Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (22): 5092-5098, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.5092.5098
© 2013 Asian Network for Scientific Information

On Remote Attestation Based on Trusted Cloud Computing

Yong Zhao and Pei-Yong Cong
College of Computer Science, Beijing University of Technology, Beijing 100124, China

Abstract: Current studies on remote attestation for cloud computing neither consider the computational nodes as a group nor support dynamic trusted verification. To overcome these shortcomings, we put forward an attestation based on Identity and State. This new scheme can prevent jointly attack so as to have better security than existing ones. We concluded that effective rate of our scheme ascends as the threshold value increases.

Key words: Remote attestation, trusted computing, threshold cryptography, bilinear pair

INTRODUCTION

As the information technology develops, cloud computing becomes the most attractive word after personal computer and Internet (Dahbur *et al*., 2011). IaaS (Infrastructure as Service) lies in the base of the architecture of cloud computing which is the basics of higher lays such PaaS (platform as service) and SaaS (software as service). So the security of IaaS determines whether the whole service of cloud computing (Brassil, 2010). In this paper, we focus on the core technologies of remote attestation in the environment of IaaS. The typical IaaS services are like Elastic Computing proposed by Amazon, blue cloud by (International Business Machine, 2008), cloud platform by Sun (Sun Microsystems, 2009) and so on. IaaS is able to provide all kinds of virtual machine instances with various computational capabilities. User virtual machine runs on cluster computer owned and maintained by cloud service provider which lowers the cost of computation of either a user or a company. In this new computing paradigm, data and computing are outsourced to remote computer that is not controlled by a user. Naturally, the user concerns the trustworthiness and security of both his data and computing environment very much. So it is an urgent need for cloud service provider to prove to a user that his computing environment and stored data are in trusted state.

Production information system is referred to that kind of information system that serves specific organizer. High requirements on trustworthiness and security are the fundamental characteristics (Shen, 2004). Additionally, in production information system, for example of national defense, the behaviors of users are fixed usually; the system manager is trusted by default. How to adapt cloud computing to production information system is not studied too much. In this study, we conduct the deep research.

Based on identity and state of a virtual machine, our scheme focuses on individuals in the group of virtual machines in cloud computing. We consider virtual machines as a whole. Similar to a person in a group, what every virtual machine does has influence on the whole group definitely. To be explicit, a malicious virtual machine who does frequent remote attestation to outside surely affects the reputation of its whole group.

RELATED WORK

The literature, Dawoud *et al*. (2010) analyzes the potential security problems existing in current IasS and then puts forward an improved security scheme. In this scheme, the security of IAAS is enhanced in terms of hardware, network connection, platform virtualization, software for cloud computing, utility computing, service level agreement etc. For example, there are 5 malicious virtual machines in a trusted group that consists of 100 virtual machines in cloud computing.

The literature, Santos *et al*. (2009) introduced trusted computing into IaaS firstly and then proposed a concept called trusted cloud computing platform. It can be guaranteed that all virtual machines are trusted using configuration based remote attestation. However, this scheme can only prove the static state of the latest restart while cannot handle the dynamic attacks such as buffer overflow, DMA attack and so on. Moreover, Endorsement Key of TPM is used to sign which leads to leakage of privacy because the verifier is able to track the usage of Endorsement Key.

In the literature, Goldman *et al*. (2010), to support remote attestation for virtual machine, virtual TPM is improved to update attestation by the means of the

Corresponding Author: Yong Zhao, College of Computer Science, Beijing University of Technology, Beijing 100124, China

following events such as changing, updating, patching the configuration of virtual platform and so on. However, it is actually a static remote attestation based on configuration while it cannot attest the running states of virtual platform. Additionally, this scheme only deals with the trust root based on software and lacks both trusted guarantee provided by TPCM and attestation of physical platform on which the virtual machines are running.

The literature, Schiffman *et al*. (2010) proposes a trusted attestation scheme for cloud computing based on validation centre where the trustworthiness of physical platform in cloud computing can be validate. However, this scheme is also based on the static configuration of platform, that is, the dynamic properties of platform cannot be checked.

Overall, the following two shortcomings exist if we directly apply the current remote attestation schemes in IaaS into the trusted cloud computing for production information system:

- Lack of study on the trusted group in IaaS. Current research focuses on attestation of a certain virtual machine in cloud computing, it is neglected to consider the virtual machine as a group
- Lack of study on dynamic trusted attestation in computing platform of IaaS. Current attestation for static or startup configuration only proves firmware and software configuration involved in the latest startup of computing platform. Challenger pays more attention to whether the platform is trusted when it running

To cope with the problems mentioned above, this study conducts a thorough study on the remote attestation for the startup of virtual machine. In our scheme, dynamic measurement over running platform is taken into account. And then, based on the results of dynamic measurement, we investigate the impact of computing-node group on the individual virtual machine in terms of trustworthiness. When it comes to the other types of remote attestation in cloud computing such as management of nodes, the migration of virtual machine, they are similar to the one for startup of virtual machine. Due to the space limitation, we do not give the detailed description.

SCHEME DESIGN

In our scheme of remote attestation, the basic framework of remote attestation in literature (Santos *et al*., 2009) is adopted. Further, trusted coordinator is responsible that honest user's identity is bound with security attributes of its corresponding state. After the startup of trusted platform, the important system services, data file and application program are measured before they are executed. Measurement process and results are stored in the logs of TPM while the cumulative digest is stored in the extended PCR. To be intuitive, the concept of virtual machine in the literature (Santos *et al*., 2009) is hereinafter referred to virtual computing node.

The following are the steps when a virtual computing node that attests itself to outside initiator after receiving remote attestation request. Firstly, the virtual computing node measures itself according to the trustworthiness function $E(x_i)$ presented in the following sub-section called state evaluation function. Secondly, it broadcasts the value of its trustworthiness function to anyone who interacts with itself while it receives the values of trustworthiness from virtual computing nodes that interact with it. Thirdly, based on aforementioned two steps, this virtual computing node cooperates with at least t interacted virtual computing nodes to conduct remote attestation. To be explicit, this virtual computing node gets the values of trustworthiness function of at least t interacted virtual computing nodes to calculate the value of state evaluation function. Moreover, the number t is threshold value for our scheme. Fourthly, the initiator determines whether to trust the virtual computing node after receiving evaluation result. In Fig. 2, the procedures of remote attestation based on identity and state is described intuitively in Fig. 1.

Parameter setup: G_a, G_b are defined as multiplicative group whose order is p and g is the generator of G_a. Bilinear map is $e(G_a, G_a) \rightarrow G_b$ where $e(g, g) = I$, H_d: $\{0, 1\}^{lid} \rightarrow G_a$. So, the public parameter is $(G_a, G_b, g, P, e(g, g) = I, H_d)$. $S = \{S_1, S_2, \ldots\ldots S_n\}$ is defined as a set of n trusted entities where ID_{si} is defined as public information of signer S_i. Every trusted entity can choose $\gamma_i \in_R Z_p^*$ to calculate $Z_{si} = g^{\gamma i}$.

Key generation: For each entity, a function $F_{si}(x) = a_{si}0x + a_{si}1\ p_0^{\times} \bmod q$ is constructed for each entity, where $a_{si} \neq 0$ and p_0 indicating a small prime. For the rest of $S_j (j \neq i)$, $f_{si}(ID_{sj}\text{-}ID_{si})$ and the check value of $g^{fsi,\,m} \bmod p$ $(m = 0, 1)$ are calculated. Then, the computational results and $Z_{si} = g_{\gamma i}$ is broadcast to everyone in the set S by means of authentication. For every S_j who receives both $f_{si}(ID_{Sj}\text{-}ID_{si})$ and Z_{si}, $= g^{\gamma i}$ it verifies whether Z_{si} match with a certain value in the list of malicious virtual computing nodes. If match, S_j will terminate the interaction with S_i. If mismatch, it verifies whether the equation of $g^{fsi(ID_{sj}-ID_{si})} = (g^{asi0})^{(ID_{sj}-ID_{si})}(g^{asi1})^{p_0^{(ID_{sj}-ID_{si})}}$ holds. If correct, it means $f_{si}(ID_{sj}\text{-}ID_{si})$ is valid. If incorrect, $f_{si}(ID_{si}\text{-}ID_{si})$ is invalid. If

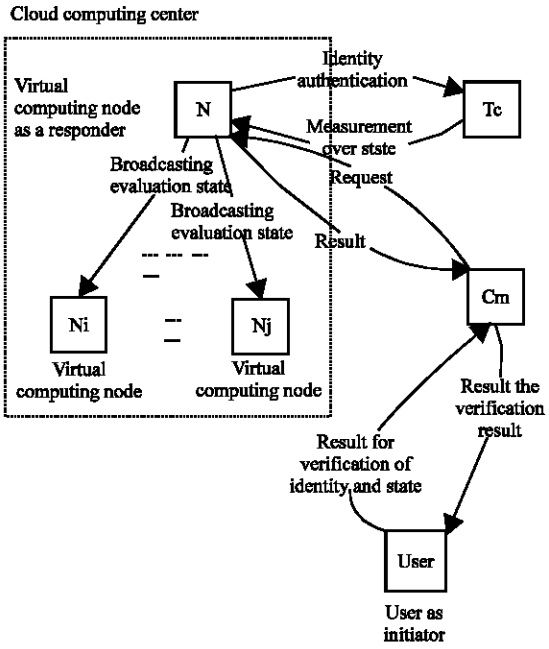

Fig. 1: Process for remote attestation

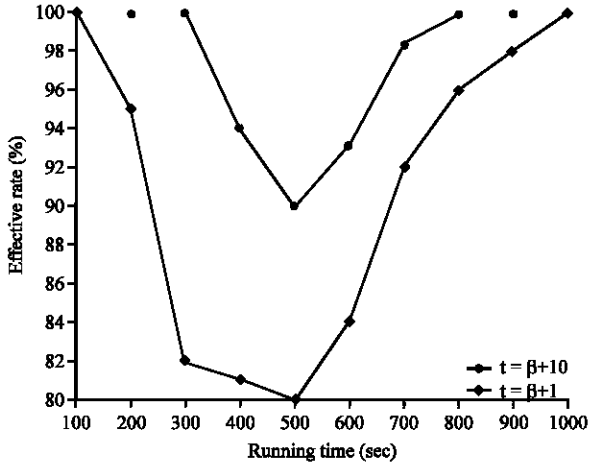

Fig. 2: Effective rate when β% = 15%

everyone completes the aforementioned verification process and receives computational information of t participants, every S_i computes its private key denoted as:

$$d_{si} = u_i + \sum_{j=1, j\neq i}^{t} f_{si}(ID_{sj} - ID_{si})$$

where, $u_i \in G_a$ is chosen from G_a. Moreover, the corresponding public key is $Pk_i = g^{dsi}$.

Identity verification: Fisrt of all, responder knows $Pk_i = g^{ds}$ that is public key of initiator's platform. The first Responder sends $Pk_i = g^{dsi}$ to initiator.

Second, the initiator chooses $e \in_R Z_p^*$ to send it to responder. Then, the responder sends both:

$$T = g^{\frac{1}{d_{si}+r_c-r}}$$

and $R = g^{r_c}$ to the initiator, where $r_c \in_R Z_P^*$ is chosen by responder.

Third, the initiator verifies whether the equation of $e(T, Rg^{-r} PK) = I$ holds to obtain whether the responder is honest.

STATE EVALUATION FUNCTION

TC (trusted coordinator) is able to obtain the snapshot of the whole group of virtual computing nodes as responder which is used to achieve dynamic measurement over virtual computing node. The different roles virtual computing nodes play in the whole group can be classified into three types which are trusted coordinator, virtual computing node as responder and virtual computing node as client respectively. The evaluation rating of a virtual computing node as responder in the group can be classified into four categories which are very trusted, general trusted, critical trusted and untrusted, respectively. From the angle of human society, how trustworthy a person is related to how trustworthy every person around him. This theory is applied to virtual computing nodes as responder in cloud computing. So, the virtual computing nodes interacted

with the responder are taken into account when we study a virtual computing node that attests itself to outside initiator in cloud computing. The realization of our scheme is based on the measurement concept (Chang *et al*., 2006). The trustworthiness function in (Chang *et al*., 2006) is denoted as $E(x_i)$ in our scheme to evaluate the virtual computing node x_i using the corresponding snapshot.

$x_j, x_i+1 \ldots\ldots x_m$, represent the virtual computing nodes who interact with the responder denoted as $x\pi$ while their corresponding evaluation values are $\{E_j, E_j+1 \ldots\ldots E_m\}$, respectively.

Let $\bar{E}$ be the average of evaluation values for the whole system. Let Se: $x \rightarrow [0, 1]$ be evaluation function to evaluate the instant state of a virtual computing node according to the group of virtual computing nodes that interacts with it. We have that:

$$Se(x) = \frac{1}{m}(\sum_{i=j}^{m}(\bar{E} - E_i)^2$$

Further, let Evaluation: $x \rightarrow [VT, GT, CT, UT]$ be the function of evaluation result for virtual computing node $x\pi$. The value of Evaluation are fore types which are Very Trusted (VT), General Trusted (GT), Critical Trusted (CT) and Untrusted, respectively (UT):

$$Evaluation = \begin{cases} VT, E_2 \le Se(x) < E_3 \\ GT, E_1 \le Se(x) < E_2 \\ CT, E_0 \le Se(x) < E_1 \\ UT, 0 \le Se(x) \le E_0 \end{cases}$$

where, $0 \le E_0 \le E_1 \le E_2 \le E_3 \le 1$. In addition, E_0, E_1, E_2, E_3 are set by the manager of trusted coordinator according to the security policy.

According to the value of Evaluation (x), we can decide in which state virtual computing node $x\pi$ is. Additionally, Evaluation is broadcast to all members in the group.

STATE VERIFICATION

- Anyone of t-1 virtual computing nodes, together with the virtual computing node as responder, can sign on Evaluation. Without loss of generality, let the t-1 virtual computing node be $S_1, S_2 \ldots S_{t-1}$ and the responder as S_t. First of all, S_i selects a private secret integer written as y_{si} where $S_i \in \{S_1, S_2 \ldots S_t\}$ and y_{si}. Using y_{si}, we can calculate $Y_{si} = g^{y_{si}}$ and $W_{si} = g^{z_{si}+y_{si}-1}$ where Z_{si}^{-1} is inverse element of Z_{si} in prime field GF(ps). Then, both D_{si} and W_{si} are broadcast to the group of interacted virtual computing nodes. After every virtual computing node receives broadcast messages sent by t-1 virtual computing nodes:

$$r = \prod_{i=1}^{t} W_{si} \bmod p_s$$

is calculated where h(x) as hash function, $S_i = Z_{si}(d_{si}\, b_{si}\, h(Evaluation(x)) - r(y_{si}+Z_{si}^{-2}) \bmod qs$ and:

$$b_{si} = \prod_{j=1, j\neq i}^{n} (1 - ID_{sj}/ID_{si})$$

Moreover, every participant of the group signature submits $\{r, S_i\}$ to responder

- After the generator of group signature receives $\{r, S_i\}$ sent by the participants, it will verify whether $g^{si}(W^{si}) = (PK_i^{Z_{si}}$ holds. If the equation holds, it means the signature is correct, otherwise incorrect. After collecting t effective signature $\{r, S_i\}$ where (i = 1, 2..t), responder will compute:

$$Z_{sa} = \prod_{i=1}^{t} (PK_i^{\pi_a b_{vi}})$$

$$s = \sum_{i=1}^{t} s_i \bmod q_s$$

$$b_{si} = \prod_{j=1, j\neq i}^{n} (1 - ID_{sj}/ID_{si})$$

Select a random secret integer d to compute $B = g_v^d$. Then, according to the public information of the participants, $W_{vi} = g^{dvi}$ and $R_{vi} = W_{vi}^d \bmod P_v$ (i = 1, 2,… m) are calculated. Finally, $\{r, s, Z_{sa}, R_{vi}$, evaluation (x)$\}$ as signature result is published

- Without loss of generality, suppose k initiators (in real case, usually one initiator). Further, anyone of k initiators can verify the correctness of signature. When $\{r, s, Z_{sa}, R_{vi}$, evaluation(x))$\}$ is received by the initiators, they select $v_1, v_2, \ldots v_k$ as verifier, respectively. ID_{vi} is public information of verifier v_i. Firstly, every v_i computes $R_{vi} = B^{dvi} \bmod p_v$. If it does not hold, the signature is ineffective, otherwise go to next step. Secondly, $g^s r^r = (Z_{sa})h(m) \bmod ps$ will be calculated. If the equation of $g_s^s r^r = (E_{si} Z_{sa} Z_{sr}) \bmod p_s$ holds, the signature is effective otherwise ineffective. Thirdly, the initiator decides whether the evaluation result Evaluation(x) in line with trusted evaluation expectation of the initiator so as to finish the process of remote attestation

SECURITY ANALYSIS

Key generation: During key generation, S_i in the set may send false information of $f_{si}(ID_{sj}\text{-}ID_{si})$ to S_j in order to cheat S_j. However, any participant S_j that receives $f_{si}(ID_{sj}\text{-}ID_s)$

will verify whether $g^{fsi(ID_{sj}-ID_{si})} = (g^{si,0})^{(ID_{sj}-ID_{si})}(g^{si,1})^{Pk^{(ID_{sj}-ID_{si})}}$ holds. Consequently, any forged $f_{si}(ID_{sj}-ID_{si})$ cannot pass the verification, that is, it is impossible for malicious member to succeed in forging $f_{si}(ID_{sj}-ID_{si})$.

Identity verification phase: Construct an attacker A to interact with honest virtual computing node with TPM. The attacker impersonates honest virtual computing node to prove the algorithm secure. Construct a function written as F which is an attacker against K-CCA hard problem. In other words, this function knows the data $\{h_1, h_2 \ldots\ldots h_k \in Z_p, S_1, S_2 \ldots\ldots S_K \in Z_p, g, g^x\}$:

$$g^{\frac{1}{h_1 dsi+s_1}}, g^{\frac{1}{h_2 dsi+s_2}} \ldots\ldots g^{\frac{1}{h_n dsi+s_n}}\}$$

Function F takes attacker A as a sub-program owned by it and impersonates the public key generated by TPM of honest virtual computing node in order to response A.

Function F impersonates TPM of an honest virtual computing node. For any participant(s) $\in \{S_1, S_2 \ldots\ldots S_n\}$, attacker A can interact with S_i concurrently; the participant(s) have the respective public key $Pk_i = g^{dsi}$. Attacker A impersonates verifier V and then sends $r_i \in_R Z_p^*$ where $I \in 1, 2 \ldots n$. S_i returns R_i or $\perp$ as response to $r_i \in_R Z_p^*$. If $r_i \in \{h_1, h_2 \ldots\ldots h_k\}$, C sends:

$$g^{\frac{1}{r_i dsi+s_i}}$$

and g^{si} to attacker A, otherwise sends $\perp$. Because attacker A knows the public key of TPM, $e(T, Pk^r r) = I$, that is, attacker A cannot differentiate between the ideal environment and real environment. As a result, interaction process is safe.

Attacker A impersonates honest TPM. During this phase, function F impersonates honest verifier. It sends $r \notin \{r_1, r_2 \ldots r_n\}$ to A with the reference of (PK, r_i, T_i, R_i). If attacker A outputs the result successfully which makes both $s/r \notin \{S_i/r_i\} | i = 1 \ldots, n$ and $e(T, Pk^r R) = I$ hold. In other words, attacker A finds a solution to K-CCA hard problem. Obviously, it is impossible within computational ability of polynomial.

The attacker eavesdrops session script:

$$(PK r_i, T_i = g^{\frac{1}{x dsi+s_i}}, R_i = g^{s_i})$$

during the identity verification. However, the eavesdropper is unable to obtain dS_i related to the information of private key from:

$$(PK, r_i, T_i = g^{\frac{1}{x dsi+s_i}}, R_i = g^{s_i})$$

Further, the attacker cannot impersonate honest virtual computing node. The only way the attacker can do is to send the verifiers:

$$(PK, r_i, T_i = g^{\frac{1}{x dsi+s_i}}, R_i = g^{s_i})$$

continuously. The attacker expects the challenger values:

$$T = g^{\frac{1}{xdsi+s}}$$

and $R = g_s$ to be the same with counterparts used in session script. This kind of attack is similar to birthday attack, that is, the success probability is about $p \le 1/2^{\sqrt{n}}$ that can be neglected.

To wrap up, it is secure during identity verification phase.

State verification phase: During this process, the generator of every g^{si} $(W_{si})^r = (PK_i^{f_1(zsi)})^{h(M)bsi}$ mod ps part of signature can verify whether individual signature is true by verifying whether g^{si} $(W_{si})^r = (e_{sii}^{zsi})^{h(se)(x)bsi}$ mod ps holds, due to the fact that g_i^{si} $(W_{si}^r) = g^{f_1(zsi)(dsi\ bsi\ h(se(x))-rf_2(ysi,\ zsi))\ mod\ qs}$ $g^{rf_1(zsi)\ f2(ysi,\ zsi)} = g^{f_1(zsi\ dsi\ bsi\ h\ (se\ (x)} = (Pk_i^{f_1(Zsi)h(se(x))bsi}$. Because the information except S_i is either public or can be inferred by public information, it is impossible to find a forged y_{si} that makes $gsi(w_{si})^r = (Pk_i^{f_1(zsi))h(M)bsi}$ mod ps hold. As a result, any signer in the set $S = \{S_1, S_2, \ldots\ldots S_n\}$ fails to provide false individual signature $\{r, S_i\}$.

If an attacker infers y_{si} of S_i from public key of S_i, it means it have to solve the discrete logarithm problem. Even if during the key generation phase, messages are transferred in plaintext between users, an attacker definitely fails. Because the attacker is unable to know the secret integer $\gamma_i \in_R Z_p^*$ of participant S_i. As a result, the attacker fail to get private key y_{si} of S_i and $\gamma_i \in_R Z_p^*$.

In order to impersonate the signature of S_i, an attacker has to choose an random number $y_d \in [0, 1]$ and calculate $W_{Si} = g^{f_1(zsi)f2(yd,\ zsi)}$ to be broadcast later. Because the attacker does not know the private key of S_i, it needs to calculate S_i' that satisfies $g^{si}(W_{si})^r = (Pk_i^{f_1(zsi)})^{h(M)bsi}$ mod ps. However, the attacker computes S_i' successfully which means the attacker is able to solve the discrete logarithm problem. Obviously, this cannot happen. According to signature equation:

$$s = \sum_{j=1, i \ne j}^{n} S_j + S_i'$$

the attacker need to compute S_i' to satisfy:

$$g^{s_i} r^r \prod_{j=1, i\neq j}^{n} g^{s_j} = (Z_{s_i})^{h(s\|e(x))}$$

Similarly, this is a discrete logarithm problem. Because the attacker fails to obtain private key y_{si} of S_i and $\gamma_i \in_R Z_p^*$, the false individual signature cannot pass the corresponding verification. As a result, in the case that private key of S_i is unknown, it is impossible to counterfeit the effective signature of S_i.

To wrap up, less than t participants fail to generate signature which means our scheme satisfies the requirements of threshold signature. Consequently, state verification phase is secure.

SIMULATION AND ANALYSIS

We use NetLogo to simulate the remote attestation for the scenario in cloud computing under study. In the simulation scenario, we adopt conditions as follows: 2.36G CPU of intel double cores, 4G memory and Win7 operating system. The parameters in simulation are shown in the following Table 1.

Suppose the group of virtual computing nodes in cloud computing platform conduct X times of remote attestation during the evaluation period of system running. Out of X times, malicious virtual computing nodes conduct Y times. So the effective rate of remote attestation under study is defined as:

$$\mu = \frac{X - Y}{X}$$

Suppose malicious virtual computing nodes account for β% in the entire group and let the threshold T be β+1 and β+10.

We simulate and study the effective rates when β% is 15, 35 and 50%, respectively. The experimental data are shown in the following Fig. 2-4, respectively.

For each of Fig. 2-4, we have that as threshold value increases, the effective rate becomes bigger. Naturally, consider an extreme example like t is equal to the number of all virtual computing nodes in the cloud computing platform, no malicious virtual computing node can conduct remote attestation to outside if there is only one honest one. The more the threshold is, the more the effective rate is.

Because the malicious virtual computing node can be limited dynamically, the effectiveness of our scheme becomes more and more effective as the time passes.

Overall, our scheme is not only secure but also can prevent group cheat effectively.

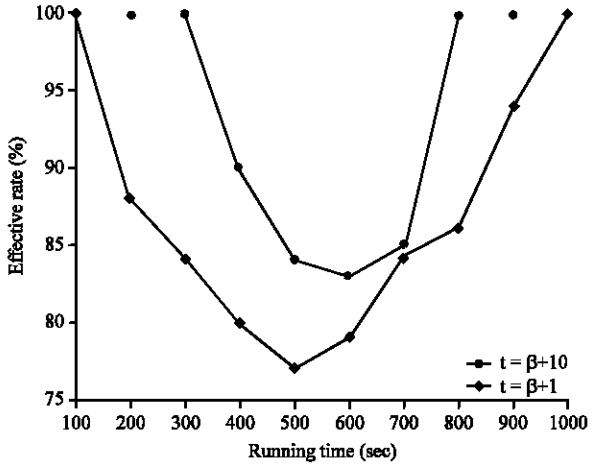

Fig. 3: Effective rate when β% = 35%

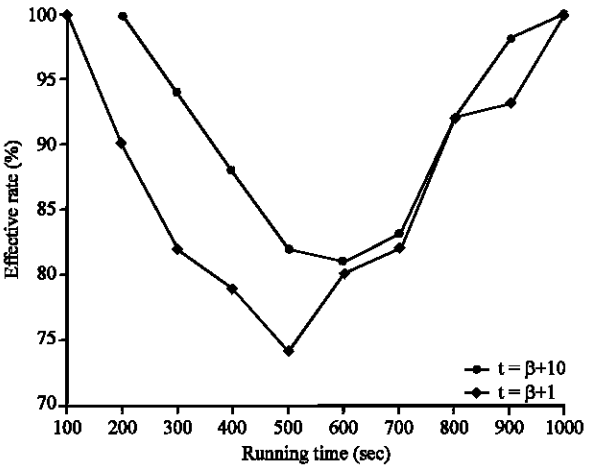

Fig. 4: Effective rate when β% = 50%

Table 1: Threshold values and initial values for simulation

Variables	Value	Sensitivity (mv/T)
N	100	The No. of virtual computing nodes in cloud computing
M	300	The times of snapshots for the virtual computing nodes
E_0	0.3	Threshold for untrusted
E_1	0.5	Threshold for critical trusted
E_2	0.7	Threshold for general trusted
E_3	0.9	Threshold for very trusted
Time	1000 (s)	Running time of simulation system
β%	Seen below	The percentage of malicious virtual computing nodes versus the whole ones.
t	Seen below	The least No. of virtual computing nodes to cooperate to achieve the remote attestation

CONCLUSION

The remote attestation put forward in this paper can prevent virtual computing nodes from conducting collusion attack. Moreover, there is no need for trusted centre to manage all keys of signers. Group cheat in cloud computing can be eliminated greatly.

ACKNOWLEDGMENTS

Work partially supported by the program "Core Electronic Devices, High-end General Purpose Chips and Basic Software Products" (No. 2010ZX01037-001-001), "Study on Policy-driven Data Privacy Protection under Cloud Computing" (No. 4122012). Thanks for the valuable contribution of both Bei GONG and Zhenhu NING to the experiment.

REFERENCES

Brassil, J., 2010. Physical layer network isolation in multi-tenant clouds. Proceedings of the 30th International Conference on Distributed Computing Systems Workshops, June 21-25, 2010, Genoa, Italy, pp: 77-81.

Chang, J., H. Wang and G. Yin, 2006. A time frame based dynamic trust model for P2P systems. Chin. J. Comput., 29: 1301-1307.

Dahbur, K., B. Mohammad and A.B. Tarakji, 2011. A survey of risks, threats and vulnerabilities in cloud computing. Proceedings of the 2nd International Conference on Intelligent Semantic Web-Services and Applications, April 18-20, 2011, Amman, Jordan.

Dawoud, W., I. Takouna and C. Meinel, 2010. Infrastructure as a service security: Challenges and solutions. Proceedings of the 7th International Conference on Informatics and Systems, March 28-30, 2010, Cairo, Egypt, pp: 1-8.

Goldman, K., R. Sailer, D. Pendarakis and D. Srinivasan, 2010. Scalable integrity monitoring in virtualized environments. Proceedings of the 5th ACM Workshop on Scalable Trusted Computing, October 4-8, 2010, Chicago, IL., USA., pp: 73-78.

International Business Machine, 2008. IBM blue cloud solution. http://www-31.ibm.com/ibm/cn/cloud/

Santos, N., K.P. Gummadi and R. Rodrigues, 2009. Towards trusted cloud computing. Proceedings of the Conference on Hot Topics in Cloud Computing, June 14-19, 2009, Berkeley, CA, USA.

Schiffman, J., T. Moyer, H. Vijayakumar, T. Jaeger and P. McDaniel, 2010. Seeding clouds with trust anchors. Proceedings of the ACM Workshop on Cloud Computing Security Workshop, October 4-8, 2010, Chicago, IL., USA., pp: 43-46.

Shen, C., 2004. Build active and comprehensive protection system. China Inform. Secur., 41: 18-20.

Sun Microsystems, 2009. Introduction to cloud computing architecture. http://www.techrepublic.com /resource-library/whitepapers/introduction-to-cloud-computing-architecture/