International Journal of Security and Its Applications
Vol.9, No.10 (2015), pp.395-404
http://dx.doi.org/10.14257/ijsia.2015.9.10.36

Client Oriented Remote Attestation Model in Cloud Environment

Liu Zhenpeng[1,a],Wang Xu[1,b], LiuYifan[2],Guo Ding[1],Zhu Xianchao[1]

[1]*College of Information and Electrical Engineering, Hebei University, Baoding Hebei 071000, China*
[2]*School of Computer Science and Technology, China University of Mining and Technology, Xuzhou Jiangsu 221008, China*
[a]*lzp@hbu.edu.cn*
[b]*freeman200608@126.com*

Abstract

In the field of cloud security, the cloud provider don't disclose any internal configuration information to protect itself, so the client know nothing about their data stored in the cloud and security status of the node providing services for them, thereby it causes the client's worry whether to adopt cloud computing services. So that the trust between client and cloud computing provider become one of the biggest obstacles hindering the development of the cloud computing. Based on Direct Anonymous Attestation (DAA) and Dynamic Property Trusted Attestation (DPTA), we propose a client oriented remote attestation (CORA) model in cloud environment, client can select a node in the cloud at corresponding security level according to their own needs and can dynamically verify the node's security status. At the same time, because the use of anonymous method it will not expose classified information of the node, such as configuration and location information etc. Furthermore we add service life of certificates to update certificates regularly, which enhanced the security of the attestation.

Keywords: *cloud security , remote attestation, client oriented*

1. Introduction

With the advantages of on-demand self-service, ubiquitous network access, physical location transparency, rapid scalability and pay-for-use, Cloud computing sets off a new wave of information tide and is widely used in many fields of society. For the enterprise users, it can significantly reduce the storage and maintenance cost of information. For the personal users, it will relieve the restraint of personal users' limited storage and computing resources, if the storage and computing of information is put in the cloud [1]. Although cloud computing has great features, but trust and security problem has become the bottleneck for its development.

Cloud service provider rarely disclose its internal configuration as to ensure its own safety, so client do not know the security situation of its data and the node providing services, which consequently causes the client worry about adopting cloud computing services. Thus trust problem has become one of the biggest obstacles in cloud computing' development. To solve this problem, trusted computing technology proposed by TCG (Trusted Computing Group) has integrated into cloud computing environment reliably and been the focus in cloud security field [2]. Trusted computing is based on Trusted Platform Module, through the trusted measurement of software and hardware to complete the extension of trust chain and to ensure the system boot credibly. The verifier can judge the trusted state of platform by remote attestation. Based on Direct Anonymous Attestation [3] (DAA) and Dynamic Property Trusted Attestation [4] (DPTA), we propose a client oriented remote attestation (hereafter referred to as CORA) model in cloud environment, client can select a node in the cloud at corresponding security grade

ISSN: 1738-9976 IJSIA

according to their own needs and can dynamically verify the node's security status. At the same time, because the use of anonymous method it will not expose classified information of the node, such as configuration and location information etc. Furthermore we add service life of certificates to update certificates regularly, which enhanced the security of the attestation.

2. Related Works

By far, there are two main schemes proposed by TCG, one is Privacy CA (Privacy Certification Authority) scheme in TPM1.1 standard, another is DAA (Direct Anonymous Attestation) scheme in TPM1.2 standard. Although DAA has solved the CA bottleneck problem in Privacy CA, but they are all binary attestation scheme, which will easily disclose the system configuration of platform leading to threat to platform security.

To prevent the exposure of platform's privacy, Poritz *et al.* [5] propose PBA (Property-Based Attestation) from system architecture level. Sadeghi *et al.* [6] propose the concrete realization progress of PBA. Chen *et al.* [7] propose the protocol of PBA. Qin Yu et al. [8] propose a scheme called Component property base remote attestation, component as the granularity, every component's security attribute can be attested, which has better flexibility and expandability. Yan Jianhong [4] proposes DPTA(Dynamic Proper ty Trusted Attestation) method based on PBA. The verifier analogously compute PCR value and compare with the PCR value in the certificate, so attestor can prove that the attested platform satisfies a certain security attribute. The experiment shows that this method can protect the platform's privacy and overcome the PBA's static feature, so it has both real-time and security feature.

Under the cloud environment, Xin *et al.* [9] put the property-based attestation mechanism into cloud environment, based on the character in the cloud, proposed property-based remote attestation oriented to cloud computing. Client can verify node attributes effectively without disclosing the node's information. Xu [10] *et al.* propose real-time remote attestation based on IaaS cloud, which uses virtualization and trusted computing to verify whether the behavior of software is trusted in real time. Zhang Yan *et al.* [11] propose a scheme based on virtual machine identity attestation, under the original authentication and trust relationship, concealing the authoritative identity information in the process of identity attestation, avoided disclosing the organization structure and location at el. information and supporting cloud environment's transparent structure and location-independent feature.

3. The Overall Structure of CORA Model

The overall structure is as shown in Figure 1. Our model is based on Eucalyptus architecture. The Eucalyptus architecture mainly include 3 roles: common cloud node, called Node Controller (NC), accept responsibility for booting, sleeping, closing and migrating VM. Cluster control node, called Cluster Controller (CC), simultaneously connect server between intranet and extranet and manage several NCs. Cloud control node, called Cloud Controller (CLC), which is the entrance for client and the global decision analysis component, accept responsibility for managing the request of client and administrator, making high level VM dispatch decision, disposing service level agreement and maintaining enduring metadata of system and client.

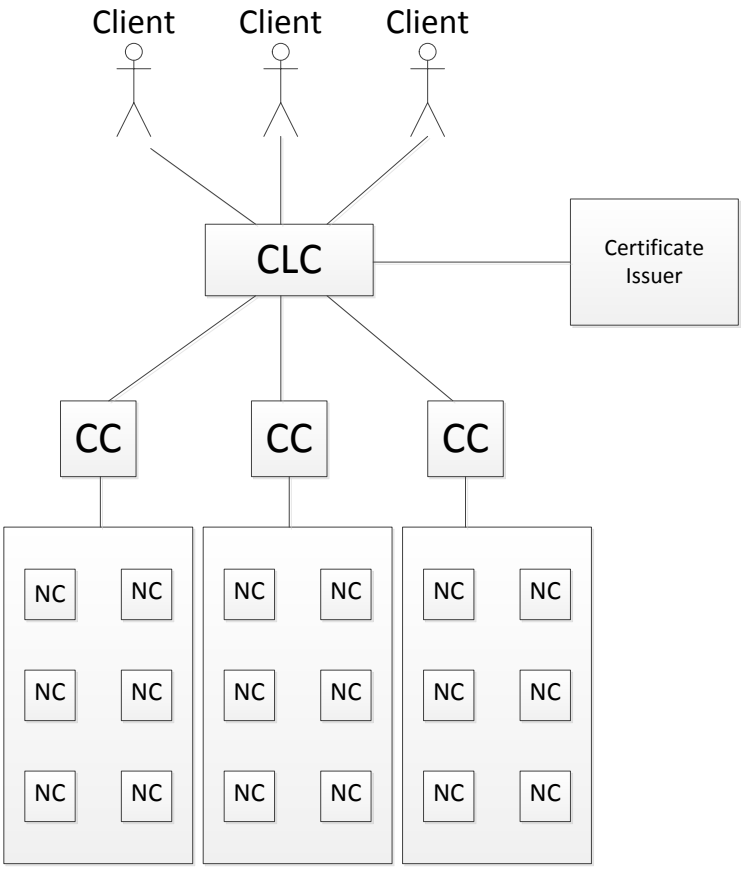

Figure 1. The Overall Structure of CORA Model

In this model we add a trusted third party called Certificate Issuer (CI). CI is responsible for issuing DAA certificate and security level attribute (hereafter referred to as SLA) certificate to NC and communicating with CLC in security. In addition CI maintain a security level attribute configuration database, NC use its trusted platform module to measure its configuration, then CI map the measurement data sent by NC into corresponding security level and issue SLA certificate to NC. In CLC, after NC get the security level attribute certificate, CLC verifies the certificate and whether the node is in trusted state, then create a security level list of all NC. The list will be updated at fixed period. When client submit applications, CLC select a NC in the list according to the security level requested by client, then client verifies NC's SLA certificate to decide whether to adopt the service provided by the NC.

4. The Protocol of CORA Model

This model is based on DAA and DPTA, we add service life of certificates on the basis of DPTA, the setting of service life of certificates is according to the different cloud platform's character, so that we can update security level list in the service life of certificates.

Security parameters used in the protocol is shown in the TABLE I.

Table 1. Security Parameters

Parameter	Length	Definition
l_n	2048	The length of RSA key
l_f	104	The length of secret message f_0 、 f_1
$l_\varnothing$	80	The length of security parameter in Zero-knowledge proof
l_{sv}	160	The length of security level attribute configuration *sv*
l_{cv}	160	The length of security configuration *cv*
l_H	160	The length of hash function's output in Fiat-Shamir heuristic method
l_e	368	The length of index *e* in the certificate
$l_e^{'}$	120	The length of interval for selecting *e*
l_v	2536	The length of random number *v* in the certificate
l_r	80	Safety parameter needed in reducing trusted attestation
l_p	1632	The length of coefficient *P* in Pedersen Commitment
l_{exdate}	160	The length of service life of certificate *exdate*

l_{exdate} is the length of the added parameter service life of certificates (represented by *exdate*).

Moreover: l_e, l_v in DAA satisfy:

$l_e > l_\varnothing + l_H + \max(l_f + 4, l_e^{'} + 2)$, $l_v > l_n + l_H + \max(l_f + l_r + 3, l_\varnothing + 2)$

l_e, l_v in security level attribute certificate satisfy:

$l_e \geq \max(l_{cv}, l_{sv}) + 2$, $l_v \geq l_n + \max(l_{cv}, l_{sv}) + l_\varnothing$, and $l_Q > l_{cs} + l_H + l_\varnothing + 2$. l_Q represents the length of subgroup order *Q* of Z_p^* and *P*, *Q* satisfy: $2^{l_Q} > Q > 2^{l_{Q-1}}$, $2^{l_P} > Q > 2^{l_{P-1}}$.

Our protocol adopts Pedersen Commitment[12], CL signature scheme[13], Zero-knowledge proof and DAA scheme [3]. Pedersen Commitment is used to conceal and binding NC's configuration. CL signature scheme group signature policy, it permit efficient protocol to sign knowledge proof and restore secret information of the signature effectively according to discrete logarithm based on knowledge proof. Zero-knowledge proof is a method by which attestor can prove to verifier that a given statement is true, without conveying any information apart from the fact that the statement is indeed true. DAA scheme is based on CL signature scheme and Zero-knowledge proof, furthermore it will use Fiat-Shamir heuristic method to transform knowledge proof to non-interactive signature of Knowledge.

4.1 Issuing DAA Certificate and SLA Certificate Protocol

All NCs in the cloud platform already acquire the public key published by CI $PK_I = (n, g^{'}, g, h, S, Z, R_0, R_1, R_2, \mathrm{R}_3, \gamma, \Gamma, \rho)$. We add two parameters R_2 , R_3 different with DAA's PK_I which are used in SLA certificate. Generation method is same with R_0 , R_1 .

i) NC verifies PK_I in the same way with DAA.

ii) NC makes a request to CI.

iii) CI generates a random number n and send to NC to prevent replay attack.

iv) NC sends $\{U, N_I, n, \{\{cv_i, sml, ML_{NC}\}_{SK_{EK}}, H(PK_{EK})\}_{PK_I}\}$ to CI. cv_i is the value of platform security configuration, *sml* is platform storage measurement log, ML_{NC} is platform integrity measurement list, SK_{EK} is private key of endorsement key, PK_{EK} is public key of endorsement key, $H(PK_{EK})$ is hash value of PK_{EK} . Specific process is as follows:

a) NC's TPM generates a secret message f and a random integer v' ($v' \in_R \{0,1\}^{l_n+l_\varnothing}$) and splits f into two messages $f_0 = LSB_{l_f}(f)$, $f_1 = CAR_{l_f}(f)$. Compute:

$\zeta_I = (H_\Gamma(1 \| bsn_I))^{(\Gamma-1)/\rho} \bmod \Gamma$, $U = R_0^{f_0} R_1^{f_1} S^{v'} \bmod n$, $N_I = \zeta_I^{f_0 + f_1 2^{l_f}} \bmod \Gamma$, bsn_I is the basename of CI.

b) TPM executes Quote command and then gets ML_{NC}, *sml*. After that, encrypt ML_{NC}, cv_i and *sml* with SK_{EK} ,encrypt $H(PK_{EK})$, $\{cv_i, sml, ML_{NC}\}_{SK_{EK}}$ with PK_I.

c) At last, NC sends $\{U, N_I, n, \{\{cv_i, sml, ML_{NC}\}_{SK_{EK}}, H(PK_{EK})\}_{PK_I}\}$ to CI.

v) CI judges whether NC is the black list by calculating $N_I \stackrel{?}{=} \zeta_I^{f_0' + f_1' 2^{l_f}} \bmod \Gamma$. f_0', f_1' are leaked secret message.

vi) CI verifies random number n. Then CI decrypt $\{\{cv_i, sml, ML_{NC}\}_{SK_{EK}}, H(PK_{EK})\}_{PK_I}$ with SK_I. Verify whether NC's TPM is the valid TPM by compare $H(PK_{EK})$ to the hash value of EK public key list. Then get cv_i, *sml*, ML_{NC} with valid PK_{EK}. Use *sml* recalculate platform integrity measurement ML_{NC}', compare ML_{NC}' with ML_{NC}. At last verify whether NC is at trusted state with ML_{NC}.

vii) CI verifies U and N_I through Zero-knowledge proof.

viii) CI generates DAA certificate. Generate a random integer $\hat{v}$ and a random prime e and compute:

$$v'' = \hat{v} + 2^{l_v - 1}$$

$$A = \left(\frac{Z}{US^{v''}}\right)^{1/e} \bmod n$$

ix) CI generates SLA certificate. Compare cv_i with security level attribute configuration database, get the corresponding value of security level attribute configuration sv_i. Generate random primes e_i, v_i , their length require that:

$l_e \geq \max\{l_{sv}, l_{cv}, ML_{NC}\}$ $l_v \geq l_n + \max\{l_{sv}, l_{cv}, ML_{NC}\} + l_\varnothing$

Compute:

$$A_i = \left(\frac{Z}{UR_0^{cv_i} R_2^{ML_{NC}} R_3^{exdate} R_1^{sv_i} S^{v_i}}\right)^{1/e_i} \bmod n$$

exdate is the service life of certificate, $exdate \in_R \{0,1\}^{l_f}$.

x) CI encrypts DAA certificate, SLA certificate, sv_i and *exdate* with PK_{EK} then send $\{\{A, e, v''\}, \{A_i, e_i, v_i\}, sv_i, exdate\}_{PK_{EK}}$ to NC.

4.2 SLA Signing Protocol

The signature procedure of SLA certificate is between TPM and host. TPM input SK_{EK} .host input $\{A_i, e_i, v_i\}$. Common input: $PK_I = (n, g', g, h, S, Z, R_0, R_1, R_2, R_3, \gamma, \Gamma, \rho)$, commitment parameter: $par_{com} = (g, h, P, Q)$, cv_i, ML_{NC},

sv_i, *exdate*, n_v (n_v is provided by the verifier).

i) The TPM performs as follows:

a) Generate a random $N_t \in_R \{0,1\}^{l_\varnothing}$.

b) Generate a random $r \in_R \{0,1\}^{l_Q}$ and compute the commitment $C_{cv_i} = g^{cv_i} h^r \bmod P$.

c) Generate a TPM signature

$\sigma_M => Sign_M(SK_{AIK}, par_{com} \| C_{cv_i} \| ML_{NC} \| exdate \| n_v \| N_t)$

Send to host σ_M with the values C_{cv_i}, ML_{NC}, N_t and exdate.

ii) The host performs as follows:

a) Generate a random $\omega \in \{0,1\}^{l_n}$.

b) Compute: $\hat{A} = A_i S^{\omega} \bmod n$, $\hat{v} = v_i - \omega e_i$.

c) Create a masked signature: $\hat{\sigma}_{CI} => (\hat{A}, e_i, \hat{v})$.

iii) The host computes the signature of knowledge protocol:

$$SPK\{(cv_i, ML_{NC}, \mathrm{e}_i, \hat{v}, r) : \frac{Z}{R_1^{sv_i}} \equiv \pm \hat{A}^{e_i} R_0^{cv_i} R_2^{ML_{NC}} R_3^{exdate} S^{\hat{v}} (\bmod n) \wedge C_{cv_i} = g^{cv_i} h^r$$

$$(\bmod P) cv_i \in \{0,1\}^{l_{cv}+l_{\varnothing}+l_H+2} \wedge (e_i - 2^{l_e}) \in \{0,1\}^{l_e^{'}+l_{\varnothing}+l_H+1}\}(n_v, N_t)$$

steps as follow:

a) Compute $Z^{'} = \frac{Z}{R_1^{sv_i}} \bmod n$.

b) Generate random integers:

$r_v \in_R \{0,1\}^{l_e+l_n+2l_{\varnothing}+l_H+1}$, $r_e \in_R \{0,1\}^{l_e^{'}+l_{\varnothing}+l_H}$,

$r_{cv} \in_R \{0,1\}^{l_{cv}+l_{\varnothing}+l_H}$, $r_r \in_R \{0,1\}^{l_Q+l_{\varnothing}+l_H}$,

$r_{ML_{NC}} \in_R \{0,1\}^{l_{ML_{NC}}+l_{\varnothing}+l_H}$, $r_{exdate} \in_R \{0,1\}^{l_{exdate}+l_{\varnothing}+l_H}$.

c) Compute: $\tilde{Z}^{'} = \hat{A}^{r_e} R_0^{r_{cv}} R_2^{r_{ML_{NC}}} R_3^{r_{exdate}} S^{r_v} \bmod n$

$\tilde{C}_i = g^{r_{cv}} h^{r_r} \bmod P$.

$c = Hash(PK_I \| par \| sv \| C_{cv_i} \| ML_{NC} \|$

$exdate \| \tilde{Z}^{'} \| \tilde{C}_i \| n_v \| N_t)$

$s_v = r_v + c \cdot \hat{v}$, $s_{cv} = r_{cv} + c \cdot cv_i$

$s_e = r_e + c \cdot (e_i - 2^{l_e - 1})$, $s_r = r_r + c \cdot r$

$s_{ML_{NC}} = r_{ML_{NC}} + c \cdot ML_{NC}$ $s_{exdate} = r_{exdate} + c \cdot exdate$

Generate the signature of SLA certificate:

$$\sigma_{MPBA} => (\hat{A}, \sigma_M, N_t, C_{cv_i}, ML_{NC}, exdate, c, s_v, s_{cv}, s_e, s_r, s_{ML_{NC}}, \mathrm{s}_{exdate})$$

4.3 The Verification Protocol

Verifier acquires PK_I published by CI and PK_{AIK}, par_{com}, σ_{MPBA}, DAA certificate, $\{PCRs\}_{PK_{AIK}}$, $sml^{'}$, n_v sent by NC. The verifier performing as follows:

i) Verify σ_M with PK_{AIK}:

$(par_{com} \| C_{cv_i} \| ML_{NC} \| exdate \| n_v \| N_t)$ and verify the whether the exdate is exceed the time limit. If positive go to the next step.

ii) Compute:

$\hat{Z}^{'} = Z^{'-c} \hat{A}^{s_e + c2^{l_e - 1}} R_0^{s_{cv}} R_2^{s_{ML_{NC}}} R_3^{s_{exdate}} S^{s_v} \bmod n$

$\hat{C}_i = C_{cv_i}^{-c} g^{r_{sv}} h^{s_r} \bmod P$

iii)Verify:

$c \overset{?}{=} hash(PK_I \| par \| sv_i \| C_{cv_i} \| ML_{NC} \| exdate \| \hat{Z}^{'} \| \hat{C}_i \| n_v \| N_t)$

$s_{ML_{NC}} \overset{?}{\in} \{0,1\}^{l_{ML_{NC}}^{'}+l_{\varnothing}+l_H+1}$, $s_{cv} \overset{?}{\in} \{0,1\}^{l_{cv}^{'}+l_{\varnothing}+l_H+1}$

$s_e \stackrel{?}{\in} \{0,1\}^{l'_e + l_\varnothing + l_H + 1}$, $s_{exdate} \stackrel{?}{\in} \{0,1\}^{l'_{exdate} + l_\varnothing + l_H + 1}$

$PCRs \stackrel{?}{=} ML_{NC}$

4.4 Registration Protocol

NC registers to CLC. NC prove its security attribute level to CLC, CLC add NC into its security attribute level list. As follows:

i) NC sends request to CLC. Encrypt cv_i, NC's *id* with PK_{CLC}, then send them to CLC.

ii) After CLC receives request,generates a random n_v, encrypts n_v with SK_{CLC}, then sends to NC.

iii) After the decryption,NC gets and generates a attestation identity key (*AIK*),signs (public key of *AIK*) with DAA certificate. Execute Quote command to get *PCRs* (the value of real-time ML_{NC}). Encrypt *PCRs*, SLA certificate with SK_{AIK} (private key of *AIK*), and encrypt DAA certificate with PK_{CLC}, send them to CLC.

iv) CLC verifies DAA certificate , acquire PK_{AIK}. Then get *PCRs* and SLA certificate using PK_{AIK}. Execute verification protocol in *C* and compare ML_{NC} with *PCRs*, if they have equal value, then the NC is at trusted state now. At last CLC add NC into security attribute level list.

4.5 Request and Verification Protocol for Client

When client requires cloud service, client submits a service request to CLC and verifies NC's security attribute level and trusted state. Specific process is as follows:

i) Client requests a NC to CLC, which security attribute level is sv.

ii) CLC chooses a NC in the security attribute level list, which security attribute level is sv.

iii) NC generates an AIK, sign PK_{AIK} with DAA certificate, then sends it to the client.

iv) Client verifies DAA certificate and gets PK_{AIK}. Generate a random number, encrypt it with PK_{AIK} and send to NC.

v) After NC gets $\{n_v\}_{PK_{AIK}}$, decrypt it with SK_{AIK} Execute Quote command to get the real-time PCRs. Encrypt SLA certificate and PCRs with SK_{AIK} and send to client.

vi) Client executes verification protocol. Verify whether NC's security attribute level is sv, whether ML_{NC} is equal to PCRs and the service life of certificate. If pass the verification, NC sends the order to CLC, that run the service on the NC.

5. The Whole Process of CORA Model

From NC request certificates to client submit service request and running service is shown in Figure 2.

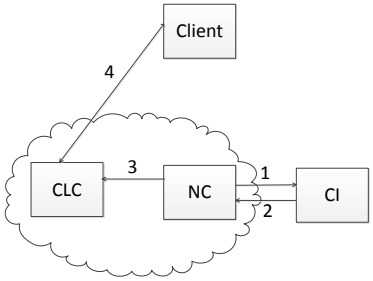

Figure 2. The Whole Process of CORA Model

The specific process is as follows:

i) NC sends $\{U, N_I, n, \{\{cv_i, sml, ML_{NC}\}_{SK_{EK}}, H(PK_{EK})\}_{PK_I}\}$ to CI and requests SLA certificate and DAA certificate.

ii) CI executes Issuing DAA Certificate and SLA certificate protocol in A, NC executes SLA Signing protocol in 4.2.

iii) Execute registration protocol in 4.4.

iv) Execute request and verification protocol for Client in 4.5.

6. Performance and Security Analysis

6.1 Performance Analysis

In this paper, as our model is based on DAA and DPTA, so there is no bottleneck problem of third party. However, this will increase the complexity of the protocol, and therefore the request and issuing process of the certificate can be selected at a lower user visit (e.g. morning) time to reduce the impact on the user experience.

During the verification process, the client can dynamically verifies the node's trusted status based on needs, this may affect the operation of the node. But nodes only need to execute extend and load PCR command and respond in millisecond level, client will not notice. The clients need to compare the real-time PCRs and ML_{NC} in the SLA certificate and verify the service life of certificate. Since only execute binary compare operating, so the execution time is negligible.

6.2 Security Analysis

When issuing certificates, CI verifies ML_{NC} and the formula $N_I \stackrel{?}{=} \zeta_I^{f_0' + f_1' 2^{l_f}} \bmod \Gamma$, to ensure that the node is in trusted state. Because of the certificate is no longer a one-time certificate, by adding the service life of certificate to specify the period of the certificate. So the node update certificate regularly to further ensure its safety. The correctness and unforgeability of the protocol have been proved in the literature [3] [4].

In the signing process, after the node receives the SLA certificates, node configuration is hidden by using the Pedersen commitment and zero-knowledge proof. Client can't know the specific configuration of the node, thus the node configuration concealed. Moreover, as this process uses attestation identity key AIK, so it meet un- connectivity, users can not connect to unreliable attestation process, to ensure the integrity of the attestation.

In the client and CLC verification process, they can dynamically verify the node trusted state, by compare the real-time PCRs and ML_{NC} in the SLA certificate. If PCRs is equal to ML_{NC}, then the node is trusted. If not, just indicate the node may have been attacked; client can stop using this node to run the service.

7. Conclusion

In cloud environment based on Direct Anonymous Attestation (DAA) and Dynamic Property Trusted Attestation (DPTA), we propose a clients oriented remote attestation (CORA) model, enables clients to select the node in corresponding security level according to their needs and verify the node's trusted state dynamically. And on the basis of DPTA adds the service life of certificate parameters. Attribute service life to SLA certificates. By regularly updating the certificate can enhance security of attestation. Because clients know the security level and trusted state of the node, so it enhances the clients' trust to the cloud service provider trust and confidence to use the cloud services.

Acknowledgements

This work was supported by National Science and Technology Support Program (2013BAK07B04).

References

[1] N. H. Yu, Z. Hao, J. Xu, "Review of Cloud Computing Security," Chinese Journal of Electronics(**2013**), vol. 41, no. 2, pp. 371-381.
[2] D.G. Feng, M. Zhang, Y. Zhang, "Study on Cloud Computing Security," Journal of Software(**2011**), vol.22(1), pp. 71-83.
[3] E. Brickell, J. Camenisch, L. Chen "Direct anonymous attestation," Proceedings of the 11th ACM conference on Computer and communications security. ACM,(**2004**) , pp.132-145.
[4] J.H. Yan, "Trusted Dynamic Attestation Mechanism Based on Property Certificate," Journal of Chinese Computer Systems(**2013**), vol.34(10), pp. 2349-2353.
[5] J. Poritz, M. Schunter, E.V. Herreweghen. Property attestation—Scalable and privacy-friendly security assessment of peer computers. Technical Report RZ 3548, IBM Zurich Research Laboratory(**2004**).
[6] A.R. Sadeghi and C. Stuble, "Property-based attestation for computing platforms: caring about properties, not mechanisms," Proceedings of the **2004** workshop on New security paradigms. ACM, pp. 67-77.
[7] L. Chen,R. Landfermann,H. Löhr, "A protocol for property-based attestation," Proceedings of the first ACM workshop on Scalable trusted computing. ACM(**2006**) pp. 7-16.
[8] Y. Qin and D.G. Feng , "Component property base remote attestation," Journal of Software(**2009**), vol.20(6), pp. 1625-1641.
[9] Y.S. Xin, Y. Zhao,Y. Li, "Property-Based Remote Attestation Oriented to Cloud Computing," Computational Intelligence and Security (CIS), 2011 Seventh International Conference on. IEEE,(**2011**),pp.1028-1032
[10] Z. Xu, A. Yu,W. Yang "Real-time remote attestation of IaaS cloud," Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on. IEEE(**2012**) pp. 1028-1032.
[11] Y. Zhang, D.G. Feng,A.M. Yu, "Virtual Machine Anonymous Attestation in Cloud Computing," Journal of Software (**2013**), vol.24(12), pp. 2897-2908.
[12] T.P Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," Advances in Cryptology—CRYPTO'91. Springer Berlin Heidelberg, **1992**,pp. 129-140.
[13] J. Camenisch, A. Lysyanskaya, "A signature scheme with efficient protocols," Security in communication networks. Springer Berlin Heidelberg(**2003**)pp. 268-289.

Authors

Liu Zhenpeng, he is the director of Hebei University computer network information center. He has research interest in cloud computing and information security.

Wang Xu, he is the postgraduate student of Hebei University. He has research interest in cloud computing and information security.