Wang and Liu *EURASIP Journal on Information Security* (2018) 2018:1
https://doi.org/10.1186/s13635-018-0071-1

EURASIP Journal on Information Security

RESEARCH Open Access

A trusted measurement model based on dynamic policy and privacy protection in IaaS security domain

Liangming Wang* and Fagui Liu

Abstract

In Infrastructure as a Service (IaaS) environments, the user virtual machine is the user's private property. However, in the case of privacy protection, how to ensure the security of files in the user virtual machine and the user virtual machine's behavior does not affect other virtual machines; it is a major challenge. This paper presents a trusted measurement model based on dynamic policy and privacy protection in IaaS security domain, called TMMDP. The model first proposed a measure architecture, where it defines the trusted measurement of the user virtual machine into the trust of files in the virtual machine and trusted network behavior. The trusted measure was detected through the front-end and back-end modules. It then describes in detail the process of the trusted measurement in the two modules. Because the front-end module is in the guest virtual machine, it also describes the protocol to ensure the integrity of the module. Finally, the model proved to address security challenges of the user virtual machine in IaaS environments by a security analysis.

Keywords: IaaS, Privacy protection, Trusted measurement, Dynamic policy

1 Introduction

Recently, the application of cloud computing is becoming more and more popular. Cloud computing integrate separate information resources and supply on demand. It is on behalf of the information technology development trend towards intensification and large scale. Cloud computing consists of three levels of service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). As the lowest level of service in cloud computing, IaaS provide customers with the CPU resource, storage resource, network, and other basic computing resources to support the underlying functionality built on its various services. Compared to traditional data center, hosting, etc., the virtual machine's resources in an IaaS environment belong to a variety of tenancy, but outside of the virtual machine software and hardware, resources belong to IaaS operators. Trust between IaaS virtual machine operators and between users becomes very important.

The Trusted Computing Group (referred to as TCG) [1] proposed trusted computing technology, trying to provide an endpoint trusted for distributed computing platform. Trusted computing technology platform in computing hardware layer is introduced, trusted platform module (referred to as TPM), actually to provide a trusted root (root of trust, referred to as RoT) for computing platform. Based on the trusted root, using the trust chain delivery mechanism, trusted computing technology implements integrity measurement to the local hardware and software layer by layer. The measurement results are saved in the TPM platform configuration registers (referred to as PCR). Thereafter, remote computing platform via remote authentication mechanism (remote attestation) compare the local PCR measurement results in order to verify the trust of the local computing platform. Trusted computing technology gets rid of dependency on the central server for distributed nodes, directly through the TPM chip to build trust on the user's machine, to create better scalability, higher reliability, availability, and enhanced security of distributed applications platform.

* Correspondence: lmwang@scut.edu.cn
School of Computer Science & Engineering, South China University of Technology, Guangzhou, China

The core mechanism of trusted computing is remote attestation; each node in the cloud computing environment is by remote attestation mechanism to build mutual trust and to guarantee the security of application. The validity of remote attestation is based on integrity measurement. In the TCG specifications, it measures the integrity only from BIOS to operating system, not including the application layer. In fact, the trustworthiness of the application layer is critical to cloud computing security. Therefore, the researchers proposed many enhanced integrity measurement mechanisms, for instance, IMA (integrity measurement architecture) [2] and PRIMA [3]. But in IaaS environment, the user virtual machine is a private property, the privacy protection is very important in the process of integrity measurement.

In this paper, we proposed a trusted measurement model based on dynamic policy and privacy protection in IaaS security domain that can implement the effective trust measurement and protect the privacy of user's virtual machine.

The remaining part of this paper is organized as follows. Section 2 describes background. Then, the architecture of TMMDP is given in Section 3. Section 4 describes the detailed design and implementation of the model. Section 5 describes integrated measurement algorithm. Section 6 analyses the security of the model. Finally, we summarized the paper and outline the future work.

2 Background

2.1 IaaS

Comparing to SaaS and PaaS, as the lowest level of service in cloud computing, IaaS provide the basic computing resources to clients. Typical applications include Amazon EC2, VMWare, and Google Compute Engine (GCE). In this paper, we focus on that IaaS provide virtual machine service to public users.

Our IaaS platform built on XenServer, which is an open source IaaS platform. Based on the powerful open source Xen Project Hypervisor [4, 5], XenServer provides efficient management of Windows and Linux Virtual Machines (VMs) and delivers an extremely cost-effective platform for application, desktop, and server consolidation.

2.2 Bayesian theory-based trust model

Bayesian inference is a kind of conditional inference. People know cognition, processing of probability information and its law more deeply through discussion and exploration in the field, directing people to learn and make decision more effectively. The Bayesian theory is applied to the trust model, providing solid theoretical foundation for calculation of degree of trust, predicting future possible results through prior probability and new measurement. The most typical distributions in Bayesian family include Beta and Dirichlet distribution.

Beta distribution is typically used to describe binary measurement system <α, β>. The degree of trust is expressed through expected value of probability distribution function of α and β Beta distribution.

$$\mathrm{f}(p|\alpha,\beta)=\frac{p^{\alpha-1}(1-p)^{\beta-1}}{\int x^{\alpha-1}(1-u)^{\beta-1}dx},\alpha\geq 0,\beta\geq 0 \tag{1}$$

$$\mathrm{f}(x)=\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1},0\leq x\leq 1,\alpha\geq 0,\beta\geq 0 \tag{2}$$

Mathematical expectation:

$$\mathrm{Exp}(p|\alpha,\beta)=\frac{\alpha}{\alpha+\beta} \tag{3}$$

Beta distribution is only suitable for binary measurement system, and Dirichlet distribution is suitable for multiple measurement result.

Suppose that a measurement has k results and total n interaction, where every interaction has a measurement result and the number of measurement in No. m ($m = 1, 2,\ldots, k$) is n_{m}. Then, the posterior probability distribution of estimated parameter p is:

$$\mathrm{f}(p,n,k)=\frac{1}{\int_0^1\prod_{m=1}^{k}x^{\left(n_m+\frac{C}{k-1}\right)}dx}\prod_{m=1}^{k}p_m{}^{\left(n_m+\frac{C}{k-1}\right)} \tag{4}$$

Mathematical expectation:

$$\mathrm{Exp}(p_{\mathrm{m}})=\frac{\mathrm{n_m}+\frac{\mathrm{C}}{\mathrm{k}}}{\mathrm{C}+\sum_{\mathrm{m}=1}^{\mathrm{k}}\mathrm{n_m}} \tag{5}$$

2.3 Security challenges of virtual machines in IaaS security domain

Users' virtual machines are their private property, just like the property stored in a safety deposit box in a bank, which cannot be accessed by any other person or authority. But because it is stored on public IaaS environments and therefore must abide by certain rules, you cannot install programs that do not meet the requirements as these might pose a security risk. In addition, users in the virtual machine cannot have security-related effects on other clients' virtual machines.

2.4 Integrity measurement and remote attestation

Integrity measurement is the foundation of remote attestation. In TCG specifications, integrity measurement is defined as the process of obtaining metrics of platform characteristics that affects the integrity (trustworthiness) of a platform and putting digests of those metrics in PCRs. Based on the integrity measurement, the TCG solution of remote attestation for platform

authentication is sometimes called binary attestation. The advantages of binary attestation process are simple and reliable, with no other trusted third party involved. IMA is the solution of IBM to the binary attestation [2]. As the same as IMA, [6–8] are typical of static measurement method. To solve the problem of TOCTOU [9, 10], a dynamic measurement method is proposed.

A more flexible extension to the binary attestation is property-based attestation (PBA): on a higher system level, attestation should only determine whether the system has a desired property [11–14]. PBA verifies the integrity of a system with regard to certain policies.

Traditional remote attestation focused on the single physical machine. The cloud computing and IaaS environment [15–23] presented many new solutions. Many of these solutions did not care for the user's privacy. Our solution mainly focuses on the public IaaS cloud and privacy protection.

3 TMMDP overview

3.1 IaaS security domain model and entities included

The IaaS in this security domain architecture is shown in Fig. 1, and our security domain model contain the following entities:

- Security Management Server (SMS): The server is responsible for the safety management in the security domain.
- Entity Server (ES): The physical server hosts user's virtual machines.
- Management Virtual Machine (MVM): Running on the hypervisor and directly interacting with physical hardware. In Xen environment, it is commonly referred to as Dom0.
- User Virtual Machine (UVM): User's virtual machine, in Xen environment, is commonly referred to as DomU.
- User: Owner of one or more virtual machines in a security domain.

3.2 Measurement architecture design

The trust measurement architecture includes the front-end module in a user's virtual machine and the back-end module in a management virtual machine. Because it is running on a hypervisor on the same physical host, front-end and back-end measurement modules are respectively the security drivers by being the communication channel of communication within the hypervisor, as shown in Fig. 2.

The back-end measurement module setup is on Dom0, so it is controlled on the IaaS platform. But as the front-end measurement module is run on the client's virtual machine, there is a risk of being tampered with, so you need to measure real-time monitoring front-end module integrity to ensure that the front-end measurement module of the returned data is authentic.

Back-end measurement module includes front-end module integrity assurance and network behavior measurement in two parts. Network behavior measurement includes network behavior policy receiving, network behavior analysis, and network trust report. Network behavioral information is collected by a completed netfilter module. Front-end measurement module includes VM policy receiving, fingerprint lib of local files, and the local files' trust report.

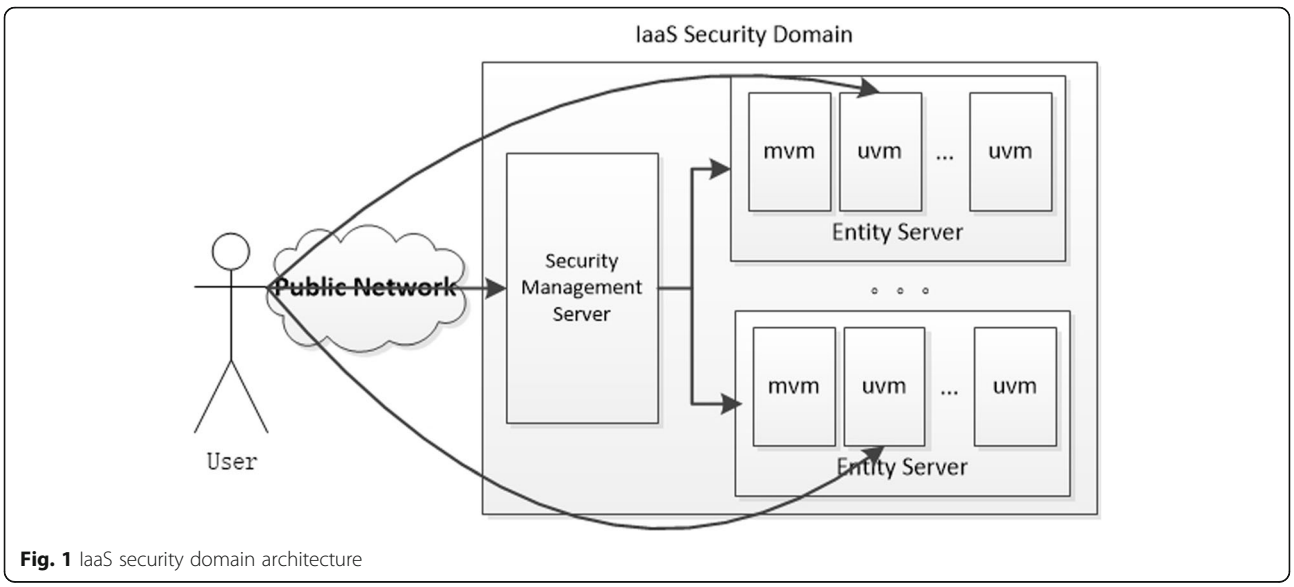

Fig. 1 IaaS security domain architecture

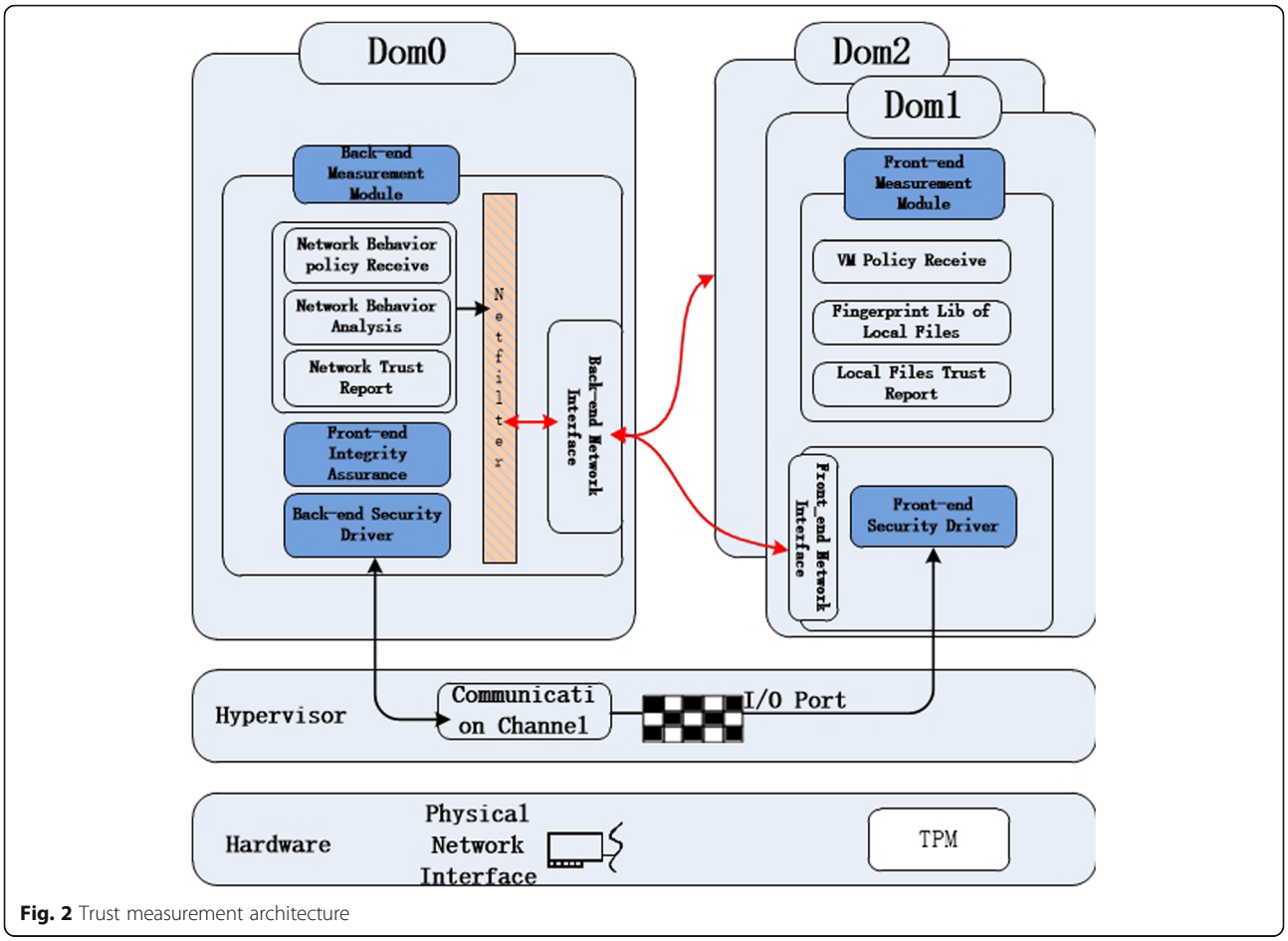

Fig. 2 Trust measurement architecture

4 Design of TMMDP

4.1 Measurement scheme to user's virtual machine based on privacy protection

4.1.1 Measurement strategy

The trust measurement to UVM includes security detection of local files and network behavior. File security for UVM files meet the security standard of IaaS when all file sources are the credible. Network behavior security is that UVM can not affect other users, especially a security threat to other VMs of local IaaS.

For the trust measurement of files, in this paper, we divide UVM files into three types, including system files, application files, and data files. For data files, we will not check the security; we focus the trust measurement to the system and application files. Behavior security check focuses on interacting with the outside, not including the VM internal behavior.

Check time: File security checking is done when there is a change in the system or application files while network behavior checking when VM interacts with the outside.

4.1.2 Measurement processes

Definition

1. Measure policy: It includes measuring module measures on a different file or network event timing and the frequency of measurement requirements. According to the different measurement results, the measure policy is updated dynamically.
2. Fingerprint lib of files: In the hash value lib of files, measurement module should synchronize between SMS and local. For personal application file, you should validate by the trusted third party and submit the hash value to SMS.
3. Trust report: Including the trust report of files or network behavior without privacy. The report is an important basis to generate dynamic strategy.

The initialization process

1. Front-end measurement module synchronizes measurement policy of virtual machine files from the Security Management Server

2. Front-end measurement module synchronizes fingerprint lib of files from the Security Management Server
3. Back-end measurement module synchronizes measurement policy of network behavior from the Security Management Server

The process of local files trust measurement

1. Register listening events of file modification (ensure that all changed files can be measured)
2. Generate measurement list of files based on the measurement policy of virtual machine files (including file name and measure frequency)
3. Determine the measurement timing of files in the list according to the measurement policy
4. Calculate the hash value of the file
5. Find the fingerprint database, checking whether the calculated hash value exist
6. Calculate the file integrity trust value by the file integrity trust measurement algorithm in 5.1
7. Generate trust report
8. Encrypt trust report and upload it

The process of network behavior trust measurement

1. Register listening events of network data packets transform
2. Generate monitoring scheme according to the measurement policy (IP and port, measure frequency, etc.)
3. Calculate the network behavior's trust value by the network behavior trust measurement algorithm in Section 5.2
4. Generate trust reports on monitoring results
5. Encrypt trust report and upload it

4.2 The integrity assurance protocol of front-end measurement module

1. Generate symmetric key by TPM of MVM and distribute them to back-end measurement module in MVM and front-end measurement module in UVM
2. Generate nonce by TPM of MVM and send integrity measurement command with nonce to front-end measurement module
3. Front-end module calculate the hash value of self, encrypt the value using a secret key and send it to MVM
4. MVM decrypt the value and compare it to the standard value
5. Generate nonce by TPM of MVM and send confused command with nonce to front-end measurement module
6. Front-end module generate random string and send to MVM

5 Integrated measurement algorithm

5.1 File integrity trust measurement

During the measurement, file integrity has only two results: integrity and non-integrity, we use Beta distribution to describe it. Let m represent the number of measurement result of integrity and n represent the number of measurement of non-integrity. As in Eq.(3), P represents the probability of measurement result of non-integrity, let $\alpha = m + 1$ and $\beta = n + 1$. When 10 files are integrity measured, if eight results are with integrity, two results are with non-integrity, then the probability density distribution of probability p that measured the result as with integrity is $f(p|(8+1),(2+1) = f(p|9,3))$, according to the mathematical expectation of Beta distribution equation $\mathrm{Exp}(p|9,3) = 0.75$. Here, 0.75 represents that the probability of integrity of the file during measurement is 0.75.

During the actual measurement of our system, we divide the files into system file and application file; typically, we think that the influence of the system file on safety is greater than that of the general application file, as system file brings safety risk more possibly than general application file does when non-integrity occurs. So, a weighing factor μ ($\mu >= 1$) is added when non-integrity occurs in the system file. M_{sf} represents the number of system file measurement results as being with integrity, m_{af} represents the number of application file measurement results as being with integrity, n_{sf} represents the number of system file measurement results being non-integrity, n_{af} represents the number of application file measurement results as being with non-integrity, and T_{f} represents the file integrity trust value. So,

$$a = m_{\mathrm{sf}} + m_{\mathrm{af}} + 1, b = \mu \times n_{\mathrm{sf}} + n_{\mathrm{af}} + 1 \tag{6}$$

According to Eq. (3), it can be known that the file integrity trust value is:

$$T_f = \frac{m_{\mathrm{sf}} + m_{\mathrm{af}} + 1}{m_{\mathrm{sf}} + m_{\mathrm{af}} + \mu \times n_{\mathrm{sf}} + n_{\mathrm{af}} + 2} \tag{7}$$

It can be found from the above equation that when measuring file integrity, results of with integrity and non-integrity have the same influence on results of degree of trust, but actually, the measurement result of non-integrity has influence on the degree of trust far greater than the results of integrity. We need to introduce a penalty mechanism when a measurement result is of non-integrity. When a measurement result is of non-integrity, counting adopts exponential function with e as the base number, and an exponential part needs to consider proportional relation between the number of

measurement as being of non-integrity and integrity. The final penalty mechanism added trust value calculation equation is:

$$\mathrm{T_f} = \frac{m_{\mathrm{sf}} + m_{\mathrm{af}} + 1}{m_{\mathrm{sf}} + m_{\mathrm{af}} + e^{(\mu \times n_{\mathrm{sf}} + n_{\mathrm{af}})\left(1 + \frac{\mu \times n_{\mathrm{sf}} + n_{\mathrm{af}}}{\mu \times n_{\mathrm{sf}} + n_{\mathrm{af}} + m_{\mathrm{sf}} + m_{\mathrm{af}}}\right)} + 2} \tag{8}$$

5.2 Network behavior trust measurement

Network behavior trust measurement has three results: legal, illegal, and uncertain, according to Dirichlet distribution. As in Eq.(5), k value is 3; in addition, the selected value of constant C is determined as 3, the same as that of k. So, the corresponding probability distribution function is:

$$\mathrm{f}(p, n, 3) = \frac{1}{\int_0^1 \prod_{m=1}^3 x^{\left(n_m + \frac{3}{2}\right)} dx} \prod_{m=1}^3 p_m^{\left(n_m + \frac{3}{2}\right)} \tag{9}$$

The corresponding mathematical expectation is:

$$\mathrm{Exp}(p_m) = \frac{n_m + 3/2}{3 + \sum_{m=1}^3 n_m} \tag{10}$$

n_1,n_2, and n_3 represent the numbers of network behavior detection result being legal, illegal, and uncertain, respectively. T_n represents the trust value of the network behavior. So, according to Eq. (10), it is known that network behavior trust value is:

$$T_n = \mathrm{Exp}(p_1) = \frac{n_1 + 3/2}{3 + \sum_{m=1}^3 n_m} \tag{11}$$

Namely:

$$T_n = \frac{n_1 + 3/2}{3 + n_1 + n_2 + n_3} \tag{12}$$

5.3 Overall comprehensive weighing

Trust measurement value of virtual machine in IaaS consists of file integrity trust measurement value (T_f) and network behavior trust measurement value ($\mathrm{T_n}$); overall trust measurement value of virtual machine is obtained by using the simple weighing:

$$T = a \times T_\mathrm{f} + b \times T_\mathrm{n} \tag{13}$$

Put them into Eqs. (8) and (12) to obtain the overall trust measurement value formula after the simple weighing:

$$T = a \times \frac{m_{\mathrm{sf}} + m_{\mathrm{af}} + 1}{m_{\mathrm{sf}} + m_{\mathrm{af}} + e^{(\mu \times n_{\mathrm{sf}} + n_{\mathrm{af}})\left(1 + \frac{\mu \times n_{\mathrm{sf}} + n_{\mathrm{af}}}{\mu \times n_{\mathrm{sf}} + n_{\mathrm{af}} + m_{\mathrm{sf}} + m_{\mathrm{af}}}\right)} + 2} + b \times \frac{n_1 + 3/2}{3 + n_1 + n_2 + n_3} \tag{14}$$

6 Security analysis of the model

TMMDP check the security of local files by means of installing the front-end trust measurement module in the user's virtual machine and check security of network behavior by means of monitoring the network data packet in MVM. The model focus on the security challenge proposed in Section 2.2.

Because the front-end measurement module is installed on the user's virtual machine, the user has full control over the entire virtual machine, thus ensuring that the integrity of the front-end measurement module is the front-end measurement basis for security. The integrity assurance protocol of front-end measurement module effectively ensures the integrity of the front-end measurement module. In the protocol, the security key and random numbers are generated by TPM of entity server and confused commands are mixed with normal commands.

By adding the listener in the event of creating or updating a file, the model measures all of the modified files. To other files, the model will measure them at random on different frequencies according to the policy scheme, in case of failure in listening events. The model will adjust dynamically the measurement frequency of files based on the measurement result. Fixed measure based on event listening and measurement dynamically at random will guarantee the integrity of the user's files.

In IaaS platform, the user's virtual machine communicate by the network driver installed on the management virtual machine. We monitor the network data in the back-end measurement module by a netfilter module and an analysis of the security of network behavior, ensuring the security of other users.

Because the front-end measurement module run on the user virtual machine, the entire measurement process runs guest virtual machine environments. Front-end measurement modules documented the results of measuring trust in the report on the report server, the report does not relate to a specific measurement process. IaaS platform as a whole does not know the customer-specific files in a virtual machine, effectively protecting the customer's privacy.

Through analysis we can see that (1) measurement procedure does not expose user's privacy about files in virtual machine, (2) a model can monitor user file integrity in virtual machines to prevent the presence of an

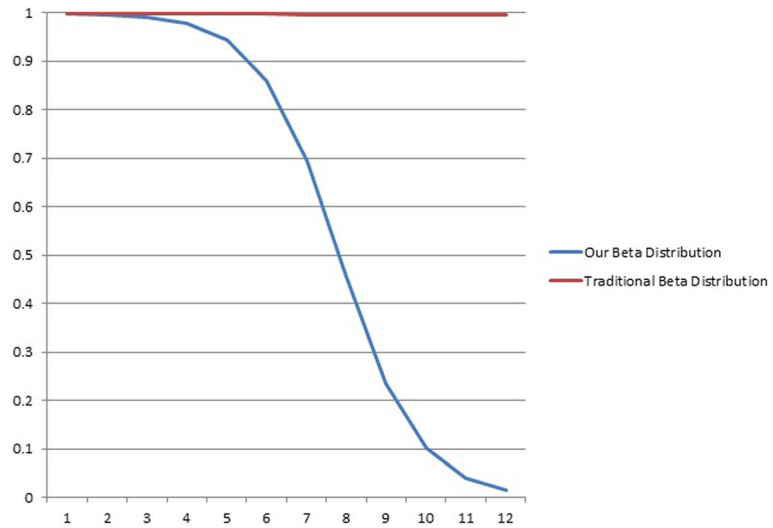

Fig. 3 Beta distribution-based file trust measurement

illegal file, and (3) a model can detect violations of network behavior to other IaaS users' security.

In order to assess the degree of trust measurement model proposed in the paper, we simulate the environment with multiple untrusted files on virtual machine of IaaS. Because Beta distribution and Dirichlet distribution have the same characteristics, here, we mainly conduct an experimental analysis of Beta distribution-based file integrity trust measurement.

6.1 Beta distribution-based file trust measurement result

The experiment set basic data as having 500 system integral files and 2000 application integral files.

As in Fig. 3, it can be seen that the file trust measurement curve directly using traditional Beta distribution varies very slowly, while trust measurement curve using Beta distribution improved by us varies more quickly. It also more conforms to the characteristic that the integrity of few files can have greater influence on trust in virtual machine.

6.2 Analysis of influence of integrity of different type of files on trust measurement

The experiment set basic data as having 500 system integral files and 2000 application integral files.

In Fig. 4, T1 represents variation of degree of trust when the measured incomplete file is an application file,

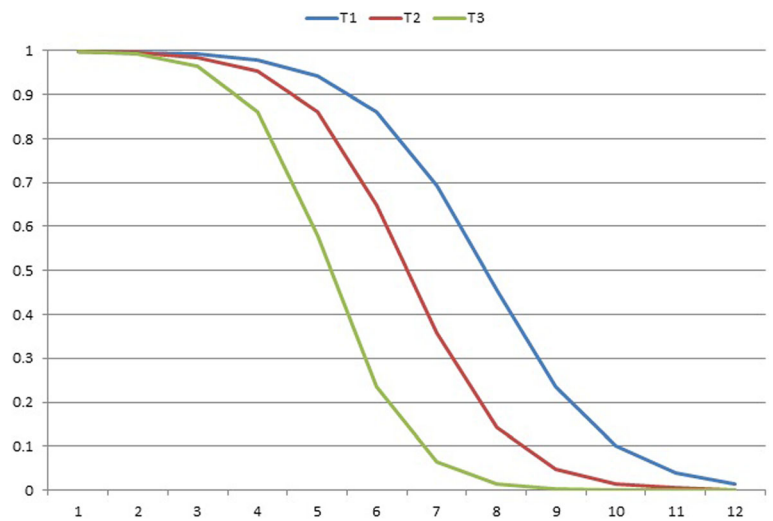

Fig. 4 Comparison of influence of system file and application file on trust

T2 represents that when the measured incomplete file is a system file and the importance of weighing factor of μ is taken as 1.2, and T3 represents that when the measured incomplete file is a system file and the importance weighing factor of μ is taken as 1.5.

As in Fig. 4, it can be found that the greater the system file importance weighing μ value is, the more the degree value of trust decreases dramatically, which also indicates that the more important a system file is, the greater the influence on degree of trust when system file measurement is incomplete.

7 Conclusions

This paper analyzes virtual machine security challenges encountered in an IaaS environment, including file security and network behavior to other IaaS virtual machine security. TMMDP, a trusted measurement model based on dynamic policy and privacy protection in IaaS security domain, is proposed, divided into front-end and back-end measure modules in the model, front-end measurement module to detect virtual machine file security and back-end measurement module to detect network behavior security. By the security analysis to the model, it achieves the desired results.

In the future, we will enhance the scheme about generating policy in the security management server. Therefore, we will improve the existing trust report on the basis of protecting the privacy and calculate a more precise level of trust by the trust report.

Acknowledgements
We must thank the effort of the reviewers and the editors.

Funding
This paper is supported by the key production-study-research combination project of 2012 Guangdong Province(project number: 2012B091000109), the second batch of core technology research project on strategic emerging industry of Guangdong Province(project number: 2012A010701005), and the Fundamental Research Funds for the Central Universities(project number:2012ZM0051).

Availability of data and materials
We can provide it.

Authors' contributions
WL is the corresponding author of the paper. LF contributed the idea of the paper. Both authors read and approved the final manuscript.

Authors' information
Wang Liangming is a Lecturer in South China University of Technology, China. His main research interests include network security and the Internet of things. Liu Fagui is a professor in South China University of Technology, China. She received the Ph.D. degree from South China University of Technology. Her main research interests include data mining, semantic web, and the Internet of things.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 1 November 2017 Accepted: 12 February 2018
Published online: 23 February 2018

References

1. Trusted computing group, TCG Specification Architecture Overview version 1.4, [2008–04-20]. https://trustedcomputinggroup.org/about/.
2. Sailer, R, Zhang, X, Jaeger, T, van Doorn, L (2004). Design and implementation of a TCG-based integrity measurement architecture. In *Proceedings of the 13th USENIX Security Symposium*.
3. Jaeger, T, Sailer, R, Shankar, U (2006). Prima: policy-reduced integrity measurement architecture. In *Proceedings of the 2007 ACM workshop on scalable trusted computing (SACMAT '06)*.
4. Barham, P, Dragovic, B, Fraser, K, et al. (2003). Xen and the art of virtualization. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles(SOSP '03), New York, USA*, (pp. 164–177).
5. Pratt, I, Fraser, K, Hand, S, et al. (2005). Xen 3.0 and the art of virtualization. XEN 3.0 and the art of virtualization. In *Proceedings of the Linux symposium*, (pp. 65–77).
6. Li Xiaoyong, Han Zhen, Shen Changxiang. Transitive trust to executables generated during runtime. Proceedings of ICICIC2007, Washington DC: IEEE Computer Society, 2007:518–521.
7. Yang, Y, Huanguo, Z, Wan, L, et al. (2008). Design and implementation of an integrity measurement system based on windows trusted computing platform. In *Proceedings of the 9th International Conference for Young Computer Scientists*, (pp. 229–233). Washington DC: IEEE Computer Society.
8. Shi, E, Perrig, A, Van Doorn, L (2005). BIND: a fine-grained attestation service for secure distributed systems. In *Proceeding of the IEEE symposium on security and privacy*, (pp. 154–168). Oakland: IEEE Press.
9. Loscocco, PA, Wilson, PW, Pendergrass, JA, et al. (2007). Linux kernel integrity measurement using contextual inspection. In *Proc of STC 2007*, (pp. 21–29). New York: ACM.
10. Thober, M, Pendergrass, JA, Mcdonell, CD. (2008). Improving coherency of runtime integritymeasurement. ACM Workshop on Scalable Trusted Computing, Stc 2008, Alexandria, Va, Usa, October (pp. 51–60). DBLP.
11. Kuhn, U, Selhorst, M, Stuble, C (2007). Realizing property-based attestation and sealing with commonly available hard- and software. In *ACM STC 2007*, (pp. 50–57). ACM.
12. M. Manulis and M. Steiner. UPBA: User-authenticated property-based attestation, PST 2011. Full Version.
13. Nagarajan, A, Varadharajan, V, Hitchens, M, Gallery, E. (2009) *Property Based Attestation and Trusted Computing: Analysis and Challenges*. International Conference on Network and System Security. IEEE, pp. 278–285.
14. Feng, DG, Yu, Q. (2010). A property-based attestation protocol for TCM. Science China Information Sciences. **53**(3), 454–464.
15. Shen Changxiang. System behavior based trustworthiness attestation for computing platform, 2007
16. Wang, C, Wang, Q, Ren, K, Lou, W (2010). Privacy-preserving public auditing for data storage security in cloud computing. In *IEEE INFOCOM*.
17. De Souza, WAR, & Tomlinson, A (2015). SMM-based hypervisor integrity measurement. In *2015 Int. Conf. On cyber security and cloud computing*, (pp. 362–367).
18. Mei, S, Wu, J, Cheng, Y, Ma, J, Ren, J, Li, X (2011). Trusted bytecode virtual machine module: towards dynamic remote attestation in cloud computing. In *Proc. - 2011 Int. Symp. Intell. Inf. Process. Trust. Comput. IPTC 2011*, (pp. 19–23).
19. Awad, A, Kadry, S, Lee, B, Zhang, S (2014). Property based attestation for a secure cloud monitoring system. In *Proc. - 2014 IEEE/ACM 7th Int. Conf. Util. Cloud Comput. UCC 2014*, (pp. 934–940).
20. Berger, S, Goldman, K, Pendarakis, D, Safford, D, Valdez, E, Zohar, M (2015). Scalable attestation: a step toward secure and trusted clouds. In *Proc. - 2015 IEEE Int. Conf. Cloud Eng. IC2E 2015*, (pp. 185–194).
21. Pawloski, A, Wu, L, Du, X, Qian, L (2015). A practical approach to the attestation of computational integrity in hybrid cloud. In *2015 Int. Conf. Comput. Netw. Commun. ICNC 2015*, (pp. 72–76).
22. Rajendran, VV, & Swamynathan, S. (2016). Hybrid model for dynamic evaluation of trust in cloud services. *Wirel. Netw*, *22*(6), 1807–1818.
23. Manzoor, S, Taha, A, Suri, N. (2017). Trust Validation of Cloud IaaS: A Customer-centric Approach. Trustcom/bigdatase/ispa. IEEE. pp 97–104