Computers, Materials & Continua
DOI: 10.32604/cmc.2022.028382
Article

Tech Science Press

Root-Of-Trust for Continuous Integration and Continuous Deployment Pipeline in Cloud Computing

Abdul Saboor[1,*], Mohd Fadzil Hassan[2], Rehan Akbar[1], Erwin Susanto[3], Syed Nasir Mehmood Shah[4], Muhammad Aadil Siddiqui[5] and Saeed Ahmed Magsi[5]

[1]High-Performance Cloud Computing Centre (HPC3), Department of Computer & Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Perak, Malaysia
[2]Centre for Research in Data Science (CeRDaS), Department of Computer & Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Perak, Malaysia
[3]School of Electrical Engineering, Telkom University, Bandung, Indonesia
[4]KICSIT, Institute of Space Technology (IST), Islamabad, Pakistan
[5]Department of Electrical and Electronics Engineering, Universiti Teknologi PETRONAS, Seri Iskandar, Perak, Malaysia
*Corresponding Author: Abdul Saboor. Email: abdul_19001745@utp.edu.my

Received: 08 February 2022; Accepted: 30 March 2022

Abstract: Cloud computing has gained significant use over the last decade due to its several benefits, including cost savings associated with setup, deployments, delivery, physical resource sharing across virtual machines, and availability of on-demand cloud services. However, in addition to usual threats in almost every computing environment, cloud computing has also introduced a set of new threats as consumers share physical resources due to the physical co-location paradigm. Furthermore, since there are a growing number of attacks directed at cloud environments (including dictionary attacks, replay code attacks, denial of service attacks, rootkit attacks, code injection attacks, etc.), customers require additional assurances before adopting cloud services. Moreover, the continuous integration and continuous deployment of the code fragments have made cloud services more prone to security breaches. In this study, the model based on the root of trust for continuous integration and continuous deployment is proposed, instead of only relying on a single sign-on authentication method that typically uses only id and password. The underlying study opted hardware security module by utilizing the Trusted Platform Module (TPM), which is commonly available as a cryptoprocessor on the motherboards of the personal computers and data center servers. The preliminary proof of concept demonstrated that the TPM features can be utilized through RESTful services to establish the root of trust for continuous integration and continuous deployment pipeline and can additionally be integrated as a secure microservice feature in the cloud computing environment.

Keywords: Root of Trust (RoT); Trusted Platform Module (TPM); cryptoprocessor; microservices; Hardware Security Modules (HSM); DevOps

1 Introduction

The digital revolution and hyper-connectivity are indeed impacting and will continue to influence our economy and society in different manners. Cyber-physical systems (CPS), edge computing, cloud computing, internet of things (IoT), embedded systems, service-oriented architecture (SOA), and other technological advancements provide all of the enabling factors for the industrial revolution, which is changing the industrial landscape environment. Cyber security is a vital part of this industrial revolution, and its knowledge domain is not restricted to certain applications areas. Some of the primary areas of cyberspace that have to be secured include enterprise computing infrastructure [1,2], IoT [3,4], telemedicine [5,6], and eventually the internet of everything [7].

The transition towards cloud/edge computing has contributed to the growth of data centers as they are the infrastructure that offers this modern model for computing and information management [8]. The term cloud computing, as defined by the American National Institute of Standards and Technology (NIST), allows universal and flexible on-demand access to a shared pool of configurable computing resources (e.g., storage, networks, compute nodes, applications, and services), which can be easily supported and delivered with nominal management effort and will require minimum service provider contact [9].

In the recent past, the shift towards the use of microservices in cloud computing environment also came with several challenges, including security vulnerabilities [10,11], finding the right size and number of services [12], reducing energy consumption and carbon footprint [13], better response time [14] and many more. Such cloud computing issues compromise resources for running services and lose potential customers at peak times or even over-provision resources that contribute to wasted capital costs.

Cloud security is one of the core challenges as businesses now have easier access to a wide range of computational resources, such as memory, hosts, and services, due to the cloud infrastructure. Leveraging cloud services also offer extra benefits in terms of cost reductions, performance gains, accessibility, adaptability, and scalability [15]. Such benefits have prompted organizations to rely on cloud computing to supply services to their clients. Therefore, it is vital to provide secure cloud infrastructure against internal and external threats. A substantial collection of studies on the security issues of cloud computing and security solutions are used to mitigate or eliminate those security threats [16].

Furthermore, the emergence of agile development models and vast adoption of DevOps lead to a new domain of security assurances in continuous integration and continuous deployment (CI/CD) processes [17,18]. However, numerous attacks have been carried out by exploiting flaws in software-based security systems. It means that solely software-based security solutions can no longer provide foolproof protection [19]. As a result, the developers and researchers are moving towards hardware-based security solutions such as trusted computing to provide more solid security. The trusted computing group (TCG) has developed trusted computing technology (TCT) to address computer security issues through system-on-chip advancements. It is accomplished by integrating a hardware chip called a trusted platform module (TPM) [20] that secures the apps and services running on multiple systems from tampering, thereby assuring that the platform operates as expected. The core components of TPM (refer: Fig. 1) consists of cryptographic processor (includes: random number generator, Rivest-Shamir-Adleman (RSA) key generator, encryption-decryption engine, etc.), persistent memory (includes: storage root key-SRK, endorsement key-EK), and versatile memory (includes: storage keys, attestation identity keys-AK, and platform configuration registers-PCR) [21].

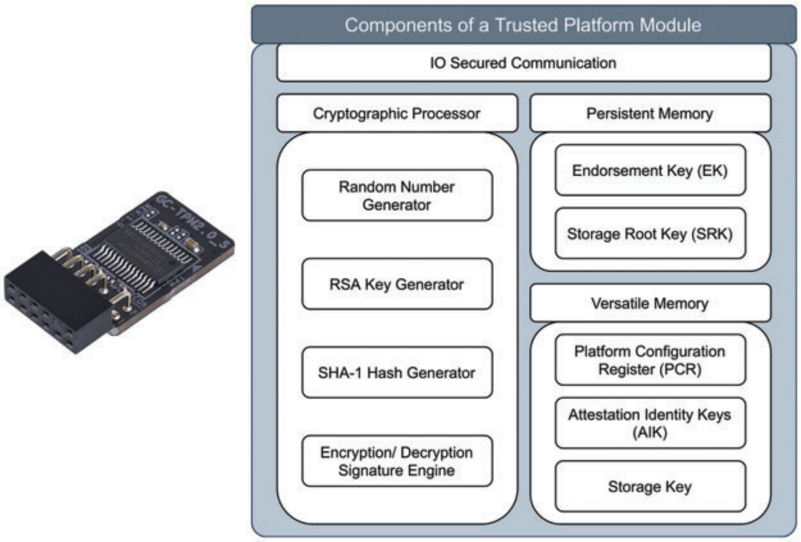


Figure 1: Main components of Trusted Platform Module (TPM)

TPM establishes trust for data storage, data integrity protection, measurement, and reporting [22]. It allows a device to be authenticated, identified, verified for integrity, and encrypted. The critical feature underlying trust computing technology is remote attestation which steadily increases trusted computing from the bottom to the middleware and upper application layers; this is known as trust chain. Among a chain of trust scenario, all systems that will be launched are considered unverified and should be therefore reviewed before loading. TPMs feature certain memory regions known as platform configuration registers (PCR), storing crucial security information such as measured values. In addition to safe storage, the TPM keeps one unique endorsement key (EK) for cryptographic operations. This key is created during the TPM manufacturing process, and the private part of the key has never been removed from the TPM. At the time of remote attestation, attestation identity keys (AK) are employed to ensure the confidentiality of the system's identification. When an attester gets a request for a remote attestation system (from a verifier), it generates an authenticity report that includes PCR values and digital signatures created using an AK. The fact that the private component of the AK has never been removed from the TPM guarantees the report's legitimacy and authenticity [22].

This research study develops an artifact model that specifies the security actions applicable to CI/CD process while utilizing the TPM. This artifact contributes to a solution to the problem of integrating security in DevOps. As a result, design science methodologies were chosen as the basis for this research work. The core contributions of this article can be summarized as:

a) Propose a TPM-based authentication model for CI/CD process.
b) Propose a platform remote attestation technique using RESTful services to safeguard against user identify theft.
c) Validate the working principle and implementation feasibility by evolving a proof-of-concept prototype.

The remaining paper is structured as follows: The literature review is presented in Section 2, which discussed the existing solutions and identified the deficiencies. Next, the details of our proposed model are given in Section 3, which addresses root-of-trust for CI/CD processes. To validate our solution's working principle and feasibility by employing a prototype is described in Section 4. Finally, Section 5 concluded the study and gave direction for the future.

2 Literature Review

This study addresses the use of TPM for cloud architecture/environment to have the root of trust, thus increasing the CI/CD process security in a cloud computing environment to reduce the risk of user identity theft and optimize single sign-on service security. However, there are other alternative methods of hardware security [19] that are not considered due to the limited scope of the study. Such methods are Platform Trust Technology (PTT), Software Guard Extensions (SGX), and Memory Protection Extensions (MPE). Although SGX has been widely utilized to solve numerous security problems, it still includes flaws that might be abused to compromise its users' privacy. One of such vulnerabilities is the successful side-channel assaults that have resulted in the leakage of secrets from the SGX enclaves [23]. As a result, employing alternative technologies like TPM, merging TPM and SGX [24], embracing other trusted computing services, and investigating their applicability in situations like cloud computing has become an important study area.

To enhance trust in software as a service (SaaS) of cloud services, Hedabou, Azougaghe, and Bentajer proposed a design for a trustworthy SaaS model that evokes greater confidence in cloud users' applications [25]. The trust authorities and the underlying cloud service provider verified the SaaS services application source code using a multi-signature mechanism based on TPM. The proposed protocol of the architecture demonstrated that the service's validity could be ensured both before and after it is launched on a cloud service infrastructure.

Mo et al. [26] addressed remote user security authentication schema, where the cloud users are registered with the trusted certificate authority to get the certified authority issued certificate. These certificates are then used to validate the users.

The identity theft risk for single sign-on authorization was evaluated by Cusack et al. [27]. They proposed involving an external trusted party (such as TPM) as an authorized entity to mitigate such threats. A working solution was provided that can minimize the trade-off among the risk of disclosure, the risk of human users, and the security of services.

Muthiya [28] proposed a solution that first checks the user's authentication and then separates the user's information using a pattern matching methodology to enable secure cloud data operations. Blowfish computation is utilized to encrypt the data. After that, the ideal location of a data center is determined using the cross grey wolf optimization and firefly approach. Finally, the encrypted data is stored in the cloud at an ideal place. The data is then split column by column and segregated at an ideal location in the cloud. It was concluded that this approach is very secure because the user cannot get the file without authentication verification.

A flexible and safe access control system for a single sign-on that enables a security-level-based authorization methodology is proposed by Badirova et al. [29]. According to the study, the attribute mapping strategy simplifies accessibility management by classifying a wide variety of characteristics and authentication mechanisms (e.g., PKI, FIDO U2F, LoA, and so on); the use of a multi-factor authentication architecture helps to maintain security consistency.

A study by Yang et al. [30] proposed an anonymous signature schema named direct anonymous attestation (DAA) that used TPM to test the host state. They reduced the TPM signing workload where TPM was utilized to take single exponentiation for signature generation. The test results showed that the proposed schema took less signing time.

Khan et al. [31] offered a thin client-friendly methodology for assessing the trustworthiness of cloud providers to establish trust between both the user and the cloud service provider. The cloud provider platform was validated by a Trusted Party (TP), which serves the purpose of verifying the security characteristics of a cloud platform. The proposed architecture attests, ranks, and finally certifies the cloud platform with the TP leveraging attestation based on the Trusted Platform Module (TPM). The cloud users then have an added advantage to choosing a cloud platform based on the security requirements.

The performance analysis of a secure cloud computing model was done by Alotaibi et al. [32]. A cloud security performance model (CSPM) was proposed, and the evaluation of performance was done for different data sizes. The comparison of RSA and ECC asymmetric encryption for storing and retrieving different-sized data was performed.

A schema-based on TPM to reinforce security by using identity-based encryption (IBE) key manager was introduced by Igarramen et al. [33]. The authors proposed a design where IBE is used to perform the key management. The system prototype implementation demonstrated that such a design could offer value-added to the security. And to our understanding, such schema can be extended to cloud architecture.

The technologies of ARM Trust Zone, TXT, AMD SEV, and SGX are provided in a detailed review as four essential industrial-scale commercial hardware-based solutions offered that can be used by cloud security providers in the data centers [19]. The research also provided a guideline for IT administrators in assessing which solution is the most appropriate for their specific security requirements and future cloud deployments. The authors concluded that all four solutions in particular settings can provide security services and can be used to ensure the security of data centers. However, the authors of the paper omitted the TPM because it is mainly implemented as a dedicated, special-purpose chip that can be used in combination with Intel TXT. However, contrary to their statement, the proposed model uses the TPM feature set to provide security for the microservices-based application running in the cloud environment. To sum up, the key highlights of the literature review are presented in Tab. 1.

Table 1: Highlights of the literature review

S. No.	Author(s)/ Reference	Technique/Algorithm/ Methodology	Finding of the research	TPM usage
1	Hedabou et al. [25]	Used Diffie-Hellman oracle procedure to amend the Boldyreva's scheme to meet the RSA-based signature requirements of the TPM.	The proposed protocol of the architecture demonstrates that the service's validity is ensured both before and after it is launched on a cloud service infrastructure.	✓

(Continued)

Table 1: Continued

S. No.	Author(s)/ Reference	Technique/Algorithm/ Methodology	Finding of the research	TPM usage
2	Mo et al. [26]	Generated the temporary identification ID by using a third-party CA for cloud users, thus leading to mutual authentication.	The agreement process led to fewer interaction trips and have less computational complexity, and lower communication delays.	✓
3	Muthiya et al. [28]	Blowfish computation is utilized to encrypt the appropriated data. The ideal location of a data center is determined using the cross grey wolf optimization and firefly approach.	This approach is very secure since the data is split column by column and isolated at the best possible place in the cloud. Because the user cannot access the file without first confirming identity, this approach is extremely safe.	✗
4	Badirova et al. [29]	Security level-based access design and attribute mapping approaches were coupled, leveraging on the multi-factor authentication.	Simplify the identification and authorization procedure while increasing the security of the services provided.	✗
5	Yang et al. [30]	Used Direct Anonymous Attestation (DAA) to provide the anonymous and pseudonymous signatures.	It reduced the TPM signing workload where TPM was utilized to take single exponentiation for signature generation, thus requiring less signing time.	✓
6	Khan et al. [31]	Ranking and verifying the cloud vendor attributes through a trusted computing component without the client's direct involvement.	Support thin clients (mobile clients), thus relieving the client of the computationally intensive task of attestation and verification.	✓
7	Igarramen et al. [33]	Use of Identity-based encryption (IBE) along with TPM and decentralized private key generator (PKG).	It improved the security of PKG and its master secret key.	✓
8	Muñoz et al. [34]	Trusted integrity platform (TIP) server utilized in P2ISE tool for integrity and validation proofs	CPU usage and memory utilization verified that the solution does not exhaust developers' resources or cause deployment delays.	✓

A trusted platform module can be used in a cloud computing environment for various objectives, including adding a layer of protection to operations, storage, communications, and monitoring. The rest of the section summarizes the potential of TPM utilization in the cloud computing environment and the security issues mitigation using TPM.

Key generation and providing secure storage of the key: The TPM can generate cryptographic keys and save them securely within the TPM [35]. These keys can be later on used by other entities, for example, virtual machines, operating systems, etc. In TPM, keys can then be produced in one of three ways: via seed, via random number generator, or importing them into the TPM.

Integrity measurement: The newer type of TPM (version 2.0) has agility in hash functions. Now the TPM can use virtually any hash algorithm such as SHA-256, advanced encryption standard (AES) symmetric algorithm [20], and elliptic curve cryptography (ECC) asymmetric algorithm [36]. The TPM is capable of calculating and storing hash values for various components of the system in its registers. Several hashes can be combined. The value of the given hash is compared against the stored value; thus, it is used to detect any unauthorized modifications and guarantee system integrity [37].

Remote attestation: The mechanism of remote attestation/ distributed attestation can be provided using TPM. It enables an entity to authenticate itself with another remote entity. Remote attestation has several implications in the cloud. In this sense, an entity demonstrates to the petitioner with its state acknowledging that it’s trustworthy and the underlying system is not compromised and tampered with [38].

Cryptographic operations: TPM co-processor can perform cryptographic operations by providing features including generating a random number, hashing, performing symmetric/ asymmetric encryption, and generating keys [21].

Trusted boot: This feature can be provided by the TPM through integrity measurement. Whenever a system begins the booting process, the TPM evaluates and performs an integrity measurement on the various hardware, software, firmware components to guarantee that the system was not altered or tampered with before the booting process [22].

VM monitoring: A typical service supplied by TPM in cloud computing infrastructure is the secure monitoring of virtual machines. TPM can offer a secure, energy-efficient, reliable, and efficient protocol for migrating virtual machines between multiple cloud providers [39] by considering essential security services such as privacy, authenticity, and security [40].

According to the literature review, it is evident that prevailing trusted computing-based methods are inadequate for cloud computing (particularly for CI/CD pipelines) because they require the complicated operation of authentication and authorization on the client-side while potentially disclosing too much information about the underlying infrastructure to service users. Furthermore, the solutions provided do not use the emerging microservices architecture and do not provide abstraction as RESTful services.

3 Methodology

Cloud computing is centered around a variety of deployment and service models for many users while delivering a wide range of services. Complex applications with many interconnected microservices demand a sophisticated framework for the development and operation (DevOps). A pipeline notion that facilitates the CI/CD of microservices is one of the necessary elements of a DevOps ecosystem. These pipelines for CI/CD guide a developer’s source code across multiple phases, such as building, testing, packaging, deployment, and operations, using automated systems with regulatory mechanisms. This section explains the proposed methodology for TPM-based design to establish the root of trust for microservices source code deployment during the CI/CD process. Contrary to the classical approach, which uses ID and password for source authentication, a service-based identity authentication model based on a hardware security module (HSM) is proposed to check the legitimacy of the source from where the microservice source code is deployed into VM/Container hosted on the cloud data center.

3.1 TPM as Root of Trust for Cloud Computing Environment

In computer systems, the idea of a root of trust (RoT) is the first link in a trust chain that ensures secure boot, secure communication, and mitigating unauthorized access to systems and services. If the credentials of the first code run have been confirmed as valid, the operation of the following pieces of code will trust those credentials. A hardware root of trust is the basis upon which a computing system's secure operations are based. Within a cryptographic system, the root of trust is the component that can always be trusted. RoT methods often contain a hardened hardware module to generate crypto keys because cryptographic security relies on keys for encryption and decryption operations on data and perform activities like creating digital signatures and validating signatures [41]. The hardware security module (HSM), for example, produces and protects keys while also performing cryptographic activities within a secure environment.

This research study utilized Trusted Platform Module (TPM) to establish RoT. TPM is a System on Chip (SoC) hardware module that is separate from the CPU. The two variants of TPM are termed TPM-1.2 and TPM-2.0. The TPM 1.2 specification specifies only the usage of the RSA and SHA-1 hashing algorithms [42]. NIST has already directed several government agencies to shift to SHA-256. Therefore, several organizations are abandoning SHA-1 due to security concerns. The industry giants such as Google, Amazon, Microsoft have stated that they phased out support for SHA-1-based signatures and certificates in 2017.

On the other hand, the TPM-2.0 standard is ISO compliant (according to ISO/IEC 11889-1:2015,104) and has been approved by many countries. The TPM Software Stack (TSS) is a model that establishes a consistent API for interacting with the TPM's functionality. The TSS software specifications may be used by the developers to create interoperable client apps for better tamper-resistant computing. Infineon Technologies (a German semiconductor manufacturer founded in 1999 and producer of TPM add-on modules) has released an open-source middleware for use with TPM-2.0 in 2018. What would be needed now is a straightforward implementation and incorporation of TPM for the cloud, not only of dedicated middleware direct interaction with hypervisors and virtual machines but also in the form of consumable microservices that can be deployed with less management and technical efforts. This study took advantage of TPM to provide source authorization for microservices source-code deployment in a cloud environment.

3.2 TPM-Based Security for CI/CD in Cloud Computing Environment

Companies dedicate greater attention and resources to building and creating sustainable software at a faster pace as the software business becomes more competitive. Continuous integration, delivery, and deployment are software development industry processes that allow developers to release new features more often and reliably. Fig. 2 shows the process in detail where code, build, and testing stages of the software release process are referred to as continuous integration. Each committed revision of the code starts an automatic build and test process. Code changes are usually automatically created, tested, and then packaged to release for production deployment using continuous delivery. Continuous delivery complements continuous integration by providing all code changes to a production environment once the build stage is completed. For such a CI/CD process, the legitimacy check of the source is based on user ID and password. Although the use of id and password is a simple, quick, and convenient method to use and implement, it is also considered a poor form of security.

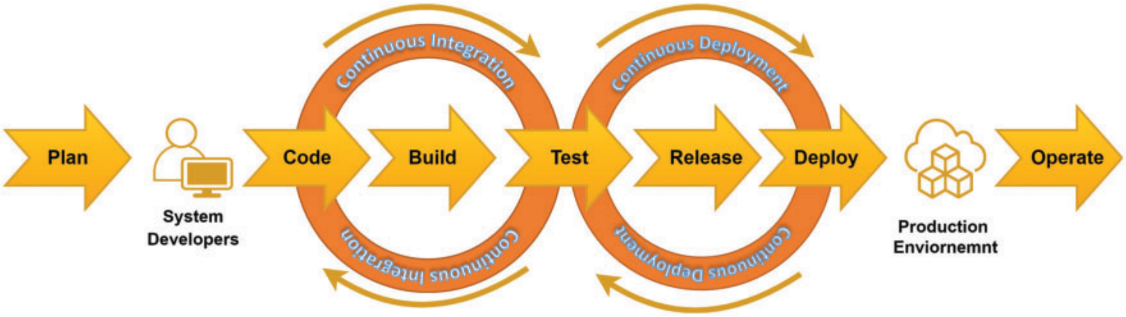


Figure 2: Continuous Integration (CI) and Continuous Delivery (CD) process

To provide a better way to check the legitimacy of the source system, this study proposes a novel model to establish RoT for CI/CD process in Cloud Computing Environment. The model relies on TPM to define root-of-trust for CI/CD in the cloud computing model and provides access to TPM features using RESTful services. To provide the greatest degree of abstraction to the lowest level, the TPM software stack (TSS) is made up of the following layers [21]: Device Driver (DD), Resource Manager (RM), TPM Access Broker (TAB), TPM Command Transmission Interface (TCTI), System API (SAPI), Enhanced System API (ESAPI), and Feature API (FAPI), as shown in Fig. 3. FAPI is intended to be a very high-level API to stage everything required by most of the programmers that build a program utilizing the TPM. If needed, it can complement this set of APIs using the ESAPI or SAPI to provide access to DD, RM, TAB, and TCTI.

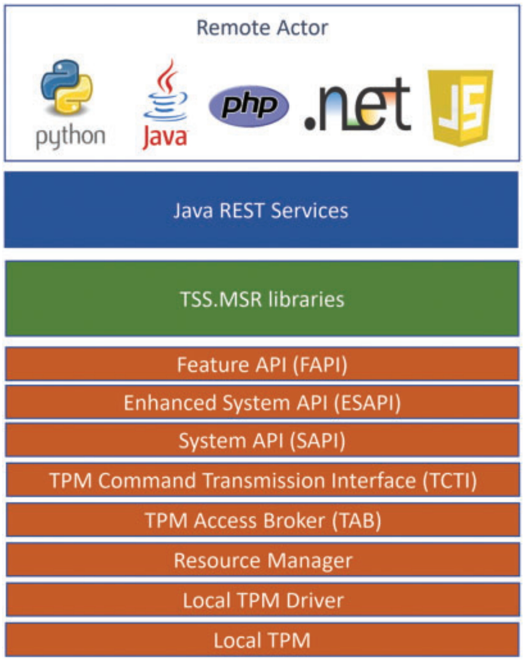


Figure 3: Layered representation of the proposed framework

The TSS.MSR API was used on top of these layers (refer to Fig. 3), which is Microsoft's TPM Software Stack (TSS) implementations. Microsoft has created several TSS implementations for several high-level programming languages to make the development of TPM 2.0 based applications and services easier. All of these implementations use the relevant languages to give a comprehensive implementation of the TPM 2.0 API (data structures, commands, enumerations, and unions). In addition to real TPM devices, the TSS.MSR libraries give access to a TPM simulator, allowing software development and experimentation on systems without a TPM 2.0 device. The simulator is linked through a TCP/IP socket, allowing it to run on a distant machine or within another process running on the same machine.

The TPM software stack (TSS), which provided an application programming interface, was used to access the TPM function. The TSS.MSR, which provided the TSS with the Java abstraction layer, was stacked among the RESTful function. The final output was retrieved in JavaScript Object Notation (JSON) which is a lightweight data exchange standard and easily accessible by cloud-based applications. The JSON response is then utilized by the remote actors (i.e., end-user application or service to service requests). The defined root-of-trust using TPM 2.0 provided authorization of the source from where the microservice is committed to the cloud for production/developed purposes. The proposed RoT layer, as shown in Fig. 4, used the TPM chip of the source system to generate and match attestation keys and provide platform attestation.

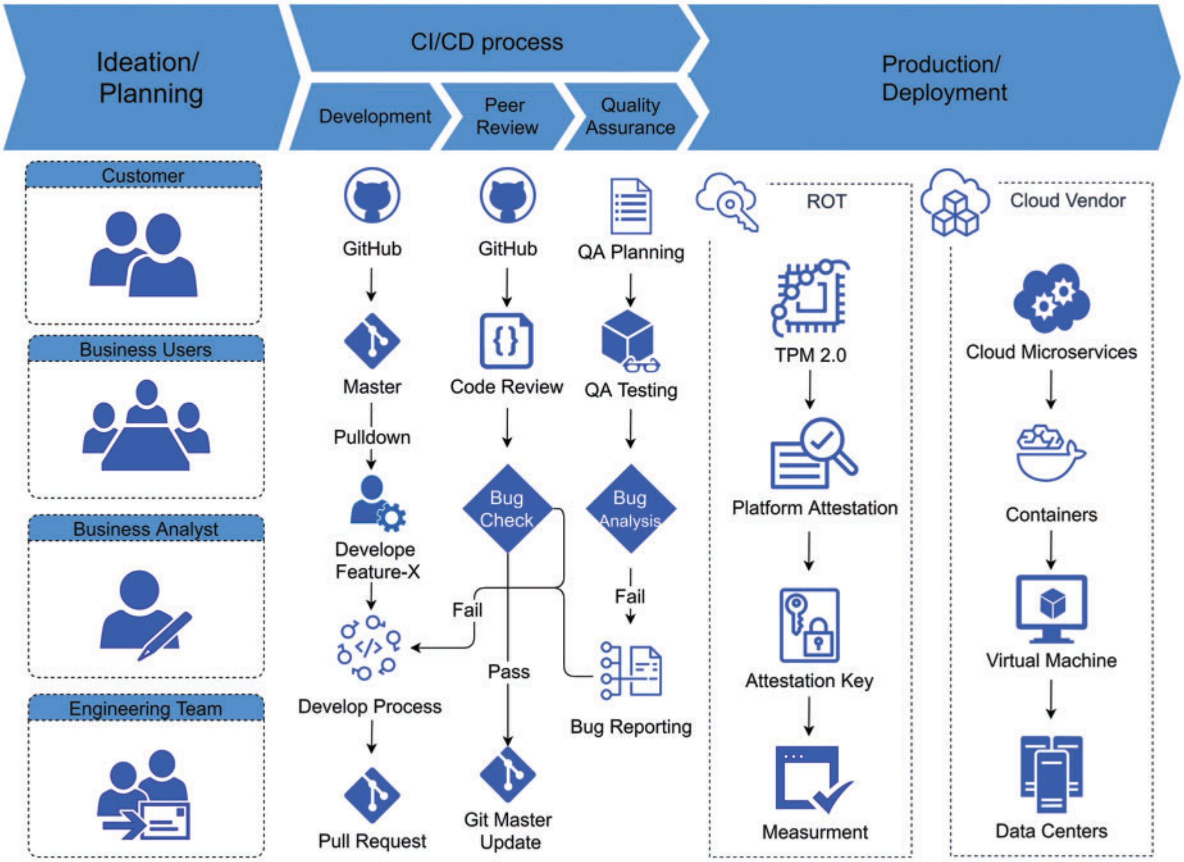


Figure 4: Proposed root of trust (RoT) layer

The sequence diagram of the proposed model is shown in Fig. 5 to elaborate on the working principle of the model. It illustrates the steps to generate credentials, store credentials, and verify credentials. Several components are in action, including the TPM, the Attestor (source system), the Verifier (Java-based web server hosting RESTful Service), and the TPM CA.

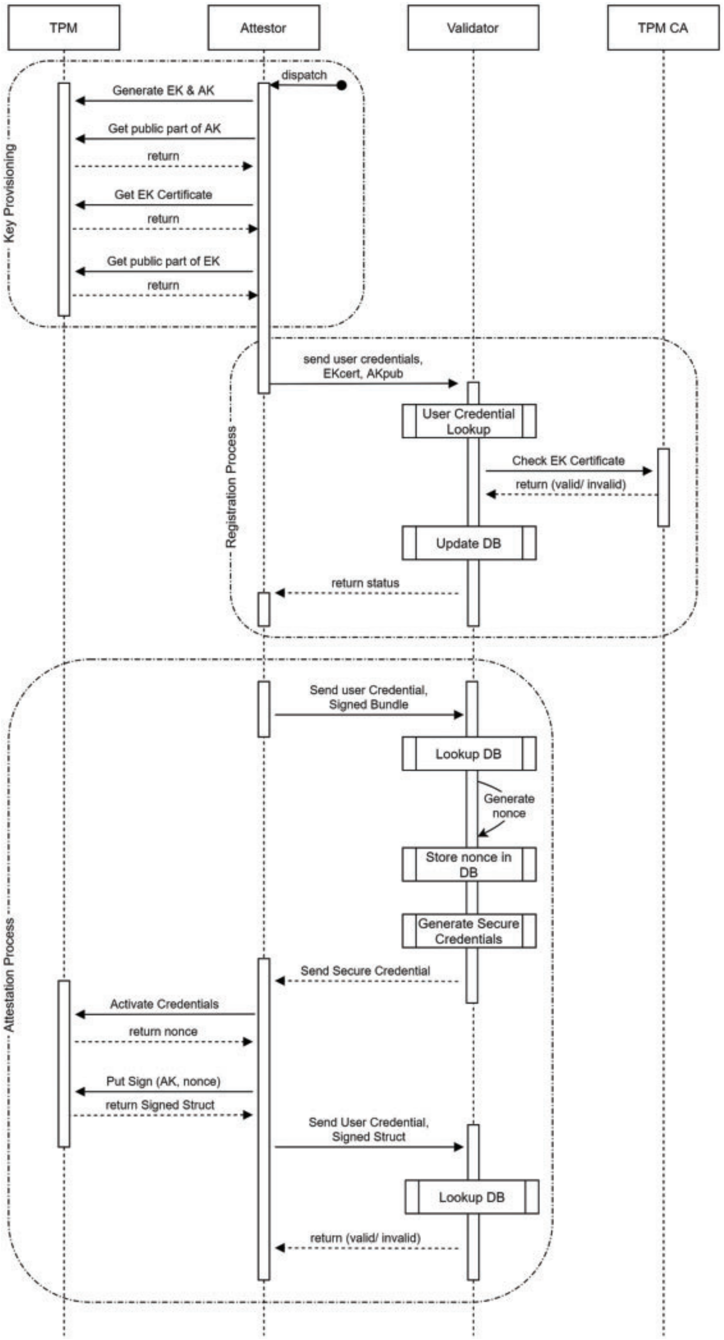


Figure 5: The proposed model's sequence diagram for creating, storing, and verifying credentials

The process starts from the attestor who originates the key provisioning request. The key provisioning process produces an endorsement key (EK) and an attestation key (AK) on a TPM. The endorsement key is an asymmetric key that is persistently incorporated in the Trusted Platform Module (TPM) security hardware, typically during the manufacturing process. Outside of the TPM, the private element of the endorsement key is never disclosed. The public part of the endorsement key assists in the verification of a real TPM. In contrast, the AK is a non-duplicatable restricted signing key. The attestor reads and stores the AK public key and EK certificate on the platform.

In the registration process, the credentials are stored with the verifier. The generated EKpub, AK, and EK certificate (EKcert) are passed to the verifier along with the user credentials (generally consisting of id and password). The validator will run a user lookup in DB, where the query result will correlate with a user in the database. The verifier then validates the EK cert by verifying the certificate chain with the TPM CA. This ensures that the TPM is a real hardware TPM and not a TPM emulator. Once the EKcert is verified, the registration processes will record relevant platform parameters to the leading edge/data layer. These values are recorded into the validator database (DB). Meanwhile, the status will be returned to show that the registration request is completed and ready to perform the source authentication.

During the authentication process, the attester submits an authentication request to the validator. The validator gets the generated nonce. Then it encrypts the generated nonce, which results in the production of the security credentials. These secure credentials are then passed back to the attestor.

When the attestor receives a secure credentials blob, the attestor may decrypt the credentials blob by using the AK, which will extract the nouns, along with the AK handle. The attestor then creates a structure consisting of the nonce along with Akpub and sends it back to the validator. The received structure is then compared to the expected metrics and stored information by the validator. If there is a match, the validator will authorize the user and create the user session for the CI/CD process. Meanwhile, it will also return the authentication status to trigger the CI/CD process. If a mismatch happens, the validator will reject the session request and send back the appropriate error message to the attestor.

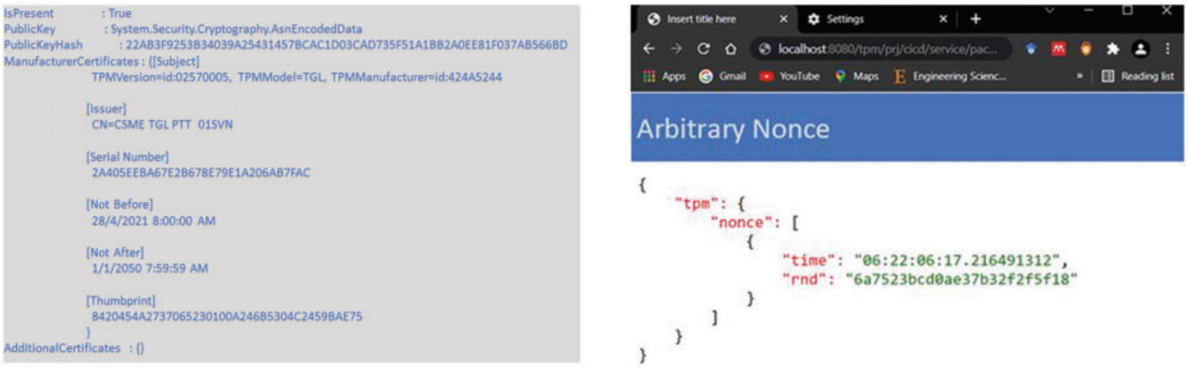
4 Results and Discussion

The core objective of this study was to define TPM based design to establish a root of trust for continuous integration and deployment process in the cloud computing environment. Generally, the single sign-on service concept is applied for the authorization of users in the CI/CD pipeline [43]. A consistent pattern of login credentials might be used to authenticate the user. The same may be utilized to access more than one application with the assistance of cookies produced and session control [44,45]. The proposed model (as stated in the Methodology Section) was implemented and tested to offer the proof of concept (which is typically produced from an experimentation or trial project and indicates that a concept design, business proposition, or other similar notion is possible) to deliver an improved root of trust in CI/CD pipeline. The proposed model was deployed in a real-time system running service-oriented architecture (SOA) that uses both RESTful and SOAP Web services. The hardware and software configuration employed for this proof of concept (PoC) is given in Tab. 2.

The execution process started with key provisioning. The EK and AK keys were generated stored on the attestor machine. The communication between the entities was based on RESTful service calls hosted on a local Tomcat server. The JSON responses were also captured for ease of use. The EK information, as shown in Fig. 6a, provided the details about EK public key hash, manufacture information, serial number, etc. The actual values of EK and AK in the figures are changed due to security reasons.

Table 2: Test environment configurations for PoC

Hardware/ Software	Configuration
Processor	11th Gen Intel(R) Core (TM) i7-11390H @ 3.40GHz
Memory	8 GB RAM; 512 GB SSD
VMware Workstation Pro	16.1.2
VM Operating System	Ubuntu 18.04
Web Server	Apache Tomcat 9.0.46
Services	Jakarta RESTful Web Services (JAXRS), Java Native Access (JNA)
TPM	Version 2.0; Intel (INTC)



```
IsPresent              : True
PublicKey              : System.Security.Cryptography.AsnEncodedData
PublicKeyHash            : 22AB3F9253B34039A25431457BCAC1D03CAD735F51A1BB2A0EE81F037AB566BD
ManufacturerCertificates : {[Subject]
                    TPMVersion=id:02570005, TPMModel=TGL, TPMManufacturer=id:424A5244

                   [Issuer]
                    CN=CSME TGL PTT  01SVN

                   [Serial Number]
                    2A405EEBA67E2B678E79E1A206AB7FAC

                   [Not Before]
                    28/4/2021 8:00:00 AM

                   [Not After]
                    1/1/2050 7:59:59 AM

                   [Thumbprint]
                    8420454A2737065230100A246B5304C2459BAE75
                   }
AdditionalCertificates  : {}
```

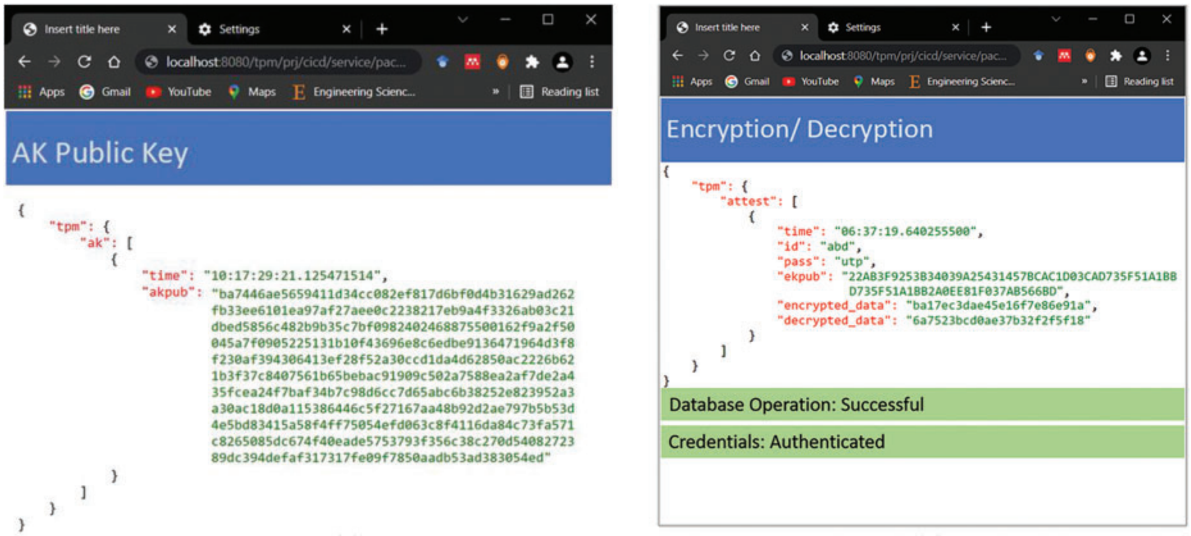
(a)

```
{
    "tpm": {
        "nonce": [
            {
                "time": "06:22:06:17.216491312",
                "rnd": "6a7523bcd0ae37b32f2f5f18"
            }
        ]
    }
}
```

(b)

Figure 6: (a) Endorsement key information; (b) Arbitrary nounce generated

The local RESTful services were able to access the TPM features such as true random number generator (refer to Fig. 6b), data encryption, and data decryption. The TPM software stack (TSS) was used, which provided an API to access the TPM function and built a mutual authentication scheme that uses keys/certificates stored in the TPM to authenticate the compute nodes and their parents natively. In the PoC run, the RESTful function successfully communicated with the TSS.MSR, which provided the abstraction layer for the TSS. The developed RESTful services in java utilized the Jakarta RESTful Web Services (JAXRS). Also, the Java Native Access (JNA) was used to access the core TPM libraries, which were available as C/C++ libraries. The final result fetched was provided in a lightweight data-interchange format. The fetched result of the AK public part is shown in Fig. 7a. The result of the encryption and decryption process is shown in Fig. 7b. The successful operation of these services indicates that such services can be successfully implemented as RESTful services and have the ability to provide the root of trust for continuous integration and continuous deployment process in the cloud computing environment.



AK Public Key

```
{
    "tpm": {
        "ak": [
            {
                "time": "10:17:29:21.125471514",
                "akpub": "ba7446ae5659411d34cc082ef817d6bf0d4b31629ad262
                         fb33ee6101ea97af27aee0c2238217eb9a4f3326ab03c21
                         dbed5856c482b9b35c7bf0982402468875500162f9a2f50
                         045a7f0905225131b10f43696e8c6edbe9136471964d3f8
                         f230af394306413ef28f52a30ccd1da4d62850ac2226b62
                         1b3f37c8407561b65bebac91909c502a7588ea2af7de2a4
                         35fcea24f7baf34b7c98d6cc7d65abc6b38252e823952a3
                         a30ac18d0a115386446c5f27167aa48b92d2ae797b5b53d
                         4e5bd83415a58f4ff75054efd063c8f4116da84c73fa571
                         c8265085dc674f40eade5753793f356c38c270d54082723
                         89dc394defaf317317fe09f7850aadb53ad383054ed"
            }
        ]
    }
}
```

(a)

Encryption/ Decryption

```
{
    "tpm": {
        "attest": [
            {
                "time": "06:37:19.640255500",
                "id": "abd",
                "pass": "utp",
                "ekpub": "22AB3F9253B34039A25431457BCAC1D03CAD735F51A1BB
                          D735F51A1BB2A0EE81F037AB566BD",
                "encrypted_data": "ba17ec3dae45e16f7e86e91a",
                "decrypted_data": "6a7523bcd0ae37b32f2f5f18"
            }
        ]
    }
}
```

Database Operation: Successful

Credentials: Authenticated

(b)

Figure 7: (a) AK public key information (b): Encryption/ Decryption operation

The overall working of successful operation of PoC proved that TPM-based proposed model could establish root-of-trust for continuous integration and continuous deployment process in a cloud computing environment. The underlying TPM based RESTful services can improve security inside a microservices-based cloud application even if it is divided and dispersed over two or more cloud service providers, which is one of the core benefits of adopting cloud-based applications (the ability to divide and distribute the application over several platforms as a Service supplier such as Amazon AWS, Google App Engine, Microsoft Azure, and others). In such an instance, the lightweight communication between the participating service providers may occur directly over the RESTful service layer.

The proposed schema to establish the root of trust for continuous integration and continuous deployment pipeline in the cloud computing environment is effective against security attacks such as dictionary attacks, brute force, cryptanalysis, and reply attacks. A dictionary attack occurs when an attacker attempts to steal user credentials, primarily passwords, by employing a dictionary of commonly used passwords. The PoC showed that the proposed schema could use the TPM owner authorization to mitigate dictionary attacks. It is also effective against brute force (when an attacker attempts every conceivable key and password combination in an attempt to guess the design secrets) and cryptanalysis (where hackers may try to compromise the integrity of a cryptographic design by learning the mathematics of the cryptographic algorithms and utilizing that understanding to search for design faults). The use of a nonce in the proposed methodology also protected against reply attacks because the nonce is used only once in a cryptographic operation and never repeated.

5 Conclusion and Future Work

This study proposed a TPM-based design to establish root-of-trust for the continuous integration/ deployment process. The remote attestation mechanism was developed and tested to guard against user identity theft. Instead of just relying on user id and password, the remote attestation involved the TPM generated EK and AK keys to check the source's legitimacy, which initiated the code change process widely sent through the automated procedure that will reach the production environment using the

CI/CD pipelines. The developed proof of concept demonstrated the core functionality of the proposed model, and successful working provided evidence for its feasibility to be used in the cloud environment.

The suggested schema's findings can be summed by stating that, rather than designing and developing entirely new protocols for TPM inclusion in the CI/CD pipeline, a solution based on a proven technology of RESTful services can be employed. The TPM endorsement key and unique AK keys that cannot be replicated are used to authenticate the final deployment procedure in the CI/CD pipeline. As a result, a layer of protection based on the root of trust is added to assure the authentication and authorization of CI/CD pipeline activities.

Once considering adopting the proposed attestation model in the corporate infrastructure, it is vital to prepare for platform firmware upgrades, operating system, or user application updates. It is also crucial to design recovery strategies in the event of catastrophic failures. Because, in the event of a loss, it may be necessary to re-instantiate the impacted TPM keys. It might need to measure the system hardware and software configuration state and alter the database, which stores the measurements, keys, and critical certificates. Therefore, future work will assess the framework for acute failure and recovery procedures. It will do the performance analysis to check for CPU utilization, memory consumption, and RESTful service execution time.

Acknowledgement: The resources and computing environment was provided by the High-Performance Cloud Computing Center (HPC3) at Universiti Teknologi PETRONAS (UTP), Perak, Malaysia. We are thankful for their support.

Funding Statement: The research work was supported by UTP-Universitas Telkom, Indonesia International Collaborative Research Funding (ICRF) 015ME0-153 and Center for Graduate Studies (CGS), Universiti Teknologi PETRONAS (UTP), Perak, Malaysia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. I. Beer and M. F. Hassan, "Adaptive security architecture for protecting RESTful web services in enterprise computing environment," *Service Oriented Computing and Applications*, vol. 12, no. 2, pp. 111–121, 2018.

[2] M. I. B. Mohamed, M. F. Hassan, S. Safdar and M. Q. Saleem, "Adaptive security architectural model for protecting identity federation in service oriented computing," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 5, pp. 580–592, 2021.

[3] T. Rajmohan, P. H. Nguyen and N. Ferry, "A decade of research on patterns and architectures for IoT security," *Cybersecurity*, vol. 5, no. 1, pp. 1–29, 2022.

[4] A. Rehman, M. F. Hassan, Y. K. Hooi, M. A. Qureshi, T. D. Chung *et al.,* "Context and machine learning based trust management framework for internet of vehicles," *Computers, Materials & Continua*, vol. 68, no. 3, pp. 4125–4142, 2021.

[5] X. Zhang, X. Sun, X. Sun, W. Sun and S. K. Jha, "Robust Reversible Audio Watermarking Scheme for Telemedicine and Privacy Protection," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3035–3050, 2022.

[6] X. Zhang, W. Zhang, W. Sun, X. Sun and S. K. Jha, "A robust 3-D medical watermarking based on wavelet transform for data protection," *Computer Systems Science and Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.

[7] S. P. Mohanty, "Security and privacy by design is key in the Internet of Everything (IoE) Era," *IEEE Consumer Electron. Mag*, vol. 9, no. 6, pp. 4–5, 2020.

[8] Q. Zhang, L. Cheng and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[9] E. A. O. Simmon, "Evaluation of cloud computing services based on NIST SP 800-145," *NIST Special Publication*, vol. 500, pp. 322, 2018.

[10] D. Yu, Y. Jin, Y. Zhang and X. Zheng, "A survey on security issues in services communication of Microservices-enabled fog applications," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 22, pp. e4436, 2019.

[11] C. Esposito, A. Castiglione and K.-K. R. Choo, "Challenges in delivering software in the cloud as microservices," *IEEE Cloud Computing*, vol. 3, no. 5, pp. 10–14, 2016.

[12] M. Amaral, J. Polo, D. Carrera, I. Mohomed, M. Unuvar *et al.,* "Performance evaluation of microservices architectures using containers," in *2015 IEEE 14th Int. Symp. on Network Computing and Applications*, Cambridge, MA, USA, pp. 27–34, 2015.

[13] A. Saboor, A. K. Mahmood, A. H. Omar, M. F. Hassan, S. N. M. Shah *et al.,* "Enabling rank-based distribution of microservices among containers for green cloud computing environment," *Peer-to-Peer Networking and Applications*, vol. 15, pp. 1–15, 2021.

[14] A. Saboor, A. K. Mahmood, M. F. Hassan, S. N. M. Shah, F. Hassan *et al.,* "Design pattern based distribution of microservices in cloud computing environment," in *2021 Int. Conf. on Computer & Information Sciences (ICCOINS)*, Kuching, Malaysia, pp. 396–400, 2021.

[15] L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, "A break in the clouds: Towards a cloud definition," *SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2009.

[16] S. N. Mthunzi, E. Benkhelifa, T. Bosakowski, C. G. Guegan and M. Barhamgi, "Cloud computing security taxonomy: From an atomistic to a holistic view," *Future Generation Computer Systems*, vol. 107, no. 3, pp. 620–644, 2020.

[17] Y. Bobbert and M. Chtepen, "Findings and core practices in the domain of CI/CD and DevOps on security compliance," *Strategic Approaches to Digital Platform Security Assurance*, vol. 9, pp. 308–313, 2021.

[18] D. Verslegers, "Research findings in the domain of security assurance in DevOps," *Strategic Approaches to Digital Platform Security Assurance*, vol. 11, pp. 322–377, 2021.

[19] O. Demigha and R. Larguet, "Hardware-based solutions for trusted cloud computing," *Computers & Security*, vol. 103, no. 1–2, pp. 102117, 2021.

[20] J. D. Osborn and D. C. Challener, "Trusted platform module evolution," *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, vol. 32, pp. 536–543, 2013.

[21] W. Arthur, D. Challener and K. Goldman, "A practical guide to TPM 2.0: Using the new trusted platform module in the new age of security," *Apress Open*, vol. 1, pp. 392, 2015.

[22] S. Hosseinzadeh, B. Sequeiros, P. R. M. Inácio and V. Leppänen, "Recent trends in applying TPM to cloud computing," *Security and Privacy*, vol. 3, no. 1, pp. e93, 2020.

[23] S. Fei, Z. Yan, W. Ding and H. Xie, "Security vulnerabilities of SGX and countermeasures: A Survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–36, 2021.

[24] Z. Hongwei, K. Zhipeng, Z. Yuchen, W. Dangyang and Y. Jinhui, "TSGX: Defeating SGX side channel attack with support of TPM," in *2021 Asia-Pacific Conf. on Communications Technology and Computer Science (ACCTCS)*, Shenyang, China, pp. 192–196, 2021.

[25] M. Hedabou, A. Azougaghe and A. Bentajer, "TPM based design for enhanced trust in SaaS services," *CS & IT Conference Proceedings*, vol. 10, no. 5, pp. 217–226, 2020.

[26] J. Mo, Z. Hu and Y. Lin, "A user authentication scheme based on trusted platform for cloud computing," in *Int. Conf. on Security, Privacy and Anonymity in Computation, Communication and Storage*, Zhangjiajie, China, pp. 122–130, 2016.

[27] B. Cusack and E. Ghazizadeh, "Evaluating single sign-on security failure in cloud services," *Business Horizons*, vol. 59, no. 6, pp. 605–614, 2016.

[28] D. E. R. Muthiya, “Design and implementation of crypt analysis of cloud data intrusion management system,” *The International Arab Journal of Information Technology*, vol. 17, pp. 895–905, 2020.

[29] A. Badirova, S. Dabbaghi, F. F. Moghaddam, P. Wieder and R. Yahyapour, “An optimized single sign-on schema for reliable multi-level security management in clouds,” in *2021 8th Int. Conf. on Future Internet of Things and Cloud (FiCloud)*, Rome, Italy, pp. 42–49, 2021.

[30] K. Yang, L. Chen, Z. Zhang, C. J. P. Newton, B. Yang *et al.,* “Direct anonymous attestation with optimal TPM signing efficiency,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2260–2275, 2021.

[31] I. Khan, H.-U Rehman, M. H. F. Al-Khatib, Z. Anwar and M. Alam, “A thin client friendly trusted execution framework for infrastructure-as-a-service clouds,” *Future Generation Computer Systems*, vol. 89, no. 6, pp. 239–248, 2018.

[32] A. F. Alotaibia, M. A. AlZaina, M. Masuda and N. Z. Jhanjhib, “Performance evaluation and analysis of CSPM: A secure cloud computing model,” *Turkish Online Journal of Qualitative Inquiry*, vol. 12, pp. 3288–3306, 2021.

[33] Z. Igarramen, A. Bentajer and M. Hedabou, “TPM based schema for reinforcing security in IBE’s key manager,” in *Int. Conf. on Model and Data Engineering*, Toulouse, France, vol. 1085, pp. 146–153, 2019.

[34] A. Muñoz, A. Farao, J. R. C. Correia and C. Xenakis, “P2ISE: Preserving project integrity in CI/CD based on secure elements,” *Information-an International Interdisciplinary Journal*, vol. 12, no. 9, pp. 357, 2021.

[35] C. Chen, H. Raj, S. Saroiu and A. Wolman, “cTPM: A cloud TPM for cross-device trusted applications,” in *11th USENIX Symp. on Networked Systems Design and Implementation (NSDI 14)*, Seattle, Washington, USA, pp. 187–201, 2014.

[36] L. Chen and R. Urian, “Algorithm agility-discussion on TPM 2.0 ECC functionalities,” in *Int. Conf. on Research in Security Standardisation*, Gaithersburg, MD, USA, pp. 141–159, 2016.

[37] W. Ozga, D. L. Quoc and C. Fetzer, “TRIGLAV: Remote attestation of the virtual machine’s runtime integrity in public clouds,” in *2021 IEEE 14th Int. Conf. on Cloud Computing (CLOUD)*, Chicago, IL, USA, pp. 1–12, 2021.

[38] B. Kuang, A. Fu, W. Susilo, S. Yu and Y. Gao, “A survey of remote attestation in Internet of Things: Attacks, countermeasures, and prospects,” *Computers & Security*, vol. 112, no. 3, pp. 102498, 2022.

[39] S. K. Abd, D. A. Khalid, M. M. Jaber, R. Hassan and A. Meri, “Using energy efficient security technique to protect live virtual machine migration in cloud computing infrastructure,” *Journal of Engineering Science and Technology*, vol. 16, pp. 2629–2651, 2021.

[40] T. Zeb, A. Ghafoor, A. Shibli and M. Yousaf, “A secure architecture for inter-cloud virtual machine migration,” in *Int. Conf. on Security and Privacy in Communication Networks*, Beijing, China, pp. 24–35, 2014.

[41] V. A. W. M. Gligor, “Requirements for root of trust establishment,” in *Cambridge Int. Workshop on Security Protocols*, Cambridge, UK, pp. 192–202, 2018.

[42] A. Tomlinson, “Introduction to the TPM,” *Smart Cards, Tokens, Security and Applications*, pp. 173–191, 2017.

[43] T. Pulli, “CI/CD pipeline for SSO service,” M.S. dissertation, Aalto University Learning Centre, Finland, pp. 1–47, 2021.

[44] P. Pandey and T. N. Nisha, “Challenges in single sign-on,” *Journal of Physics: Conference Series*, vol. 1964, pp. 42016, 2021.

[45] J.-M. Belmont, “Hands-on continuous integration and delivery: Build and release quality software at scale with Jenkins, Travis CI, and CircleCI,” *Packt Publishing Ltd*, vol. 1, pp. 27–84, 2018.