Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.028612

ARTICLE

Enhancing the Trustworthiness of 6G Based on Trusted Multi-Cloud Infrastructure: A Practice of Cryptography Approach

Mingxing Zhou[1,2], Peng Xiao[3], Qixu Wang[1,2,*], Shuhua Ruan[1,2], Xingshu Chen[1,2] and Menglong Yang[4]

[1]School of Cyber Science and Engineering, Sichuan University, Chengdu, 610207, China

[2]Cyber Science Research Institute, Sichuan University, Chengdu, 610207, China

[3]Information Center, China Southern Power Grid Yunnan Power Grid Co., Ltd., Kunming, 650000, China

[4]School of Aeronautics and Astronautics, Sichuan University, Chengdu, 610065, China

*Corresponding Author: Qixu Wang. Email: qixuwang@scu.edu.cn

Received: 29 December 2022 Accepted: 05 May 2023 Published: 22 September 2023

ABSTRACT

Due to the need for massive device connectivity, low communication latency, and various customizations in 6G architecture, a distributed cloud deployment approach will be more relevant to the space-air-ground-sea integrated network scenario. However, the openness and heterogeneity of the 6G network cause the problems of network security. To improve the trustworthiness of 6G networks, we propose a trusted computing-based approach for establishing trust relationships in multi-cloud scenarios. The proposed method shows the relationship of trust based on dual-level verification. It separates the trustworthy states of multiple complex cloud units in 6G architecture into the state within and between cloud units. Firstly, SM3 algorithm establishes the chain of trust for the system's trusted boot phase. Then, the remote attestation server (RAS) of distributed cloud units verifies the physical servers. Meanwhile, the physical servers use a ring approach to verify the cloud servers. Eventually, the centralized RAS takes one-time authentication to the critical evidence information of distributed cloud unit servers. Simultaneously, the centralized RAS also verifies the evidence of distributed RAS. We establish our proposed approach in a natural OpenStack-based cloud environment. The simulation results show that the proposed method achieves higher security with less than a 1% system performance loss.

KEYWORDS

6G; multi-cloud; trusted Infrastructure; remote attestation; commercial cipher

1 Introduction

With the standardization of the 5G standard and its commercial application on a global scale starting in 2019, 6G has quickly become a research hotspot [1–3]. The real physical and virtual digital worlds will be connected by 6G in the future, which will significantly impact our lives [4]. In order to cover the global space, air, ground, and sea scenarios, the 6G network architecture must have the characteristics of distributed autonomy. The architecture of the distributed cloud computing platform can better manage homogeneous or heterogeneous networks [5,6].

Although cloud computing technology is developing and becoming more sophisticated, its security issues still need to be considered [7–9]. Nalini used a security tree to illustrate the significance

of cloud security and summarize security challenges at three levels: virtualization, applications, and networks [10]. In the future, 6G technology will tightly integrate networks and computing [11], necessitating the realization of unified hosting and the use of infrastructure resources. Virtualization technology will be of great technical value [12]. However, its security risks, such as virtual network vulnerabilities, virtual machine vulnerabilities, and hypervisor vulnerabilities [13], can pose deadly threats. At present, research on 6G security mainly covers physical layer security, network slicing security, platform security, and artificial intelligence security [14]. Ismaeel integrates zero trust architecture into the 6G network supporting digital twins, achieving the purpose of protecting data, equipment and users [15]. Frehat proposed a mitigation method for adversarial attacks against 6G machine learning model [16,17]. Security for 6G virtualization has received little research to date. 6G is considered promising for providing deep learning to aid in the virtualization of security functions [18]. DTCPN focuses on solving the complex and inaccurate modeling processes of existing network platforms and monitoring the security of virtual network resources [19]. The software-based RINA solution ensures high-performance connectivity and meets isolation requirements for virtual network functions [20]. A delay-aware dual hypervisor placement and control path approach enables the virtualization layer to adapt to sudden load changes [21].

The trend involves multiple cloud service providers (CSPs) participating due to the large layout and extensive business capabilities of the 6G network architecture, which will involve how trust relationships are established between various clouds. Kurdi et al. proposed a lightweight trust management algorithm based on subjective logic, building mutual trust relationships based on the system ratings received by CSPs for their past behavior and the ratings given by other CSPs [22]. For the lack of trust management models for cross-cloud federation scenarios, Ahmed et al. proposed a cloud-to-cloud trust paradigm based on trust bidirectionality, trust portfolio, delegated control, and resource awareness [23]. The fog-based hierarchical trust mechanism Wang et al. proposed [24] considers the trust in the underlying architecture and the trust in CSPs and SSPs (sensor service providers). In the literature [25], dynamic interactions between cloud tenants and CSPs, and QoS (quality of service) parameters of cloud service have also been used to establish trust. All of these trust models aim to show users that CSPs are reliable, but they all call for the selection of metrics, which has the drawback that it is challenging to find metrics that accurately reflect the trust relationship.

A strong foundation of trust for 6G network can be created by establishing a secure and reliable cloud infrastructure based on trusted computing technology. Hardware-based security, the comprehensive approach of trusted computing and secure areas, will become the cornerstone of future computing networks, as Dr. Mikael Prytz of Ericsson believes. Smart Habitat defines several protection levels for 5G and 6G, such as multi-level isolation and protection of the integrity of SDN and VNF components, and recommends using trusted hardware environments [26]. According to the 6G white paper [27], the integrity of remote platforms, including operating systems, virtual machines, and services, can be addressed by tying critical platform operations to hardware TPM. In addition to designing a remote proof approach for 5G core services and implementing trust and remote proof in the cloud and mobile infrastructures, Oliver also summarized a proof of concept for the plan in the healthcare industry [28]. All of the above studies point out, at a theoretical level, that applying trusted computing-related technologies to 6G networks can improve security based on the fact that remote proofs can address the integrity of the platform. However, there is a gap in establishing trust relationships between multiple cloud environments.

The risks of distributed cloud deployment in 6G network are shown in Fig. 1. It consists of many small distributed cloud units, including distributed cloud units (DCU) and a centralized cloud unit (CCU). The lack of trust between cloud units in this architecture and the security risks associated

with virtualization will inevitably prevent its widespread use. Motivated by the research, we propose a trust construction method for multi-cloud scenarios in 6G network architecture. The trustworthy state of multiple cloud units is divided into the trust within the cloud unit and the trust between cloud units to achieve dual-level verification and solve the weak trust problem among them. This method can detect system components whose integrity is compromised in time, including attacks introduced through virtualization threats. The work is based on our previous work [29], but the previous work only focused on extending the cryptographic algorithm to the vTPM so that the virtual machine could use SM2/SM3/SM4 algorithm through the vTPM. Firstly, we enabled the PCR bank for SM3 algorithm and modified the BIOS and kernel source code to use SM3 algorithm in the trusted boot phase. Then the centralized RAS (the remote attestation server of CCU) verifies the distributed RAS (the remote attestation server of DCU), the distributed RAS verifies the physical server (PS), and the PS verifies the integrity of the cloud servers circularly. Finally, critical evidence information of all servers is verified by the centralized RAS at one time. In summary, this paper makes the following contributions:

- We propose a method for building trust between cloud units. This method leaves the verification of servers in DCU to the centralized RAS, the distributed RAS, and PS, and the centralized RAS verifies key evidence for all servers at one time, greatly reducing the verification burden on the centralized RAS.
- A method of applying commercial cipher SM2/SM3/SM4 algorithms to vTPM is proposed. Extended the cryptographic algorithms supported by vTPM to enable virtual machines to use them. The chain of trust based on SM3 algorithm can be established by modifying the BIOS and IMA (Integrity Measurement Architecture).
- We conducted related experiments on a real OpenStack cloud platform, showing that the proposed approach has better effectiveness and efficiency. Only less than 1% system performance loss of the host is impacted when the chain of trust is created using SM3 algorithm. The computation time for the additional step is less than 3 seconds, even when validating a cloud unit with 10,000 server sizes.

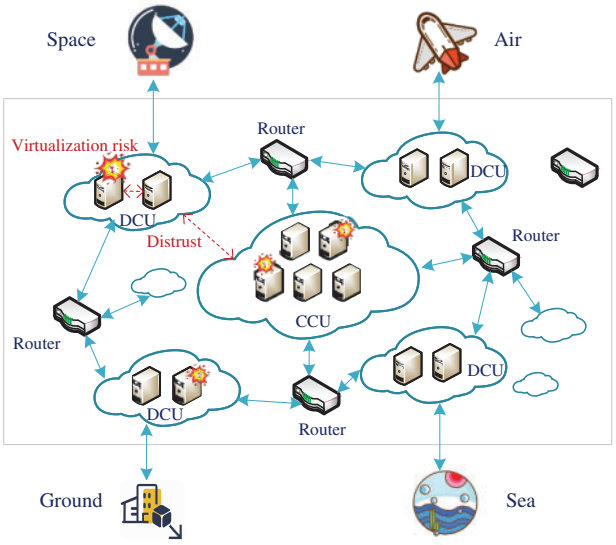

Figure 1: Risks of distributed cloud deployment in 6G network

The rest of this paper is organized as follows. Section 2 introduces the related work of cloud-to-cloud trust, trusted computing technology, and the 6G security scheme. We overview the system architecture, assumptions, and design requirements in Section 3. Section 4 introduces the design and implementation of each part of the proposed scheme. The experimental results and performance evaluation are given in Section 5. Finally, we conclude the work in Section 6.

2 Related Work

This section summarizes and reviews related work on cloud trust assessment, trusted computing technology, and 6G security solutions.

2.1 Trust Evaluation for Cloud

At present, the schemes for establishing or improving the trust of a single cloud can be mainly divided into three categories: (1) improving the credibility based on anomaly detection of the cloud environment [30,31]; (2) based on reputation, quality of service (QoS), feedback rating and other indicators to establish trust evaluation methods [32,33]; (3) building a trusted cloud based on trusted computing technology [34,35]. Solutions related to anomaly detection mainly detect security threats in the cloud environment promptly by monitoring system performance data, behavioral feature data, and training models through machine learning and deep learning. The trust evaluation method mainly provides cloud users with an evaluation model method for CSP. Such methods, such as literature [34], evaluate the credibility of cloud services based on quality feedback ratings and cloud-specific security indicators. The related method based on trusted computing combines trusted computing with cloud computing technology, which can effectively verify the behavior of the cloud and prove the trust relationship.

As for the trust relationship between clouds and clouds, Kurdi et al. [22] proposed the InterTrust scheme to improve trust in interconnected clouds. The approach builds mutual trust based on system ratings of CSPs' past behavior and ratings given by other CSPs, which is used to address the interconnected cloud computing paradigm in which multiple CSPs participate. Aiming at the lack of a trust management model in cross-cloud joint scenarios, Ahmed et al. [23] identified trust bidirectionality, trust composition, delegated control, and resource awareness as the theoretical principles that constitute the interconnected cloud computing paradigm based on a large number of literature surveys. Wang et al. [24] proposed a fog-based layered trust mechanism that considers the trust of the underlying architecture and the trust of CSPs and SSPs, focusing on the real-time comparison of service parameters, collection of behavior monitoring exception information, and quantitative evaluation of entities. Other related studies are basically based on the dynamic interaction between cloud users and cloud service providers in, QoS, and other parameters to establish trust [25]. These trust models need to use the index characteristics of the system and cloud services as model parameters, but the defect is that it is challenging to propose indicators that can fully describe the trust relationship.

2.2 Trusted Computing Technology

Trusted computing technology [36] has the characteristics of measurement, storage, and reporting, which provide security functions such as trusted boot, remote attestation, integrity checking, and encryption and decryption. From the time of system power-on to the running process, the trusted boot [37] measures the components and loads them in a specific order. Extending the measurement results into the PCR of the TPM builds a chain of trust from the root of trust to system applications [38].

For the Linux operating system, the trust chain construction process is measured by BIOS and IMA based on the SHA1 algorithm, but the SHA1 algorithm has been proven not to have strong collision resistance [39]. Although the SHA256 algorithm can be used for measurement, SM3 algorithm is more secure and effective against boomerang attacks [40].

Remote attestation techniques declare the properties of the target by providing evidence to the evaluator, which can be used to verify the secure state of a viable execution environment. In recent years, many scholars have focused on solving the application in the Internet of Things, and some have researched the efficiency of the remote attestation process [41–44]. Some representative remote attestation tools, such as OpenCIT, OAT, OpenPTS, and Keylime. provide attestation frameworks to implement request initiation, evidence return, and integrity verification. Existing verification tools can work well for a single cloud computing environment but will face more significant challenges once applied to scenarios with complex cloud environments such as 6G.

2.3 Security Solution for 6G

In order to promote the development of next-generation wireless communication networks, the potential security challenges of 6G are studied to provide valuable security considerations for 6G standardization work [45,46]. Domestic and foreign researchers and related institutions propose to enhance the security and privacy of 6G in terms of physical layer security, network slicing security, platform security, and artificial intelligence security, which can utilize distributed ledgers, quantum computing, platform monitoring and detection, identity authentication, privacy protection technology [47,48]. Although using these techniques can improve security, their essence is to secure certain parts of the architecture. Trusted computing technology can establish the security cornerstone of the basic environment. When combined with other security protection technologies, it will perform better in improving the security of the 6G architecture.

Therefore, Smart Habitat [26] defines three protection solutions for the security problems in 5G/6G networks and outlines the security solutions in detail. It is proposed to protect the integrity of hypervisors, virtual machines, operating systems, controllers and containers for building network infrastructure, which can apply secure boot devices and feasible execution environment technologies. The white paper [27] summarized the research challenges of 6G in terms of trust, security and privacy. For example, for the security of telecom cloud convergence, the integrity of the computing domain of virtual services that may change dynamically needs to be solved, which can provide verifiable remote proof in some cases, combining platform operations with tamper-resistant hardware TPM to address remote platform integrity issues. Digitization of the medical and railway sectors through cloud computing and network technologies such as 5G or 6G brings additional security challenges. Authentication and integrity of devices, services, and other functional components need to be addressed, and the introduction of trusted computing technology can solve this problem based on remote attestation [28].

3 Framework and Requirements

In this section, we present our approach's framework, assumptions and design requirements.

3.1 Framework Overview

The proposed method aims to establish the trust relationship within and between the cloud units for the 6G network architecture. Fig. 2 shows the overall architecture of this method, which divides the trust relationship into two levels trust within cloud units and trust between cloud units.

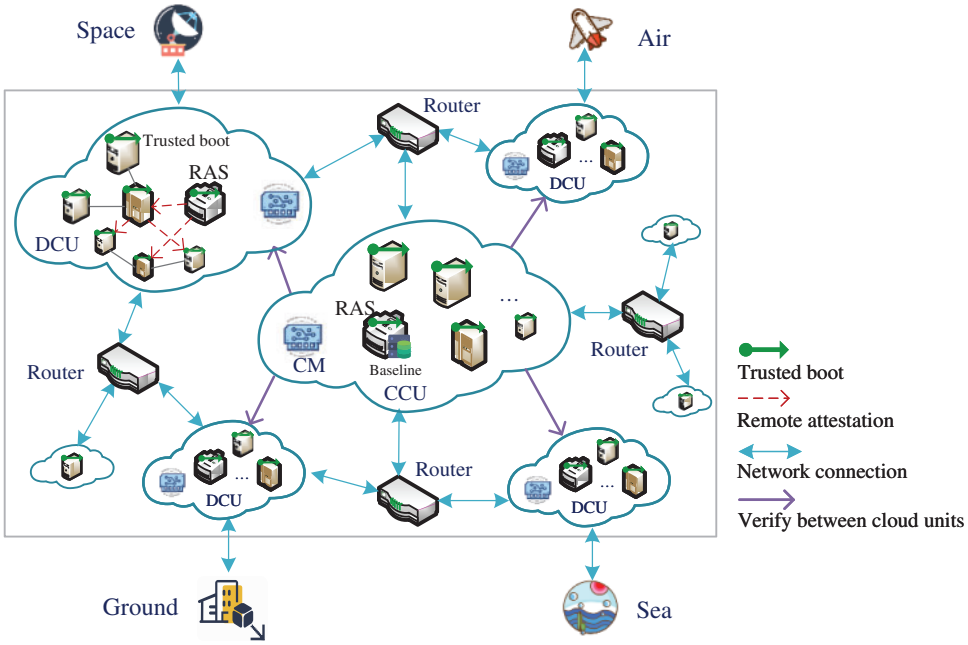

Figure 2: Architecture overview

1) Internal trust in the cloud unit

The physical host in cloud unit is configured with a hardware TPM that supports SM3 algorithm. The virtual machine is configured with a vTPM that supports the SM3 algorithm, whose integrity is protected. From power-on to running, the system completes the trusted boot based on the SM3 algorithm, establishing the server's trust. At the same time, a remote attestation server is set in each distributed cloud unit to verify the integrity status of other servers.

2) Trust between cloud units

The 6G network involves multiple cloud units cooperating, so it is necessary to ensure the authenticity and credibility of the cloud units. The communication module (CM) is used to establish the connection between cloud units, and the centralized RAS maintains a reference value library to store the key evidence of each DCU. The centralized RAS initiates two remote certifications to the remaining distributed cloud nodes, first verifying the integrity of the distributed RAS and then verifying the integrity of the remaining servers.

What's more, a baseline value library (Baseline) that stores key evidence information of all DCU units is maintained in the centralized RAS. Considering the huge scale of servers in multi-cloud architecture of 6G, the key evidence information refers to the boot_aggregate value of each server in the proposed approach. The boot_aggregate is a cumulative hash over TPM registers 0 to 7.

As shown in Fig. 3, the process of establishing the trust relationship between the multi-cloud environments of the 6G network can be decomposed into the following steps:

1) Trusted boot: All servers measure the system's critical components based on trusted boot and extend the results into the PCR registers of the TPM/vTPM.

2) Verification of the distributed RAS: The centralized RAS initiates a certification request to the distributed RAS of the cloud unit through the communication module and performs an integrity verification based on the current IMA measurement log and PCR register information of the distributed RAS.

3) Internal verification of the cloud unit: The distributed RAS initiates a certification request to all physical servers (PS) and compares and verifies the evidence information with the benchmark value database. The PS verifies the cloud server (S) of another PS according to the rules.

4) One-time verification of key evidence: The centralized RAS obtains the key evidence information of all servers in the current cloud node through the communication module and completes the integrity verification of all servers at one time.

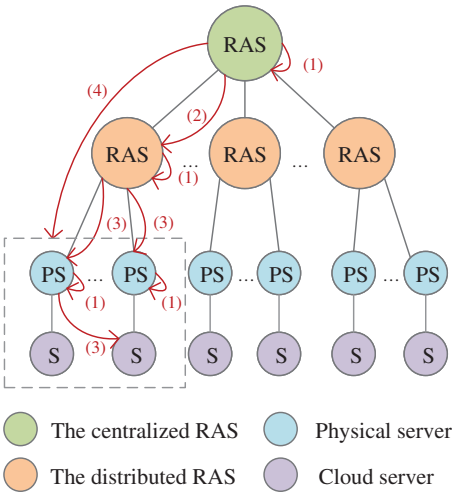

Figure 3: The process of establishing trust relationship between multiple clouds

3.2 Assumptions

Our approach is developed with the following assumptions:

1) Cloud servers are all equipped with hardware TPM or vTPM, and an attacker can't destroy the server physically.

2) The virtual machine images and vTPM instances may be tampered with before starting, such as adding malware, Trojan backdoor, etc., to them.

3) The remote attestation server in CCU guarantees integrity based on trusted boot, and the centralized RAS is always authentic and credible.

3.3 Design Goals

We aim to achieve the following goals for the multi-cloud environment in the 6G network architecture.

1) Practical goals

Based on the large scale, low communication delay, and diversified customization of 6G networks, our goal is to provide a practical method for establishing trust relationships to ensure the authenticity

of all cloud units. The proposed method needs to have the characteristics of flexible deployment and bring as little performance overhead as possible.

2) Security goals

Our security goal is to guarantee the integrity of each cloud unit's infrastructure and verify the trusted state of each distributed cloud unit, which is resistant to impersonation and sybil attacks.

Resistance to impersonation attack: When using the proposed method, attackers cannot masquerade new distributed cloud units into the architecture.

Resistance to sybil attack: The attacker cannot hijack the server through the vulnerability of the virtualization layer. Once the attacker hijacks the server and pretends to be a legitimate user, the integrity verification of the server will inevitably fail.

4 The Proposed Approach

In this section, we present the critical techniques of the proposed method. Our approach can be divided into three main components: (1) Establishment of trust relationship between cloud units; (2) Trusted boot within cloud units; (3) Expansion of virtual trusted platform modules.

4.1 Chain of Trust for Multi-Cloud

A single cloud unit contains many servers, so the cloud unit can be completely trusted only when all servers are trusted. For the trust relationship between multiple cloud units, it is necessary to ensure that the cloud unit itself is trustworthy and that the CCU passes the integrity verification of its distributed cloud unit (DCU). For convenience of expression, CRAS denotes the centralized RAS and DRAS denotes the distributed RAS. In this section, multi-cloud trust is divided into the trust within and between cloud units, and the weak trust problem is solved through dual-level verification.

Definition 1: The trust relationship of multiple cloud units means that the status of all servers in the cloud unit is trusted, and other clouds successfully verify its integrity. That is to say, the trust relationship of multiple cloud units is divided into cloud unit internal trust and cloud unit trust. The trust relationship of multiple clouds is recorded as C_{multi}, the internal trust relationship of cloud units is respectively recorded as C_{CCU} (centralized cloud unit) and C_{DCU} (distributed cloud unit), and the trust relationship between cloud units (that is, the trust of CCU to DCU) is recorded as $C_{C\to D}$. The trust relationship model is

$$\mathrm{C_{multi}} = (\mathrm{C_{CCU}}, \mathrm{C_{DCU}}, \mathrm{C_{C\to D}}) \tag{1}$$

$$\mathrm{C_{CCU}} = (\mathrm{C_{CRAS}}, \mathrm{C_{server}}) \tag{2}$$

$$\mathrm{C_{DCU}} = (\mathrm{C_{DRAS}}, \mathrm{C_{server}}) \tag{3}$$

$$\mathrm{C_{C\to D}} = (\mathrm{CRAS} \overset{\mathrm{RA}}{\to} \mathrm{DRAS}, \mathrm{CRAS} \overset{\mathrm{RA}}{\to} \mathrm{servers}) \tag{4}$$

Among them, the trust chain of the server (such as C_{CRAS}, C_{DRAS} and C_{server}) is established based on trusted boot, and the trust chain of CCU to DCU is established through remote certification.

1) Internal trust in the cloud unit

All servers in the cloud unit establish their trust chain based on a trusted boot and then establish the trust relationship of the entire cloud unit through remote certification. The process of establishing the internal trust relationship of the cloud unit is shown in Fig. 4. The distributed RAS verifies the integrity of all physical servers, and the physical servers then verify the integrity of the virtual machines. Considering that if an attacker hijacks the physical server (host), the verification results of the PS on all the above cloud servers are not credible. Therefore, we designed a ring verification method, where the physical server verifies the cloud server on the next physical server in sequence. The validation model is as follows:

$$\mathrm{DRAS} \xrightarrow{\mathrm{RA}} \mathrm{PS_i}\ (1 \le \mathrm{i} \le \mathrm{n}) \tag{5}$$

$$\mathrm{PS_i} \xrightarrow{\mathrm{RA}} \mathrm{S_{PS_{(i+1)\ mod\ n}}}\ (1 \le \mathrm{i} \le \mathrm{n}) \tag{6}$$

PS_i refers to the ith physical server (host) of the cloud unit, and $S_{PS_{(i+1)\ mod\ n}}$ refers to all the cloud servers on the (i+1) mod n host.

This method requires that the reference values of all PS_i cloud servers be stored in the $PS_{(i+1)\ mod\ n}$, and the corresponding reference values must be updated in time when operations such as VM migration, creation, and destruction occur. The advantage is that it can reduce the verification burden of the RAS and simultaneously avoid the virtual machine's untrustworthy verification result caused by the hijacking of the host machine (referring to PS). Only by hijacking all the hosts in the cloud unit can the attacker avoid the detection of the attack, which is more difficult.

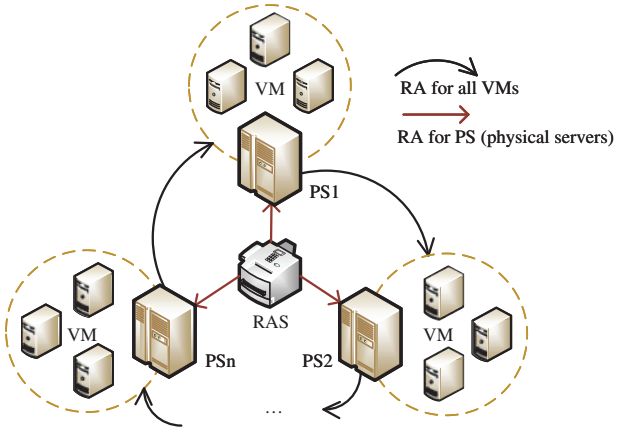

Figure 4: The establishment of trust relationship within the cloud

2) Trust between cloud units

In order to avoid the significant performance overhead required by the centralized RAS to verify all servers in DCU, we designed the trust relationship establishment method between cloud units, as shown in Fig. 5. The distributed RAS in the DCU is responsible for collecting the key evidence of the server in the unit and sending it to the benchmark library (baseline) of the centralized RAS. Here we use the boot_aggregate value of the virtual machine as the key evidence. Boot_aggregate indicates the result of aggregation of key component measurements from BIOS to kernel during the trusted boot process of the server. The centralized RAS maintains a baseline with DCU flags and corresponding

key evidence. The communication module (CM) is responsible for establishing the communication connection between the two clouds and undertakes the task of transmitting data.

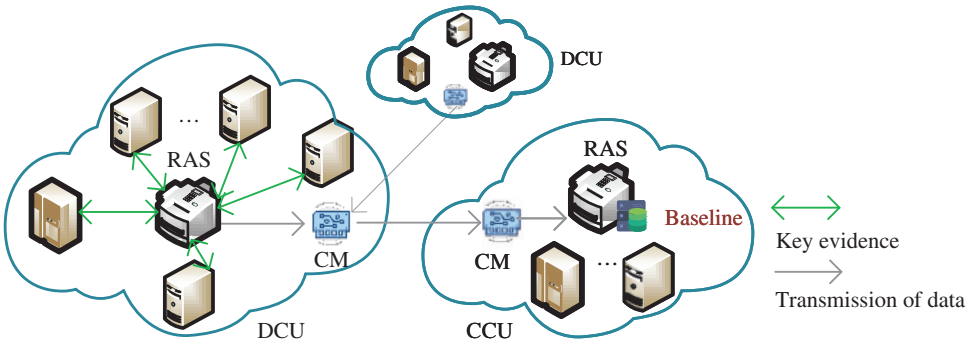

Figure 5: Establishment of trust relationship between cloud units

When the CCU verifies the state of the DCU, it first verifies the integrity of the distributed RAS server. Then request the distributed RAS to send the key evidence information of the rest of the servers, calculate the aggregation result by the method such as formula (7), and send badcu to the centralized RAS. The centralized RAS reproduces the aggregation process based on the evidence information stored in the baseline and compares and verifies the status of the DCU.

$$\begin{cases} ba_{old} = ba_{cur}\ (1 \le i \le m-1) \\ ba_{cur} = SM3(ba_{old} \parallel ba_i)\ (1 \le i \le m) \\ ba_{DCU} = ba_{cur}\ (i = m) \end{cases} \tag{7}$$

where there are m servers in total, ba_{cur} represents the result after the ba_i (boot_aggregate) of the evidence information of the i server is aggregated, and ba_{DCU} represents the result of all aggregation the evidence information of DCU. $\parallel$ means to concatenate two strings.

Algorithm 1: Trusted state between clouds

Input: ba_i, PCR10, ima.log
Output: trusted_state

```
trusted_state = false && t1 = RA(the distributed RAS)
while i in (0, n) do
  t2 += RA(PS_i)
end while
while j in (0, m) do
  t2 += RA(server_j)
end while
if (t1 + t2) > 0 then // 0 represents success, 1 means failure
  return trusted_state
end if
while i in (0, m) do
  ba_DCU = SM3(ba_DCU||ba_i)
end while
```

(Continued)

Algorithm 1 (continued)

14: if $ba_{CCU} == ba_{DCU}$ then
15: t4 = 0
16: end if
17: if t1 == 0 && t4 == 0 then
18: trusted_state=true
19: end if
20: **return** trusted_state

The complete process of establishing a trust relationship between the CCU and the DCU is shown in Algorithm 1. This includes the internal trust of the DCU and the verification of the CCU to the DCU: (1) Firstly, the CCU initiates a verification request to the target DCU, and the centralized RAS verifies the integrity of the distributed RAS. (2) When the verification is passed, the distributed RAS verifies the integrity of all internal PSs, and the PS verifies the integrity of the virtual machine. (3) Then the distributed RAS collects key evidence from other servers to calculate ba_{DCU}, and the centralized RAS calculates based on the evidence information of the baseline, and compares it with the received ba_{DCU} to verify whether it passes. (4) Judge whether the trust relationship can be established according to the results of each step.

4.2 Trusted Boot of OS in Cloud

In the trusted boot phase, we take the TPM or vTPM as the trust starting point. The SHA1 algorithm is used to measure the application in the order of CRTM, BIOS, Grub, and OS, and then the IMA subsystem of OS measures the application based on the default SHA1 algorithm.

Definition 2: The critical components at each stage of the trusted startup process are marked as entity E, and the trust relationship of entities is marked as T. Starting with entity E_1 as the source of trust establishment, the key components are measured and verified in the sequence of installation and startup. If the entity E_i passes the verification of E_{i+1}, the trust is passed to E_{i+1}. When all the essential components E_i in the startup phase are verified as credible, the startup process is credible. The trust relationship transfer model is as follows:

$$\mathrm{T_{E_i}} \rightarrow \mathrm{T_{E_{i+1}}} (1 \leq \mathrm{i} \leq \mathrm{n}) \tag{8}$$

Definition 3: The trusted boot process of a physical server takes the hardware TPM as the starting point of the trust relationship. After the startup is completed, only the hypervisor and other applications closely related to the virtual machine will run. The trust model is

$$\mathrm{T_{E_{TPM}}} \rightarrow \mathrm{T_{E_{BIOS}}} \rightarrow \mathrm{T_{E_{Grub}}} \rightarrow \mathrm{T_{E_{OS}}} \rightarrow \mathrm{T_{E_{VM\_envi}}} \tag{9}$$

where E_{VM_envi} represents critical applications such as hypervisor, vTPM software, and vBIOS program. The integrity of the E_{VM_envi} component is one of the prerequisites for the trustworthiness of the virtual machine.

Definition 4: The trusted boot process of a cloud server (virtual machine) takes vTPM as the starting point of trust, and the trust model is as follows:

$$\mathrm{T_{E_{vTPM}}} \rightarrow \mathrm{T_{E_{vBIOS}}} \rightarrow \mathrm{T_{E_{Grub}}} \rightarrow \mathrm{T_{E_{OS}}} \rightarrow \mathrm{T_{E_{APP}}} \tag{10}$$

Definition 5: Since the trust starting point of the virtual machine is vTPM, which does not have the characteristics that the hardware is difficult to be tampered with, we design an instance security module (ISM) to verify the integrity of the vTPM. The trust model is as follows:

$$T_{E_{TPM}} \rightarrow T_{E_{ISM}} \rightarrow T_{E_{vTPM}} \tag{11}$$

where E_{ISM} represents the security management module (ISM) in the cloud environment, its function is to protect the integrity of the vTPM instance.

For the chain of trust in a single cloud, we use SM3 algorithm to measure the critical components in the trusted boot process, maintain the trust chain of the host machine and virtual machine, respectively, based on the layered idea, and establish the trust transfer relationship between them. As shown in Fig. 6, this method enables SM3 PCR bank of the vTPM, which makes it possible to store the hash value during the trusted measurement process into SM3 PCR bank. Then the BIOS program of the virtual machine and the measurement algorithm used by Kernel IMA are modified to SM3, which involves the BIOS extension module, IMA extension module and instance security management.

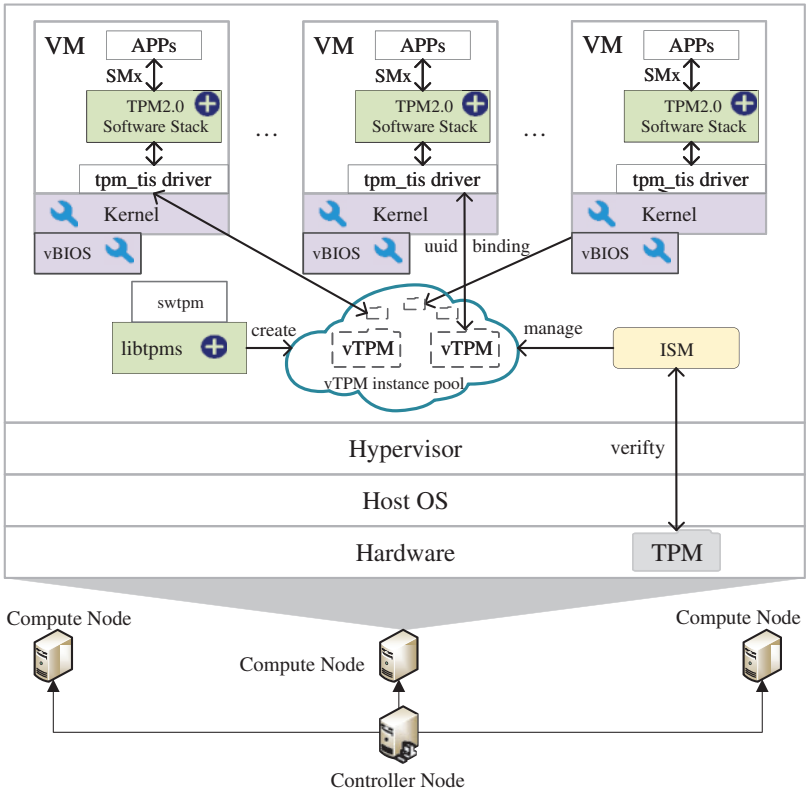

Figure 6: Trusted boot chain of trust in a cloud unit

Algorithm 2: BIOS extension for SM3 measurement

Input: VM startup command

Output: bios_measurements // bios measurement log

1: u8 digest[sm3_bufsize]

(Continued)

Algorithm 2 (continued)

```
tpm_startup()
tpm_option_rom()
sm3((const u8 *)addr, len, pcctes.digest);
while 1 do
  u8 hashalg_flag = TPM2_ALG_SM3_256_FLAG
  if (suppt_banks & (1<<(flagnum - 1))) then
    break;
  end if
end while
sm3(hashdata, hashdata_length, hash)
 memcpy(v→hash, sm3, max(hsize, sm3_bufsize)
tpm20_extend()
return bios_measurements
```

1) BIOS extension module

The BIOS extension module needs to fulfil the following objectives: (i) Supporting SM3 algorithm. (ii) The BIOS trusted measurement function is based on SM3 algorithm. (iii) The trusted measurement results were extended to SM3 PCR bank. Algorithm 2 demonstrates the process of the BIOS expansion module.

2) IMA extension module

IMA is the integrity subsystem in kernel. When the operating system starts, the file integrity is measured according to the measurement policy, and the measurement results are recorded in the log file ima.log. The IMA extension module needs to meet the following requirements: (i) Kernel support for SM3 algorithm, (ii) trust measurement process of IMA using SM3 algorithm, and (iii) measurement results extended to SM3 PCR bank. The critical function of the IMA extension module is shown in Algorithm 3.

Algorithm 3: IMA extension for SM3 measurement

Input: OS boot instructions
Output: ascii_runtime_measurements // ima measurement log

```
#define IMA_DIGEST_SIZE sm3_digest_size
ima_hash_algo = HASH_ALGO_SM3_256
tpm_buf_append_u16(&buf, TPM2_ALG_SM3_256)
memcpy(res_buf, out→digest, sm3_digest_size)
memcpy(digest_list[i].digest, hash, TPM_DIGEST_SIZE)
tpm2_pcr_extend()
return ascii_runtime_measurements
```

3) Instance security management

The vTPM instance and virtual machine image on the host are at risk of being tampered with. Although the related method [49] proposed to store the hash value of the instance in the hardware TPM by means of a measurement list, so as to maintain the trust relationship between the TPM and the vTPM. However, this method does not take into account that in actual scenarios, multiple cloud servers running on the same physical machine may need to be turned on and off frequently, which causes great

difficulties in the maintenance of the measurement list. Therefore, we designed the instance security management (ISM) module to manage instances and image files.

The flow of the instance security management module is shown in Fig. 7, and its purpose is to protect the integrity of the vTPM and image files. Before starting the virtual machine, first use the hardware TPM to verify the integrity of the ISM, and here we store the integrity measurement result of the ISM in PCR 11 of the TPM. Then verify the integrity of the virtual machine image file and vTPM instance through the ISM module, and the virtual machine can start only after the comparison with the basevalue is verified. When the virtual machine is running, the vTPM is in the occupied state, so there is no need to consider the instance being tampered with at this time. After shutdown, ISM measures the virtual machine image file and vTPM instance and updates the stored basevalue results. Since the size of the virtual machine image file is measured in GB, the hash operation is time-consuming. Therefore, we cut the image file into multiple small files of equal size and then perform a hash operation, and finally perform a hash operation on the hash values of all the small files.

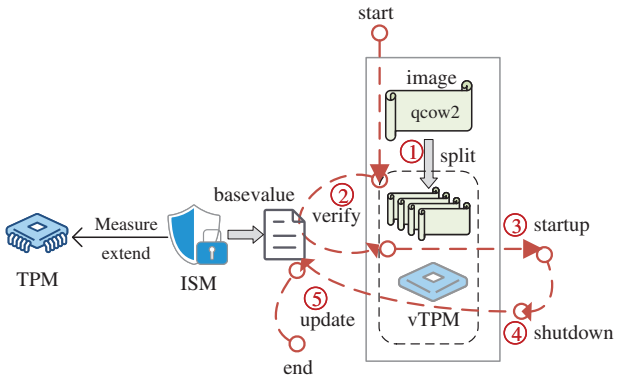

Figure 7: Instance security management module

4.3 Algorithms Extension in vTPM

The SM2/SM3/SM4 series algorithm (referring to SMx) proposed by China has been incorporated into the international standard ISO/IEC. They are improved on the basis of ECC/SHA-256/AES-128, which can resist universal key replacement attacks, boomerang attacks, and key leaking Trojans [40,50,51], respectively. Currently, vTPM does not support the SMx algorithm, so neither trusted boot nor encryption services could use the more secure SMx algorithm.

Considering the security of the algorithm used in the measurement process, we use SM3 algorithm to complete the measurement operation in trusted boot. However, vTPM does not have the PCR bank of SM3 algorithm for the time being, and the rest of the encryption algorithms have certain defects. Therefore, we extended the SMx algorithm to the vTPM and enabled SM3 PCR bank.

We transform the vTPM function library libtpms of the software implementation and the TPM2.0 software stack used inside the trusted virtual machine. Fig. 8 shows the vTPM extend method and depicts the interaction of the qemu virtual machine process with the vTPM instance through the software stack. We added SM2/SM4 call module and response result processing module to the vTPM library (libtpms) and the SMx support module to TPM2.0 software stack.

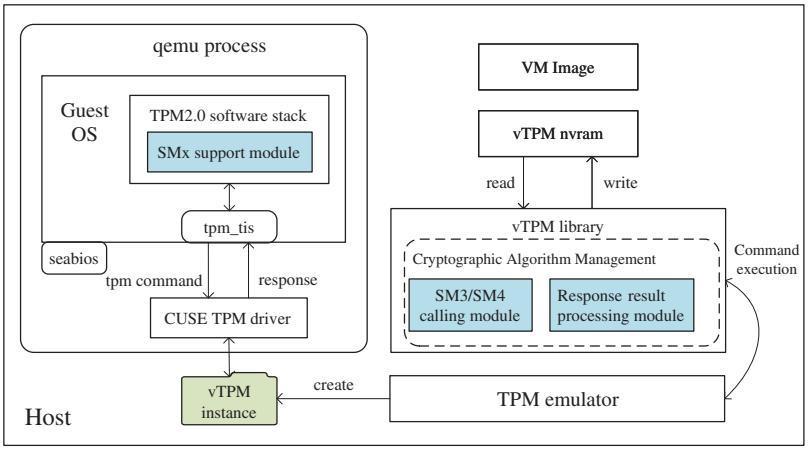

Figure 8: The vTPM extend method

1) SM3/SM4 calling module

This module adds the registration and definition of SM3/SM4 algorithm data structure to the function library. Towards SM3 algorithm, data structure SM3_256_Def is added, and functions for processing messages are defined, including sm3_init, sm3_update, sm3_final, memcpy, data length and algorithm identifier TPM_ALG_SM3_256. For SM4 algorithm, the data structure is added to the symmetric encryption algorithm selector SELECT, and functions such as SM4_encrypt, SM4_decrypt, and SM4_KEY, the encryption key setting function TpmCryptSetEncryptKeySM4, and the decryption key setting function TpmCryptSetDecryptKeySM4 are defined.

2) Response result processing module

This module is used to process the result of SM3/SM4 algorithm operations. For SM3 algorithm, add the encoding interface tpmHashStateSM3_256_Marshal, which is designed to load SM3 message bytes and sort them, splicing the message streams into strings, and writing them into SM3 operation result data structure. For SM4 algorithm, add the key encoding interface TPMI_SM4_KEY_BITS_Marshal to encode the key type and key value.

3) SMx support module

Modify the FAPI interface in the tpm2-tss software stack, and add the calling options of the SM2/SM3/SM4 algorithm according to the specification in the function interface involving signature, hash and symmetric encryption. Besides, added calling options towards SM2/SM3/SM4 algorithms for ECC, hash, and symmetric encryption algorithms in tpm2-tools, respectively.

The timing diagram of invoking SMx algorithm based on virtual trusted computing technology is shown in Fig. 9. vTPM provides SM2 signature and signature verification services as an example to introduce the execution process:

1) Generate an algorithm key pair. The VM requests to generate a SM2 algorithm key pair, and the TPM2.0 software stack generates a public-private key pair. When the VM requests to load the public-private key pair, the software stack loads the key into the vTPM.

2) Issue a signature or signature verification request. When the VM issues a SM2 algorithm signature or verification request, the software stack processes the request, and uses the set_key_algorithm function to set the ECC algorithm interface to use the SM2 algorithm. The corresponding API

component in the software stack sends the TPM command stream to the vTPM, and the algorithm identification is extracted by the asymmetric cryptographic algorithm module of the vTPM.

3) Perform a signature or verification operation. The ECC algorithm interface parses the TPM command stream and obtains the signature frame SM2 identifier. For the signature request, the signature function CryptEccSign calls the BnSignEcSm2 function to sign. For the signature verification request, the signature verification function CryptEccValidateSignature calls the BnValidateSignatureEcSm2 function for verification.

4) Return the signature or verification result. The signature or signature verification result is encoded by the TPMS_SIG_SCHEME_SM2_Marshal and TPMS_SIGNATURE_SM2_Marshal interfaces. The obtained command response stream is sent to the software stack through the tpm_tis driver, and the software stack parses the result and returns it to the VM.

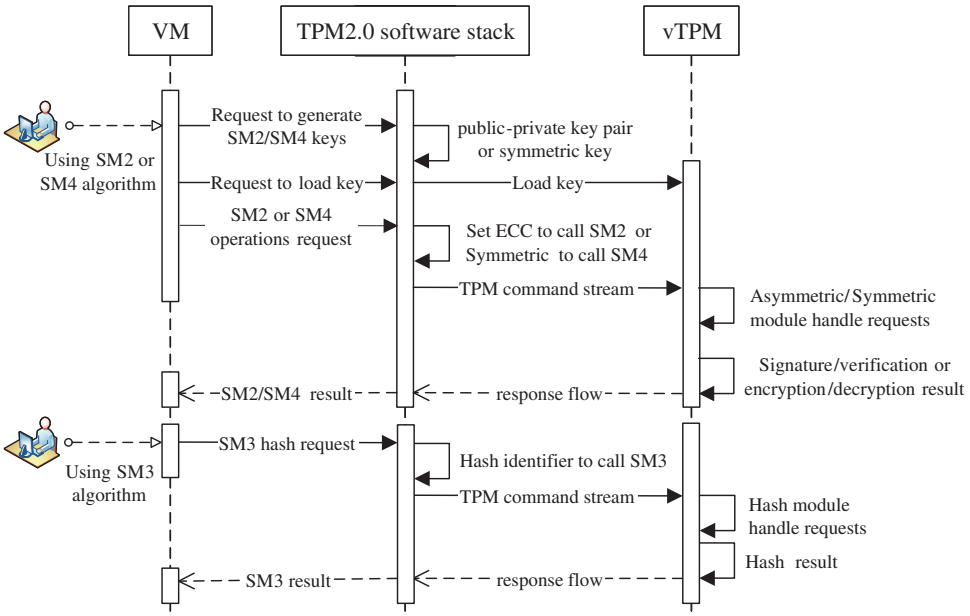

Figure 9: Timing diagram for using SM2/SM3/SM4 via vTPM

In addition, when vTPM provides SM3 hash service and SM4 algorithm encryption and decryption service, the execution process is similar to that of SM2 algorithm.

5 Experiments and Evaluation

Our ultimate goal is to establish a trust relationship for multiple cloud units in a distributed cloud computing scenario to enhance the credibility of the 6G network architecture. In this section, we provide relevant experiments to evaluate the effectiveness and performance of the proposed approach. The system of the physical server is CentOS 7.8, and the configuration is Xeon(R) Silver 4216 CPU @ 2.10 GHz/DDR4 32G∗8 memory/1.3T disk. The configuration of cloud servers is Qemu virtual CPU/2G memory/20G disk, with the OS version of CentOS 7.8 and kernel version of 3.10. The experimental environment is based on the open-source OpenStack cloud platform, in which each physical server is equipped with a hardware TPM, and the cloud server can be configured with a vTPM on the host machine.

The trusted platform module or the trusted cryptographic module has functions for cryptographic calculation and measurement storage. Therefore, we compared the proposed scheme with the existing TPM, vTPM, and TCM, and the supported functions are shown in Table 1. In contrast, this solution not only supports the trusted boot process to extend the measurement results to SM3 PCR bank of the vTPM device but also provides vTPM-based SM2/SM3/SM4 cryptographic algorithm services for virtual machines.

Table 1: Supported function comparison

Supported function	TPM	vTPM	TCM	The proposed approach
SHA1 PCR	√	√		√
SHA256 PCR		√		√
SM3 PCR			√	√
SHA1, SHA256, AES, ECC algorithms	√	√		√
SM2, SM3, SM4 algorithms			√	√

We have carried out experiments in CentOS system to verify that the modified BIOS and kernel can establish trust chain based on SM3 algorithm. It turns out that our approach can store the trusted measures into SM3 PCR bank, and the results of the SM3 hash measures are stored in the IMA measure log. For the time efficiency of trusted boot, we calculate the startup time of virtual machines without trusted platform modules (denoted as VM-null), trusted virtual machines with trust chains based on the SHA1 algorithm (denoted as TVM-SHA1), and trusted virtual machines with trust chains based on SM3 algorithm (denoted as TVM-SM3). The method for calculating the virtual machine startup time is that the execution of the qemu command is used as the start time, and the start of the network service process is used as the end time. The startup time overhead of the three virtual machines is shown in Fig. 10. The data shows that the startup time are 25.508, 30.783 and 35.499 s, respectively, and TVM-SHA1 and TVM-SM3 both enable IMA integrity measurement. Compared with VM-null, TVM-SM3 increases the time overhead because it needs to measure key components and applications such as BIOS and GRUB during the trusted boot process, which increases the time consumption by 39.17%. Compared with TVM-SHA1, TVM-SM3 increases the time overhead by 15.32%, because the calculation of SM3 algorithm takes more time than SHA1 algorithm. Although the calculation efficiency of SM3 is not as good as that of SHA1, its security is higher, and the added time overhead does not exceed 5 s. In order to ensure the safety and reliability of the virtual machine trust chain construction process, we believe that the increased performance overhead is within the acceptable range.

We designed three sets of experiments to test the performance overhead of the host machine through SM3 algorithm to achieve trusted boot. These three sets of experiments repeatedly started VM-null, TVM-SHA1 and TVM-SM3, and tested the performance overhead of the host through Unixbench during the period. Fig. 11 shows the performance overhead brought by the three types of virtual machine startup process to the host, and the final System Benchmarks Index Scores are 864.1, 856.6, and 856.1, respectively. It shows that when starting TVM-SM3, the total score of the system is only reduced by 0.93% and 0.06% compared with when starting VM-null and TVM-SHA1 and the scores of other benchmark items have little difference, which will cause performance loss to the host system Negligible.

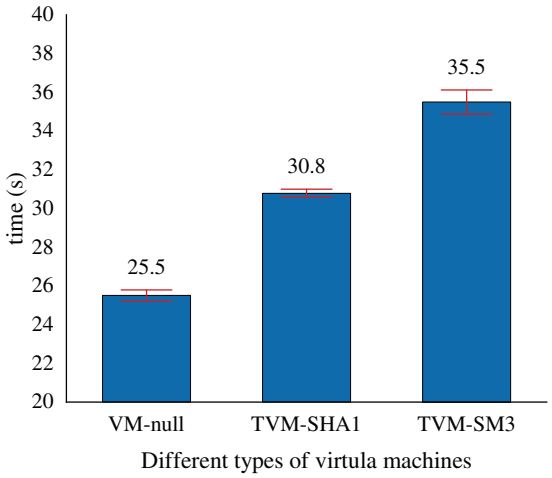

Figure 10: Three types of virtual machine startup time overhead

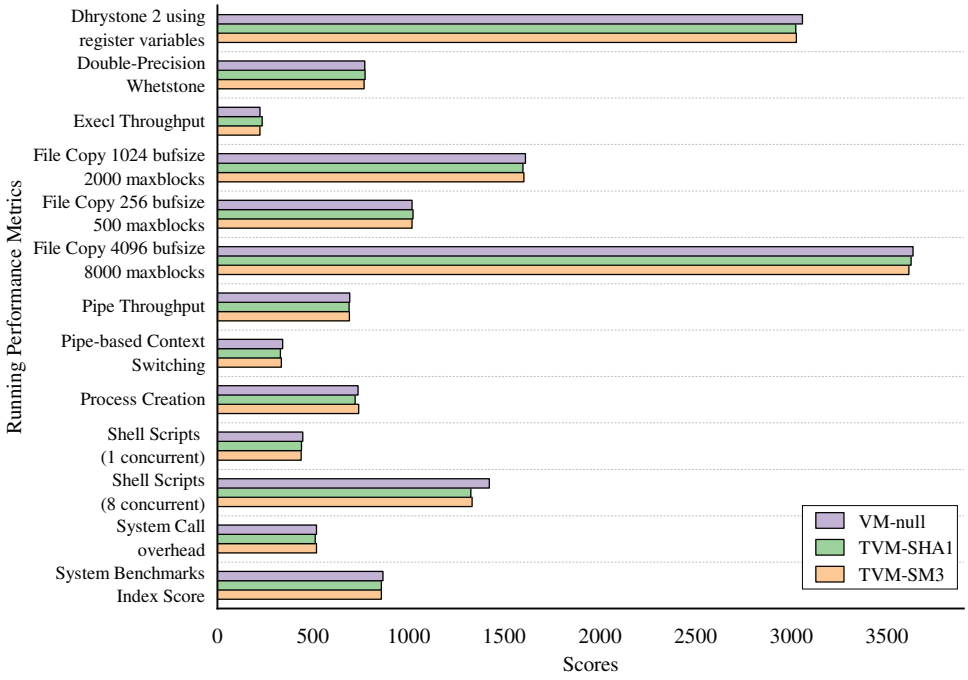

Figure 11: Performance overhead of starting three types of virtual machines under Unixbench

Next, we compared the most time-consuming virtual machine image verification and update operations in the management process of the ISM module. The regular approach is to verify integrity by computing a hash of the entire image file. However, the proposed method firstly divides the image file into several small files of equal size, uses SM3 algorithm to hash all the small files, and finally hashes all the hash values again to obtain the final image base value. As shown in Fig. 12, the regular method counts the time consumption of calculating the hash value of the image file, and the proposed approach counts the time consumption of splitting the image file, calculating the hash value of all small files, and calculating the total hash value. The data clearly shows that the time overhead increases with

the file size, but the proposed method is much less time-consuming, taking only 20% of the regular method.

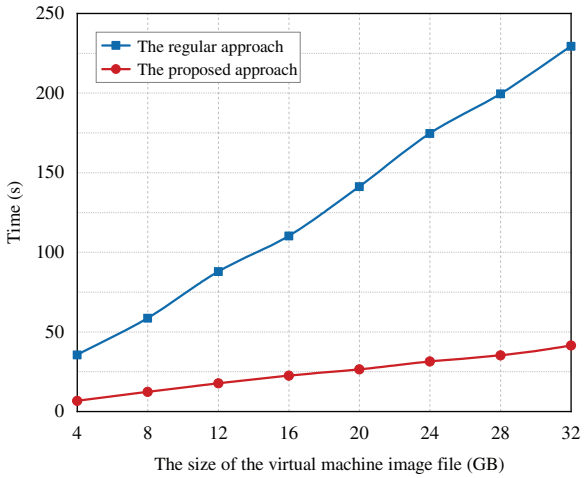

Figure 12: Time cost comparison of IMS module measurement verification stage

When establishing the trust relationship between CCU and DCU, we hand over the remote atteattestation of the server in DCU to the centralized RAS, the distributed RAS, and PS in a layered manner, which can reduce the verification burden of the centralized RAS. Regardless of the direct network transmission delay and other losses of different cloud units, the method proposed in this paper only introduces the time overhead caused by the one-time verification of the key evidence (boot_aggregate) of all servers in the DCU by the centralized RAS. Fig. 13 shows the time overhead as the number of servers in the cloud cell increases. Since the centralized RAS only aggregates the key evidence of each server, even if the number of servers reaches 10000, the time overhead is less than 3 seconds.

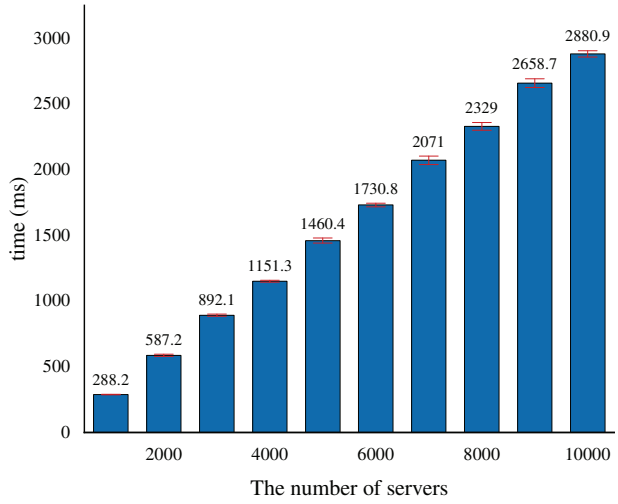

Figure 13: The calculation time of the centralized RAS varies with the number of servers

6 Conclusion

We suggest a trust-building method based on trusted computing for multi-cloud scenarios in 6G architectures to improve the trustworthiness of 6G networks. In this study, we have extended the supported cryptographic algorithms of vTPM and enabled the PCR bank of SM3 algorithm, which enabled SM3 algorithm to be used in the trusted boot phase and brought about only 15.32% of the startup time overhead. The proposed trust establishment method first establishes the OS trust chain by trusted boot, and then the centralized RAS, the distributed RAS, and PS verify different objects by remote attestation, respectively. Regardless of the verification overhead of the distributed RAS and PS, it costs no more than 3 seconds to finish the one-time verification of the key evidence towards 10,000 servers by the centralized RAS.

Acknowledgement: We are very grateful to the editors and reviewers for their valuable comments on the experimental design and English writing, which have been very helpful in improving the quality of the manuscript.

Funding Statement: This work was supported by the Ministry of Education and China Mobile Research Fund Project (MCM20200102), the 173 Project (No. 2019-JCJQ-ZD-342-00), the National Natural Science Foundation of China (No. U19A2081), the Fundamental Research Funds for the Central Universities (No. 2023SCU12129), the Science and Engineering Connotation Development Project of Sichuan University (No. 2020SCUNG129).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Mingxing Zhou, Qixu Wang; data collection: Peng Xiao, Menglong Yang; analysis and linterpretation of results: Shuhua Ruan, Xingshu Chen; draft manuscript preparation: Mingxing Zhou, Qixu Wang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The relevant data and code of this manuscript can be accessed: https://github.com/ittlelearnlive/cmes28612.

Conflicts of Interest: The preliminary work of this paper is published in 2022 IEEE 15th International Conference on Cloud Computing (IEEE CLOUD 2022). The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Banafaa, M., Shayea, I., Din, J., Azmi, M. H., Alashbi, A. et al. (2023). 6G mobile communication technology: Requirements, targets, applications, challenges, advantages, and opportunities. *Alexandria Engineering Journal, 64,* 245–274.
2. Hosseinzadeh, M., Hemmati, A., Rahmani, A. M. (2022). 6G-enabled internet of things: Vision, techniques, and open issues. *Computer Modeling in Engineering & Sciences, 133(3),* 509–556. https://doi.org/10.32604/cmes.2022.021094
3. Li, X., Liu, S., Kumari, S., Chen, C. M. (2023). PSAP-WSN: A provably secure authentication protocol for 5G-based wireless sensor networks. *Computer Modeling in Engineering & Sciences, 135(1),* 711–732. https://doi.org/10.32604/cmes.2022.022667
4. Khan, L. U., Saad, W., Niyato, D., Han, Z., Hong, C. S. (2022). Digital-twin-enabled 6G: Vision, architectural trends, and future directions. *IEEE Communications Magazine, 60(1),* 74–80.
5. Tomkos, I., Klonidis, D., Pikasis, E., Theodoridis, S. (2020). Toward the 6G network era: Opportunities and challenges. *IT Professional, 22(1),* 34–38.

6. Wang, S., Sun, T., Yang, H., Duan, X., Lu, L. (2020). 6G network: Towards a distributed and autonomous system. *2020 2nd 6G Wireless Summit (6G SUMMIT)*, pp. 1–5. Levi, Finland, IEEE.
7. Kumar, R., Goyal, R. (2019). On cloud security requirements, threats, vulnerabilities and countermeasures: A survey. *Computer Science Review, 33,* 1–48.
8. Zhang, L., Xiong, H., Huang, Q., Li, J., Choo, K. K. R. et al. (2022). Cryptographic solutions for cloud storage: Challenges and research opportunities. *IEEE Transactions on Services Computing, 15(1),* 567–587.
9. Yang, M., Huang, G., Liu, J., Gui, Y., Wang, Q. et al. (2023). PIMS: An efficient process integrity monitoring system based on blockchain and trusted computing in cloud-native context. *Computer Modeling in Engineering & Sciences, 136(2),* 1879–1898. https://doi.org/10.32604/cmes.2023.026371
10. Subramanian, N., Jeyaraj, A. (2018). Recent security challenges in cloud computing. *Computers & Electrical Engineering, 71,* 28–42.
11. Tang, X., Cao, C., Wang, Y., Zhang, S., Liu, Y. et al. (2021). Computing power network: The architecture of convergence of computing and networking towards 6G requirement. *China Communications, 18(2),* 175–185.
12. Je, D., Jung, J., Choi, S. (2021). Toward 6G security: Technology trends, threats, and solutions. *IEEE Communications Standards Magazine, 5(3),* 64–71.
13. Firoozjaei, M. D., Jeong, J. P., Ko, H., Kim, H. (2017). Security challenges with network functions virtualization. *Future Generation Computer Systems, 67,* 315–324.
14. Salahdine, F., Han, T., Zhang, N. (2023). Security in 5G and beyond: Recent advances and future challenges. *Security and Privacy, 6(1),* e271.
15. Ridhawi, I. A., Otoum, S., Aloqaily, M. (2023). Decentralized zero-trust framework for digital twin-based 6G. arXiv preprint arXiv:2302.03107.
16. Catak, F. O., Kuzlu, M., Catak, E., Cali, U., Unal, D. (2022). Security concerns on machine learning solutions for 6G networks in mmwave beam prediction. *Physical Communication, 52,* 101626.
17. Catak, E., Catak, F. O., Moldsvor, A. (2021). Adversarial machine learning security problems for 6G: mmWave beam prediction use-case. *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 1–6. Bucharest, Romania, IEEE.
18. Rahman, M. A., Hossain, M. S. (2022). A deep learning assisted software defined security architecture for 6G wireless networks: IIoT perspective. *IEEE Wireless Communications, 29(2),* 52–59.
19. Wang, X., Gao, Y., Deng, L., Chen, M. (2022). DTCPN: A digital twin cyber platform based on NFV. *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 579–583. Belfast, UK, IEEE.
20. Antón, S. G., Grasa, E., Fernández, C., Siddiqui, M. S. (2022). RINA-based virtual networking solution for distributed VNFs: Prototype and benchmarking. *2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, pp. 369–374. Grenoble, France, IEEE.
21. Mogyorósi, F., Babarczi, P., Zerwas, J., Blenk, A., Pašić, A. (2022). Resilient control plane design for virtualized 6G core networks. *IEEE Transactions on Network and Service Management, 19(3),* 2453–2467.
22. Kurdi, H., Alfaries, A., Al-Anazi, A., Alkharji, S., Addegaither, M. et al. (2019). A lightweight trust management algorithm based on subjective logic for interconnected cloud computing environments. *The Journal of Supercomputing, 75(7),* 3534–3554.
23. Ahmed, U., Raza, I., Hussain, S. A. (2019). Trust evaluation in cross-cloud federation: Survey and requirement analysis. *ACM Computing Surveys, 52(1),* 1–37.
24. Wang, T., Zhang, G., Bhuiyan, M. Z. A., Liu, A., Jia, W. et al. (2020). A novel trust mechanism based on fog computing in sensor–cloud system. *Future Generation Computer Systems, 109,* 573–582.
25. Mrabet, M., Saied, Y. B., Saidane, L. A. (2017). Modeling correlation between QoS attributes for trust computation in cloud computing environments. *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pp. 488–497. Madrid, Spain, IEEE.

26. Bogdanov, A., Shchegoleva, N., Dik, G., Khvatov, V., Dik, A. (2022). "Smart Habitat": Features of building it infrastructure, main problems of building data networks using 5G (6G) technologies. *Computational Science and Its Applications–ICCSA 2022 Workshops*, pp. 628–638. Malaga, Spain, Cham, Springer International Publishing.
27. Ylianttila, M., Kantola, R., Gurtov, A., Mucchi, L., Oppermann, I. et al. (2020). 6G white paper: Research challenges for trust, security and privacy. arXiv preprint arXiv:2004.11665.
28. Oliver, I. (2021). Trust, security and privacy through remote attestation in 5G and 6G systems. *2021 IEEE 4th 5G World Forum (5GWF)*, pp. 368–373. Montreal, QC, Canada, IEEE.
29. Zhou, M., Ruan, S., Liu, J., Chen, X., Yang, M. et al. (2022). vTPM-SM: An application scheme of SM2/SM3/SM4 algorithms based on trusted computing in cloud environment. *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*, pp. 351–356. Barcelona, Spain, IEEE.
30. Wang, Y., Wang, Q., Chen, X., Chen, D., Fang, X. et al. (2022). Containerguard: A real-time attack detection system in container-based big data platform. *IEEE Transactions on Industrial Informatics, 18(5),* 3327–3336.
31. Rabbani, M., Wang, Y. L., Khoshkangini, R., Jelodar, H., Zhao, R. et al. (2020). A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing. *Journal of Network and Computer Applications, 151,* 102507.
32. Singh, S., Sidhu, J. (2017). Compliance-based multi-dimensional trust evaluation system for determining trustworthiness of cloud service providers. *Future Generation Computer Systems, 67,* 109–132.
33. Somu, N., MR, G. R., Kirthivasan, K., VS, S. S. (2018). A trust centric optimal service ranking approach for cloud service selection. *Future Generation Computer Systems, 86,* 234–252.
34. Li, X., Wang, Q., Lan, X., Chen, X., Zhang, N. et al. (2019). Enhancing cloud-based IoT security through trustworthy cloud service: An integration of security and reputation approach. *IEEE Access, 7,* 9368–9383.
35. Santos, N., Gummadi, K. P., Rodrigues, R. (2009). Towards trusted cloud computing. *Workshop on Hot Topics in Cloud Computing (HotCloud '09),* pp. 1–5. Berkeley, CA, USENIX.
36. Ibrahim, F. A., Hemayed, E. E. (2019). Trusted cloud computing architectures for infrastructure as a service: Survey and systematic literature review. *Computers & Security, 82,* 196–226.
37. Hosseinzadeh, S., Sequeiros, B., Inácio, P. R., Leppänen, V. (2020). Recent trends in applying TPM to cloud computing. *Security and Privacy, 3(1),* e93.
38. Akram, R. N., Ko, R. K. (2014). Digital trust-trusted computing and beyond: A position paper. *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 884–892. Beijing, China, IEEE.
39. Wang, X., Yin, Y. L., Yu, H. (2005). Finding collisions in the full SHA-1. *Advances in Cryptology, CRYPTO 2005*, pp. 17–36. Santa Barbara, California, USA, Springer Berlin Heidelberg.
40. Bai, D., Yu, H., Wang, G., Wang, X. (2015). Improved boomerang attacks on round-reduced SM3 and keyed permutation of BLAKE-256. *IET Information Security, 9(3),* 167–178.
41. Xiong, H., Mei, Q., Zhao, Y., Peng, L., Zhang, H. (2019). Scalable and forward secure network attestation with privacy-preserving in cloud-assisted Internet of Things. *IEEE Sensors Journal, 19(18),* 8317–8331.
42. Nunes, I. D. O., Eldefrawy, K., Rattanavipanon, N., Steiner, M., Tsudik, G. (2019). VRASED: A verified Hardware/Software Co-Design for remote attestation. *USENIX Security Symposium*, pp. 1429–1446. Santa Clara, CA, USA.
43. Wang, Q., Chen, X., Jin, X., Li, X., Chen, D. et al. (2022). Enhancing trustworthiness of Internet of Vehicles in space–air–ground-integrated networks: Attestation approach. *IEEE Internet of Things Journal, 9(8),* 5992–6002.
44. Gaber, C., Arfaoui, G., Carlinet, Y., Perrot, N., Valleyre, L. et al. (2022). The owner, the provider and the subcontractors: How to handle accountability and liability management for 5G end to end service. *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pp. 1–7. Vienna, Austria.

45. Porambage, P., Gür, G., Osorio, D. P. M., Liyanage, M., Gurtov, A. et al. (2021). The roadmap to 6G security and privacy. *IEEE Open Journal of the Communications Society, 2,* 1094–1122.
46. Chorti, A., Barreto, A. N., Köpsell, S., Zoli, M., Chafii, M. et al. (2022). Context-aware security for 6G wireless: The role of physical layer security. *IEEE Communications Standards Magazine, 6(1),* 102–108.
47. Porambage, P., Gür, G., Osorio, D. P. M., Livanage, M., Ylianttila, M. (2021). 6G security challenges and potential solutions. *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, pp. 622–627. Porto, Portugal, IEEE.
48. Guo, H., Li, J., Liu, J., Tian, N., Kato, N. (2022). A survey on space-air-ground-sea integrated network security in 6G. *IEEE Communications Surveys & Tutorials, 24(1),* 53–87.
49. Jin, X., Wang, Q., Li, X., Chen, X., Wang, W. (2019). Cloud virtual machine lifecycle security framework based on trusted computing. *Tsinghua Science and Technology, 24(5),* 520–534.
50. Yang, A., Nam, J., Kim, M., Choo, K. K. R. (2014). Provably-secure (Chinese government) SM2 and simplified SM2 key exchange protocols. *The Scientific World Journal, 2014*.
51. Wang, D., Wu, L., Zhang, X. (2018). Key-leakage hardware trojan with super concealment based on the fault injection for block cipher of SM4. *Electronics Letters, 54(13),* 810–812.