Identity-Based Authentication for Cloud Computing

Hongwei Li[1], Yuanshun Dai[1,2], Ling Tian[1], and Haomiao Yang[1]

[1] Collaborative Autonomic Computing Lab, School of Computer Science and Engineering, University of Electronic Science and Technology of China
hongwei-li@tom.com, ruan052@126.com, yanghaomiao@sohu.com
[2] Innovative Computing Lab, Department of Electronic Engineering & Computer Science, University of Tennessee, Knoxville, USA
ydai1@eecs.utk.edu

Abstract. Cloud computing is a recently developed new technology for complex systems with massive-scale services sharing among numerous users. Therefore, authentication of both users and services is a significant issue for the trust and security of the cloud computing. SSL Authentication Protocol (SAP), once applied in cloud computing, will become so complicated that users will undergo a heavily loaded point both in computation and communication. This paper, based on the identity-based hierarchical model for cloud computing (IBHMCC) and its corresponding encryption and signature schemes, presented a new identity-based authentication protocol for cloud computing and services. Through simulation testing, it is shown that the authentication protocol is more lightweight and efficient than SAP, specially the more lightweight user side. Such merit of our model with great scalability is very suited to the massive-scale cloud.

Keywords: cloud computing, identity-based cryptography, authentication.

1 Introduction

Cloud computing is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. Users need not have knowledge of, expertise in, or control over the technology infrastructure 'in the cloud' that supports them [1,2]. Authentication, thus, becomes pretty important for cloud security. Applied to cloud computing and based on standard X.509 certificate-based PKI authentication framework, SSL Authentication Protocol (SAP) [3] is low efficient. The authors of Grid Security Infrastructure (GSI) conceded that the current GSI technique has a poor scalability [4]. W.B. Mao analyzed that this scalability problem is an inherent one due to the use of SAP [5].

Grid computing and cloud computing are so similar that grid security technique can be applied to cloud computing. Dai et al. made great contribution to Grid security [6-9]. Recently, identity-based cryptography (IBC) is developing very quickly [10-12]. The idea of applying IBC to grid security was initially explored by Lim (2004) [13]. Mao et al. (2004) proposed an identity-based non-interactive authentication framework for grid [5]. The framework is certificate-free. But the unique Private Key

M.G. Jaatun, G. Zhao, and C. Rong (Eds.): CloudCom 2009, LNCS 5931, pp. 157–166, 2009.

Generator (PKG) becomes the bottleneck of framework. Lim and Robshow (2005) proposed a hybrid approach combining IBC [14]. The approach solves escrow and distribution of private key. However, the non-interactive and certificate-free quality is lost. Chen (2005) revisited the GSI in the GT version2 and improved the GSI architecture and protocols [15]. It is significant to study IBC and cloud computing.

In this paper, based on identity-based hierarchical model for cloud computing (IBHMCC) and corresponding encryption and signature schemes, an identity-based authentication for cloud computing (IBACC) is proposed. IBACC is more efficient and lightweight than SAP, specially the more lightweight user side, which contributes good scalability to the much larger cloud systems.

The remaining of the paper is organized as the following. Section 2 introduces the identity-based hierarchical model for cloud computing (IBHMCC). In section 3, we propose identity-based encryption and signature technology for the IBHMCC. Section 4 proposes identity-based authentication mechanism for cloud computing. Section 5 makes the performance analysis for our new protocols and did simulated experiments to validate the techniques.

2 Identity-Based Hierarchical Model for Cloud Computing

As shown in Fig.1, IBHM for cloud computing (IBHMCC) is composed of three levels. The top level (level-0) is root PKG. The level-1 is sub-PKGs. Each node in level-1 corresponds to a data-center (such as a Cloud Storage Service Provider) in the cloud computing. The bottom level (level-2) are users in the cloud computing. In IBHMCC, each node has a unique name. The name is the node's registered distinguished name (DN) when the node joins the cloud storage service. For example, in the Fig.1, DN of the root node is DN_0, DN of node M is DN_M and DN of node N is DN_N. We define the identity of node is the DN string from the root node to the current node itself. For example, the identity of entity N is $ID_N = DN_0 \| DN_M \| DN_N$. " $\|$ " denotes string concatenation. We further define $ID_N|_0 = DN_0$, $ID_N|_1 = DN_0 \| DN_M$, $ID_N|_2 = DN_0 \| DN_M \| DN_N$ .The rule is applicable to all nodes in the hierarchical model.

The deployment of IBHMCC needs two modules: Root PKG setup and Lower-level setup.

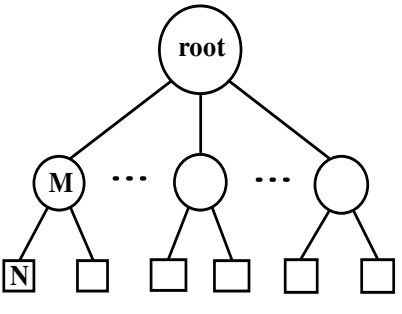

Fig. 1. IBHM for cloud computing

Root PKG setup: Root PKG acts as follows:

1. Generate group G_1, G_2 of some prime order q and an admissible pairing $\hat{e}: G_1 \times G_1 \to G_2$;
2. Choose an arbitrary generator $P \in G_1$;
3. Choose cryptography hash functions $H_1: \{0,1\}^* \to G_1, H_2: G_2 \to \{0,1\}^n$ for some n;
4. Pick a random $\alpha \in \mathbb{Z}_q^*$ and set $Q_0 = \alpha P,\ P_0 = H_1(DN_0), S_0 = \alpha P_0$. The root PKG's master key is S_0 and the system parameters are $< G_1, G_2, \hat{e}, Q_0, P, P_0, H_1, H_2 >$.

Lower-level setup

1. Assume there are m nodes in the level-1. For each node, the root PKG acts as follows (let X be an arbitrary node in the m nodes):
2. Compute the public key of node $X: P_X = H_1(ID_X)$, where $ID_X = DN_0 \| DN_X$;
3. Pick the secret point $\rho_X \in \mathbb{Z}_q^*$ for node X. ρ_X is only known by node X and its parent node;
4. Set the secret key of node $X: S_X = S_0 + \rho_X P_X$;
5. Define the Q-value: $Q_{ID_X \|} = \rho_X P . Q_{ID_X \|}$ is public.

After the above five steps are finished, all nodes in the level-1 get and securely keep their secret keys and the secret points. On the other hand, the public key and the Q-value are publicized.

Then, Each node in the level-1 similarly repeats the above steps (2-5). Similarly, all nodes in level-2 keep the secret keys and the secret point while publicizing the public key and Q-value.

3 Identity-Based Encryption and Signature for IBHMCC

In the cloud computing, it is frequent for the entities to communicate mutually. To achieve the security in the communication, it is important to propose an encryption and signature schemes. Therefore, we propose an identity-based encryption (IBE) and identity-based signature (IBS) schemes for IBHMCC in the following.

3.1 Identity-Based Encryption

IBE is based on the above Root PKG setup and Lower-level setup algorithms. It is composed by two parts: Encryption and Decryption.

Encryption: Assume E_1 and E_2 are two entities in the cloud computing. The identity of entity E_2 is $ID_{E_2} = DN_0 \| DN_1 \| DN_2$. To encrypt message m with ID_{E_2}, E_1 acts as follows:

1. Compute

$$P_1 = H_1(DN_0 \| DN_1) \tag{1}$$

$$P_2 = H_1(DN_0 \| DN_1 \| DN_2) \tag{2}$$

2. Choose a random $r \in \mathbb{Z}_q^*$;

3. Output the ciphertext

$$C = < rP, rP_1, rP_2, H_2(g^r) \oplus m > \tag{3}$$

where $g = \hat{e}(Q_0, P_0)$ which can be pre-computed.

Decryption: After receiving the ciphertext $C = < U_0, U_1, U_2, V >$, entity E_2 can decrypt C using its secret key $S_{E_2} = S_0 + \rho_1 P_1 + \rho_2 P_2$, where ρ_1 is the secret point of node $DN_0 \| DN_1$, ρ_2 is the secret point of node $DN_0 \| DN_1 \| DN_2$:

1. Compute

$$d = \frac{\hat{e}(U_0, S_{E_2})}{\prod_{i=1}^{2} \hat{e}(Q_{ID_{E_2}|i}, U_i)} \tag{4}$$

where $Q_{ID_{E_2}|1} = \rho_1 P, Q_{ID_{E_2}|2} = \rho_2 P$;

2. Output the message $m = H_2(d) \oplus V$.

3.2 Identity-Based Signature

IBS is also based on Root PKG setup and Lower-level setup algorithms. It incorporates two algorithms: signature and verification.

Signature: To sign message m, entity E_2 acts as follows:

1. Compute $P_m = H_1(DN_0 \| DN_1 \| DN_2 \| m)$;
2. Compute $\delta = S_{E_2} + \rho_2 P_m$, where ρ_2 is the secret point of entity E_2;
3. Output the signature $< \delta, P_m, Q_{ID_{E_2}|1}, Q_{ID_{E_2}|2} >$.

Verification: Other Entities can verify the signature by acting as follows: Confirm

$$\hat{e}(P, \delta) = \hat{e}(P, \rho_2 P_m)\ \hat{e}(Q_0, P_0) \prod_{i=1}^{2} \hat{e}(Q_{ID_{E_2}|i}, P_i) \tag{5}$$

if the equation is true, the signature is validated.

4 Identity-Based Authentication for Cloud Computing

In this section, based on the former IBE and IBS schemes, an identity-based authentication for cloud computing (IBACC) is proposed.

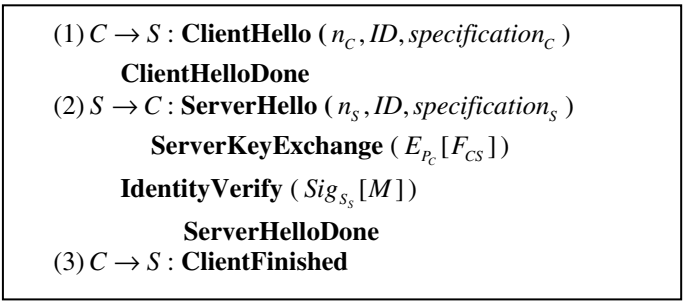

Fig. 2. Identity-based Authentication Protocol

where

n_C, n_S : the fresh random number
ID : the session identifier

$specification_C$: the cipher specification of C
$specification_S$: the cipher specification of S
F_{CS} : a pre-master secret used to generate the shared key
$E_{P_C}[F_{CS}]$: encrypt F_{CS} with the public key P_C of entity C using the encryption algorithm of IBE
M : all handshake messages since the ClientHello message
$Sig_{S_S}[M]$: sign M with the private key S_S of entity S using the signature algorithm of IBS

In step (1), the client C sends the server S a ClientHello message. The message contains a fresh random number n_C , session identifier ID and $specification_C$. $Specification_C$ extends from TLS to handle the IBE and IBS schemes. For example, $Specification_C$ could be the form $TLS\_IBE\_IBS\_WITH\_SHA\_AES$. IBE and IBS are used as secure transporting and authentication. SHA is the hash function. AES is the symmetric encryption algorithm.

In step (2), the server S responds with a ServerHello message which contains a new fresh random number n_S , the session identifier ID and the cipher specification $specification_S$. The $specification_S$ is S 's supporting ciphersuite. Then C chooses a pre-master secret F_{CS} and encrypts it with the public key P_C of entity C using the encryption algorithm of IBE. The ciphertext is transmitted to C as ServerKeyExchange message. Then S generates a signature $Sig_{S_S}[M]$ as the IdentityVerify message to forward to C . Finally, The ServerHelloDone message means the step (2) is over.

In step (3), C firstly verifies the signature $Sig_{S_S}[M]$ with the help of ID_S . Pass of verification means S is the valid owner of ID_S . This completes authentication form

S to C .Then C decrypts the $E_{P_C}[F_{CS}]$ with its private key S_C . Because of the fresh F_{CS} , the correct decryption indicates C is the valid owner of ID_C . This step authenticates the validity of C . The ServerFinished message means the step (3) finishes.

Eventually, a shared secret key between C and S is calculated by $K_{CS} = PRF(F_{CS}, n_C, n_S)$, where PRF is pseudo-random function.

5 Performance Analysis and Simulation

In this section, performance comparisons between SAP and IBACC are firstly discussed. Then simulation experiment gives precise results.

5.1 Communication Cost

The comparison of communication cost between the two different protocols is shown in table 1. Note that only dominant communication is considered, i.e. certificate, signed or encrypted messages, which may have the greatest consumptions of the network bandwidth.

Table 1. Comparison of communication cost

SAP		IBACC	
Certificate	RSA Signature	IBS Signature	IBE Ciphertext
2	2	1	1

Reference [3] shows that communication cost of SAP is two public key certificates and two RSA signatures. However, in the IBACC, the communication cost is only one IBS signature and one IBE ciphertext.

5.2 Computation Cost

The comparison of computation cost between the two different protocols is shown in table 2. Note that only dominant computation is considered, i.e. encryption, decryption and authentication.

Table 2. Comparison of computation cost

	SAP	IBACC
Client	1 ENC_R , 1 SIG_R and Authenticating server	1 ENC_I and 1 SIG_I
Server	1 DEC_R , 1 SIG_R and Authenticating client	1 DEC_I and 1 VER_I

Where

ENC_R = RSA encryption

DEC_R = RSA decryption

ENC_I = IBE encryption

DEC_I = IBE decryption

SIG_R = RSA signature

SIG_I = IBS signature

VER_I = IBS signature verification

Authenticating server=Including building certification path of server and verifying signatures.

Authenticating client= Including building certification path of client and verifying signatures.

The paper [3] showed that in the SAP, the computation cost of client was one RSA encryption, one RSA signature and Authenticating server. The computation cost of server was one RSA decryption, one RSA signature and Authenticating client. However, in the IBACC, the computation cost of client is one IBE encryption and one IBS signature. The computation cost of server is one IBE decryption and one IBS signature verification.

5.3 Simulation and Experiment Results

Simulation Platform and Reference

The platform of simulation experiment is GridSim [16] which is a simulation platform based on Java. Special users and resources can be generated by rewriting these interfaces. This aligns well with various users and resources of cloud computing. Furthermore, GridSim is based on SimJava which is a discrete event simulation tool based on Java and simulates various entities by multiple thread. This aligns well with randomness of cloud computing entity action. Therefore, it is feasible to simulate our proposed authentication protocol of cloud computing by GridSim.

The simulation environment is composed of four computers which are all equipped with P4 3.0 CPU, 2G memory. Certification chain is important for SAP. The shorter, the better. The shortest certification chain includes all 4 certifications: CA_1, client and CA_2, server. There are a cross authentication for CA_1 and CA_2. It is in this scene that SAP and IBACC are compared. Based on openssl0.9.7, SAP is implemented. Pairing computing adapts the algorithms of reference [17]. To precisely simulate the network delay, there are 25~45ms waiting time before messages are sent.

Simulation Results and Analysis

Fig.3 illustrates the authentication time of IBACC is approximately 571 ms while that of SAP is 980 ms. That is to say, authentication time of IBACC is 58% of that of SAP. Fig.4 shows thecommunication cost of IBACC is approximately 1785 bytes while that of SAP is 5852 bytes. That is to say, communication cost of IBACC is 31% of that of SAP. The simulation results confirm that the communication cost of IBACC is less and the authentication time is shorter.

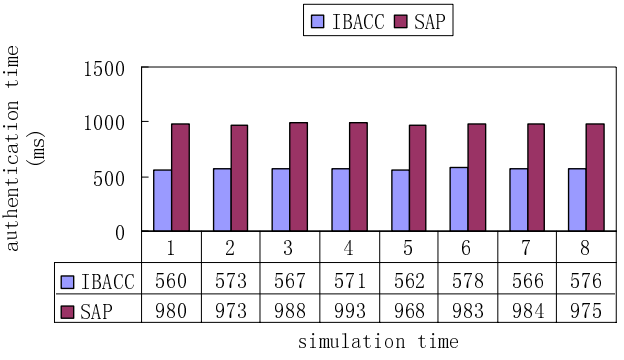

Fig. 3. Comparison of authentication time

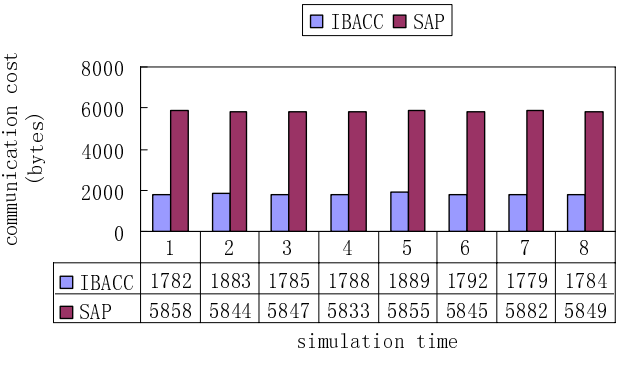

Fig. 4. Comparison of communication cost

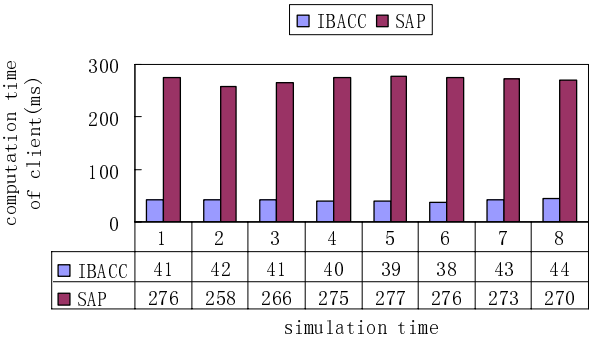

Fig. 5. Comparison of computation time of client

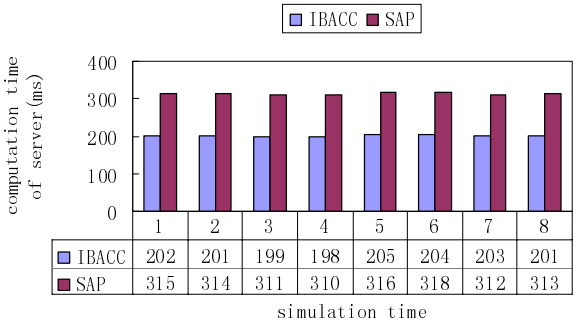

Fig. 6. Comparison of computation time of server

As shown in Fig.5, computation time of client for IBACC is approximately 41 ms while that for SAP is 272 ms. That is to say, computation time of client for IBACC is 15% of that for SAP. Fig.6 illustrates computation time of server for IBACC is approximately 202 ms while that for SAP is 313 ms. That is to say, computation time of server for IBACC is 65% of that for SAP. The simulation results confirm that both client and server of IBACC are more lightweight than those of SAP.

Furthermore, computation time of client is 20% of that of server in IBACC. This aligns well with the idea of cloud computing which allows the user with an average or low-end platform to outsource its computational tasks to more powerful servers. As a result, the more lightweight user side can connect more servers and contribute to the larger scalability.

6 Conclusion

Authentication is necessary in Cloud Computing. SSL Authentication Protocol is of low efficiency for Cloud services and users. In this paper, we presented an identity-based authentication for cloud computing, based on the identity-based hierarchical model for cloud computing (IBHMCC) and corresponding encryption and signature schemes. Being certificate-free, the authentication protocol aligned well with demands of cloud computing. Performance analysis indicated that the authentication protocol is more efficient and lightweight than SAP, especially the more lightweight user side. This aligned well with the idea of cloud computing to allow the users with an average or low-end platform to outsource their computational tasks to more powerful servers.

References

1. Erdogmus, H.: Cloud Computing: Does Nirvana Hide behind the Nebula? IEEE Software 26(2), 4–6 (2009)
2. Leavitt, N.: Is Cloud Computing Really Ready for Prime Time? Computer 42(1), 15–20 (2009)

3. Freier, A.O., Karlton, P., Kocher, P.C.: The SSL Protocol, Version 3.0.INTERNET-DRAFT (November 1996), http://draft-freier-ssl-version3-02.txt
4. Foster, I., Kesslman, C., Tsudik, G.: A Security Architecture for Computational Grids. In: ACM Conference on Computers and Security, pp. 83–90 (1998)
5. Mao, W.B.: An Identity-based Non- interactive Authentication Framework for Computational Grids, May 29 (2004), http://www.hpl.hp.com/techreports/2004/HPL-2004-96.pdf
6. Dai, Y.S., Pan, Y., Zou, X.K.: A hierarchical modelling and analysis for grid service reliability. IEEE Transactions on Computers 56(5), 681–691 (2007)
7. Dai, Y.S., Levitin, G., Trivedi, K.S.: Performance and Reliability of Tree-Structured Grid Services Considering Data Dependence and Failure Correlation. IEEE Transactions on Computers 56(7), 925–936 (2007)
8. Dai, Y.S., Levitin, G.: Reliability and Performance of Tree-structured Grid Services. IEEE Transactions on Reliability 55(2), 337–349 (2006)
9. Dai, Y.S., Xie, M., Wang, X.L.: Heuristic Algorithm for Reliability Modeling and Analysis of Grid Systems. IEEE Transactions on Systems, Man, and Cybernetics, Part A 37(2), 189–200 (2007)
10. Boneh, D., Gentry, C., Hamburg, M.: Space Efficient Identity Based Encryption without Pairings. In: Proceedings of FOCS 2007, pp. 647–657 (2007)
11. Boneh, D.: Generalized Identity Based and Broadcast Encryption Schemes. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 455–470. Springer, Heidelberg (2008)
12. Boyen, X.: General Ad Hoc Encryption from Exponent Inversion IBE. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 394–411. Springer, Heidelberg (2007)
13. Lim, H.W., Robshaw, M.: On Identity- Based. Cryptography and Grid Computing. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2004. LNCS, vol. 3036, pp. 474–477. Springer, Heidelberg (2004)
14. Lim, H.W., Robshaw, M.: A dynamic key infrastructure for GRID. In: Sloot, P.M.A., Hoekstra, A.G., Priol, T., Reinefeld, A., Bubak, M. (eds.) EGC 2005. LNCS, vol. 3470, pp. 255–264. Springer, Heidelberg (2005)
15. Chen, L., Lim, H.W., Mao, W.B.: User-friendly grid security architecture and protocols. In: Proceedings of the 13th International Workshop on Security Protocols (2005)
16. Buyya, R., Murshed, M.: GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. Journal of concurrency and computation practice and experience 14(13-15), 1175–1220 (2002)
17. Barreto, P.S.L.M., Kim, H.Y., Lynn, B., Scott, M.: Efficient algorithms for pairing-based cryptosystems. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 354–368. Springer, Heidelberg (2002)