platform software.

  The TPM functionalities have been widely used for protecting the VMs, at the launch time and migration and throughout the whole VM lifecycle. At the time of VM launch, it is essential that both the client VM and the cloud provider platform have a mutual trust in each other. This guarantees the trustworthiness and reasonable security of the computing resources/host and also guarantees the confidentiality and integrity of the VM instance. To address the limitations in the process of trusted VM launch, Paladi et al[72] have proposed a protocol that relies on TPM functionalities together with asymmetric cryptography and hashing to provide assurance on the integrity of the host and the client VM image that has requested to launch. This protocol eliminates the need of client-side prepackaging of VM images, in order to provide full scheduling flexibility on the cloud IaaS side, and enables the cloud provider to choose a target trusted host without direct involvement of the client.
- *Run time*: After a trusted boot of the system, it is essential to ensure that the system is still behaving as expected and in a trusted manner at all time during the execution. The dynamic attestation service offered by trusted computing attests the run time behavior of the system and proves it run time integrity to a remote party. Keylime[40] is a trusted cloud key management system that offers bootstrapping hardware based cryptographic identities for the nodes of IaaS, and also integrity monitoring of these nodes through periodic attestation. This periodic attestation monitors the integrity of the system continuously and polls the integrity state of the cloud nodes periodically to determine to detect whether any run time policy was violated. Trusted cloud root broker[66] is another framework for evaluating the run time trustworthiness of the system through dynamic attestation of the application.

  Direct anonymous attestation (DAA) is a cryptographic protocol designed for remote attestation, to provide signer authentication and privacy. However, this scheme is designed only for single trusted domain attestation, and therefore, it cannot be deployed in the environments in which authentication servers and users belong to various domains, such as cloud computing, mobile networks, or IoT. Yang et al[101] propose a scheme for DAA on cross trusted domains that adapts the TPM DAA to allow users authenticate a computation platform when accessing a visiting trusted domain. This delegation model encompasses the two stages of domain attestation and platform attestation, to establish a trusted relationship model between various domains.

  VM migration is the process of moving a VM from a host, platform, or a storage location to another host. This migration offers many benefits to the hosting organization, such as load balancing and disaster recovery. To support the secure migration of the VMs, many approaches and measures are available. Among all, TPM has also been used to protects the cloud instances and data at the time of migrating from a zone to another.[26,27,31,95] SIMM[27] enables the binding function of trusted computing to protect the migrating instances. It first check, the migration feasibility between the two zones on the cloud and then, before launching the live migration, encrypts the instance data on the source. This data will be decrypted on the destination using the local TPM coprocessor. In another work that addresses the issues related to secure live migration of the VMs,[95] authors use trusted computing for node attestation and guaranteeing the continuity and coherence of the trusted status throughout the lifecycle of the VMs. TPM is also used for generating EK.

- *Boot time and run time*: In the studied set of papers, a wide number of papers were using TPM and trusted computing technologies both for a trustful boot of the system and for run time integrity assurance. In Reference 42, trust computing's sealed storage is utilized to improve the chain of trust and to ensure the system integrity at boot time through dual verifiable bootstrapping. In the proposed approach, they monitor and record the run time status of the security of the critical applications on the targeted VMs. Additionally, they have defined a way to report the run time trust status of the cloud environment to its users via platform attestation. In Reference 48, an agent-based system is proposed, which is able to detect alterations, both at boot time and run time, to software or hardware done by the cloud provider to obtain access to the data of the costumers. Authors of Reference 25 present a trust capacity model to assess the trust level of services that are running on the VMs. TPM measures the integrity of loaded software on the system for execution, and for creating a chain of trust, and verification of transition of trust from the root of trust to a service.

5 | CHALLENGES, OPEN ISSUES, AND FUTURE DIRECTIONS

Trusted computing offers various techniques for trust assurance in the system. Integrity measurement (=remote attestation) is one of these techniques that guarantees that the system is in a good state.[102] In the literature reviewed, the integrity is measured at different phases and also different layers. The integrity of the system could be measured at boot time, load time, and run time. At boot time, the platform is measured to make sure it is in trusted state. First, the state of BIOS is measured and the value is sent and recorded in PCR, before it is being executed. After BIOS, the next components in the boot sequence are measured one after another (like PCI option ROM, and boot loader). In other words, a chain of measurement happens from BIOS up to the kernel.

Load time integrity checking guarantees the state of the system when the program is loaded. However, load-time integrity checking does not have a view to the code as it starts to run and the system configuration might change after the boot time. Therefore, run time verification is still necessary to measure the run time state and detect possible modifications.[103] Run time integrity measurement assures that the program is still in a safe state as it is being executed. Sianipar et al[48] propose a system that monitors the integrity of the cloud at boot time and at run time using TPM.

We also observed several potential challenges and issues, as well as future work that can be developed, on TPM usage in Cloud computing environments. For this, we propose and aim to answer five main questions that shape and present these challenges:

1. *What is needed to have a wider adoption of TPM in the cloud ecosystem?* The main steps have been taken toward the adoption of TPM devices. The TPM 2.0 specification is ISO compliant, as the ISO/IEC 11889-1:2015,[104] and it was ratified by several governments, specially some of the largest economies, such as the USA, Japan, Russia, UK, and France, among others. More recently, in 2018, an open-source middleware was made available by Infineon to use with TPM 2.0.[105] What should now be required is a simpler integration and adoption of TPM for the cloud, with specific middleware capable of interacting both with hypervisors and VMs, that can be easily implemented without requiring extensive technical knowledge, or that can be toggled as a simple configuration option.
2. *Should TPM be delivered as a service in the cloud context?* While not directly, the TPM can be used to provide security in an IaaS cloud infrastructure, as depicted in,[106] to ensure both VM and remote resources integrity. The vTPM approach is another method of applying TPM in the cloud, that works as a service that is made available to all VMs running on a given hypervisor. Through, it is possible to deliver TPM features to each individual VM, in a transparent manner to the end user. Each VM can have its unique vTPM instance, to which other VMs have no access. This could be considered the wider approach to use TPM as one would use on a regular machine/server, and will serve as a root of trust for each specific VM that has a vTPM instance associated with it.[107] However, if the TPM could be fully made into a service unto itself, it would allow to have a centralized root of trust in the cloud, from where other machines could draw from to provide TPM features to their hypervisors and VMs.
3. *Should the set of provided TPM features be extended or reduced for the cloud context?* The total TPM feature set includes random number generation, remote attestation, authentication, secure generation of cryptographic keys, binding, and sealing. Taking the cloud ecosystem into consideration, this feature set is fully applicable, and able to be used to its full extent. If possible, the main expansion that could be performed would be the capability of the TPM to be aware of the cloud infrastructure, so that it could directly adapt to enable and make its features available directly to the VMs, while keeping the information between them separate.
4. *Which layer requires more integration work?* On hardware and infrastructure levels, the TPM is already fully useable to perform its main functionalities, such as integrity measurement, as it can interact directly with the first and, as previously

mentioned, indirectly with the latter, through systems similar to the vTPM. Greater integration is needed in the platform and application layers, for the TPM to be aware of them and their specificities, so that TPM features can be applied and used in a transparent manner in these layers.

5 *How will TPM architecture in the cloud most likely evolve in the near future?* TPM is often seen as a computer system component that comes with every device. In the cloud this might not be case, as a central server might be providing attestation services for several other servers, devices and applications. In such case, a form of root of trust might still be needed in each server to support confidentiality and correctness of communications between the devices, as well as a protocol to enable interactions between it and the virtualized hosts.

6 | CONCLUSION

In this paper, we surveyed the literature for applications of TPM in the context of cloud computing, with publication dates between 2013 and 2018. As the result of this research, we collected 120 papers, from which we extracted data and analyzed them. Through this study, we were able to identify the current trends and objectives of this technology in cloud computing. We learned that Remote attestation and integrity measurement were the main purposes of utilizing TPM in cloud computing. We also presented a classification of the security threats and discussed in detail how they are thwarted by the use of TPM. Due to the fact that integrity measurement is one of the main utilities provided by TPM to assure that the system is in a good state, we gave special attention to this feature in our study. More specifically, we studied at what phase and at which software layer, the integrity is measured and at what phase/layer there is still room for research. Toward the end, the main research gaps were pinpointed and discussed.

ACKNOWLEDGMENTS

The authors gratefully acknowledge Tekes—the Finnish Funding Agency for Innovation, DIMECC Oy, the Cyber Trust research program, Nokia Foundation, and Tekniikan edistämissäätiö for their support. This work was partially performed under the scope of Project SECURIoTESIGN with funding from FCT/COMPETE/FEDER (Projects with reference numbers UID/EEA/50008/2019 and POCI-01-0145-FEDER-030657), and funding by FCT-Fundação para a Ciência e Tecnologia, through the research grant SFRH/BD/133838/2017. The authors would like to thank Samuel Laurén for sharing his valuable insights in this project.

ORCID

Shohreh Hosseinzadeh https://orcid.org/0000-0002-9282-3981

REFERENCES

1. Trusted Computing Group (TCG). http://www.trustedcomputinggroup.org/. Accessed May 26, 2019.
2. Lioy A, Su T, Shaw AL, Attak H, Lopez DR, Pastor A. *Trust in SDN/NFV Environments*. Cham: Springer International Publishing; 2017:103-124.
3. Kitchenham B, Brereton P. A systematic review of systematic review process research in software engineering. *Inform Software Technol*. 2013;55(12):2049-2075.
4. Beekman JG, Manferdelli JL, Wagner D. Attestation transparency: building secure internet services for legacy clients. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, ASIA CCS'16; 2016; New York, NY: ACM; 687-698
5. Yeluri R, Castro-Leon E. *Attestation: Proving Trustability*. Berkeley, CA: Apress; 2014:65-91.
6. Javanmard M, Salehi MA, Zonouz S. TSC: trustworthy and scalable cytometry. In: 2015 IEEE 17th International Conference on High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS); 2015; 1356-1360.
7. Park S, Yoon JN, Kang C, Kim KH, Han T. TGVisor: a tiny hypervisor-based trusted geolocation framework for mobile cloud clients. In: 2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud); 2015; 99-108.
8. Tomlinson A. *Introduction to the TPM*. Boston, MA: Springer US; 2008:155-172.
9. Davi L, Sadeghi AR, Winandy M. Dynamic integrity measurement and attestation: towards defense against return-oriented programming attacks. In: Proceedings of the 2009 ACM workshop on Scalable trusted computing; 2009; New York, NY: ACM; 49-54.
10. Tate SR, Vishwanathan R, Everhart L. Multi-user dynamic proofs of data possession using trusted hardware. In: Proceedings of the Third ACM Conference on Data and Application Security and Privacy, CODASPY'13; 2013; New York, NY: ACM; 353-364.

11. Du R, Pan W, Tian J. Dynamic integrity measurement model based on vTPM. *China Commun*. 2018;15(2):88-99. https://doi.org/10.1109/CC.2018.8300275.
12. Chakraborti A, Jain B, Kasiak J, Zhang T, Porter D, Sion R. Dm-x: protecting volume-level integrity for cloud volumes and local block devices. In: Proceedings of the 8th Asia-Pacific Workshop on Systems, APSys'17; 2017; New York, NY: ACM; Vol 16, 1-16:7
13. AlBelooshi B, Salah K, Martin T, Damiani E. Securing cryptographic keys in the IaaS cloud model. In: 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC); 2015; 397-401.
14. Cheng Y, Ding X. *Guardian: Hypervisor as Security Foothold for Personal Computers*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013:19-36.
15. Zou D, Zhang W, Qiang W, et al. Design and implementation of a trusted monitoring framework for cloud platforms. *Future Gen Comp Syst*. 2013;29(8):2092-2102. Including Special sections: Advanced Cloud Monitoring Systems & The fourth IEEE International Conference on e-Science 2011 & Cluster, Grid, and Cloud Computing. https://doi.org/10.1016/j.future.2012.12.020.
16. Singh NK, Patel YS, Das U, Chatterjee A. NUYA: an encrypted mechanism for securing cloud data from data mining attacks. In: 2014 International Conference on Data Mining and Intelligent Computing (ICDMIC); 2014; 1-6.
17. Chen C, Maniatis P, Perrig A, Vasudevan A, Sekar V. Towards verifiable resource accounting for outsourced computation. In: Proceedings of the 9th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, VEE'13; 2013; New York, NY: ACM; 167-178.
18. Thilakanathan D, Chen S, Nepal S, Calvo RA, Liu D, Zic J. Secure multiparty data sharing in the cloud using hardware-based TPM devices. In: 2014 IEEE 7th International Conference on Cloud Computing; 2014; 224-231.
19. Part TM. *1-Design Principles, Specification Version 1.2*. Trusted Computing Group; 2006.
20. Haouari A, Zbakh M, Cherkaoui R, Samadi Y, Kasmi N. TASMR: towards advanced secure mapreduc framework across untrusted hybrid clouds. In: 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech); 2017; 1-9
21. Nguyen H, Acharya B, Ivanov R, et al. Cloud-based secure logger for medical devices. In: 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE); 2016; 89-94
22. Baracaldo N, Androulaki E, Glider J, Sorniotti A. Reconciling end-to-end confidentiality and data reduction in cloud storage. In: Proceedings of the 6th Edition of the ACM Workshop on Cloud Computing Security, CCSW'14; 2014; New York, NY: ACM; 21-32.
23. Paladi N, Gehrmann C, Morenius F. *Domain-Based Storage Protection (DBSP) in Public Infrastructure Clouds*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013:279-296.
24. Yamaji K, Nakamura M, Nagai Y, Ito T, Sato H. Specifying a trust model for academic cloud services. In: 2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud); 2016; 91-99.
25. Wang Y, Chandrasekhar S, Singhal M, Ma J. A limited-trust capacity model for mitigating threats of internal malicious services in cloud computing. *Cluster Comp*. 2016;19(2):647-662.
26. Zhang F, Chen H. Security-preserving live migration of virtual machines in the cloud. *J Network Syst Manag*. 2013;21(4):562-587.
27. Syed TA, Musa S, Rahman A, Jan S. Towards secure instance migration in the cloud. In: 2015 International Conference on Cloud Computing (ICCC); 2015; 1-6.
28. Wu T, Yang Q, He Y. A secure and rapid response architecture for virtual machine migration from an untrusted hypervisor to a trusted one. *Front Comp Sci*. 2017;11(5):821-835. https://doi.org/10.1007/s11704-016-5190-6.
29. Liu A, Liu J, Uehara T. Secure streaming forensic data transmission for trusted cloud. In: Proceedings of the 2Nd International Workshop on Security and Forensics in Communication Systems, SFCS'14; 2014; New York, NY: ACM; 3-10.
30. Kanstrén T, Lehtonen S, Kukkohovi H. Opportunities in Using a Secure Element to Increase Confidence in Cloud Security Monitoring. In: 2015 IEEE 8th International Conference on Cloud Computing; 2015: 1093-1098
31. Azougaghe A, Oualhaj OA, Hedabou M, Belkasmi M, Kobbane A. Many-to-one matching game towards secure virtual machines migration in cloud computing. In: 2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS); 2016; 1-7
32. Cohen JC, Acharya S. Towards a trusted HDFS storage platform: Mitigating threats to Hadoop infrastructures using hardware-accelerated encryption with TPM-rooted key protection. *J Inform Security Appl*. 2014;19(3):224-244. https://doi.org/10.1016/j.jisa.2014.03.003.
33. Wu X, Xie X, Liu C, Li C. *An Improved Design of the Trustworthiness Authentication Mechanism of IaaS*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013:219-226.
34. Tan WY, Sinha R, Manferdelli JL, Seshia SA. *Formal Modeling and Verification of CloudProxy*. Cham: Springer International Publishing; 2014:87-104.
35. Attak H, Casassa-Mont M, Dávila C, et al. *SHIELD: Securing Against Intruders and Other Threats Through an NFV-Enabled Environment*. Cham: Springer International Publishing; 2017:197-225.
36. Parno B, McCune JM, Perrig A. Bootstrapping Trust in Commodity Computers. In: 2010 IEEE Symposium on Security and Privacy; 2010; 414-429
37. Ganapathy V. *Reflections on the Self-service Cloud Computing Project*. Cham: Springer International Publishing; 2015:36-57.
38. Demchenko Y, Ngo C, Laat dC, Lopez DR, Morales A, García-Espín JA. *Security Infrastructure for Dynamically Provisioned Cloud Infrastructure Services*. London: Springer London; 2013:167-210.
39. Demchenko Y, Ngo C, Laat dC, Membrey P, Gordijenko D. *Big Security for Big Data: Addressing Security Challenges for the Big Data Infrastructure*. Cham: Springer International Publishing; 2014:76-94.
40. Schear N, Cable PT, Moyer TM, Richard B, Rudd R.. Bootstrapping and maintaining trust in the cloud. In Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC'16. 2016; New York, NY: ACM; 65-77
41. Braga B, Santos N. P-Cop: a cloud administration proxy to enforce bipartite maintenance of PaaS services. In: 2016 IEEE 9th International Conference on Cloud Computing (CLOUD); 2016; 888-893

42. Zou B, Zhang H. Integrity protection and attestation of security critical executions on virtualized platform in cloud computing environment. In: Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing; 2013; 2071-2075.
43. Kanstrén T, Lehtonen S, Savola R, Kukkohovi H, Hätönen K. Architecture for high confidence cloud security monitoring. In: 2015 IEEE International Conference on Cloud Engineering (IC2E); 2015; 195-200.
44. Yu Z, Wang Q, Zhang W, Dai H. A cloud certificate authority architecture for virtual machines with trusted platform module. In: 2015 IEEE 17th International Conference on High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS); 2015; 1377-1380.
45. Syed TA, Jan S, Musa S, Ali J. Providing efficient, scalable and privacy preserved verification mechanism in remote attestation. In: 2016 International Conference on Information and Communication Technology (ICICTM); 2016; 236-245
46. Walsh K, Manferdelli J. Mechanisms for mutual attested microservice communication. In: Companion Proceedings of the10th International Conference on Utility and Cloud Computing, UCC'17 Companion. 2017; New York, NY: ACM; 59-64
47. Zic J, Hardjono T. Towards a cloud-based integrity measurement service. *J Cloud Comp: Adv, Syst Appl*. 2013;2(1):1-9.
48. Sianipar J, Saleh E, Meinel C. Construction of agent-based trust in cloud infrastructure. In: 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC); 2014; 941-946.
49. Jayarathna D, Varadharajan V, Tupakula U. Integrated security for services hosted in virtual environments. In: 2016 IEEE Trustcom/BigDataSE/ISPA; 2016; 82-89
50. Stallings W, Brown LB. *Computer security*. Upper Saddle River,New Jersey, United States: Prentice-Hall; 2008.
51. Velten M, Stumpf F. *Secure and Privacy-Aware Multiplexing of Hardware-Protected TPM Integrity Measurements Among Virtual Machines*. Berlin, Heidelberg: Springer; 2013:324-336.
52. Mo J, Hu Z, Lin Y. A user authentication scheme based on trusted platform for cloud computing. In: Security, Privacy, and Anonymity in Computation, Communication, and Storage. Springer International Publishing; 2016; Cham; 122-130.
53. Shi Y, Zhao B, Yu Z, Zhang H. A security-improved scheme for virtual TPM based on KVM. *Wuhan Univ J Natural Sci*. 2015;20(6):505-511.
54. Bleikertz S, Bugiel S, Ideler H, Nürnberger S, Sadeghi AR. *Client-Controlled Cryptography-As-A-Service in the Cloud*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013:19-36.
55. Zakaria I, Mustaha H. FADETPM: Novel approach of file assured deletion based on trusted platform module. In: 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech); 2017; 1-4
56. Ruan A, Martin A. *NeuronVisor: Defining a Fine-Grained Cloud Root-of-Trust: 184–200*. Cham: Springer International Publishing; 2015.
57. Cusack B, Ghazizadeh E. Evaluating single sign-on security failure in cloud services. *Business Horizons*. 2016;59(6):605-614. Cybersecurity in 2016: people, technology, and processes. https://doi.org/10.1016/j.bushor.2016.08.002.
58. Lee H, Park S, Seo C, Shin SU. DRM cloud framework to support heterogeneous digital rights management systems. *Multimedia Tools Appl*. 2015;75(22):14089-14109.
59. Futral W, Greene J. *Creating a More Secure Datacenter and Cloud*. Berkeley, CA: Apress; 2013:105-118.
60. Wang J, Zhao B, Zhang H, et al. POSTER: an E2E trusted cloud infrastructure. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, CCS'14; 2014; New York, NY: ACM; 1517-1519.
61. Khan I, Rehman uH, MHF A-k, Anwar Z, Alam M. A thin client friendly trusted execution framework for infrastructure-as-a-service clouds. *Future Generation Computer Systems*. 2018;89:239-248. https://doi.org/10.1016/j.future.2018.06.038.
62. Chen C, Raj H, Saroiu S, Wolman A. cTPM: a cloud TPM for cross-device trusted applications. In: Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation, NSDI'14; 2014; Berkeley, CA; USENIX Association; 187-201.
63. Aslam M, Gehrmann C, Björkman M. Continuous security evaluation and auditing of remote platforms by combining trusted computing and security automation techniques. In: Proceedings of the 6th International Conference on Security of Information and Networks, SIN'13; 2013; New York, NY: ACM; 136-143.
64. Wei J, Pu C, Rozas CV, Rajan A, Zhu F. *Modeling the Runtime Integrity of Cloud Servers: A Scoped Invariant Perspective*. London: Springer London; 2013:211-232.
65. Kim CH, Park S, Rhee J, Won JJ, Han T, Xu D. CAFE: a virtualization-based approach to protecting sensitive cloud application logic confidentiality. In: Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIA CCS'15. 2015; New York, NY: ACM; 651-656.
66. Ba H, Zhou H, Wang Z, Ren J, Hong T, Li Y. *Application-Assisted Dynamic Attestation for JVM-Based Cloud*. Cham: Springer International Publishing; 2015:691-700.
67. Krauß C, Fusenig V. *Using Trusted Platform Modules for Location Assurance in Cloud Networking*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013:109-121.
68. Butt S, Ganapathy V, Srivastava A. On the control plane of a self-service cloud platform. In: Proceedings of the ACM Symposium on Cloud Computing, SOCC'14; 2014; New York, NY: ACM; 1-13.
69. Zhang T, Lee RB. CloudMonatt: an architecture for security health monitoring and attestation of virtual machines in cloud computing. *SIGARCH Comput. Archit. News*. 2015;43(3):362-374.
70. Varadharajan V, Tupakula U. *Integrated Security Architecture for Virtual Machines*. Cham: Springer International Publishing; 2013:140-153.
71. Lai Y, Liu Z, Pan Q, Liu J. Study on Cloud Security Based on Trust Spanning Tree Protocol. *Int J Theoretical Phys*. 2015;54(9):3311-3330.
72. Paladi N, Gehrmann C, Aslam M, Morenius F. *Trusted Launch of Virtual Machine Instances in Public IaaS Environments*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013:309-323.
73. Sgandurra D, Lupu E. Evolution of attacks, threat models, and solutions for virtualized systems. *ACM Comput Surv*. 2016;48(3):46:1-46:38.

74. Nanavati M, Colp P, Aiello B, Warfield A. Cloud Security: A Gathering Storm. *Commun ACM*. 2014;57(5):70-79.
75. Jayaram Masti R, Marforio C, Capkun S. An architecture for concurrent execution of secure environments in clouds. In: Proceedings of the 2013 ACM Workshop on Cloud Computing Security Workshop, CCSW'13; 2013; New York, NY: ACM; 11-22.
76. Xing B, Han Z, Chang X, Liu J. OB-IMA: out-of-the-box integrity measurement approach for guest virtual machines. *Concurr Comp: Practice Exp*. 2015;27(5):1092-1109.
77. Egea M, Mahbub K, Spanoudakis G, Vieira MR. *A Certification Framework for Cloud Security Properties: The Monitoring Path*. Cham: Springer International Publishing; 2015:63-77.
78. Abbadi IM. A framework for establishing trust in Cloud provenance. *Int J Inform Secur*. 2013;12(2):111-128.
79. Wang L, Liu F. A trusted measurement model based on dynamic policy and privacy protection in IaaS security domain. *EURASIP J Inform Secur*. 2018;2018(1):1. https://doi.org/10.1186/s13635-018-0071-1.
80. Hu W, Ji D. *Formal Analysis of Dynamic Domain Establishment Protocol in Cloud Logging Service*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013:24-38.
81. Park S, Won JJ, Yoon J, Kim KH, Han T. A tiny hypervisor-based trusted geolocation framework with minimized TPM operations. *J Syst Software*. 2016;122:202-214. https://doi.org/10.1016/j.jss.2016.09.026.
82. Noman A, Adams C. Hardware-based DLAS: achieving geo-location guarantees for cloud data using TPM and Provable Data Possession. In: 2014 17th International Conference on Computer and Information Technology (ICCIT); 2014; 280-285.
83. Kashif UA, Memon ZA, Balouch AR, Chandio JA. Distributed trust protocol for IaaS Cloud Computing. In: 2015 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST); 2015; 275-279.
84. Jayaram KR, Safford D, Sharma U, Naik V, Pendarakis D, Tao S. trustworthy geographically fenced hybrid clouds. In: Proceedings of the 15th International Middleware Conference, Middleware'14. 2014; New York, NY: ACM; 37-48.
85. Aslam M, Gehrmann C, Björkman M. ASArP: automated security assessment & audit of remote platforms using TCG-SCAP synergies. *J Inform Secur Appl*. 2015;22:28-39. Special Issue on Security of Information and Networks. https://doi.org/10.1016/j.jisa.2014.09.001.
86. Yang HJ, Costan V, Zeldovich N, Devadas S. Authenticated storage using small trusted hardware. In: Proceedings of the 2013 ACM Workshop on Cloud Computing Security Workshop, CCSW'13; 2013; New York, NY: ACM; 35-46.
87. Brost GS, Huber M, Weiß M, Protsenko M, Schütte J, Wessel S. An ecosystem and iot device architecture for building trust in the industrial data space. In: Proceedings of the 4th ACM Workshop on Cyber-Physical System Security, CPSS'18. 2018; New York, NY: ACM; 39-50
88. Barreto L, Celesti A, Villari M, Fazio M, Puliafito A. An authentication model for IoT clouds. In: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, ASONAM'15; 2015; New York, NY: ACM; 1032-1035.
89. Paladi N, Michalas A, Gehrmann C. Domain based storage protection with secure access control for the cloud. In: Proceedings of the 2nd International Workshop on Security in Cloud Computing, SCC'14; 2014; New York, NY: ACM; 35-42.
90. He CG, Xu D, Fan X, Li Y. Making HCloud service trustworthy with TPM-based extension. In: International Conference on Cyberspace Technology (CCT 2013); 2013; 268-273.
91. Yeluri R, Castro-Leon E. *Boundary Control in the Cloud: Geo-Tagging and Asset Tagging*. Berkeley, CA: Apress; 2014:93-121.
92. Lauer H, Kuntze N. Hypervisor-based attestation of virtual environments. In: 2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld); 2016; 333-340
93. Varadharajan V, Tupakula U. Counteracting security attacks in virtual machines in the cloud using property based attestation. *J Network Comp Appl*. 2014;40:31-45.
94. Fan P, Zhao B, Shi Y, Chen Z, Ni M. An improved vTPM-VM live migration protocol. *Wuhan Univ J Natural Sci*. 2015;20(6):512-520.
95. He X, Tian J. *A trusted VM live migration protocol in IaaS. Trusted Computing and Information Security*. Singapore: Springer Singapore; 2017:41-52.
96. Jayaram KR, Milenkoski A, Kounev S. *Software Architectures for Self-protection in IaaS Clouds*. Cham: Springer International Publishing; 2017:611-631.
97. Pasquier TFJM, Singh J, Bacon J. Clouds of things need information flow control with hardware roots of trust. In: 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom); 2015; 467-470.
98. Luo W, Liu W, Luo Y, et al. Partial attestation: towards cost-effective and privacy-preserving remote attestations. In: 2016 IEEE Trustcom/BigDataSE/ISPA; 2016; 152-159
99. Kebbedies J, Spillner J, Braun I, Schill A. Conceptualized policy design for user-regulated trusted clouds. In: Proceedings of the 8th International Conference on Utility and Cloud Computing, UCC'15; 2015; Piscataway, NJ: IEEE Press; 601-606
100. Gopalan A, Gowadia V, Scalavino E, Lupu E. *Policy Driven Remote Attestation. Security and Privacy in Mobile Information and Communication Systems*. Berlin, Heidelberg: Springer; 2012:148-159.
101. Yang L, Ma J, Lou W, Jiang Q. A delegation based cross trusted domain direct anonymous attestation scheme. *Computer Networks*. 2015;81:245-257. https://doi.org/10.1016/j.comnet.2015.02.023.
102. Proudler G, Chen L, Dalton C. *Trusted Computing Platforms: TPM2.0 in Context*. Bristol, UK: Springer; 2014.
103. Simpson AK, Schear N, Moyer T. Runtime integrity measurement and enforcement with automated whitelist generation. *IEEE Trans*. 2013;8(7):1230-1242.
104. *ISO/IEC 11889-1:2015-Information Technology—Trusted Platform Module Library*. https://www.iso.org/standard/66510.html. Accessed July 16, 2019.
105. *Infineon Enables Open Source Software Stack for TPM 2.0—for Easier Integration of Security into Industrial and Automotive Applications*. https://www.infineon.com/cms/en/about-infineon/press/market-news/2018/INFCCS201808-075.html. Accessed July 17, 2019.

106. Bertholon B, Varrette S, Bouvry P. Certicloud: a novel TPM-based approach to ensure cloud IaaS security. In: 2011 IEEE 4th International Conference on Cloud Computing; 2011; IEEE; 121-130.

107. Perez R, Sailer R, Doorn VL, et al. vTPM: virtualizing the trusted platform module. In: Proceedings of the 15th Conference on USENIX Security Symposium; 2006; 305-320.

How to cite this article: Hosseinzadeh S, Sequeiros B, Inácio PRM, Leppänen V. Recent trends in applying TPM to cloud computing. *Security and Privacy*. 2020;3:e93. https://doi.org/10.1002/spy2.93