BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **13**, No 3

Sofia • 2013

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2013-0027

An Efficient Access Control Scheme for Cloud Environment

Shan-Shan Tu, Shao-Zhang Niu, Meng-Jiao Li

School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, 100876, China
Email: tuss008@bupt.edu.cn

Abstract: *In order to keep the confidential data in the cloud against unauthorized parties, a cryptographic access control solution based on Attribute-Based Encryption (ABE) and Identity-Based Signature (IBS) is introduced in this paper. Under the premise that cloud service provider is untrustful, the proposed scheme can ensure the data security of the cloud storage system in an open environment, as well as reduce the complexity of management. Analysis and experimental results show that the scheme can be semantically secure against adaptive chosen cipher-text attacks under the random oracle model. Our concrete access control scheme can enhance the efficiency of the cloud to a certain extent.*

Keywords: *Secure storage, attribute-based encryption, identity-based signature, cloud computing***.**

1. Introduction

Cloud computing, as one of the most exciting fields of technology, denotes an architectural shift towards thin clients and scalable centralized provision of computing and storage resources on-demand. However, to allow the Cloud Service Provider (CSP) take care of confidential corporate data will certainly raise the underlying security and privacy. For instance, an untrustworthy CSP may sell the confidential information about an enterprise to its closest business competitors for making a profit. Therefore, a natural way to keep sensitive data confidential against an untrusted CSP, is to store only the encrypted data in the cloud.

We propose a control solution based on CP-ABE and IBS scheme, achieving fine-grained cryptographic access, and then prove its security under the random oracle model. Meanwhile, the communication costs and the computation costs of our scheme should be low enough, so that the users can successfully retrieve data from the cloud, and then decrypt it by the thin client. Based on our previous work for ABE system under the standard model [1], we continue to study the simplified solution of the fine-grained access control without revocation under the random oracle model.

The rest of this paper is organized as follows: We begin with a discussion of the related work in Section 2, and present some preliminaries in Section 3. Then we outline the efficient construction based on a private cloud for our scheme in Section 4. Next, we provide the security and performance analysis respectively, in Section 5. Finally, we conclude this paper in Section 6.

2. Related work

2.1. Identity-based encryption

Shamir [2] proposed the identity-based encryption and integer factoring difficulty-based signature scheme, which was a novel efficient and secure encryption and signature scheme. In IBE system, the user's public key can be any unique string, such as name, address, identity card number or other standard mark, and the private key is preserved and generated by a trusted PKG. In 2001 Boneh and Franklin [3] constructed a bilinear map on the elliptic curve, and then Selvi et al. [4, 5] presented several encryption schemes from bilinear pairings. In a recent work, Gentry and Halevi [6] proposed a fully secure HIBE scheme by using identity-based broadcast encryption with key randomization, and Waters [7] achieved full security in systems under a simple assumption by using a dual system encryption. Boyen [8] proposed an identity-based signature and encryption scheme, the basic idea utilized two-tier design, IBE and IBS combined in a safe manner, which makes its effectiveness and security greater than the independent IBE or IBS. With advantages, such as security, compactness, availability and realizability, Xavier's idea inspired us to consider implicating this method in cloud computing networks.

2.2. Attribute-based encryption

In a recent work Chase [9] provided a construction for a multi-authority ABE system, where each authority would administer a different domain of attributes. Chase and Chow [10] provided a more practice-oriented multi-authority ABE system, which removes the trusted central authority while preserving user privacy. Reference [11] proposed an attribute-based signature (ABS, attribute-based signature), using a matrix with the properties of bilinear pairing and the structure of the monotone Boolean function, which has the advantage of being able to resist the collusion attack. Reference [12] proposed an attribute-based group signature method. Among others, Yu et al. [13] exploited and uniquely combined techniques of ABE, Proxy Re-Encryption (PRE) [14], and Lazy Re-Encryption (LRE) [15] to delegate most of the computation tasks involved in user revocation to untrusted CSPs without disclosing the underlying data contents, which might make a KP-ABE system more applicable in a cloud environment. Since each file is associated with an access control rather than a set of attributes as KP-ABE, it is harder to delegate the re-encryption operation to a third party. Then Wang et al. [16, 17] and Yu et al. [18] made the proxy re-encryption technique applied to CP-ABE.

3. Preliminaries

3.1. Bilinear pairings

Let p be a large prime number, $p \equiv 2 \bmod 3,$ and there is a large prime number q, such that $p = 6q - 1$. Let Z be the additive group of prime order q, $Z_q = \{0, \cdots, q-1\}$, Z^+ be a positive integer. Then $E/\mathrm{GF}(p)$ is the elliptic curve constructed on $\mathrm{GF}(p)$: $y^2 = x^3 + 1$, p is a point on the curve which order is q. The cyclic group generated by P is denoted by G_1, G_2 is the q-order subgroup on $\mathrm{GF}(p^2)$.

Pairing bilinear $\hat{e}: G_1 \times G_1 \to G_2$ is a map with the following properties.

1) Bilinearity: If for any $P, R, Q \in G_1$ we have

$$\hat{e}(P+Q, R) = \hat{e}(P, R)\hat{e}(Q, R),$$
$$\hat{e}(P, Q+R) = \hat{e}(P, Q)\hat{e}(P, R),$$

and for $a, b \in Z$ we have

$$\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab} = \hat{e}(P, abQ) = \hat{e}(abP, Q).$$

then the mapping $\hat{e}$ is called a bilinear map.

2) Non-degeneracy: if there are $P, Q \in G_1$, then $\hat{e}(P, Q) \neq I_{G_2}$.

3) Computability: There is a polynomial time algorithm to compute $\hat{e}(P, Q)$.

3.2. Linear secret sharing scheme

Let $P = \{P_1, P_2, \cdots, P_n\}$ be the set of the participants, an access structure $\mathbb{A}$ on P is a set of some subsets of P, that is $\mathbb{A} \subset 2^P, \phi \notin \mathbb{A},$ and meets the monotonic increase property: for any $A \in \mathbb{A}$ and $B \subset P$, if $A \subset B$, then $B \in \mathbb{A}$. A key sharing scheme to achieve the access structure $\mathbb{A}$ is by the distribution function $\Pi: S \times R \to S_1 \times \cdots \times S_n$ shared among the participants $P_1, P_2, \cdots, P_n$, $r \in R$ is a random input, each participant who grasps a sub-master key is s_i, $1 \le i \le n,$, such that:

1) for any $A \in \mathbb{A}$, the members in A can restore the master secret s, that is, $H(S \mid \Pi(S, R)\mid_A) = 0$, where $H(\cdot)$ is the entropy function;

2) for any $B \notin \mathbb{A}$, the members in B cannot obtain any information about the master secret s, that is, $H(S \mid \Pi(S,R)|_B) = H(S)$.

If $|S| = |S_i|$ for any $1 \le i \le n,$ then the secret sharing system is ideal. If $S = K$ is a finite field, R and $S_i, 1 \le i \le n,$ are the linear spaces over K, and Π is a linear function, then the secret sharing scheme is called a linear secret sharing system.

Let K be a field, $\{x_1, x_2, \cdots, x_n\}$ is the label set, $\rho : \{\text{rows of } M\} \to \{x_1, x_2, \cdots, x_n\}$ denotes that the rows of the matrix M are labeled with $x_1, x_2, \cdots, x_n$. Different rows can have one and the same label. The labeled matrix is represented by $\hat{M}(M, r) = \hat{M}$, called a Monotone Span Program (MSP). For any matrix M on K, span(M) denotes the linear space produced by row vectors of matrix M.

For each input set γ, matrix M_γ is composed of rows of members marked by γ from M. If and only if $\vec{1} \in \text{span}(M_r),$ the monotone span program $\hat{M}$ accepts γ. If the monotone span program just accepts γ, in which $f_M(\gamma) = 1,$ then calculate the Boolean function f_M. The size of $\hat{M}$ is the number of rows of matrix M.

In this article, the attributes are assumed to be participants, so the rows of the matrix will be marked by attributes.

3.3. Access tree

Let T represents an access structure tree, each non-tree leaf node represents a threshold, which is described by the threshold value and its child nodes. n_x is the number of children of non-leaf nodes x, k_x is the threshold value, then $0 < k_x \le n_x$. When $k_x = 1$, the threshold is an OR gate; when $k_x = n_x$ the threshold is AND gate. Each leaf node in the tree is described by an attribute and a threshold value $k_x = 1$. parent(x) is the parent of node x. When the node is a leaf node x, the function att(x) represents the attribute linked with the tree leaf node x. The access tree T determines each node's number from 1 to n. The function inder(x) returns the number linked with the nodes x. In any way for the given secret key, inder(x) value of node in access structure is the only designated.

Let the access tree's root be r, T_x represents the sub-tree of the root x, then T and T_r are the same. If the attribute set meets the access tree, it can be indicated by $T_x(\gamma) = 1$. $T_x(\gamma)$ is calculated as follows:

1) if X is a non-leaf node, all child nodes x' of x calculate $T_x(\gamma)$;

2) if and only if at least k_x children nodes of $T_{x'}(\gamma)$ return 1, $T_x(\gamma)$ return 1;

3) if x is a leaf node, then $T_x(\gamma)$ return 1.

3.4. Related calculation assumptions

Bilinear Diffie-Hellman (BDH) problem: For unknown $a,b,c \in Z_q^*$, given $(P,aP,bP,cP) \in G_1^4$, the BDH problem on G_1 is to calculate $\hat{e}(P,P)^{abc}$.

Definition 1. In solving the BDH problem on G_1, any probabilistic polynomial-time algorithm A's advantage $\mathrm{Adv}_A^{\mathrm{BDH}}$ is defined as:

$$\mathrm{Adv}_A^{\mathrm{BDH}} = \Pr[A(P,aP,bP,cP) = \hat{e}(P,P)^{abc} \mid a,b,c \in Z_q^*].$$

BDH assumption is that for any probabilistic polynomial time algorithm A, advantage $\mathrm{Adv}_A^{\mathrm{BDH}}$ is negligible.

Computational Diffie-Hellman (CDH) problem: For unknown $a,b \in Z_q^*$, given $(P,aP,bP) \in G_1^3$, the CDH problem on G_1 is to calculate abP.

Definition 2. In solving CDH problem on G_1, any probabilistic polynomial-time algorithm A's advantage $\mathrm{Adv}_A^{\mathrm{CDH}}$ is defined as

$$\mathrm{Adv}_A^{\mathrm{CDH}} = \Pr[A(P,aP,bP) = abP \mid a,b \in Z_q^*].$$

CDH assumption that for any probabilistic polynomial time algorithm A, advantage $\mathrm{Adv}_A^{\mathrm{CDH}}$ is negligible.

4. Our construction

4.1. Construction definition

We consider the following application scenario (Fig. 1): Company A pays a CSP for sharing corporate data in cloud servers.

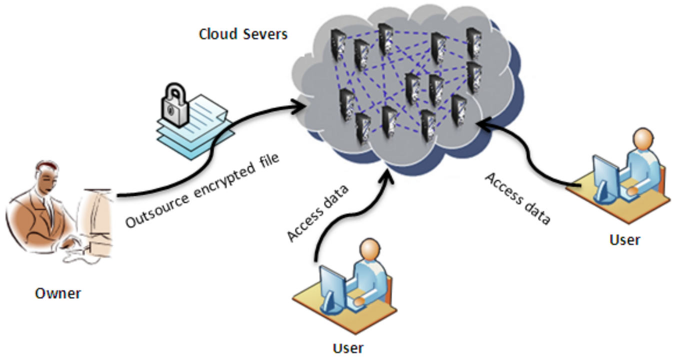

Fig. 1. Sample of an application scenario

We assume that the system is composed of the following parties: the CSP, the Trusted Third Party (TTP), enterprise users, end user and DepartMent (DM). the CSP operates a large number of interconnected cloud servers with abundant storage capacity and computational power to provide high quality services; the TTP is responsible for generating the initial parameter; company A that pays for sharing corporate data in cloud servers is an enterprise user; all personnel in the company who share data in cloud servers are end users; the department is responsible for generating keys for the end users; we use Fig. 2 as an example to illustrate these parties.

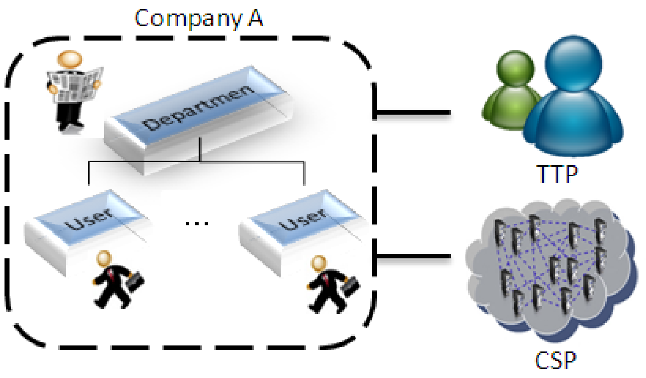

Fig. 2. Scenario model

4.2. Algorithm definition

Our scheme includes Setup, Extract, Sign, Encrypt, Key Generation, Decrypt and Verify as follows.

Setup. Let G_1 be a bilinear group of prime order q, P be a generator of G_1, $\hat{e}: G_1 \times G_1 \to G_2$ be the bilinear map**,** and k be a security parameter to determine the scale of a group. $\Delta_{i,s}(x) = \prod_{j\in S, j\neq i} \frac{x-j}{i-j}$ is the Lagrange operator, in which $i \in Z_q$, $S \subset Z_q$. Let each attribute be related to the only element in Z_q^* and choose discrete functions:

$$H_0 : \{0,1\}^* \to G_1^*,$$

$$H_1 : G_1^* \times \{0,1\}^* \to F_p^*,\ H_2 : G_2^* \to F_p^*.$$

Define $U = \{1, 2, \cdots, n\}$ to be the set of attributes, let $i \in U$, randomly choose $t_i \in Z_q$ and $y \in Z_q$.

$\text{PK} = (T_1 = t_1 P, \cdots, T_{|U|} = t_{|U|} P, Y = \hat{e}(P,P)^y, P_{\text{pub}})$ is the public key, and $\text{MK} = (t_1, \cdots, t_{|U|}, y)$ is the master key, in which $P_{\text{pub}} = yP$.

Extract. Given the strings $\text{Id}_A \in \{0,1\}^*$, generate the secret key.

Calculate $Q_A = H_0(\text{Id}_A) \in G_1$ and $K_A = y(Q_A)$ to be the secret key.

Sign. Randomly choose $l \in F_p^*$, and sign the sender identity Id$_A$, Q_A, K_A and message $m \in M$. Then calculate $j = l(Q_A) \in G_1^*$; let $h = H_1\left[j, m\right] \in F_p^*$, and calculate $v = (l+h)K_A \in G_1^*$. Therefore $\langle j, v\rangle$ is the signature.

Encrypt $\left(\langle m, \mathrm{Id}_A\rangle, \gamma, \mathrm{PK}\right)$**:** Encrypt the message $m \in G_2$ by the attributes set γ. Randomly choose $s \in Z_q$, and then ciphertext E is obtained,

$$E = (\gamma, E' = \langle m, \mathrm{Id}_A\rangle Y^s, \left\{E_i = sT_i\right\}_{i\in\gamma}, T, j, v).$$

Key generation $(T, \mathrm{MK}, \gamma, \gamma')$. An algorithm can generate the secret key, if and only if $T(\gamma') = 1$, the ciphertext can be encrypted by the attribute set γ. For each node x (including the leaf nodes, and $\mathrm{node}_{\mathrm{leaf}} \in \gamma'$) in the minimum tree T which meets $T(\gamma') = 1$, choose a Lagrange polynomial q_x.

Let $d_x = k_x - 1$ be the degree of the polynomial q_x for node x. For the root node r, let $q_r(0) = y$, and randomly choose q_r child nodes (and polynomial coefficients) to define the polynomial q_r completely (such as a child node $q_r(\mathrm{index}(x))$).

For other nodes x, let $q_x(0) = q_{\mathrm{parent}(x)}(\mathrm{index}(x))$, and randomly choose d_x child nodes (and polynomial coefficients) to define the polynomial q_r completely. Then the root node r has a master key y, the other node x has a sub-key $q_x(0)$ and the related Lagrange coefficient $\Delta_{i,s}(x)$. Moreover, for any un-leaf node, let s_x be the set of any d_x child nodes z from nodes x in the tree T, then $\sum_{z\in s_x} q_x(i)\Delta_{i,s_x'}(0) = q_x(0)$, in which $i = \mathrm{index}(z)$, $s_x' = \{\mathrm{index}(z) : z \in s_x\}$. Therefore, for any leaf node, distribute the user a secret key $D_x = \frac{q_x(0)}{t_i}P$, in which $i = \mathrm{att}(x)$.

On the other hand, for any node x in the tree T, let $\mathbb{S}_x$ be the set of all nodes on the path from node x to the root node r, and f_x be the product of all node Lagrange coefficients $\Delta_{i,s_x'}$ from $\mathbb{S}_x$.

Then $D_x = \left\{\frac{q_x(0)}{t_i}P, i = \mathrm{att}(x), i \in \gamma', f_x, x \in \gamma'\right\}$ is the decryption key.

Decrypt (E, D)**.** For the ciphertext:

$E = (\gamma, E' = \langle m, \mathrm{Id}_A\rangle Y^s, \{E_i = sT_i\}_{i\in\gamma}, T, j, v)$, the private key D, the set of attributes γ' which meets $T(\gamma') = 1$ (that is the leaf node x ($i = \mathrm{att}(x), i \in \gamma'$) in the tree T) and the root node r.

Calculate. $F_r = \prod_{x \in r} \hat{e}(D_x, E_i)^{f_x}, i = \mathrm{att}(x),$ and then $\langle m, \mathrm{Id}_A \rangle = E'/F_r.$

Verify. $\langle \hat{m}, \hat{\mathrm{Id}}_A \rangle$ is decrypted and $\langle \hat{j}, \hat{v} \rangle$ is received. Let $\hat{h} = H_1[\hat{j}, \hat{m}],$ then check

$$\hat{e}(P, \hat{v}) \overset{?}{=} \hat{e}(P_{pub}, \hat{j} + \hat{h}\hat{Q}_A), \text{ if so, then}$$

$$\langle \hat{m}, \hat{j}, \hat{v}, \hat{\mathrm{Id}} \rangle = \langle m, j, v, \mathrm{Id} \rangle.$$

4.3. Scheme construction

Based on the algorithm proposed in this paper, this section gives an integration of the key establishment, distribution and encryption of the signature for the cloud network, which consists of three parts.

1. The initialization process

First, the TTP uses a **Setup** function to calculate and output the public parameters:

$\mathrm{PK} = (T_1 = t_1 P, \cdots, T_{|U|} = t_{|U|} P, Y = \hat{e}(P, P)^y, P_{\mathrm{pub}})$ $P_{\mathrm{pub}} = yP$ and the master key $\mathrm{MK} = (t_1, \cdots, t_{|U|}, y).$

Second, the DM uses **Extract** function to generate the identity key for a given string $\mathrm{Id}_A \in \{0,1\}^*$. Calculate $Q_A = H_0(\mathrm{Id}_A) \in G_1$ and obtain the identity secret key $K_A = y(Q_A).$

The Id, K and the assigned to the attributes of the node have been written into the end users, then each user has its own attributes, identity keys, and related public parameters.

2. The signature and encryption process

If the end user A wants to send their data to the CSP, first, for a plaintext m, the A's identity Id$_A$ as the public key, Q_A and A's private key K_A, use a **Sign** function to calculate the signature $\langle j, v \rangle$, in which $j = l(Q_A) \in G_1^*$, $v = (l + h)K_A \in G_1^*$, $h = H_1[j, m] \in F_p^*$.

Second, use **Encrypt**$(\langle m, \mathrm{Id}_A \rangle, \gamma, PK)$ function, the public parameters PK and the plaintext $m \in G_2$ encrypted by the set of attributes γ, obtain the ciphertext: $E = (\gamma, E' = \langle m, \mathrm{Id}_A \rangle Y^s, \{E_i = sT_i\}_{i \in \gamma}, T, j, v).$

3. The attribute key generation, decryption and authentication process

If the end user B wants to receive the data, first, check $T(\gamma')=1$ with the attribute set γ' in its own set of attributes γ. If $T(\gamma')=1$, random choose $\mu \in Z_q^*$, calculate the public key mQ_B.

Send $(T, \gamma, \gamma', \mathrm{Id}_B, \mu Q_B)$ to the TTP.

The TTP checks the correspondence of the node Id$_B$ and γ'. If the attributes in γ' belong to the user B, use a **Key Generation** function $(T, \mathrm{MK}, \gamma, \gamma')$ to calculate the client attribute key:

$$D_X = \left\{ \frac{q_X(0)}{t_i} P, i = \mathrm{att}(x), i \in \gamma', f_X, x \in \gamma' \right\}.$$

The TTP randomly chooses $\eta \in Z_q^*$ to encrypt the attribute key D, that is, the attribute key encrypted ciphertext is $c = \{\eta Q_B, D \oplus H_2(g_B{}^{\eta})\}$ where $g_B = \hat{e}(\mu Q_B, P_{\mathrm{pub}}) \in G_2$, the ciphertext c is sent to the user B.

After receiving the ciphertext c, B uses m to decrypt the attribute key D. Let $c = \langle U, V \rangle$ be the ciphertext, calculate as follows: $D = V \oplus H_2(\hat{e}(U, P_{\mathrm{pub}})^{\mu})$.

The consistency of encryption and decryption is the same with IBE, according to the following formula:

$$\hat{e}(\mu Q_B, P_{\mathrm{pub}})^{\eta} = \hat{e}(\eta Q_B, P_{\mathrm{pub}})^{\mu} = \hat{e}(U, P_{\mathrm{pub}})^{\mu}.$$

B uses **Decrypt**(E, D) function, for the leaves of the tree node x and the root node r, calculate

$$F_r = \prod_{x \in r} \hat{e}(D_x, E_i)^{f_x}, i = \mathrm{att}(x)$$

and obtain $E'/F_r = \langle m, \mathrm{Id}_A \rangle$.

B uses **Verify** function to confirm the obtained $\langle \hat{m}, \hat{\mathrm{Id}}_A \rangle$ and the received $\langle \hat{j}, \hat{v} \rangle$.

In addition, before the construction of actual private cloud networks, the master key in the initialization process is generally generated and managed by the TTP, and the attribute key needs to be generated in the private cloud networks communication for a specific access to the tree. To secure the delivery of the attribute keys, the TTP should encrypt the attribute key in practical applications. Shamir's IBE algorithm decryption is used to encrypt the attribute key in this scheme.

A new client in the private cloud network needs to be initialized by the management system, such as TTP, to obtain the corresponding identity attributes. Similarly, if the client leaves, the management system needs to recover their occupied resources, such as the IP address and the corresponding attributes.

5. Security and performance analysis

5.1. Security analysis

1. Confidentiality

Theorem 1. If the Computational Diffie-Hellman (CDH) problem in group G_1 is difficult, then the proposed attribute-based encryption and identity-based signature algorithm in the random oracle model at any probabilistic polynomial time under an adaptively chosen ciphertext attack IND-CCA2 is secure.

Proof: Given an instance of the random CDH problem (P, aP, bP), the challenger C's goal is to calculate abP. If there is an IND-CCA2 adversary A (Let A knows the system parameters $q, G_1, G_2, \hat{e}$) can successfully attack the encryption signature scheme, then prove that C can take advantage of A to solve the CDH problem.

C makes $sP = aP$, A selects two messages m_0 and m_1, the sender Id$_A$, the receiver Id$_B$ who will be attacked and the receiver's attributes, and requests C to encrypt and sign the message. C makes $q_r(0) = bP$. Subsequently C randomly selects m_b from m_0 and m_1, $b \in \{0,1\}$, encrypts the message m_b in accordance with the signature encryption algorithm, and sends A the ciphertext E. A does not know the user's private key according ciphertext E, can only guess b', if $b' = b$, C considers:

$F_r = \hat{e}(P, P)^{sq_r(0)} = \hat{e}(aP, bP) = \hat{e}(abP, P) = \hat{e}(P, P)^{ab}$. In other words, C calculates abP successfully, and then calculates $\hat{e}(P, P)^{ab}$. C solves the CDH problem.

2. Unforgeability

Theorem 2. If the computational Diffie-Hellman 1problem in group G_1 is difficult, then the proposed attribute-based encryption and identity-based signature algorithm in the random oracle model at any probabilistic polynomial time under an adaptively Chosen Message Attack IND-CMA2 is secure.

Proof: Given an instance of the random CDH problem (P, aP, bP). The challenger C's goal is to calculate abP. If there is an IND-CMA2 adversary A (let A knows the system parameters $q, G_1, G_2, \hat{e}$), can successfully attack the encryption signature scheme, then prove that C can take advantage of A to solve the CDH problem.

C makes $P_{\text{pub}} = aP$, A selects a sender Id$_A$ who will be attacked, C makes $Q_i = r_i P$, $K_i = r_i aP = aQ_i$ and $Q_A = bP$. A's private key is unknown, A generates a forged signatures v^* for a message m^*. If v^* is a legitimate signature of the user provided by A, that is, the message m^* can be obtained by the authentication algorithm, then C generates two legitimate signatures for message m by application of random oracle technology: $v' = (l + h')K_A$ and $v'' = (l + h'')K_A$, $h' \neq h''$. Then

$$(v' - v'')/(h' - h'') = K_A = abP.$$

In other words, C calculates abP successfully. C solves the CDH problem.

3. Other safety analysis

Non-repudiation: If there are signatures $\langle j, v\rangle$ generated by A's private key K_A and an arbitrary random integer l, due to these two variables known just by A, then the encrypted signer A cannot deny their behavior. However, for the recipient, as long as his attribute set γ' meets $T(\gamma') = 1$, then with the assistance of the PKG to decrypt the ciphertext, the decipher B can deny their behavior.

Resistance collusion: After the TTP receives $(T, \gamma, \gamma', \mathrm{Id}_B, \mu Q_B)$, first it checks the correspondence of the nodes Id_B and γ', if the attributes in g' belong to node B, then with the application of the **Key Generation** function calculate the node attribute key D; if the attributes are from different nodes, then it will not send an attribute key D to any node.

5.2. Performance analysis

Considering a thin client with low computing power, small memory and other features, choose the Lagrange polynomial for each node of the tree. Lagrange coefficient calculation and attribute-key encryption are not in the client (these calculations are completed by the TTP). In the attribute key generation process, the clients only need to complete less computing.

Table 1. Comparison of the relevant schemes

Properties	IBBSC [5]	PFIBE [16]	Our scheme
User key size	$O(L)$	$O(L+I)$	$O(L)$
Ciphertext	$O(3N)$	$O(NT+n)$	$O(d+1)$
Encryption	$O(1)$ (map)	$O(NT+n)$ (exp)	$O(d+1)$ (exp)
Decryption(map)	$O(2)$	$O(1)$	$O(d+2)$
Access control over IDs or attributes	No	Yes	Yes

Table 1 gives the computational cost comparison of the main algorithms in IBBSC scheme [5], PFIBE scheme [16] and our scheme. In Table I, L is the number of attributes associated with a user, I is the maximum depth of DMs administering attributes associated with a user, N is the number of conjunctive clauses in an access structure, T is the maximum depth of DMs administering attributes in the access control, n is the number of recipients. d is the number of attributes in the set γ' which meets $T(\gamma') = 1$.

It can be seen from Table 1 that IBBSC[5] uses a bilinear operation in the encryption process which causes the greatest delay. Therefore, its computational efficiency is relatively low. PFIBE [16] uses only one time the complex bilinear operation in the decryption process, but its computation is related to the network scale, when the network size is larger, more multiplication and exponentiation computations make the cost rapidly increasing. Our scheme is much more efficient than the other two methods, being independent from the network size and fit for private cloud networks.

5.3. Experimental analysis

The experimental equipment consists of 1.73 GHz Inter(R) Core(TM), 2 GB DDR2 and Windows Server 2003. The experimental environment is constructed by Red Hat Enterprise 5 in the VMware Workstation 6.5.1 virtual machine, and distributed to 1GB RAM. Our experiments just consider a different length of the master keys, and ignore the transmission delay of data in the distributed network. The longer the sign-encryption master key is, the higher the security is. However, at the same time it will cause the corresponding sign-encryption time to become longer. For different numbers of attributes and key lengths, the computing time of the proposed algorithm and IBBSC [5] is compared in Figs 3 and 4, where d is the number of attributes, t is the number of clients, the running time is the average of the amount 10 times.

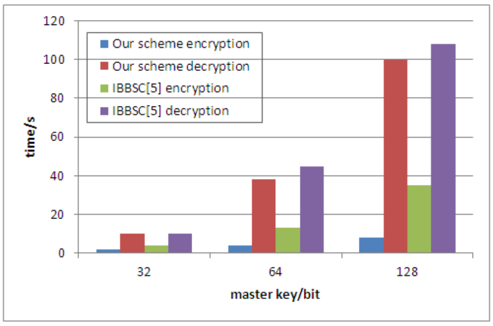

Fig. 3. Comparison of the computing time in different master keys ($d = 2$, $t = 25$)

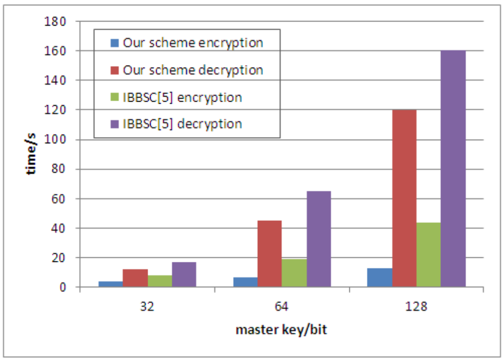

Fig. 4. Comparison of the computing time in different master keys ($d = 3$, $t = 50$)

From Figs 3 and 4 we can figure out that both the encryption and decryption time cost in our scheme is less than in IBBSC method, and as the length of the master key increases, our scheme is more efficient. Also, while the number of attributes d and clients t increases, the efficiency of our method is more obvious. This improvement is meaningful since the algorithm running time has a direct impact on the real-time encryption system.

Remark: This paper does not use the standard arithmetic library, it has not been optimized to reach the standards of the commercial library, so if using the commercial library, the average of the algorithm running time can be shortened. On the other hand, the choice of the prime number has also an important impact on the performance of the algorithm.

6. Conclusion

In this paper we proposed a cryptographic access control solution based on CP-ABE and IBS schemes in cloud computing, so that to simultaneously achieve: (1) low manage complexity; (2) fine-grained access control; (3) thin client adaptability; (4) data unforgeability. We proved the scheme, which is also collusion resistant, to be semantically secure against adaptively chosen ciphertext attacks under the random oracle model.

In future work we will design expressive and scalable user revocation schemes for cloud servers under the random oracle model or the standard model.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (No 61070207).

References

1. Tu, S., S. Niu, H. Li. A Fine-Grained Access Control and Revocation Scheme on Clouds. Concurrency Comput. Pract. Exper, 2012, ISSN: 15320626, DOI: 10.1002/cpe.2956.
2. Shamir. Identity-Based Eryptosystem and Signature Schemes. – In: Proc. of CRYPTO’84 Conf., 1984, ISBN: 978-3-540-15658-1, 47-53.
3. Boneh, D., M. Franklin. Identity-Based Encryption from the Weil Pairing. – In: Proc. of Crypto’2001 Conf., 2001, ISBN: 978-3-540-42456-7, 213-229.
4. Selvi, S., S. Vivek, N. Jain. Cryptanalysis of Li et al.'s Identity-Based Threshold Signcryption Scheme. – In: Proc. of EUC’08 Conf., 2008, ISBN: 978-0-7695-3492-3, 127-132.
5. Selvi, S., S. Vivek, R. Gopalakrishnan. Cryptanalysis of Mu et al.'s and Li et al.'s Schemes and a Provably Secure id-Based Broadcast Signcryption (IBBSC) Seheme. – In: Proc. of WISA 2008 Conf., 2009, ISBN: 978-3-642-00305-9, 115-129.
6. Gentry, C., S. Halevi. Hierarchical Identity Based Encryption with Polynomially Many Levels. – In: Proc. of TCC Conf., 2009, ISBN: 978-3-642-00456-8, 437-456.
7. Waters, B. Dual System Encryption: Realizing Fully Secure IBE and HIBE Under Simple Assumptions. – In: Proc. of CRYPTO’2009 Conf., 2009, ISBN: 978-3-642-03355-1, 619-636.
8. Boyen, X. Multipurpose Identity-Based Signcryption: A Swiss Army Knife for Identity-Based Cryptography. – In: Proc. of CRYPTO’2003 Conf., 2003, ISBN: 978-3-540-40674-7, 383-399.

9. Chase, M. Multi-Authority Attribute Based Encryption. – In: Proc. of TCC Conf., 2007, ISBN: 978-3-540-70935-0, 515-534.

10. Chase, M., S. Chow. Improving Privacy and Security in Multi-Authority Attribute-Based Encryption. – In: Proc. of ACM CCS Conf., 2009, ISBN: 978-1-60558-894-0, 121-130.

11. Maji, H., M. Prabhakaran, M. Rosulek. Attribute-Based Signature: Achieving Attribute-Privacy and Collusion-Resistance, 2008. **http://eprint.iacr.org/2008/328**

12. Khader, D. Attribute Based Group Signature With Revocation, 2007. **http://eprint.iacr.org/2007/241.**

13. Yu, S., C. Wang, K. Ren, W. Lou. Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing. – In: Proc. of INFOCOM Conf., 2010, ISBN: 978-1-4244-5836-3, 534-542.

14. Goh, E., H. Shacham, N. Modadugu, D. Boneh. Sirius: Securing Remote Untrusted Storage. – In: Proc. of NDSS Conf., 2003, ISBN: 1-891562-16-9, 131-145.

15. Blaze, M., G. Bleumer, M. Strauss. Divertible Protocols and Atomic Proxy Cryptography. – In: Proc. of EUROCRYPT'98 Conf., 1998, ISBN: 978-3-540-64518-4, 127-144.

16. Wang, G., Q. Liu, J. Wu. Achieving Fine-Grained Access Control for Secure Data Sharing on Cloud Servers. – Concurrency Comput. Pract. Exper., Vol. **23**, 2011, No 12, 1443-1464.

17. Wang, G., Q. Liu, J. Wu, M. Guo. Hierarchical Attribute-Based Encryption and Scalable User Revocation for Sharing Data in Cloud Servers. – Computers and Security, Vol. **30**, 2011, No 5, 320-331.

18. Yu, S., C. Wang, K. Ren. Attribute Based Data Sharing With Attribute Revocation. – In: Proc. of ASIACCS Conf., 2010, ISBN: 978-1-60558-936-7, 261-270.