Applications of Trusted Computing in Cloud Context

Mohammad Reza Memarian, Diogo Fernandes, Pedro Inácio, Ville Leppänen and Mauro Conti

Abstract Trusted computing is a technology that enables computer systems to behave in a given expected way. Achieving that goal happens by arming an isolated piece of hardware with embedded processing, cryptographic capabilities such as encryption key that is kept safe from software layer attacks. The mentioned module is accessible to the rest of the computer system via a well-defined and tested application programming interface. Trusted computing protects the system against external attackers and even against the owner of the system. Cloud computing enables users to have access to vast amounts of computational resources remotely, in a seamless and ubiquitous manner. However, in some cloud deployment models, such as public cloud computing, the users have very little control over how their own data is remotely handled and are not able to assure that their data is securely processed and stored. Cloud administrators and other parties can be considered threats in such cases. Given the ground that cloud has been gaining and the rate at which data is generated, transmitted, processed, and stored remotely, it is vital to protect it using means that address the ubiquitous nature of the cloud, including trusted computing. This chapter investigates applications of trusted computing in cloud computing areas where security threats exist, namely in live virtual machine migration.

M. Reza Memarian · V. Leppänen (✉)
Department of Information Technology, University of Turku, Turku, Finland
e-mail: Ville.Leppanen@utu.fi

M. Reza Memarian
e-mail: mohammad-reza.memarian@utu.fi

D. Fernandes
PepsiCo, Michrów, Poland

P. Inácio
Computer Science, University of Beira Interior, Covilhã, Portugal

M. Conti
Department of Mathematics, University of Padua, Padua, Italy
e-mail: conti@math.unipd.it

S. Chaudhary et al. (eds.), *Research Advances in Cloud Computing*,
DOI 10.1007/978-981-10-5026-8_18

1 Introduction

In computing, the term *trust* refers to establishing a high degree of confidence in the behavior of a system, provided that particular inputs are expected to render certain outputs. Trust is knowledge of the user on the precise functioning of the system. Due to the diversity of computing systems, this matter can not be addressed in a straightforward manner. The state in which a single computing system can be is determined by running a set of configurations with varying dimensionality that can be reshuffled and recombined in a multitude of ways. That set changes as the system is used throughout time due to installation, upgrade or removal of software and replacement of hardware. For example, the Linux kernel subsystem implements an Integrity Measurement Architecture (IMA) that can be explored for integrity attestation purposes [7]. Hence, identifying the entire state set of a system can be an unfeasible task. Smaller subsets of well-known configurations are more manageable, but that does not satisfy the diversity of the computing systems. Frequently, trust assurance is achieved using cryptographic proofs that testify reliability of a system, regardless of the adjacent conditions and inputs at the cost of some overhead. Other approaches consist of formally proving that software works according to requirements.

Despite concerns over the security of cloud environments [16], cloud computing has been developing and maturing. This technology enables the envisioned computing as a utility, essentially by eliminating the hassle of establishing on-site Information Technology (IT) infrastructures. It is capable of allocating, on-demand and off-site, fine-grained resources with minimal cost, by leveraging economy of scale. However, outsourcing private data and storage to providers with multi-tenant environments raises security concerns. Trusted computing is, therefore, an essential component to cloud environments that can alleviate some of those security concerns. Nevertheless, the setup of cloud infrastructures under the service delivery models translates into an interplay of different hardware, virtualization and software technologies at multiple layers. That interplay, in turn, creates computing diversity that poses as a difficulty in achieving trusted remote computing in a holistic manner.

In the light of the benefits of trusted computing to cloud environment, it is important to study current applications of one to the other, taking into consideration the challenges and requirements of cloud computing. This chapter makes that discussion by analyzing the security requirements in terms of trust to cloud services and by studying the applicability of trusted solutions to such requirements. Therefore, the contributions of this chapter are twofold. First, cloud computing is described with a focus on its trust requirements. Second, current applications of trusted computing are enumerated and weighted according to different criteria within the cloud security requirements.

Next, Sect. 2 gives an introductory overview of the cloud computing deployment models and subsequently focuses on cloud services and security requirements. Section 3 describes trusted computing and enumerates applications of that technology to cloud computing environments. Section 4 summarizes the discussion and points out open issues. Finally, Sect. 5 concludes the chapter.

2 Cloud Computing

Computing in the *cloud* emerged several years ago as a means to describe computing as a *utility* off-site. This computing model not only offloads some storage and computing responsibilities to a cloud provider, but also the burden of managing IT infrastructures and security duties. For providers, services wrap well-defined resources from elastic pools that are measured and allocated as needed to users. In turn, consumers of the services are charged per subscription, which can significantly decrease costs for all kinds of small to large businesses.

The National Institute of Standards and Technology (NIST) adds the notions of ubiquitous access, monitored, on-demand, and shift provision of resources with minimal management burden to the definition of cloud computing [28]. This computing paradigm consists of three main service delivery models, Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS), that can be set up in four deployment modes: private, public, community, and hybrid.

2.1 Cloud Services

Initially, cloud services were largely discussed according to three delivery models (as described in other chapters of this book) that illustrate the different layers of the cloud stack: IaaS, PaaS, and SaaS. Concerning trust and applications in the cloud context, the layers SaaS and IaaS are the most interesting, as those primarily provide the end-user applications and raw computing and storage resources, respectively. The description of services, however, is now often particularized under Anything-as-a-Service (XaaS), conveying the meaning that cloud services can deliver anything in the form of services. In fact, new service definitions have been made in an arbitrary way throughout time, resulting in a lack of a unified XaaS scheme [13], a view that was foreseen several years ago by Armbrust et al. [3].

The virtualization layer underpinning IaaS infrastructures brings many benefits, despite the implied overhead. A Virtual Machine Monitor (VMM) can handle several Virtual Machines (VMs), each one possibly encapsulating a different Operating System (OS) (a guest) with distinct settings. Access to the hardware is regulated by the manager according to a scheduling algorithm. This setup has noticeable advantages in terms of security by design, such as controlled isolation of the environment, regardless of vulnerabilities, and close monitoring of resource usage and communications. With the dependency between guest and native OSs removed, VMs can be rolled back to previously saved states (i.e., snapshots) or be moved around, in a process termed *migration*. This VM independence implies that, within IaaS infrastructures, data can be in one of three postures at a given time: at *transport* (data-in-motion), at *rest* (data-in-rest), and at *runtime* (data-in-processing). Migration of data between VMs or live VM migration is central to achieve energy-efficient consolidated workloads in clouds by minimizing the number of servers that are underutilized or idle [11].

2.2 Security Issues

Assuring security in all aspects of systems and end-to-end communications constitutes often a burden, because it is not granted by design. Unfortunately, security issues are likely prone to appear as a consequence. This principle holds true for cloud environments, as they are built on top of current networks and web technologies. Apart from mentioned technologies, virtualization is a vital component of cloud computing structure. OS-level virtualization provides the building blocks for running multiple OSs while sharing hardware resources, and effectively enhances isolation by means of sandboxing. Nonetheless, the virtualization technology may not be completely free of vulnerabilities, allowing adversaries to escape the controlled environment, a process known as VM *escape*. A prime real-world example of VM escape is the VENOM [9] vulnerability, identified by CVE-2015-3456. This vulnerability allows an attacker to run any code in the hypervisor process context by exploiting a buffer overflow in the Floppy Disk Controller (FDC) of the Quick Emulator (QEMU) hypervisor used by Xen and Kernel-based Virtual Machine (KVM) platforms.

Migration of VMs can always expose data as it is in motion. In the process of migration, VMMs copy memory pages of the VM to be migrated from source host to destination host seamlessly, while guest VMs are still running. This opens opportunities for attackers (especially malicious insiders) to access raw memory data of migrating VM. A myriad of information lies in the memory as everything in the OS traverses through the Random Access Memory (RAM), including passwords and cryptographic keys. In 2011, researchers employed simple forensics techniques to recover sensitive information from Xen VM snapshots, which contain copies of memory from a certain point in time in [32, 33].

Needless to say that snapshots at rest comprise tempting targets too, in case the storage media is accessible. Beyond the sensitive data they may hold, a compromised snapshot or image can be used to spread malware within the cloud environment if used as a golden image to boot up VMs.

To better arrange the discussion below, the threats and security requirements to cloud computing are discussed against the three postures the data can have in the cloud. We assume that the attacker is either an individual inside the infrastructure or has equivalent access. The threats with regard to data postures are as follows:

- When *at transport*, data potentially moves from a given (physical) system to another. As such, vulnerabilities related with networking technologies also play an additional role in such a scenario. The potential insecure communication channel is part of the attack model. If the data is moving between data centers, in an ecosystem known as *intercloud*, threats such as data leakage or modification are more prominent. Examples include the modification of VM's image during migration, namely to inject malicious software during the procedure. In this case, the data may be accessed or modified along the path from the source to the destination, which may render the intrusion or leakage more difficult to detect or account for.
- Within the context of cloud, data *at rest* may concern files and database instances of SaaS, PaaS applications or IaaS VM images. Clouds are also commonly used

to store backups of entire remote systems on demand or on a regular basis. If the data is stored in plaintext, or with insecure schemes (e.g., data may be encrypted with keys generated in the cloud itself), then it is susceptible to eavesdropping, data modification or leakage (the attacker may copy the data elsewhere). Eavesdropping may lead to compromise of private or confidential data, namely of industrial secrets, causing monetary harms. Modification of VM images may be performed with the goal of injecting malware, while eavesdropping has the intention of accessing confidential data. Modification of data at rest from PaaS or SaaS applications may be performed to induce a different behavior on the development environment or applications.

- The cloud is an ecosystem for very diverse runtime environments, and gives rise to very specific scenarios. In this case, data may leak from one execution environment to another or be injected between allegedly isolated sandboxes. Specific threats include cross-VM and container attacks [44], and malicious software installed at the hypervisor layer reading the contents of the memory from a running VM. Multi-tenancy is a core technology for the cloud, but brought its new set of security issues, especially in public clouds [27]. In such multi-tenancy scenario, two customers may be sharing the same technology, libraries, global variables and storage, which need to be adequately provisioned. At the *runtime* posture, threats are mainly coming from co-resident systems or applications [37].

2.3 Security Requirements

Two of the main security requirements of users in any secure environment are data confidentiality and integrity. It is vital for users to make sure that those properties of their data are preserved and guaranteed at any stage of operation. In addition to the mentioned requirements, trust is another factor which is more desired in the cloud than in other environments. There should be mechanisms in place to assure users that the trusted party transports and processes their data securely. In this section, we discuss security requirements for user data in the cloud with respect to cloud services mentioned in the previous section.

In most cases, confidentiality is the most important security requirement for user's data in cloud computing. It applies to any of the previously identified postures that data can be in. Privacy is also becoming more important in an age where ubiquity is increasing. Moreover, integrity of the user data is the other vital security requirement that shall be preserved in all the mentioned states, along with data authentication during transmission. At the transport state, adequate controls shall be implemented in place in order to provide a secure transport channel. Encryption mechanisms and Message Authentication Codes (MACs) are typically employed to provide a confidential and authenticated channel between cloud instances. Nonetheless, the aforementioned security mechanisms have an impact on the performance of the systems, which may hinder full deployment on every communication. Furthermore,

usage of controls such as Network Intrusion Detection System (NIDS) can help to detect network level attacks and suspicious activities.

In addition to the transport and at rest states, data can be at the processing state. While users offload their heavy computational activities on cloud resources, they need trustworthy computational activity by the service provider. Parties with adequate privileges or via exploitation of vulnerabilities can access, modify or delete other users data. As such, confidentiality and integrity of data should be preserved at storage and computation time too [42]. However, the typical ciphers and integrity mechanisms cannot be used to protect data in the processing state. Data would have to be loaded in plaintext to the memory [34] to be processed. All the mentioned concerns arise from the fact that users do not have physical access to their data and applications. Recent research lines on homomorphic encryption schemes are motivated by the mentioned scenarios.

As pointed out earlier, main security properties that shall be preserved in any secure computing environment are confidentiality, integrity, and availability which are referred to as the CIA triangle. Trusted Computing (TC) can contribute to preservation of confidentiality and integrity of the data while availability is not directly achieved by implementation of TC. Starting from the described requirements and postures, it is now possible to investigate how TC can be used to provide assurance of the properties to the user.

3 Contribution of Trusted Computing to Cloud Security

The trust issue is best put into perspective when considering the evolution of IT infrastructures throughout time. Amoroso [2] accurately described, a few years ago in the context of modern enterprise infrastructures, a decisive point in time where the transition of IT infrastructures to the cloud was accelerating. The early IT model of the 1990s considered assets to be on-site, enclosed by a well-defined and controlled perimeter. Evolving business and communication needs, however, required to open network ports into the environment. Such is the case with Virtual Private Networks (VPNs), websites and email, all still in use today, and Internet access. Email, for instance, has been and still is one of the most concerning open channels into the network as it is heavily explored by attackers to deploy malware. Eventually, this drop in trust leads to a multitude of network and host-based monitoring and detection technology.

Fast forward to the current day, with cloud computing booming, the scope of the trust issue enlarges and worsens, leapfrogging from on-site IT infrastructures into off-site cloud environments. Trusted computing technology, however, helps alleviating the problem. The first part of this section describes TC, while the remaining parts point out deployments of that technology addressing the specific security issues of cloud environments discussed before.

3.1 Definition of Trusted Computing

TC refers to a set of software-based and hardware-based definitions and technologies that enable computer systems to behave in a desired and expected way. In the TC design, systems are less dependent on their owners while, even to some extent, are protected against them. TC requires a set of public and private key pairs to be generated and fixed on the hardware at the manufacturing time. The key pair is referred to as Endorsement Key (EK). Using the hard coded EK, platforms can authenticate each other and applications running on a platform can assure other applications on other platforms about their origin platform. TC also enables running of a particular desired software only and various desired restriction can be imposed on runtime behavior of applications.

TC is specifically applicable to distributed applications in environments such as cloud computing. In such environments, applications can make sure that the other applications or platforms are the correct ones. One of the key concepts in TC design is remote attestation. Remote attestation enables authorized parties to detect unwanted changes to the computing system. It is applicable in various areas of computing such as detection of unwanted change in the licensed software and verifying the platform that an application is executing on it.

Trusted Platform Module (TPM) is a standard for a piece of hardware (microprocessor) that acts as an enabler of TC. Using TPM, a user can ensure that the application is running on the specific hardware and OS. This secure cryptographic module delivers a hardware-based method to handle authentication of user, data protection, and network access, and brings out the matter of security from the software layer only. Hardware-based TPMs are bound to a single standalone device by design. The origin of trust is therefore limited in scope, which turns out to be unsuitable for applications where sharing is desired or for cross-device scenarios. An extension of the version 2.0 of the TPM specification is presented in [8] in order to address multi-device scenarios. The extension for TPM v2.0 actually trusts and relies on the cloud to share an additional key, though it does not address any particular cloud security issue.

3.2 Trusted Cloud Computational Security

One of the critical postures that user data can be at is the processing state (while using the cloud services). At that state, data require substantial protection in order to ban privileged insiders to interfere with the user computational processes. The processing state refers to the execution of internal processes for computing over user data. It encompasses various types of calculation, simulation, data processing, and program execution. Hence, clients shall have methods in hand to verify integrity and confidentiality of their data at computation time on the cloud. That concern enforces limitations of using cloud for security-critical computations such as confidential simulations.

While the data of a user can be at processing state at any service level, the case of IaaS is the most relevant for this study. At the IaaS level, users have more control over the underlying infrastructure of the service, when compared to other service layers. That enables users to have a more deterministic role in determining the security level of their service at IaaS level while in other service levels, the providers get that role. On the other hand, applying trusted computing techniques to the PaaS and SaaS levels but not to the IaaS level would be unnatural, since trust building is transitive and one should start it from the lowest levels. Having trusted computational security for the IaaS level can be seen to implicitly provide it also for the PaaS and SaaS levels. Hence, it is no surprise that applications of TC are mostly proposed for the IaaS level. However, some papers propose additional trusted computing solutions for the PaaS and SaaS layers.

In the following, we survey some works having focus in computational issues of trusted cloud. Many of these works are actually wide in scope—describing overall trusted cloud solution with computational capabilities. The papers are summarized in Table 1.

At the IaaS level, services are provided in the form of VM. Those VMs are started based on some images. The user can either purchase the image from the image repository of the service provider or the user can upload an arbitrary image to be used for the user's VM. To verify the integrity of the started system, the user shall make sure that the started VM booted an expected image. Incorporation of TC into cloud computing platforms is an effort in that regard. Wallom et al. [41] proposed myTrustedCloud which incorporated TC into the Eucalyptus cloud platform. Trusted computing enables users of the cloud to be assured about the integrity of the VM itself and the underlying VMM. Each VM executes the desired applications on top of a commodity OS. That condition simulates a form of the open-box system. On the other hand, users can modify the settings of an OS in order to satisfy the security requirements of their applications and diminish the unrequited services from a large OS. That simulates a form of closed-box system. The closed-box setting creates an execution environment that disables malicious insiders from accidentally or

Table 1 Overview of trusted cloud papers having focus in computational issues

Paper	Layer	Overview
[41]	IaaS	An outline of trusted cloud for security-critical computation.
[17]	IaaS	Early (2003) constructive work on Terra system for trusted general-purpose computing
[22]	IaaS	Introduces open source cloud computing framework Eucalyptus.
[40]	PaaS	Trusted computing based solution for Java environment. The solution is applicable to cloud context
[6]	PaaS/SaaS	Efficient and Secure Educational Platform (ESEP) for cloud computing based on TPMs
[30]	SaaS	Provides trusted SLA (service level agreement) monitoring services as part of a cloud based billing system

intentionally accessing and tampering the user data at processing time [34]. The requirement for that is to have a VMM that supports trusted computing.

Garfinkel et al. [17] proposed Terra, a Trusted Virtual Machine Monitor (TVMM) architecture that is able to simultaneously run VMs in both open and closed-box settings. That allows each application to run on a specifically modified version of an OS. Furthermore, the architecture of Terra allows the TVMM to apply TC requirements such as remote attestation of the applications for each VM. Hence, it is effective for implementation of distributed applications in cloud environments. Using trusted computing, the user can verify the integrity of the VM itself, the Node Controller (NC) and the Elastic Block Store (EBS). In order to verify integrity of the VM, the integrity of all three mentioned elements should be verified, which is called iterative attestation. That verifies the operation of the trustworthy VM on a trusted platform [22].

As opposed to Terra, which is suitable for operation on a single platform, Trusted Cloud Computing Platform (TCCP) [34] operates on multiple platforms (data center wide) enabling VMs to move around and use the live migration feature. As such, the attestation encompasses the entire service ensuring the customer about the security of each platform that computation is taking place on. Important components of TCCP are TVMM and a third-party trusted coordinator. Nodes shall go through a secure boot process in order to install the TVMM. That trusted coordinator keeps a list of trusted nodes that the user can have for user's own VM to securely operate on. To be trusted a node shall run a TVMM and be in the secure perimeter. One of the important points here is that the VM's launch time is a critical moment requiring protection and other operations such as suspend and resume [34].

While attestation is a useful mechanism for remote verification of trust, it has two shortcomings. By attestation, some private information of the service provider such as details about the platform and the internal structure of internal systems can be uncovered. Potential malicious users can benefit from that information to form attacks. Secondly, if third parties handle the attestation [22], they become the single point of failure [41].

Even though cryptography can contribute to preserve confidentiality and integrity of data at transport and storage states, it is currently ineffective during computation time [34], as data shall be loaded in plaintext to memory. Fully Homomorphic Cryptography (FHC) allows a set of limited operations on the encrypted data, but the performance of FHC is not at a level to be operational in practice. This problem is more severe in the cloud because it has a multi-tenant environment and the infrastructure is not under control of the data owner. Cloud employees either accidental or maliciously might tamper or access data, causing violation of confidentiality and integrity. At situations where user data is unprotected in the memory for processing, anyone with privileged access level can have access to the data. A preliminary countermeasure is to limit the physical access to the hardware and servers. However, limiting the physical access only thwarts a small portion of the attacks as various other attacks take place with remote access, and existing solutions are not fully effective in mitigating attacks in that field [32].

One can also find PaaS level solutions of trusted computing. One such is trusted computing implementation for platform-independent Java environment by Toegl et al. [40]. To be precise, the paper only sees cloud computing as one possible context for their technical solution, and thus this work is only indirectly cloud related.

SaaS level solutions do also exist. In such cases, the SaaS solution has some specific data and functionality that is secured with trusted computing techniques. Brohi et al. [6] describe a secure cloud infrastructure for an Efficient and Secure Educational Platform (ESEP)—it can be seen either as a SaaS or a PaaS level solution. The actual solution also contains elements from the IaaS level. A different kind SaaS level trusted service is provided from the THEMIS system by Park et al. [30]. The THEMIS system is a billing system implemented for a cloud computing environment, but the system provides monitoring of service level agreement (SLA) properties by implementing that functionality based on TPM modules. In fact, there are several other papers that provide similar SLA related functionality based on trusted computing techniques in cloud computing contexts.

3.3 Trusted Cloud Transport Security

The attestation process can be the target of network layer attacks. Two of the related attacks in that layer are reply attacks and Man-in-the-Middle (MitM). In order to prevent reply attacks, a cryptographic nonce, which is generated by the user shall be used for the attestation session. In order to tackle the MitM, the NC shall make sure that the VM requesting attestation is running and is connected to that NC itself [41].

At the VM transport time, user data can be the target of leakage and tampering attempts [34]. In live migration, the states of a VM are transferred between two nodes, which both need to be trusted.

We have looked at papers focusing on transport security in trusted cloud context. In the following, we survey some recent such papers and summarize the results as Table 2. Almost all such papers deal with VM migration at IaaS level—such constructions are also surveyed recently in [1, 25]. This is quite natural, as considering the SaaS level, the mechanisms to securely transmit SaaS application data from one (cloud) system to another are already well understood and solved even outside the cloud context. On the other hand, sharing SaaS level data is an elementary part of

Table 2 Overview of trusted cloud papers having focus in migration issues

Paper	Layer	Overview
[10]	IaaS	Virtual TPM-based solution for VM migration in private clouds
[4]	IaaS	VM migration solution focusing on developing trust token-based protocol
[15]	IaaS	Further developed VM-vTPM solution where the focus is in TLS channel
[19]	PaaS/IaaS	Virtual TPM-based solution enabling container migration
[38]	IaaS	An OpenStack and TPM-based solution for VM migration

the whole idea of cloud computing. Migrating applications from a cloud system to another neither seems to be a popular topic in the literature. The reason perhaps is that a cloud application corresponds to a service and instead of moving services from one place to another, one can replicate the same service in several places (and then moving corresponds to setting a service up in one place and closing it down in another place—not necessarily moving any data related to the service). There is however one seemingly growing exception to this PaaS level activity—the container technology is gaining more popularity and one can think of moving a container (typically made just for one application) as a PaaS level migration activity. A virtual TPM-based such framework is described in [19].

In Danev et al. [10], three security requirements are enumerated for secure migration of VMs based on Trusted Platform Modules (vTPMs), namely VM-vTPM confidentiality and integrity, initiation authenticity (of the migration requester), and preservation of the trust chain. The last one is of particular importance when considering the different elements of the cloud stack and trust transitivity, as well as the strong association between hardware TPMs and vTPMs. To cope with these requirements, Danev et al. [10] described a protocol where migration of VM-vTPM pairs is made possible between attested nodes by introducing an additional key layer between TPMs and vTPMs, at the cost of some overhead. Moreover, Aslam et al. [4] add as a requirement that the destination of a migration should be trustworthy too. To cope with that, and other cloud requirements like scheduling, transparency, and scalability, a token-based trust scheme is described to attest that the same software state trusted by the user is found on platforms where the VM are migrated to. This scheme relies on a TPM-based communication protocol between the source and destination systems, as well as on trust tokens pre-generated by the cloud provider in a segregated network.

Another constructive solution for VM migration is given by Fan et al. [15]—their work especially focuses on development of TLS-based migration protocol. VM migration is studied in several contexts. Syed et al. [38] study the issue in OpenStack context applying TPM, libvirt, and QEMU.

3.4 Trusted Cloud Storage Security

Cloud storage is used for file, system and image backups. Guaranteeing security against confidentiality and integrity breaking attempts means usually to encrypt and authenticate the data. Depending on the usage and type of data, TPM may be used as a means to derive encryption keys, perform encryption and decryption of data, and testify the integrity of the data during retrieval.

In the case of remote storage of files and system backups (e.g., Dropbox), data should already be in an encrypted format when it reaches the cloud, though this does not always happen nowadays. If special functions over the data, such as search, are required, TPM may be used to perform them in a safe environment, returning sanitized values. In the case of image storage, TPM is particularly useful for attestation purposes.

Table 3 Overview of trusted cloud papers having focus in storage issues

Paper	Layer	Overview
[35]	IaaS	Technical solution for server and client side focusing on handling and sharing of encryption keys
[36]	IaaS	A general encryption and trusted computing based solution for cloud data
[20]	SaaS	Specific solution for trustworthy flow cytometry data analyses
[5]	SaaS	Provenance-based trusted solution for access control and provenance information provision for the users
[39]	PaaS	Provenance solution for forensics needs based on trusted computing
[43]	IaaS	An OpenStack-based cloud solution for forensics-enabled investigations
[21]	IaaS	A trust-based solution in hybrid cloud setting for geographically fenced data
[31]	IaaS	TGVisor: A storage solution for controlling geolocation of data with trusted computing and supporting especially mobile clients
[26]	IaaS	SecLoc: A solution for supporting location sensitivity of cloud data storage with trusted computing

In the following, we review a small set of rather recent works that focus on providing storage security in the trusted cloud context. Often the papers also deal with other issues besides the storage security. The papers are summarized in Table 3.

Shin et al. [35] consider the access control mechanism provided for typical cloud storage to require improvement. They propose a technical solution called DFCloud for an improved TPM-based solution of managing encryption keys and overall key sharing between dynamically defined legal users. Special focus is given for mobile devices as means to access such cloud storage. On the client side, DFCloud is based on using ARM's TrustZone technology. From the cloud point of view, the DFCloud works at IaaS level.

There are several general solutions proposed for securing cloud data using trusted computing technologies. Singh et al. [36] describe a TPM-based solution, NUYA, using Kerberos for generally securing data in the cloud context. As opposed to generic solutions, there exist also some rather specific application related data that are secured with trusted computing based techniques in the cloud context. Javanmard et al. [20] give such a solution for the medical field, specifically for flow cytometry analyses to support disease diagnosis activities. As specific solutions are more like applications, the TSC (Trustworthy and Scalable Cytometry) solution of [20] can be seen to be made for the SaaS layer.

Concerning cloud storage, there is occasionally a clear need to be able to track the usage and origins of data. *Provenance* on data is information of actions that are taken on it since the creation of data (including creation). Many cloud systems support data provenance as a feature, but technical solutions for guaranteeing trusted provenance-based access and information are also presented in the literature. A survey of provenance solutions is given in [24]. Bates et al. [5] present a trusted computing based provenance solution for access control but also provide the provenance data as a SaaS-like service for the user. Progger (Provenance Logger) is another

technical solution by Ko and Will [23] for provenance information but unlike [5] it is not really based on trusted computing but on a kernel-space solution. In many works, provenance-based solutions are developed towards auditing and forensics needs. One such paper is by Taha et al. [39], where that kind of trusted computing based solution is given. The solution is made for a set of applications and thus it can be considered as a PaaS/SaaS level solution. Another OpenStack-based solution is given by Zawoad and Hasan [43]. They describe a construction named FECloud to support forensics-enabled investigations concerning data provenance. Their solution is indirectly based on trusted computing.

One rather recent challenge for cloud computing systems has been the (often law-based) requirement to enforce governmental data privacy regulations and to ensure that data (and computations on the data) do not cross some specific geographic boundaries. There are several specific trusted computing based technical solutions provided for securing location sensitivity of the data in a cloud system. In general, the idea of such trustworthy geographically fenced hybrid clouds (TGHC) is described by Jayaram et al. [21]. TGVisor, by Park et al. [31], represents a more detailed technical IaaS solution for more or less the same problem but also supporting mobile clients. Another related solution is SecLoc by Li et al. [26]. SecLoc is specifically made for needs raising from Canadian law—to provide a location-sensitive cloud storage for example, storing health records.

4 Discussion and Open Challenges

Despite the research advancements in this field, one of the fundamental issues of trust remains open. That issue is the one revolving around the perception of trust, specifically what different individuals and groups make of it both in concept and in relation to technology. This is especially relevant to cloud environments, such as the project described in [14], which aimed at identifying issues in a pilot High-Performance Computing Cluster (HPCC) in the cloud for several stakeholders of the petrochemical industry. Their main finding is the one described as a clash between organizational behavior, a *political cloud* so to speak. Moreover, in [29], trust relates to reputation and not as in mathematical attestations using a hardware module, further highlighting the point of awareness. How TPMs and vTPMs come to address this multi-tenant scenario where users have distinct understanding of the underlying concepts is still unknown. Nevertheless, it is foreseeable that the technological solutions based on encryption will continue to be developed, not only to cope with the security, privacy, and trust needs, but also to provide a seamless cloud experience.

Another important challenge in trusted cloud computing is trust transitivity and zoning. This is well illustrated when considering the complex interaction of trust from the bare metal to the hypervisor and to the interface, in view of the IaaS hybrid interplay of multiple software instances and devices, whether virtual or physical. Here, zoning refers to the secure isolation of trust zones for and between tenants. This calls for trust assessment models such as the one described in [18], which

considers different scenarios with and without TPM availability for the processor and Basic Input/Output System (BIOS) or hypervisor signing. Furthermore, trust is an issue of source and destination, such as the works done upon live VM migration. The transitivity and zoning also encompass all that is in between, so a network building trust path [12] is needed too for intra-cloud and intercloud migrations.

5 Conclusions

Cloud computing and trusted computing are increasingly the focus on several studies to address the security issues posed by the former. Virtualization is advantageous from the computing and cost-efficiency points of view, allowing to create multi-tenant infrastructures running co-resident operating systems. Pre-packaged software development environments in the cloud are also useful centralized repositories to save time when setting up dependencies, libraries, and tools, which allow devising cloud applications. Nevertheless, a lack of trust in computing, storage, and transport is evident when considering the offload of IT responsibilities to third-party cloud providers.

A number of security requirements from the trust standpoint were discussed in this chapter. These security requirements highlight that cloud environments need improvement at several levels so that the trust chain of the cloud stack holds throughout the several heterogeneous cloud systems, such as live VM migration from one cloud platform to another. Multiple works describe ways to enhance trust attestation in certain points, but may be limited in scope and do so not without introducing additional complexity and cryptographic and communication overhead or a third-party entity. That establishes that realizing fully trusted cloud environments to users is not yet within grasp. Achieving this ideal setup would require to mimic the same levels of trust as users have with their own on-site systems.

References

1. Ahmad, R. W., Gani, A., Hamid, S. H. A., Shiraz, M., Xia, F., & Madani, S. A. (2015). Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues. *The Journal of Supercomputing*, *71*(7), 2473–2515.
2. Amoroso, E. G. (2013). From the enterprise perimeter to a mobility-enabled secure cloud. *IEEE Secur Privacy*, *11*(1), 23–31.
3. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., et al. (2010). A view of cloud computing. *Commun ACM*, *53*(4), 50–58.
4. Aslam, M., Gehrmann, C., & Björkman, M. (2012). Security and Trust Preserving VM Migrations in Public Clouds. *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (pp. 869–876).
5. Bates, A., Mood, B., Valafar, M., & Butler, K. (2013). Towards secure provenance-based access control in cloud environments. In *Proceedings of the Third ACM Conference on Data and Application Security and Privacy* (pp. 277–284). ACM.

6. Brohi, S. N., Bamiah, M. A., Chuprat, S., Ab Manan, J. L. (2012). Towards an efficient and secure educational platform on cloud infrastructure. In *2012 International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM)* (pp. 145–150). IEEE.
7. Cesena, E., Ramunno, G., Sassu, R., Vernizzi, D., & Lioy, A. (2011). On Scalability of remote attestation. In *Proceedings of the 6th ACM Workshop on Scalable Trusted Computing (STC)* (pp. 25–30). New York, NY, USA: ACM
8. Chen, C., Raj, H., Saroiu, S., & Wolman, A. (2014). cTPM: A cloud tpm for cross-device trusted applications. In: *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation (NSDI), USENIX Association, Berkeley, CA, USA* (pp. 187–201).
9. CrowdStrike. (2015). VENOM Vulnerability. Retrieved May 2016, from http://venom.crowdstrike.com/.
10. Danev, B., Masti, R. J., Karame, G. O., & Capkun, S. (2011). Enabling secure VM-vTPM migration in private clouds. In *Proceedings of the 27th Annual Computer Security Applications Conference (ASAC)* (pp. 187–196). New York, NY, USA: ACM
11. Dargie, W. (2014). Estimation of the cost of VM migration. In *23rd International Conference on Computer Communication and Networks (ICCCN)* pp. 1–8.
12. Divakarla, U., & Chandrasekaran, K. (2016). Trusted path between two entities in Cloud. In *6th International Conference on Cloud System and Big Data Engineering (Confluence)* pp. 157–162.
13. Duan, Y., Fu, G., Zhou, N., Sun, X., Narendra, N. C, & Hu, B. (2015). Everything as a service (XaaS) on the cloud: origins, current and future trends. In *IEEE 8th International Conference on Cloud Computing* pp. 621–628.
14. Eldred, M., Adams, C., & Good, A. (2014) Trust challenges in a high performance cloud computing project. In *IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 1045–1050).
15. Fan, P., Zhao, B., Shi, Y., Chen, Z., & Ni, M. (2015). An improved vTPM-VM live migration protocol. *Wuhan University Journal of Natural Sciences*, *20*(6), 512–520.
16. Fernandes, D. A. B., Soares, L. F. B., Gomes, J. V., Freire, M. M., & Inácio, P. R. M. (2014). Security Issues in Cloud Environments—A Survey. *International Journal of Information Security (IJIS): Special Issue Named Security in Cloud Computing*, *13*(2), 113–170.
17. Garfinkel, T., Pfaff, B., Chow, J., Rosenblum, M., & Boneh, D. (2003). Terra: A virtual machine-based platform for trusted computing. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, SOSP '03* (pp 193–206). ACM.
18. Gonzales, D., Kaplan, J., Saltzman, E., Winkelman, Z., & Woods, D. (2015). Cloud-trust—A security assessment model for infrastructure as a service (IaaS) clouds. *IEEE Transactions on Cloud Computing PP*(99), 1–14.
19. Hosseinzadeh, S., Laurén, S., & Leppänen, V. (2016). Security in container-based virtualization through vTPM. In *Proceedings of the 9th International Conference on Utility and Cloud Computing* pp. 214–219. ACM.
20. Javanmard, M., Salehi, M. A, & Zonouz, S. (2015). 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) (pp. 1356–1360). IEEE.
21. Jayaram, K., Safford, D., Sharma, U., Naik, V., Pendarakis, D., & Tao, S. (2014). Trustworthy geographically fenced hybrid clouds. In *Proceedings of the 15th International Middleware Conference* (pp. 37–48). ACM.
22. Khan, I., Rehman, H., & Anwar, Z. (2011). Design and deployment of a trusted eucalyptus cloud. In *2011 IEEE International Conference on Cloud Computing (CLOUD)* (pp. 380–387). IEEE.
23. Ko, R. K., & Will, M. A. (2014). Progger: An efficient, Tamper-evident Kernel-space logger for cloud data provenance tracking. In *2014 IEEE 7th International Conference on Cloud Computing (CLOUD)* (pp. 881–889). IEEE.

24. Lee, B., Awad, A., & Awad, M. (2015). Towards secure provenance in the cloud: A survey. In *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)* (pp. 577–582). IEEE.
25. Leelipushpam, P. G. J, & Sharmila, J. (2013). Live VM migration techniques in cloud environment a survey. In 2013 IEEE Conference on Information & Communication Technologies (ICT), (pp. 408–413). IEEE.
26. Li, J., Squicciarini, A., Lin, D., Liang, S., & Jia, C. (2015). SecLoc: Securing location-sensitive storage in the cloud. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies* (pp. 51–61). ACM.
27. Memarian, M. R., Conti, M., & Leppänen, V. (2015). EyeCloud: A Botcloud Detection System. In *2015 IEEE Trustcom/BigDataSE/ISPA* (Vol. 1, pp. 1067–1072).
28. NIST. (2011). The NIST definition of cloud computing. Retrieved June 2016, from http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf.
29. Noor, T. H., Sheng, Q. Z., Yao, L., Dustdar, S., & Ngu, A. H. H. (2016). CloudArmor: Supporting reputation-based trust management for cloud services. *IEEE Transactions on Parallel and Distributed Systems*, *27*(2), 367–380.
30. Park, K. W., Han, J., Chung, J., & Park, K. H. (2013). THEMIS: A Mutually verifiable billing system for the cloud computing environment. *IEEE Transactions on Services Computing*, *6*(3), 300–313.
31. Park, S., Yoon, J. N., Kang, C., Kim, K. H., & Han, T. (2015). 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)* (pp. 99–108). IEEE.
32. Rocha, F., & Correia, M. (2011). Lucy in the sky without diamonds: Stealing confidential data in the cloud. In *IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)* (pp. 129–134).
33. Rocha, F., Abreu, S., & Correia, M. (2011). The Final Frontier: Confidentiality and Privacy in the Cloud. *Computer*, *44*(9), 44–50.
34. Santos, N., Gummadi, K. P., & Rodrigues, R. (2009). Towards trusted cloud computing. In *Proceedings of the 2009 Conference on Hot Topics in Cloud Computing, USENIX Association, Berkeley, CA, USA, HotCloud'09*.
35. Shin, J., Kim, Y., Park, W., & Park, C. (2012). DFCloud: A TPM-based secure data access control method of cloud storage in mobile devices. In *2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 551–556). IEEE.
36. Singh, N. K., Patel, Y. S., Das, U., & Chatterjee, A. (2014). NUYA: An encrypted mechanism for securing cloud data from data mining attacks. In *2014 International Conference on Data Mining and Intelligent Computing (ICDMIC)* (pp. 1–6). IEEE.
37. Somani, G., Gaur, M. S., Sanghi, D., & Conti, M. (2016) DDoS attacks in cloud computing: Collateral damage to non-targets. *Computer Networks*.
38. Syed, T. A., Musa, S., Rahman, A., & Jan, S. (2015). Towards secure instance migration in the cloud. In *2015 International Conference on Cloud Computing (ICCC)* (pp. 1–6). IEEE.
39. Taha, M. M. B., Chaisiri, S., Ko, R. K. (2015). Trusted tamper-evident data provenance. In *2015 IEEE Trustcom/BigDataSE/ISPA* (Vol. 1, pp. 646–653). IEEE.
40. Toegl, R., Winkler, T., Nauman, M., & Hong, T. (2009). Towards platform-independent trusted computing. In *Proceedings Of The 2009 Acm Workshop On Scalable Trusted Computing* (pp. 61–66). ACM.
41. Wallom, D., Turilli, M., Martin, A., Raun, A., Taylor, G., Hargreaves, N., et al. (2011). myTrustedCloud: Trusted cloud infrastructure for security-critical computation and data management. In *IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom)*. (pp. 247–254).
42. Wei, L., Zhu, H., Cao, Z., Jia, W., & Vasilakos, A. V. (2010). SecCloud: Bridging secure storage and computation in cloud. In *IEEE 30th International Conference on Distributed Computing Systems Workshops* (pp. 52–61).

43. Zawoad, S., & Hasan, R. (2015) FECloud: A trustworthy forensics-enabled cloud architecture. In *Proceedings of 11th Annual International Federation for Information Processing WG 11.9 International Conference on Digital Forensics* (pp. 271–285).
44. Zhang, R., Su, X., Wang, J., Wang, C., Liu, W., & Lau, R. W. H. (2015). On Mitigating the Risk of Cross-VM Covert Channels in a Public Cloud. *IEEE Transactions on Parallel and Distributed Systems*, *26*(8), 2327–2339.