Tsinghua Science and Technology

Volume 19 | Issue 5 Article 6

2014

Dynamic Measurement Protocol in Infrastructure as a Service

Shuang Xiang
School of Computer, Wuhan University, Wuhan 430072, China.

Bo Zhao
School of Computer, Wuhan University, Wuhan 430072, China.

An Yang
School of Computer, Wuhan University, Wuhan 430072, China.

Tao Wei
School of Computer, Wuhan University, Wuhan 430072, China.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 06/13 pp470-477
Volume 19, Number 5, October 2014

Dynamic Measurement Protocol in Infrastructure as a Service

Shuang Xiang, Bo Zhao*, An Yang, and Tao Wei

Abstract: Infrastructure as a Service (IaaS) has brought advantages to users because virtualization technology hides the details of the physical resources, but this leads to the problem of users being unable to perceive their security. This defect has obstructed cloud computing from wide-spread popularity and development. To solve this problem, a dynamic measurement protocol in IaaS is presented in this paper. The protocol makes it possible for the user to get the real-time security status of the resources, thereby solving the problem of guaranteeing dynamic credibility. This changes the cloud service security provider from the operator to the users themselves. This study has verified the security of the protocol by means of Burrow-Abadi-Needham (BAN) logic, and the result shows that it can satisfy requirements for innovation, privacy, and integrity. Finally, based on different IaaS platforms, this study has conducted a performance analysis to demonstrate that this protocol is reliable, secure, and efficient.

Key words: Burrow-Abadi-Needham (BAN) logic; real-time security; trusted dynamic measurement

1 Introduction

With the rapid development of cloud computing, a variety of cloud computing service models have been proposed by NIST, including Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS)[1] which are currently the most widely accepted. IaaS is the most basic kind of cloud service model, which can be customized according to users' needs and decreases the overhead of deploying hardware. The concept of virtualization with flexibility and freedom has been widely accepted. Recently, some large IT companies have begun to offer IaaS services to the public: Amazon's EC2 is a representative product[2]. However, with the popularity of IaaS applications, its potential security problems have gradually drawn researchers' attention. The main threats in IaaS include three aspects: infrastructure sharing will lead to uncontrolled access of users and operating systems, the integrity of the resources used by the user cannot be proven, and there are no relevant security protocols to protect the communication process between users and cloud. These threats have seriously blocked the popularization and development of cloud computing.

Researchers have launched a series of studies to solve these problems. Nuno et al.[3] proposed a TCCP architecture, Khan et al.[4] proposed the credible Eucalyptus cloud, while Cui et al.[5] proposed a trusted IaaS framework. The above researches all started from the perspective of service providers by designing a series of secure protocols and architectures to constrain the node behavior, so as to ensure the credibility of the service providing process. However, these studies are based on a series of protocols focusing on constraining the process of Virtual Machine (VM) booting and migration, which means that users are unable to master the resource security and flexibility is affected. Furthermore, users cannot perceive whether a running node environment is safe, and these designs cannot resist against a TOCTOU attack.

Bertholon et al.[6] proposed a TCRR protocol from the users' considerations for security, which

• Shuang Xiang, Bo Zhao, An Yang, and Tao Wei are with School of Computer, Wuhan University, Wuhan 430072, China. E-mail: xiangshuang1984@163.com; zhaobo@whu.edu.cn; allenan@tom.com; taowhu@ 163.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2014-07-15; revised: 2014-07-21; accepted: 2014-08-22

can validate specific VMs, according to users' requirements. However, this process does not involve a trusted third party and does not consider the dynamic measurement of runtime VMs, thus being unable to ensure dynamic credibility. In addition, protocol security analysis and proof is an important part of the design. Only through integrated formal verification can potential security issues and defects be detected in the protocols and thereby avoid security attacks, but the majority of the above-mentioned research work has no strict formal analysis or verification, and there may be potential security threats.

We have designed a provably secure IaaS layer protocol that allows users to put forward measure requests and to obtain the real-time trusted status of the resources they are using. The cloud service provider is responsible for measurement implementation in the protocol, and a trusted third party is responsible for the certificate and the verification of measurement results. The separation of measurement and validation processes increases the users' perception of credibility. Compared with previous works, the protocol allows users to manage measurement points and measurement components by themselves, which enhances management flexibility. The dynamic measurement of optional components avoids static metrics TOCTOU attacks. Finally, the introduction of a trusted third party increases the credibility of the results. Based on different IaaS management platforms, this paper has also implemented the specific protocol including the integrated process from dynamic measurement to verification, and conducted performance analysis.

2 Protocol Design

The security of registers, user initiation, and migration of virtual machines[3-5] has been improved through IaaS layer management software platforms in the cloud, but the run-time security of user resources cannot be guaranteed. The IaaS layer structure itself determines whether the user is vulnerable to attacks when using resources, and because service providers often provide a shared infrastructure, physical node hardware resources for system users are not completely isolated. Therefore, when one user is attacked, other active users are vulnerable, so they need to know the real-time security status of resources in order to protect the security of their own resources. This process is often achieved by dynamic measurement.

To measure all user resources in a cloud environment dynamically is inefficient and unnecessary, and does not address how to ensure the legitimacy of the user's identity nor the authenticity of the result. Based on these challenges, this paper designs a dynamic measurement protocol in the IaaS layer. It guarantees the dynamic credibility of resources and customizes the component list according to user requirements, thus changing the provider of cloud services from the operator to the users.

2.1 Key components

The IaaS layer of a cloud management software platform is the core environment management control component; the users are the clients who use it; while the node physical machine in the cloud environment, the operating system installed on the physical machine, the virtual machine monitor, and the virtual machine are the direct providers of user resources. The node's physical machine is the actual carrier of the operating systems, virtual machines, and virtual machine monitors, so from the physical point of view, they are a unit which we can consider as a whole in the protocol design called cloud nodes. A virtual machine disk image storage management area is used to store and manage virtual machine templates, users' virtual machine images, snapshots, and so on.

Users read into the concrete physical machine when needed, so we can assume that the data in the virtual machine disk image storage management zone is static. They are secure, and their security can be protected by other means such as encryption, so the design of our protocol does not involve this component. Finally, this protocol requires a trusted third party for certificate issuance, as well as management and verification of the results from the Cloud Node (CN) metric. In summary, this protocol mainly involves four components, and their relationships are shown in Fig. 1.

User. The users of IaaS resources use various resources in cloud environments through the cloud management software platform. They hope to obtain security of their own resources.

Iaas Cloud Management Software Platform (CMSP). The IaaS layer management software platform is the core component of the entire IaaS layer. It manages and controls the node physical machine, existing disk image area, virtual machine

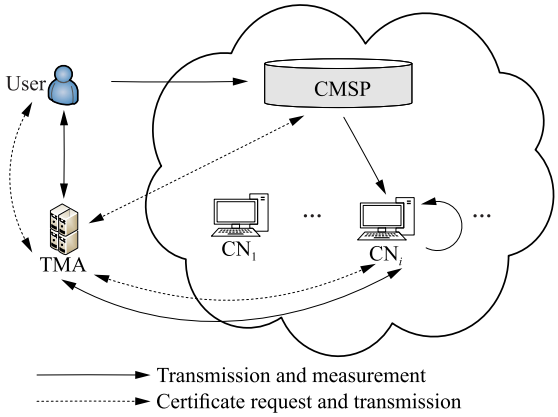

Fig. 1 Protocol components' relationship.

monitor, and virtual machines running on the physical machine, and it also provides interfaces for users to access and use. Current open source IaaS management software platforms include Eucalyptus[7], OpenStack[8], and OpenNebula[9].

CN. The CN is the physical machine that stores the resources that the user desires to keep in a safe state. The process of accepting a measure request sent by the CMSP, launching dynamic metrics, and producing a report that is sent to a Trusted Measure Agent (TMA), a trusted third party, requires the key support of the TPM.

TMA. We propose an additional component based on common IaaS architecture called the TMA. The TMA is a trusted third party that is in charge of certification and responsible for the management and validation of the metric results from the CN. The report will be returned to the user by the TMA.

2.2 The main interaction process

Based on the four components of Section 2.1, the interactive process of the design of this protocol is shown in Fig. 2. The process enables the user to know the security status of resources in real-time according to their own needs through public key encryption and signature technology. In the meantime, it ensures the legitimacy of user identity, the confidentiality of data transmission, and the authenticity of dynamic measurement results.

(1) USR→ CMSP: $\mathrm{SIG_{Usr}}\{\mathrm{CN_{ID}}, \{\mathrm{User_{ID}}, \text{nonce}, \mathrm{mList}(\mathrm{item_1}, \mathrm{item_2}, \cdots)\}\ \mathrm{AIK_{CN}^{PUB}}\}$

According to user's need, we first use $\mathrm{AIK^{PUB}}$ of the corresponding node to encrypt the component list whose security status the user wants to obtain. The user then uses their private key to sign the encrypted component list and sends it to CMSP.

(2) CMSP → $\mathrm{CN}_i\{\mathrm{User_{ID}}, \text{nonce}, \mathrm{mList}(\mathrm{item_1}, \mathrm{item_2}, \cdots)\}\mathrm{AIK_{CN}^{PUB}}$

CMSP uses $\mathrm{AIK_{CN}^{PUB}}$ to verify the user's identity and message integrity. If the message is correct, CMSP sends the encrypted information to the corresponding CN node, according to the encrypted message.

(3) CN_i → TMA: $\mathrm{SIG}_N\{\mathrm{User_{ID}}, \text{nouce}, \mathrm{CN_{ID}}, \mathrm{TPM_{ID}}, \mathrm{SML}, \mathrm{mList}(\mathrm{item_1}, \mathrm{item_2}, \cdots)\}$

The node receives the request and decrypts it using the $\mathrm{AIK_{CN}^{PVR}}$. Then, the node initiates a dynamic measurement according to mList. After the process of dynamic measurement, CN makes a measurement report. The report includes Storage Measurement Log (SML) stored in TPM, name of CN, user ID, user nonce, user measurement list, and version. Finally, CN node signs the report and sends it to TMA.

(4) TMA → User : $\{\text{result}, \text{nonce}\}\mathrm{AIK_{User}^{PUB}}$

After receiving all the information, TMA extends and restores the order of all the recording points in the log according to the measurement Log( ). By calculating $\mathrm{Hash}(\mathrm{User_{ID}}, \text{nonce}, \mathrm{CN_{ID}}, \mathrm{TPM_{ID}}, \mathrm{PCR_{value}^{\mathit{i}}})$. TMA verifies the integrity of the information. If verification is successful, then it returns result = ok, otherwise it returns result = false. Finally, TMA uses $\mathrm{AIK_{User}^{PUB}}$ to encrypt the result and nonce, then sends it to the user.

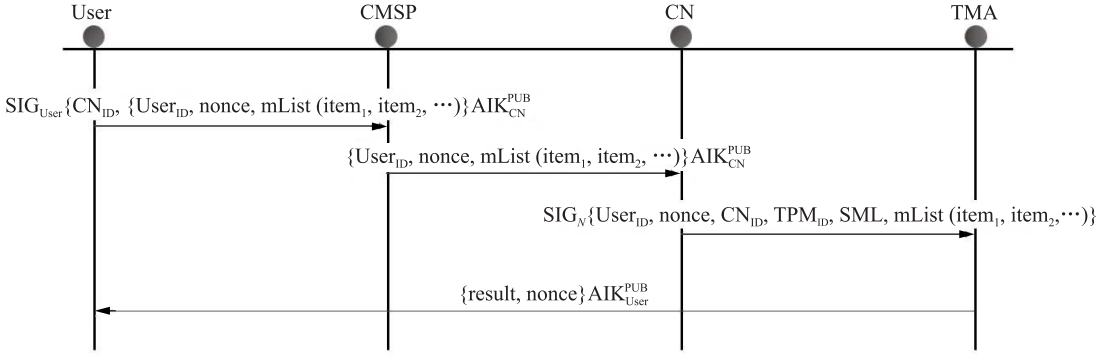

Fig. 2 Interaction process of protocol.

3 Security Proof of Protocol

3.1 Terms and rules of BAN logic

Burrow-Abadi-Needham (BAN) is a modal logic based on knowledge and belief[10]. In the BAN logic ratiocination, we first convert the protocol messages to idealized protocol, then we make a reasonable postulate according to specific circumstances, and ratiocinate based on protocol and postulate. The main beliefs that participate in the protocol constantly change with the message. Finally, we check whether the protocol can achieve the desired objectives[11].

In addition, we use the following terms:

$P| \equiv X$: P believes X;

$P \lhd X$: P sees x;

$P| \sim X$: P once said X;

$p| \Rightarrow X$: P has jurisdiction over X;

$\sharp(X)$: The formula X is fresh;

$| \xrightarrow{K} P$: P has K as a public key;

$\{X\}_K$: This represents the formula X encrypted under the key K;

$P \xleftrightarrow{K} Q$: P and Q may use the shared key K to communication;

$P \overset{K}{\Longleftrightarrow} Q$: The formula K is a secret known only to P and Q;

$< X >_Y$: This represents X combined with the formula Y, it is intended that Y be a secret, and that its presence prove the identity of whoever utter $< x >_y$.

There are 19 logical rules in BAN logic. The following are the rules that we use in the proof procedure:

R1: $\dfrac{P| \equiv Q \overset{K}{\Longleftrightarrow} P \;\; P \lhd X_K}{P| \equiv Q| \sim X}$;

R2: $\dfrac{P| \equiv | \xrightarrow{K} Q \;\; P \lhd X_{K^{-1}}}{P| \equiv Q| \sim X}$;

R3: $\dfrac{P| \equiv Q \overset{K}{\Longleftrightarrow} P \;\; P \lhd X_K}{P \lhd X}$;

R4: $\dfrac{P| \equiv \sharp(X)}{P| \equiv \sharp(X, Y)}$;

R5: $\dfrac{P| \equiv \sharp(X), P| \equiv Q| \sim X}{P| \equiv Q| \equiv X}$;

R6: $\dfrac{P| \equiv Q| \sim (X, Y)}{P| \equiv Q| \sim X}$;

R7: $\dfrac{P| \equiv Q| \Longrightarrow X, \;\; P| \equiv Q| \equiv X}{P| \equiv X}$.

3.2 Security analysis of protocol

3.2.1 The idealized protocol

According to the protocol process described in Section 2.2 and the general analysis method of BAN, the idealized protocol is shown below:

M1: User $\rightarrow$ CMSP: $\{\text{CN}_{\text{ID}}, \sharp(\text{nonce}), \{\text{User}_{\text{ID}}, \text{nonce}, \text{mList}(\text{item}_1, \text{item}_2, \cdots)\}K_{\text{CN}}\}K^{-1}_{\text{User}}$;

M2: CMSP $\rightarrow$ CN : $\{\text{User}_{\text{ID}}, \text{nonce}, \text{mList}(\text{item}_1, \text{item}_2, \cdots)\}K_{\text{CN}}$;

M3: CN $\rightarrow$ TMA: $\{\text{User}_{\text{ID}}, \text{nonce}, \text{CN}_{\text{ID}}, \text{TPM}_{\text{ID}}, \text{SML}, \text{mList}(\text{item}_1, \text{item}_2, \cdots)\}K^{-1}_{\text{CN}}$;

M4: TMA$\rightarrow$ User : {result nonce} K_{User}.

3.2.2 The idealized assume

A1:User$| \equiv \sharp(\text{nonce})$;

A2:CMSP$| \equiv | \xrightarrow{K_{\text{User}}}$ User;

A3:TMA$| \equiv | \xrightarrow{K_{\text{CN}}}$ CN;

A4:CN$| \equiv$ User $\overset{K_{\text{User}}}{\Longleftrightarrow}$ CN;

A5:User$| \equiv$ TMA $\overset{K_{\text{User}}}{\Longleftrightarrow}$ User.

3.2.3 Security objectives and analysis

The protocol in Section 2 consists of two parts. The first part is that the user sends the message to the CMSP, and then the CMSP sends the message to the CN before the beginning of the measurement. Based on the security requirements of the protocol, the goals of this part are for the CMSP to trust the legitimacy of measurement information sources, and for the CN to trust the confidentiality of the information. The second part is that the CN sends the measurement report to the TMA after the measurement procedure. The goals of this part are that the TMA trusts the authenticity of the measurement information sources, and the user trusts the authenticity and novelty of the result. The security of the measurement process depends on the TPM on the CN.

According to BAN logic syntax, we abstract the security objectives and prove them as follows.

(1) CMSP$| \equiv$ User$| \sim \text{CN}_{\text{ID}}$

Using M1, We have

CMSP$\lhd\{\text{CN}_{\text{ID}}, \sharp(\text{nonce}), \{\text{User}_{\text{ID}}, \text{nonce}, \text{mList}(\text{item}_1, \text{item}_2, \cdots)\}K_{\text{CN}}\}K^{-1}_{\text{User}}$.

By initialization postulate A1 and using rule R2, we have

CMSP$| \equiv$ User$| \sim (\text{CN}_{\text{ID}}, \sharp(\text{nonce}), \{\text{User}_{\text{ID}}, \text{nonce}, \text{mList}(\text{item}_1, \text{item}_2, \cdots)\})$,

Using rule R5, We can prove that

CMSP$| \equiv$ User$| \sim \text{CN}_{\text{ID}}$.

(2) CN$| \equiv$ User$| (\text{User}_{\text{ID}}, \text{nonce}, \text{mList}(\text{item}_1, \text{item}_2, \cdots))$

Using M1, we have

CN $\lhd \{\text{User}_{\text{ID}}, \text{nonce}, \text{mList}(\text{item}_1, \text{item}_2, \cdots)\}$

Using postulate A4 and rule R1, we have

CN| ≡ User| ∼ (User_{ID}, nonce, mList(item_1, item_2, $\cdots$)).

(3) TMA| ≡ CN| ∼ (User_{ID}, nonce, CN_{ID}, TPM_{ID}, SML, mList($\text{item}_1, \text{item}_2, \cdots$))

Using M3, we have

TMA ◁ {User_{ID}, nonce, CN_{ID}, TPM_{ID}, SML, mList($\text{item}_1, \text{item}_2, \cdots$)}$K_{\text{CN}}^{-1}$.

Using postulate A3 and rule R5, we have our proof.

(4) User| ≡ TMA| ∼ (result, nonce)

Using M4, A4, and rule R1, we have

User| ≡ TMA| ∼ (result, nonce).

(5) User|♯(result, nonce)

Using M4, we have

User ◁ (result, nonce).

Using A4 and rule R3, we have

User ◁ (result, nonce).

Using A1 and rule R4, we can prove

User|♯(result, nonce).

With the above proof, we learn that the design of the protocol can satisfy our security requirements. That is to say, with the premise that the TPM could guarantee the security of the measurement process, our design of the protocol is secure and reliable. It can satisfy all the requirements of originality, confidentiality, and integrity.

4 Experiment and Performance Analysis

In the design of the protocol described in Section 2, dynamic measurement is a fundamental component of this protocol. The cloud node initiates the dynamic measurement by accepting the user's measurement request and the list of measuring, and guarantees the run-time security of user resources by dynamic measurement techniques. So in the concrete implementation, we need to design and realize the dynamic measurement process of cloud nodes as well as the interaction process of the protocol.

4.1 Dynamic measurement technology

TPM-based static integrity measurement methods ensure the integrity protection of platform boot time, but cannot ensure runtime security of the platform. Based on this, Intel and AMD have respectively put forward their own dynamic measurement root technology. Their core idea is that we can allow the system to measure the integrity at the current time from any untrusted status as a starting point of measurement. The trusted computing specification[12] has also distinguished between static PCR and dynamic PCR. It defines that PCR17-23 is a dynamic trusted root and can only be reset by a dynamic trusted root.

The technologies proposed by Intel and AMD are Intel TXT[13] and AMD SVM[14] respectively, and they have almost the same principle. In this paper, we used Intel TXT technology.

4.2 Implementation of trusted measurement in cloud

A full virtualized environment running on CN node includes HypervisorDomain 0 and Domain U. The structure of our design of the trusted measurement module is shown in Fig. 3.

We set a measurement module in Hypervisor and a control module in Domain 0 of each node. The

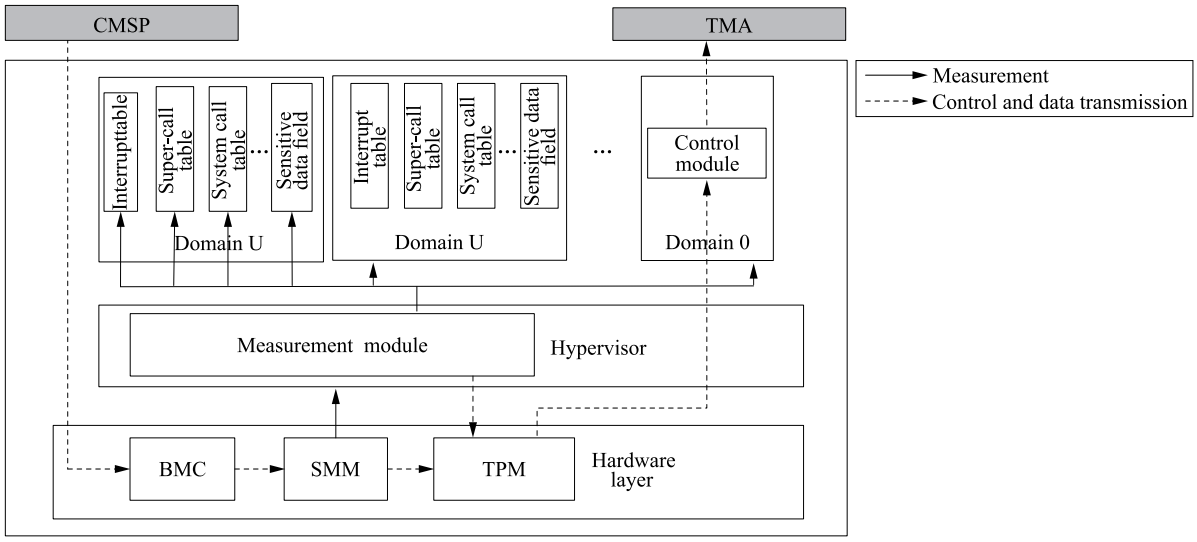

Fig. 3 Dynamic measurement process.

measurement module is designed to measure the virtual domain as well as its internal components. While the System Management Mode (SMM) guarantees the security of both the measurement modules and Hypervisor itself, the control module is intended to obtain the metrics reports from the TPM module and to send the report to the TMA.

The measurement process is as follows: (1) Initialization of the operation starts the process. In the process of starting Domain U, the process of setting the interrupt table is intercepted in order to preserve the assigned address of the virtual machine vCPU structure and mark the virtual machine that is running; (2) The CMSP triggers the SMI remotely via IPMI and BMC, and then the SMI dynamically executes the measurement to the VMM and the measurement modules within the VMM; (3) After the measure gets passed, the metrics program is triggered via a super call created by the measurement module. According to the user's measure requests to the kernel image, system call table, interrupt table, and the exception table, as well as the super-sensitive data field, a part or all of the dynamic integrity measurement is conducted. (4) After the measurement, the control module in Domain 0 obtains the relevant PCR values and produces a measure status report that will be sent to the TMA.

4.3 Performance analysis

The security and effectiveness of the protocols in the practical application is the crucial factor whether the protocols can be widely used or not[15], as well as whether the new protocol will affect the performance of the original platform. Based on the two points above, this paper has carried out a testing analysis for the effectiveness and security of the protocol, as well as the effect that the protocol has on the original platform.

Our experimental environment is as follows: Three servers equipped with Intel 2.0 GHz X5620*2 processors, 128 GB memory, and Western Digital 4 TB Raid 3 hard disks. Each server carries an ST TPM1.2 security chip for building the CMSP and CN as well as the trusted third party, respectively. For software, the operating system referred to in this paper uses SELS 11 sp1 or Ubuntu 12.04 Server Edition, while Xen 4.0 is chosen to be the virtual machine monitor. The IaaS layer management software platform is Eucalyptus 3.1, OpenStack E3, and OpenNebula 3.6 for conducting the implementation analysis.

4.3.1 Effectiveness

In order to verify whether the protocol is able to work effectively, the paper carried out an implementation of the protocol on three servers to judge whether the protocol can perform securely and effectively. We have established a remote login environment on a TMA server with JSP displaying pages and postgreSQL storing encrypted measure results. The components that users can select include kernel images, the system call table, interrupt table, and super call table. After the measurement of the selected component, users can log into the TMA to view the results.

In our experiment, users use four different virtual machine instances to measure different components. Before users start the measurement, we attacked the third virtual machine instances to modify the kernel permissions table, and the measurement results after the attack are shown in Fig. 4. The components marked red indicate that they are under attack while those marked blue are those not selected by users for measuring. It indicates that the experiment has reflected the trusted status of a different virtual machine instance in a timely and effectively manner.

4.3.2 Security

In order to verify whether this protocol can resist existing common attacks against the trusted measurement platform, we conducted a secure test on the protocol by simulating replay attacks, forgery attacks, and channel stealing. The additional premise of our attack test is that the TMA and the TPM, as well as all the private keys, are safe.

In the replay attack, we assume that there exist some security problems in Hypervisor. The attacker can make use of the communication module deployed in Xen Hypervisor to obtain the measuring control information of the CMSP and deliver the information out of date to the TMA. For a forgery attack, we assume that the

Hosts and Instances

InstanceID	HostIp	HostName	Kernel_State	SystemCallTable_State	IDT_State	HyperCall_State	UserInfo
i-35C505D8	192.168.1.124	liss-nc1	trust	trust	trust	trust	seucalyptus-1
i-37BC17A4	192.168.1.124	liss-nc1	trust	trust	trust	unknwn	seucalyptus-1
i-23D56C8E	192.168.1.125	liss-nc2	untrust	trust	unknown	trust	seucalyptus-1
i-D36A1E7B	192.168.1.125	liss-nc2	trust	unknown	unknown	unknown	seucalyptus-1

Welcome to Trusted Eucalyptus CLoud Administration System! Migrate Instance Stop Instance

Fig. 4 Report of measuring results.

attacker can get part of the CN permissions, thereby being able to try to falsify measurement results. In a channel attack, we assume that the attacker can intercept data transmission at any time and try replacing part of the data. The results of the experiment are shown in Table 1.

The results show that in the replay attacks, since the authentication information has utilized random numbers to guarantee data freshness, the TMA finds the random number in the report is inconsistent when verifying the measurement report, thus detecting that the platform has been attacked. The success of forgery attacks and channel attacks depends on the safety of modern cryptography, but we can assume that an attacker cannot achieve the purpose of attack due to the existing public key cryptography system.

4.3.3 Effect on the performance of the cloud platform

The protocol designed in this paper has improvements that have added some communication interaction and platform measurement, which may lead to an effect on performance. The communication interaction mainly involves package sending and receiving, as well as some simple encryption and decryption. Compared with the overhead of platform measurement, the impact of communication interaction on the cloud platform can be ignored, so we will focus on the evaluation of the measurement's performance overhead of the original platform. In addition, we also measure the time cost from user-initiated dynamic measurement request to receiving results under laboratory conditions.

We have tested performance overhead for management software in different IaaS layers. Since in this protocol, users can selectively measure the component list, during our measurement tests, we just consider the maximum overhead situation, i.e., selecting all components for measurement to verify whether the measurement overhead of the system will have impacts on user experience. In this paper, we have gathered the statistical data of measurement time for five times and the results are shown in Fig. 5.

Table 1 The results of attack experiment.

Attack	Analysis of the results	Effective or not
Replay	When TMA verifies the measuring report, the inconsistent nonce is found which means that attack is detected.	No
Forgery	Since the attacker cannot obtain private keys and the right of controlling TPM, he is unable to make a valid report.	No
Channel	The data packets have been signed or encrypted, thus cannot be changed.	No

The experiments show that for different IaaS platforms, the overhead is between 600 and 800. The measurement time cost from users initiating dynamic measurement to receiving the final result is between 1000 and 1600 ms. We can see that there is a tiny system overhead for the protocol on different platforms which will not impact the performance of the original cloud platform and the user's actual experience.

5 Conclusions

Existing technologies and methods cannot guarantee the dynamics and credibility of the IaaS layer resources. To solve this problem, we propose a dynamic trust measurement protocol in the IaaS layer that can be proved to be secure. It allows users to take the initiative for dynamic measurement requests for resources used in cloud environments according to their needs, and can customize the measurement list to improve flexibility. Meanwhile, the protocol solves the problem whereby traditional research cannot guarantee the dynamics and credibility of user resources, and changes the cloud service security provider from the operator to the users. This has greatly improved the flexibility. In this paper, the process for measuring and controlling and the process for verifying and managing measurement results are separated, so that the administrator of the cloud CLC cannot know or tamper with the final results, improving the credibility of the results for the user. In this paper, the protocol is implemented on different IaaS platforms. The results show that it can dynamically protect the integrity of user

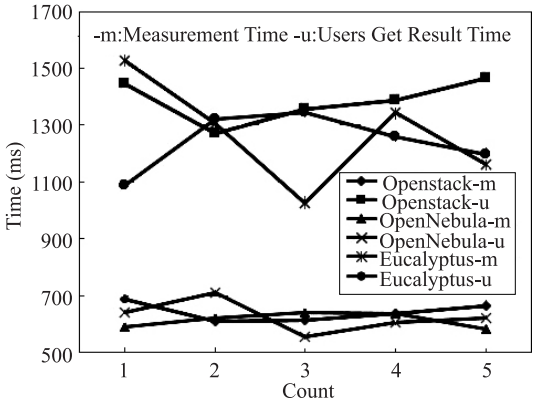

Fig. 5 Time overhead.

resources and will not impact the performance of the original cloud platform and user experience.

Acknowledgements

This work was supported by the National Basic Research Program of China (No. 2014CB340600), the National Natural Science Foundation of China (Nos. 61332019, 61173138, 6127245, and 91118003), and the New Products and Technology Research and Development Projects of Hubei Province (No. 2012BAA03004).

References

[1] NIST definition of cloud comp., http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, 2012.

[2] Amazon elastic compute cloud (EC2), http://aws.amazon.com/ec2/, 2006.

[3] S. Nuno, G. Krishna, and R. Rodrigo, Towards trusted cloud computing, in *Proc. the 2009 Conference on Hot Topics in Cloud Computing*, 2009, pp. 22-27.

[4] I. Khan, H. Rehman, and Z. Anwar, Design and deployment of a trusted eucalyptus cloud, in *Proc. 2011 IEEE International Conference on Cloud Computing (CLOUD)*, 2011, pp. 380-387.

[5] W. Cui, Y. Li, and X. Si, The protocol design of a eucalyptus-based infrastructure-as-a-service (IaaS) cloud framework, *Journal of Electronics & Information Technology*, vol. 34, no. 7, pp. 1748-1754, 2012.

[6] B. Bertholon, S. Varrette, and P. Bouvry, CertiCloud: A novel TPM-based approach to ensure cloud IaaS security, in *Proc. IEEE International Conference on Cloud Computing (CLOUD)*, 2011, pp. 121-130.

[7] D. Daniel, W. Rich, G. Chris, G. Chris, O. Graziano, S. Sunil, Y. Lamia, and Z. Dmitrii, The eucalyptus opensource cloud-computing system, in *Proc. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2009, pp. 124-131.

[8] The OpenStack Community, OpenStack cloud software, http://www.openstack.org/, 2011.

[9] The OpenNebula Project, OpenNebula: The open source toolkit for cloud computing, http://opennebula.org/, 2011.

[10] M. Burrows, M. Abadi, and R. Needham, A logic of authentication, *ACM Transactions in Computer System*, vol. 8, no. 1, pp. 18-36, 1990.

[11] S. S. Ahamad, V. N. Sastry, S. K. Udgata, Secure mobile payment framework based on UICC with formal verification, *Int. J. of Computational Science and Engineering*, vol. 9, no. 4, pp. 355-370, 2014.

[12] TCG, TCG specification architecture overview, https://www.Trustedcomputinggroup.org, 2010.

[13] Intel TXT, http://www.intel.com/content/www/us/en/data-security/security-overview-general-technology.html, 2007.

[14] AMD SVM, http://www.anandtech.com/show/2480/9, 2007.

[15] C. Wang, H. Leung, S. C. Cheung, and Yumin Wang, Use of cryptographic technologies for privacy protection of watermarks in internet retails of digital contents, *Int. J. of High Performance Computing and Networking*, vol. 3, no. 5/6, pp. 385-394, 2005.

Shuang Xiang received the BS and MS degrees in information security from Wuhan University in 2006 and 2009 respectively. He is currently working toward the PhD degree at Wuhan University. His research interests include cloud computing and trusted computing.

Bo Zhao received the PhD degree in information security from Wuhan University in 2006. He is currently a professor in Wuhan University. His research interests include information security and trusted computing.

Yang An received the PhD degree in the State Key Laboratory of Information Engineering from Wuhan University in 2005. She is currently an associate professor in Wuhan University. Her research interests include information security and computer application.

Wei Tao is currently working toward the MS degree in Wuhan University. He received the BS degree from Wuhan Textile University in 2013. His research interests include information security and trusted computing.