International Journal of Security and Its Applications
Vol. 10, No. 2 (2016), pp.217-236
http://dx.doi.org/10.14257/ijsia.2016.10.2.20

Effective and Secure Access Control for Multi-Authority Cloud Storage Systems

Lin Xin[1], Xingming Sun[1], Zhangjie Fu[1], Liang-Ao Zhang[1] and Jie Xi[1]

[1]School of Computer and Software & Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, 210044, China
nj03xinlin@163.com, sunnudt@163.com, wwwfzj@126.com, zlo2010@163.com, jeese1226@163.com

Abstract

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is a promising cryptographic tool to implement access control for secure cloud storage systems. However, most existing access control schemes based on CP-ABE for the multi-authority cloud storage systems rely on a fully trusted global certificate authority. It is just an ideal assumption while there never exists a fully trusted global certificate authority in reality. In this paper, we construct a system with multiple certificate authorities (CA). The parameters of those CAs could be verified when an authority suspects the messages received from the correlative certificate authority. Besides, we construct a verifiable secret sharing (VSS) scheme to realize the decentralization of the certificate authority in our scheme. The scheme adopts the Pedersen commitment in combination with the properties of bilinear-pairs on elliptic curve and bilinear Diffie-Hellman problem. The analysis shows that our scheme is highly efficient, authentic and provably secure under the security model. Our scheme simultaneously supports efficient attribute revocation.

Keywords*: Access Control, Verifiable Secret Sharing, CP-ABE, Cloud Storage*

1. Introduction

The Internet technology develops rapidly nowadays. With the fast-changing distributed application, the demand to implement safe, reliable, efficient data sharing and processing is increasingly strong. Cloud computing provides a pay-per-use model for enabling available, convenient and on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services). These resources can be rapidly provisioned and released with minimal management effort or service provider interaction.

With the aid of cloud computing, small- and medium-size businesses go from being constrained to certain geographies due to budget limitations to having the ability to scale globally with significantly reduced overhead costs. At the academia side, there are also notable examples such as VCL and FutureGrid to provide support to academia research projects. However, the security and privacy of the sensitive data in the cloud have risen the users' great concerns as the cloud serves cannot be fully trusted. To prevent the untrusted servers from accessing sensitive data, data access control is an efficient way to ensure the data security in the cloud.

ISSN: 1738-9976 IJSIA

As the service providers cannot be entirely trusted, traditional server-based access control methods no longer apply to cloud storage systems. In recent access control systems, cryptographic techniques are well applied to access control for remote storage systems [2]–[4]. Some works [5]-[6] deliver the key management and distribution from the data owners to the remote server under the assumption that the server is trusted or semi-trusted. Attribute-based Encryption (ABE) is a promising technique that is designed for access control of encrypted data. Goyal et al. [7] formulated the ABE into two complimentary forms: Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE).

Extensive research has been done for CP-ABE. Because CP-ABE does not require the data owner to distribute keys and gives them more direct control on access policies. However, a user may hold attributes distributed by multiple authorities which is responsible for attribute management and key distribution in CP-ABE schemes. For another, the data owner may share his data to users managed by different authorities. Existing CP-ABE schemes cannot be directly applied to the access control for multi-authority cloud storage systems for lack of efficiency. Yang et al. [1] proposed a scheme for effective data access control for multi-authority cloud storage systems. The aim of our paper is to study the security issue about its fully trusted single global certificate authority. To properly improve its security, we apply a verifiable secret sharing scheme.

Threshold secret sharing is an important means in information security and data security. It plays a critical role in safely storing, transmitting and legally using important information as well as secret data. In recent years, many researchers mainly focus on studying against fraud problems in secret sharing scheme. One after another, they respectively propose threshold secret sharing schemes based on Lagrange interpolation polynomial, projective geometry, Chinese Remainder Theorem and matrix multiplication. In general secret sharing scheme there exist two problems: on the one side, whether the distribution center sends the real shadows to each member, and how each member verifies whether their received shadow is real; on the other side, how to identify the correctness of the shadows provided by members in the secret recovery phase. This paper proposes a verifiable secret sharing scheme based on bilinear pairings to be applied to DAC-MACS.

In this paper, we focus on solving the fully trusted global problem while the global certificate authority can never be fully trusted. The scheme in this paper takes advantage of Pedersen commitment as well as the properties of bilinear-pairs on elliptic curve and bilinear Diffie-Hellman problem to propose a verifiable secret sharing scheme. Then it will be applied to implement a ciphertext-policy ABE which enables the cloud to verify whether there is any fraudulent conduct in certificate authorities. If every CA successfully verifies the message sent by another CA using the latter CA’s public commitment, the cloud is confident that the CAs faithfully perform its operations. If a malicious CA attempts to participate in the generation procedure of important parameters, the other CAs will find that the verification equation disconfirm and then refuse any value from the malicious CA. As a result, the security of the system is further improved.

Our Contributions. We formulate the security problem when the global certificate authority collapses or breaks down, or even is motivated to defect. Our constructions are as follows:

1. We construct an additional verifiable secret sharing (VSS) scheme to decentralize the power of the certificate authority to reduce the influence when it was compromised. The VSS scheme adopts the Pedersen commitment in combination with the properties of bilinear-pairs on elliptic curve and bilinear Diffie-Hellman problem.

2. We construct CP-ABE based secure access control for cloud storage system with multiple certificate authorities (CAs). The parameters of those CAs could be verified when an authority suspects the messages received from the correlative certificate authority. The cost is to introduce little overhead for multi-CAs. Nevertheless, the security of our scheme is improved and the latter parts will confirm it.

1.1. Related Work

Cloud storage service allows owners to outsource their data to cloud servers for storage and maintenance. Users may resort to cloud computing for its ability on cloud storage, ranked search over encrypted cloud data [32]-[33]. However, data outsourcing also eliminates owners' ultimate control over their data. In [8]-[9], they indicate many security issues urgent to be solved in the current cloud storage service. For sensitive data, the data owner has to implement access control via encrypting data and controlling the user's ability of decryption, which is called cryptographic access control.

Shamir [10] proposed the concept of identity-based cryptography and Boneh et al. [11] constructed the first practical system identity-based cryptography. Sahai et al. [12] presented a fuzzy identity-based encryption scheme which is the earliest prototype of attribute-based encryption (ABE). Goyal et al. [7] further clarified the concept of ABE and proposed two complimentary forms of ABE: key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). According to Goyal's KP-ABE scheme, Bethencourt et al. [13] proposed a CP-ABE scheme that was closer to real access control systems. CP-ABE relates the user's secret key with a set of attribute and associates the ciphertext with an access structure tree. If the attribute set satisfies the access structure tree, then the user has the ability to decrypt the data. As CP-ABE schemes [13]-[20] are more natural to accomplish access control, we focus on the CP-ABE to realize our scheme.

In the paper [21]-[23], they discussed the usage of ABE to realize fine-grained access control for outsourced data. In these schemes, a trusted single authority is responsible for the management of attribute and the key distribution. Nevertheless, this setting easily leads to data leakage and the single authority becomes a bottleneck in the large scale cloud storage systems. There are many papers proposed some new encryption methods to solve problems about multi-authority ABE. Chase [14] proposed a solution that introduced a global identifier to tie users' keys together. This scheme applies to strict one strategy and relies on a central authority for key management and thus it would become the vulnerable point for security attacks and a bottleneck in large scale systems. To improve these shortcomings, Muller et al. [24] proposed a multi-authority ABE scheme with a centralized authority that could deal with any expressions in LSSS access policy. To remove the central authority, Chase et al. [25] used a distributed PRF (pseudo-random function) while Lin et al. [26] adopted threshold mechanism. In the latter scheme, the set of authorities is pre-determined and it requires the interaction among the authorities during the system setup. But in the scheme, the ability to resist collusion attacks is limited for a threshold parameter which is chosen at installation time. Lewko et al. [18] proposed a comprehensive scheme constructed in Composite order bilinear groups that incurs heavy computation cost. However, it is secure against any collusion attacks and requires no central authority.

Extensive research has be carried out on the revocation in ABE systems. Some of them are designed only for the single authority systems and do not support for the multi-authority systems [17]. Some of them are only for KP-ABE systems [28]. Some of them supported attribute revocation but incurred a heavy communication overhead for various reasons according to their own conceptual designs. In [1], Yang et al. designed an efficient attribute revocation method that can achieve both forward security and backward security while only incurred less communication cost and less computation cost of the revocation, where only those components associated with the revoked attribute in the secret keys and the ciphertext need to be updated. The scheme is designed for multi-authority cloud storage system. However, the global certificate authority in the system model is set to be trusted. However, in real storage systems, the authority can fail or be corrupted, which may leak out the data since the authority masters some important information.

The above observation motivates us to develop a new method to improve the security of the data access control for multi-authority cloud storage system while the scheme is firstly supposed to be effective and support effective decryption and revocation. We emphasize that the global certificate authority is hypothesized not to be fully trusted as in Yang et al. scheme [1]. In our scheme the global certificate authority may fail or be corrupted in the cloud storage system for defective server equipment or for the business profits.

1.2. Organization

The rest of this paper is organized as follows: We first give some necessary preliminary information in section II. Then it describes the system model and security model of our scheme in section III. Then we present a new secure data access control scheme for multi-authority cloud storage without a global fully trusted certificate authority in section IV. We give a comprehensive analysis of our scheme in security and performance in section V. Finally, we state our conclusion in section VI.

2. Preliminaries

2.1. Bilinear Pairings

Let G_1 and G_2 be two multiplicative cyclic groups of large prime order p . Let g be a generator G_1 of and let e be a bilinear map $e: G_1 \times G_1 \to G_2$. The bilinear map has the following properties:

1) Bilinearity: for all $u, v \in G_1$ and $a, b \in Z_p$, we have $e(u^a, v^b) = e(u^b, v^a) = e(u,v)^{ab}$;

2) Non-degeneracy: $e(g,g) \neq 1$;

3) Computable: efficient computability for any input pair.

Bilinear maps can be generally constructed from certain elliptic curves [29]. There is no need for readers to learn the technical details about how to build bilinear maps from certain elliptic curves. Understanding the properties of bilinear maps described above is sufficient enough for readers to access the design of our scheme.

2.2. Diffie-Hellman Problem

Definition 1 (Diffie-Hellman Problem). *Let* $G = < \mathrm{g} >$ *be a finite additive cyclic group, and* $g_1 := a \cdot g$ *as well as* $g_2 := b \cdot g$ *be two element of* G . *Given* g *as a generator, compute* $g_3 := (ab) \cdot g$ *with no information about* a *and* b .

For security analysis of our proposed scheme, we summarize some important security problems and assumptions for bilinear pairings on elliptic curves as follows.

Computational Diffie-Hellman Problem (CDHP) Given $P, aP, bP \in G_1$ for some $a, b \in Z_q^*$, the CDH problem is to compute $abP \in G_1$.

CDHP assumption No Probabilistic Polynomial Time (PPT) algorithm with a non-negligible advantage can solve the CDH problem.

Bilinear Diffie-Hellman Problem (BDHP) Given $P, aP, bP, cP \in G_1$ for some $a, b, c \in Z_q^*$, the BDH problem is to compute $e(P, P)^{abc} \in G_2$.

BDH assumption No PPT algorithm with a non-negligible advantage can solve the BDH problem.

According to the definition above, the security of our scheme is based on the computational intractability of Diffie-Hellman Problem. In one sense, our scheme proposed in the paper is provably secure.

2.3. Pedersen Commitment

The Pedersen Commitment scheme is first introduced in [30], It's an unconditionally hiding and computationally binding commitment scheme which is based on the intractability of the discrete logarithm problem. We give the adapted Pedersen Commitment scheme in a more general language as follows [31].

Setup

A trusted third party T chooses a finite cyclic group G of large prime order p so that the computational Diffie-Hellman problem is hard in G. Write the group operation in G as multiplication. The party T chooses an element $g \in G$ as a generator, and another element $h \in G$ such that it is hard to find the discrete logarithm of h with respect to g, i.e., an integer α such that $h = g^{\alpha}$. The party T may or may not know the number α. Then T publishes G, p, g and h as the system's parameters.

Commit

The domain of committed values is the finite field F_p of p elements, which can be represented as the set of integers $\mathrm{F}_p = \{0, 1, \cdots, p-1\}$. For a party U to commit a value $x \in \mathrm{F}_p$, it randomly chooses $r \in \mathrm{F}_p$, and computes the commitment $c = g^x h^r \in G$.

Open

The party U shows the values x and r to open a commitment c. And then the verifier checks whether $c = g^x h^r$.

3. System and Security Model

In this paper, we perform an analysis on the security of the certificate authority. There are two kinds of possible threats related to the CA. On the one hand, an adversary may try to corrupt it. On the other hand, the CA may inadvertently corrupt in the storage system due to hardware failures and human errors. To make things worse, the CA is economically motivated, which means it may be comprised in order to obtain confidential information in the storage system.

In order to decentralize the power of one certificate authority to reduce the influence when it was compromised, this paper adopt multiple CAs. As is shown in Fig.1, assume that there are three CAs in the storage system. This paper constructs an additional verifiable secret sharing scheme to define and manage the communication among them. If one CA breaks down, the other two CAs could still work out one secret as long as the threshold value is less than three. In addition to this, every CA in the VSS scheme publishes his commitment. Any CA else can verifies the correctness of the message sent by it and thus this scheme could identify the comprised CA.

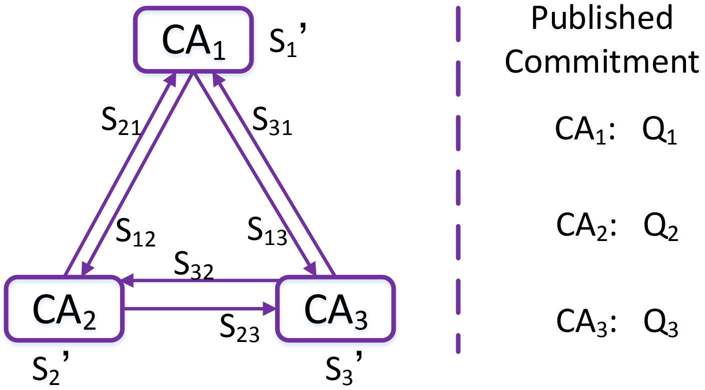


Figure 1. Model of Our VSS Scheme

3.1. System Model

In this section, we consider a secure cloud storage system for multiple authorities, as shown in Fig.2. The system model in this paper involves five different entities: the global certificate authorities (CAs), the attribute authorities (AAs), the cloud server (server), the data owners (owners) and the data consumers (users).

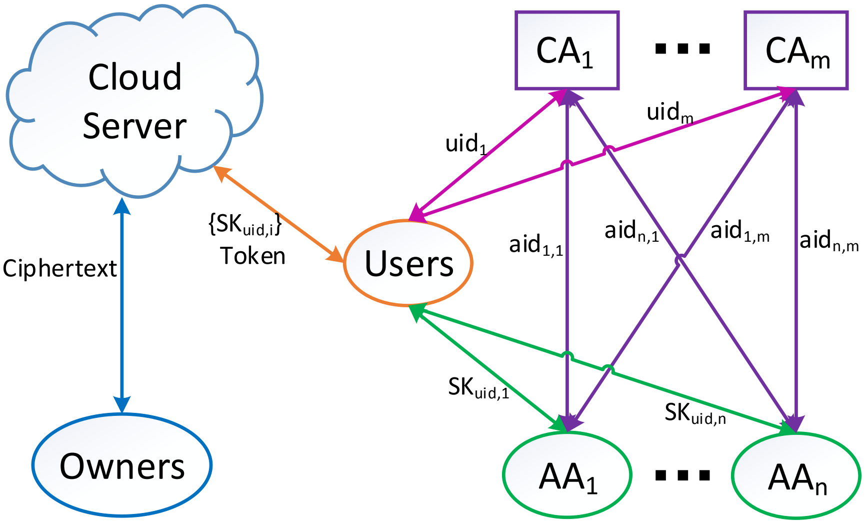


Figure 2. System Model of our Scheme

CA: Every CA is a global trusted certificate authority in the system. They accept the registration of all the users and AAs in this system. Besides, the CAs are responsible for the distribution of global secret key and global public key for each legal user in the system. However, they are not involved in any attribute management and the creation of secret keys that are associated with attributes.

AA: Every AA is an independent attribute authority. Every AA is responsible for issuing, revoking and updating user's attributes according to their own role or identity in its domain.

Every attribute is associated with one single AA. But each AA can manage an arbitrary number of attributes. It is responsible for generating a public attribute key for each attribute it manages and a secret key for each user associates with their attributes. Every AA has positive control over the structure and semantics of its attributes.

Cloud server: The cloud server stores the owners' data and provides data access service to users. In this paper, the cloud server generates the decryption token of a ciphertext for the user by using the user' secret keys issued by the AAs. In addition, the server also does the update operation of the ciphertext when an attribute revocation happens.

Data owner: The data owners in this system define the access policies of data. Under the policies, the data owners encrypt the data before outsourcing them in the cloud. Without relying on the server to obtain the data access control, all the legal users in the system can access the ciphertext. However, the access control happens inside the cryptography. Only when the user's attributes satisfy the access policy defined in the ciphertext, can the user decrypt the ciphertext.

User: A cloud user could be an enterprise or one single user. Each user in the system is assigned with some shares of an identity from the CAs, which can be gathered and calculated as its special global user identity.

To decrypt a ciphertext that can be accessed freely from the cloud server, each user may submit their secret keys issued by some AAs together with its global public key to the server. Then the system asks it to generate a decryption token for some ciphertexts. Upon receiving the decryption token, the user can decrypt the ciphertext using its global secret key. The server can generate the correct decryption token, only when the user's attributes satisfy the access policy defined in the ciphertext. To store the secret keys and the global user's public key on the server, subsequently, if no secret keys are updated for the further decryption token generation, the user need not submit any secret keys.

In order to meet the security requirements, our data access control scheme is a collection of algorithms combining a set of CP-ABE algorithms: CASetup, AASetup, UserRegister, KeyGen, Encrypt, TKGen, Decrypt and a set of attribute revocation algorithms: UKeyGen, KeyUpdate, CiphertextUpdate.

- CASetup $(\lambda) \rightarrow (MSK_i, SP_i, sk_{CA_i}, pk_{CA_i})$. This is the CA setup algorithm. It takes no input other than the implicit security parameter λ. Then it outputs the share of master key MSK_i, the share of the system parameter SP_i, as well as the pair of CA_i's secret key and public key (sk_{CA_i}, pk_{CA_i}).
- AASetup $(\{aid_i\}) \rightarrow (SK_{aid}, \{VK_{x_{aid}}, PK_{x_{aid}}\})$. This is the authority generation algorithm. It takes the authority id aid_i as input. Upon calculating the very id aid, it outputs the authority secret key SK_{aid}, the set of version keys and public attribute keys $\{VK_{x_{aid}}, PK_{x_{aid}}\}$ for all attributes x issued by the AA_{aid}.
- UserRegister $(\{sk_{CA_i}\}) \rightarrow (uid, GPK_{uid}, Sig_{sk_{CA}}(uid))$. This is the user registration algorithm. It takes the input as the set of the CA_i s' secret key $\{sk_{CA_i}\}$. Upon calculating the global

secret key sk_{CA} , for each legal user in the system, it outputs some shares of an id uid_i , and then it will computer the very global user id uid . In addition, it outputs the pair of global public key and secret key (GPK_{uid}, GSK_{uid}) and a user certification $Sig_{sk_{CA}}(uid)$.

- KeyGen $(S_{uid,aid}, SK_{aid}, \{PK_{x_{aid}}\}, \{SP_i\}, Sig_{sk_{CA}}(uid)) \rightarrow (PK_{aid}, SK_{uid,aid})$. This is the key generation algorithm. It takes five parts as inputs: a set of attributes $S_{uid,aid}$ that describes the secret key, the authority secret key SK_{aid} , the set of public attribute keys $\{PK_{x_{aid}}\}$, the shares of the system parameter $\{SP_i\}$ and the certification of the user $Sig_{sk_{CA}}(uid)$. It outputs the public key PK_{aid} and a secret key $SK_{uid,aid}$ for the user with uid .
- Encrypt $(\{PK_k\}_{k\in I_A}, \{PK_{x_k}\}_{k\in I_A}, m, \mathrm{A}) \rightarrow CT$. This is the encryption algorithm. It takes a set of public keys $\{PK_k\}_{k\in I_A}$ from the involved authority set I_A , a set of public attribute keys $\{PK_{x_k}\}_{k\in I_A}$, a message m and an access structure A over all the selected attributes from the involved AA s as inputs. According to the access structure, the algorithm encrypts m and outputs a ciphertext CT . This paper assumes that the ciphertext implicitly contains the access structure A .
- TKGen $(CT, GPK_{uid}, \{SK_{uid,k}\}_{k\in I_A}) \rightarrow TK$. This is the decryption token generation algorithm. It takes as input the ciphertext CT which contains an access structure A , user's global public key GPK_{uid} and a set of user's secret keys $\{SK_{uid,k}\}_{k\in I_A}$. If the set of attributes S satisfies the access structure A , the algorithm can successfully compute the decryption token TK of the ciphertext.
- Decrypt $(CT, TK, GSK_{uid}) \rightarrow m$. This is the decryption algorithm. It takes as inputs the ciphertext CT , the decryption token TK and the user's global secret key GSK_{uid} . It outputs the message m .
- UKeyGen $(SK_{aid}, \{u_j\}_{j\in S_U}, VK_{\tilde{x}_{aid}}) \rightarrow (UUK_{j,\tilde{x}_{aid}}, CUK_{\tilde{x}_{aid}})$. This is the update key generation algorithm. It takes as inputs the authority secret key SK_{aid} , a set of user's secret $\{u_j\}$ and the previous version key of the revoked attribute $VK_{\tilde{x}_{aid}}$. It outputs the User Update Key $UUK_{j,\tilde{x}_{aid}}$ $(j \in S_U, j \neq \mu, \tilde{x}_k \in S_{j,aid})$ and the Ciphertext Update Key $CUK_{\tilde{x}_k}$.
- KeyUpdate $(SK_{j,aid}, UUK_{j,\tilde{x}_{aid}}) \rightarrow SK'$. This is the user's secret key update algorithm. It takes as inputs the non-revoked user's current secret key $SK_{j,aid}$ and the user update key $UUK_{j,\tilde{x}_{aid}}$. It outputs a new secret key SK' to this non-revoked user.
- CiphertextUpdate $(CT, CUK_{\tilde{x}_{aid}}) \rightarrow CT'$. This is the ciphertext update algorithm. It takes as inputs the current ciphertext CT and the ciphertext update key $CUK_{\tilde{x}_{aid}}$. It outputs a new ciphertext CT' .

The former seven algorithms constitute a traditional CP-ABE scheme in our model and the latter three algorithms actually form an attribute revocation transaction. Based on DAC-MACS, the scheme proposed in this paper modifies the setting of one global trusted certificate authority in the system in case of outages and security breaches of CAs.

3.2. Security Model

We consider the case that the server may send the owners' data to the users who do not have access permission in cloud storage systems. We assume that the server will execute correctly the task assigned by the attribute authority but the server is also curious about the content of the encrypted data. The users who are dishonest may collude to obtain unauthorized access to data. The AA can be corrupted or compromised by the attackers. The CA may come across outage and security breaches in the cloud storage systems.

This section describes the security model for multi-authority CP-ABE systems by the following game between a challenger and an adversary. Similar to the identity-based encryption schemes [10]–[11], the security model allows the adversary to query for any secret keys that cannot be used to decrypt the challenge ciphertext. We assume that the adversaries can corrupt authorities only statically similar to [18], but key queries are made adaptively. Let S_A denote the set of all the authorities. Then the security game is defined as follows.

Setup. Each CA runs the CASetup and each AA runs the AASetup algorithm. The adversary specifies a set $S'_A \subset S_A$ of corrupted authorities. The challenger generates the pairs of public key and the secret key by running the key generation algorithm. For uncorrupted authorities in $S_A - S'_A$, the challenger sends only the public keys to the adversary. For corrupted authorities in S'_A, the challenger sends both the public keys and secret keys to the adversary.

Secret Key Query Phase 1. The adversary makes secret key queries by submitting pairs $(Sig_{sk_{CA}}(uid), S_{aid,1}),...,(Sig_{sk_{CA}}(uid), S_{aid,q_1})$ to the challenger, with $S_{aid,i}$ a set of attributes belonging to an uncorrupted AA_{aid} , and $Sig_{sk_{CA}}(uid)$ is a user certificate. The challenger gives the corresponding secret keys $\{SK_{uid,aid,i}\}_{i\in[1,q_1]}$ to the adversary.

Challenge. The attacker specifies two equal length messages m_0 and m_1. In addition, the adversary gives a challenge access structure (M^*, ρ^*) which must satisfy the following constraints. We let V denote the subset of rows of M^* labeled by attributes controlled by corrupted AAs. For each uid , we let V_{uid} denote the subset of rows of M^* labeled by attributes that the adversary has queried. For each uid , we require that the subspace spanned by $V \cup V_{uid}$ must not include $(1, 0,..., 0)$. In other words, the adversary cannot ask for a set of keys that allow decryption, in combination with any keys that can obtained from corrupted AAs. The challenger then flips a random coin b , and encrypts m_0 under the access structure (M^*, ρ^*) . Then, the ciphertext CT^* is given to the adversary.

Secret Key Query Phase 2. The adversary may query more secret keys, as long as they do not violate the constraints on the challenge access structure (M^*, ρ^*) .

Guess. The attacker submits a guess b' of b . If $b' = b$, the attacker wins this game. The advantage of an adversary A in this game is defined as $\Pr[b' = b] - \frac{1}{2}$.

Definition 3. *A multi-authority CP-ABE scheme is secure against static corruption of authorities if all polynomial time adversaries have at most a negligible advantage in the above security game.*

4. Attributed-based Access Control with Multiple Certificate Authorities

In this section, we first give a description of a verifiable secret sharing scheme based on bilinear-pairs without distribution center, which is used to improve the security of DAC-MACS. Then, we propose the detailed construction of our access control scheme.

4.1. A verifiable secret sharing scheme based on bilinear-pairs

We assume that there is a secret to be shared among n global certificate authorities $CA_i\,(i=1,2,...,n)$. Only when t or more than t CAs combine their shares together to recover the secret. Any combination of less than t CAs can hardly obtain any information about the shared secret. Generally speaking, in order to realize the distribution of the secret, there must be an honest and credible secret distribution center in the system. However, it is quite hard to find such an honest center, which is a bottleneck of secret sharing scheme. In this paper, we propose a verifiable secret sharing scheme based on bilinear-pairs without any honest center.

Let S , S_i and Q_j denote the secret to be shared, the share of secret of $CA_i\,(i=1,2,...,n)$, and the public information. Let $f(x)$ be the function selected by the honest center.

Our verifiable secret sharing scheme consists of four phases: System Initialization Phase, Secret Distribution Phase, Shares Calculation Phase and Secret Reconstruction Phase.

System Initialization Phase: In the system initialization phase, an elliptic curve of the additive group and a multiplicative group G_1 and G_2 are chosen with the same prime order p and there exits bilinear map among the two group, which can be effectively calculated. Let the secret to be shared be $S \in G_1$.

Secret Distribution Phase: In this section, the member $CA_i\,(i=1,2,...,n)$ randomly selects $\overline{S}_i \in_R G_1$ and a polynomial $f_i(x)=\sum_{j=0}^{t-1} a_{ij}x^j \in_R G_1[x]$, so as to satisfy $f_i(0)=a_{i0}=\overline{S}_i$. Meanwhile, CA_i assures the confidentiality of the polynomial $f_i(x)$ and publish the public information $Q_{ij}=e(P,a_{ij}),0 \le j < t$. Afterwards, CA_i secretly sends $S_{ik}=f_i(k)$ to $CA_k\,(0 \le k \le n)$, and the correctness of $f_i(k)$ can be verified by the receivers.

Shares Calculation Phase: In this section, when all of the members successfully send their sub keys, every participant computes its secret sharing share S_i, and

$$S_i=\sum_{j=1}^{n} F_j(i) \tag{1}$$

Secret Reconstruction Phase: In this section, when and only when t members $CA_i\,(i=1,2,...,t, i \in B, |B|=t)$ have successfully produced their respective sub keys S_i, the secret S can be calculated with Lagrange interpolating polynomial:

$$S=\sum_{i \in B}[F_{Bi}(i) \cdot S_i] \tag{2}$$

In the expression, $F_{Bi}(i)$ is the Lagrange interpolation coefficient and

$$F_{Bi}(x)=\prod_{j \in B \backslash \{i\}} \frac{x-x_j}{x_i-x_j}\,. \tag{3}$$

Record as $S \cong \sum_{i=1}^{n} \overline{S}_i$, $Q \cong \prod_{k=1}^{n} Q_{ki}$ and $f(x) \cong \sum_{i=1}^{n} f_i(x)$. Afterwards, the correctness of the secret S can be verified with the published public information C_0 , where $C_0 = e(P, S)$.

In the secret sharing scheme without any honest center, we can realize its verifiability via bilinearity of bilinear pairings, with no need for implementation of any complex interactions. Without enough numbers of members that share the very secret collaborate with others, the process of sharing a secret could not be set up, and the sharing of a secret does not be controlled by any single member. As a consequence, this scheme work with high communication efficiency.

4.2. Construction of the Proposed Access Control Scheme

We construct a secure access control system for multi-authority cloud storage without a global fully trusted certificate authority based on an adapted CP-ABE scheme in [1].

Let S_{CA} , S_A and S_U denote the set of global trusted certificate authorities, the set of attribute authorities and the set of users in the system respectively. Let G and G_T be the multiplicative groups with the same prime order p and $e: G \times G \to G_T$ be the bilinear map. Let g be the generator of G . Let $\mathrm{H}:\{0,1\}^* \to G$ be a hash function such that the security is in the random oracle.

Our access control scheme consists of five phases: System Initialization, Key Generation, Encryption, Decryption and Attribute Revocation.

4.2.1. System initialization phase: The system initialization phase generates system global parameters and authority secret and public key pairs. It consists of two algorithms: CA setup and AA setup.

CA Setup: All the CAs run the CA setup algorithm CASetup. First, it will take a security parameter as input. Then the CAs chooses a random number $a_i \in \square_p$ respectively as the master key MSK_i of the system and then compute the global master key MSK from the verifiable secret sharing scheme based on bilinear-pairs mentioned above, as well as the system parameter $SP = g^a$. Afterwards, the CAs apiece generate a pair of secret key and public key (sk_{CA_i}, pk_{CA_i}) , which will then be computed into a global pair of keys (sk_{CA}, pk_{CA}) .

The CAs accept both User Registration and AA Registration. They authenticates the users and each AA firstly. If a user is legal in the system, CA_i assigns a unique user id uid_i which will be then calculated into a global id uid , and randomly chooses two figures $u_{uid_i}, z_{uid_i} \in Z_p$ with which there could be two certain numbers u_{uid}, z_{uid} from the proposed verifiable secret sharing scheme above. After that, CA_i generates a global public key $GPK_{uid} = g^{uid}$, a global secret key $GSK_{uid} = z_{uid}$ as well as a certificate $Sig_{sk_{CA}}(uid, u_{uid}, g^{\frac{1}{z_{uid}}})$ by using the global secret key sk_{CA} for CAs to the user. If an AA is a legal authority in the system, CA_i assigns a unique user id aid_i which will be then calculated into a global id aid . Together with the global id, the public key pk_{CA} and the system parameter SP are sent to this AA.

AA Setup: All the AAs in S_A run the AA setup algorithm AASetup. Let S_{A_k} be the set of all attributes managed by AA_k, for each attribute $x_k \in S_{A_k}$, AA_k selects three random numbers $\alpha_k, \beta_k, \gamma_k \in \square_p$ as its secret key $SK_k = (\alpha_k, \beta_k, \gamma_k)$. By implicitly choosing an attribute version number v_{x_k}, AA_k generates its public attribute key as

$$PK_{x_k} = (g^{v_{x_k}} H(x_k))^{\gamma_k} \tag{4}$$

All the public attribute keys are published on the board of AA_k.

4.2.2. Key Generation Phase: Each AA runs the key generation algorithm KeyGen. The AA_k generates its authority public key as

$$PK_k = (e(g,g)^{\alpha_k}, g^{\frac{1}{\beta_k}}, g^{\frac{\gamma_k}{\beta_k}}) \tag{5}$$

Each owner can construct the full public key as

$$PK = (g, g^a, \{PK_k\}_{k \in S_A}, \{PK_{x_k}\}_{x_k \in S_{A_k}}^{k \in S_A}) \tag{6}$$

Meanwhile, for each user U_j in S_U, every $AA_k (k \in S_A)$ firstly decrypts the certificate $Sig_{sk_{CA}}(uid_j, u_j, g^{\frac{1}{z_j}})$ and then authenticates the legal user. If the user is legal, the AA_k assigns a set of attributes $S_{j,k}$ to the user in accordance with its role or identity in its administration domain. Afterwards, for $j \in S_U$ and $k \in S_A$, the AA_k generates a secret key $SK_{j,k}$ for the user as

$$\begin{aligned} SK_{j,k} = (K_{j,k} = g^{\frac{\alpha_k}{z_j}} \cdot g^{au_j}, L_{j,k} = g^{\frac{\beta_k}{z_j}}, \\ \forall x_k \in S_{j,k} : K_{j,x_k} = g^{\frac{\beta_k \gamma_k}{z_j}} \cdot (g^{v_{x_k}} \cdot H(x_k)^{\gamma_k \beta_k u_j})). \end{aligned} \tag{7}$$

4.2.3. Data Encryption Phase: Similarly with the data encryption phase of DAC-MACS, the data owner firstly encrypts the corresponding data component with a content key via using symmetric encryption methods. To implement the encryption algorithm Encrypt to encrypt the content key above, it takes the public key PK, the content keys k and an access structure (M, ρ) over all the selected attributes from the involved set of authorities I_A as inputs. Let M be a $\ell \times n$ matrix, where ℓ denotes the total number of all the attributes. The function ρ associates rows of M to the attributes.

To give a detailed description about the encryption algorithm, it first chooses a random encryption exponent $s \in \square_p$ and a random vector $\vec{v} = (s, y_2, ..., y_n) \in \square_p^n$, where $y_2, ..., y_n$ are used to share the encryption exponent s. For $i = 1$ to ℓ, it computes $\lambda_i = \vec{v} \cdot M_i$, where M_i is the vector corresponding to the i-th row of M. Then, it randomly chooses $r_1, r_2, ..., r_\ell \in \square_p$ and computes the ciphertext as

$$\begin{aligned} CT = (C = K \cdot (\prod_{K \in I_A} e(g,g)^{\alpha_k})^s, C' = g^s, \\ \forall i = 1 to l : C_i = g^{a\lambda_i} \cdot ((g^{v_{\rho(i)}} H(\rho(i)))^{\gamma_k})^{-r_i}, \\ D_{1,i} = g^{\frac{r_i}{\beta_k}}, D_{2,i} = g^{-\frac{\gamma_k}{\beta_k} r_i}, \rho(i) \in S_{A_k}) \end{aligned} \tag{8}$$

4.2.4. Data Decryption Phase: This phase consists of two steps: Server Token Generation and User Data Decryption.

1) Server Token Generation: As the background postulates, let I be $\{I_{A_k}\}_{k\in I_A}$, where $I_{A_k} \subset \{1,...,l\}$ is defined as $I_{A_k} = \{i : \rho(i) \in S_{A_k}\}$. Let $N_A = |I_A|$ be the number of AAs involved in the ciphertext. It chooses a set of constants $\{w_i \in \square_p\}_{i\in I}$ and reconstructs the encryption exponent as $s = \sum_{i\in I} w_i\lambda_i$ if $\{\lambda_i\}$ are valid shares of the secret s according to M.

Firstly the user U_j sends its secret keys $\{SK_{j,k}\}_{k\in S_A}$ to the server and then commands the server to compute a decryption token TK for the ciphertext CT as

$$TK = \prod_{k\in I_A} \frac{e(C', K_{j,k})}{\prod_{i\in I_{A_k}} (e(C_i, GPK_{U_j}) \cdot e(D_{1,i}, K_{j,\rho(i)}) \cdot e(D_{2,i}, L_{j,k}))^{w_i N_A}}$$

$$= \frac{e(g,g)^{au_j s N_A} \cdot \prod_{k\in I_A} e(g,g)^{\frac{\alpha_k}{z_j}s}}{e(g,g)^{u_j a N_A} \sum_{i\in I} \lambda_i w_i} \qquad (9)$$

$$= \prod_{k\in I_A} e(g,g)^{\frac{\alpha_k}{z_j}s}$$

By running the token generation algorithm TKGen, the server successfully calculates the decryption token when and only when the user possesses attributes that satisfy the access structure defined in the ciphertext CT in the prior phase. In the case of success, the server sends the decryption token TK to the user U_j.

2) User Data Decryption: Once there is a proper decryption token TK available, the user U_j can encrypt the ciphertext and access the content key with its global secret key $GSK_{U_j} = z_j$ as

$$k = \frac{C}{TK^{z_j}}. \qquad (10)$$

In this case, the user uses the content key to further decrypt the encrypted data component.

Obviously, the process of this phase and the prior one is working in the opposite direction. As we can see, the data owner only does the minimum computations since most of the pairing computations are moved to the cloud server in our scheme.

4.2.5. Attribute Revocation Phase: Considering an attribute $\tilde{x}_k$ of a user U_μ is revoked from AA_k, there are three phases included in the process of the attribute revocation: Update Key Generation, Key Update and Ciphertext Update.

1) Update Key Generation: By running the update key generation algorithm UKeyGen, AA_k generates a new attribute version key $v'_{\tilde{x}_k}$ to substitute the previous one, taking the authority secret key SK_k, the current attribute version key $v_{\tilde{x}_k}$ and the user's global public keys GPK_{U_j} as inputs. It first calculates the Attribute Update Key as $AUK_{\tilde{x}_k} = \gamma_k (v'_{\tilde{x}_k} - v_{\tilde{x}_k})$, and then further applying this $AUK_{\tilde{x}_k}$ to compute the User Update Key $UUK_{j,\tilde{x}_k} = g^{u_j \beta_k \cdot AUK_{\tilde{x}_k}}$ and the Ciphertext Update Key as $CUK_{\tilde{x}_k} = \frac{\beta_k}{\gamma_k} \cdot AUK_{\tilde{x}_k}$.

Besides, the AA_k updates the public attribute key of the revoked attribute $\tilde{x}_k$ as $PK'_{\tilde{x}_k} = PK_{\tilde{x}_k} \cdot g^{AUK_{\tilde{x}_k}}$ and broadcasts a message for all the owners that the public attribute key of the revoked attribute $\tilde{x}_k$ is updated. Then, all the owners can update their public key by getting the new public attribute key.

2) Key Update: The key update can prevent the revoked user from decrypting the new data which is encrypted by the new public keys (Forward Security). For each non-revoked user $U_j (j \in S_U)$ who has the attribute $\tilde{x}_k$, the AA_k sends the corresponding user update key $UUK_{j,\tilde{x}_k}$. Upon receiving it, the user $U_j (j \in S_U)$ updates its secret key via running the key update algorithm KeyUpdate as

$$\begin{aligned} SK'_{j,k} = (K'_{j,k} = K_{j,k}, L'_{j,k} = L_{j,k}, \\ K'_{j,\tilde{x}_k} = K_{j,\tilde{x}_k} \cdot UUK_{j,\tilde{x}_k}, \\ \forall x \in S_u, x \neq \tilde{x} : K'_{j,k} = K_{j,k}) \end{aligned} \tag{11}$$

3) Ciphertext Update: The ciphertext update can make sure that the newly joined user can still access the previous data which is published before it joins the system, when its attributes satisfy the access policy associated with the ciphertext (Backward Security). Upon receiving a ciphertext update key $CUK_{\tilde{x}_k}$ from the AA_k, the server runs the ciphertext update algorithm CiphertextUpdate to update those components of the ciphertext which are associated with the revoked attribute $\tilde{x}_k$ as

$$\begin{aligned} CT' = (C = k \cdot (\prod_{k \in I_A} e(g,g)^{\alpha_k})^s, C' = g^s, \\ \forall i = 1\ to\ l : C_i = g^{a\lambda_i} \cdot ((g^{v_{x_k}} H(x_k))^{\gamma_k})^{-r_i}, D_{1,i} = g^{\frac{r_i}{\beta_k}}, \\ D_{2,i} = g^{-\frac{\gamma_k}{\beta_k} r_i}, if\ \ \rho(i) \neq \tilde{x}_k, \\ C'_i = C_i \cdot D_{2,i}^{CUK_{\tilde{x}_k}}, D_{1,i} = g^{\frac{r_i}{\beta_k}}, \\ D_{2,i} = g^{-\frac{\gamma_k}{\beta_k} r_i}, if\ \ \rho(i) = \tilde{x}_k) \end{aligned} \tag{12}$$

Our scheme only requires to update the revoked attribute associated component of the ciphertext, while the other components which are not related to the revoked attributes are not changed. Thus, this can greatly improve the efficiency of attribute revocation.

5. Analysis of Our Scheme

In this section, we give the analysis of our proposed scheme to show its security. Then, we will also give the analysis of performance in our scheme.

5.1. Security Analysis

We will give the security analysis under the security model defined in Section **Error! Reference source not found.-Error! Reference source not found.**. Under the the security model, we conclude the security analysis into the following theorems:

Theorem 1. The verifiable secret sharing scheme based on bilinear-pairs in our scheme is secure.

Proof: In this scheme, the security of secret sharing scheme without a fully trusted central authority is equivalent to the security of the verifiable secret sharing scheme based on bilinear-pairs mentioned above. They are all based on the intractability of bilinear Diffie-Hellman hypothesis. Taking advantage of the bilinearity of bilinear pairings, Computational Diffie-Hellman Problem and Judgmental Diffie-Hellman Problem, it is easy to prove the security of the scheme which is equivalent with the intractability of bilinear Diffie-Hellman hypothesis.

The correctness of the message $S_{ik} = f_i(k)$ received by CA_k can be verified. Because there be $f_i(k) = a_{ik}$, CA_k can calculates $Q'_{ik} = e(P, a_{ik})$, and then compares the message Q_{ik} published by CA_i with the Q'_{ik} computed by itself. If the two messages are the same, then it shows that CA_i did not deceive CA_k . Otherwise, the message $f_i(k)$ received by CA_k may be wrong.

Theorem 2. When the decisional q-parallel BDHE assumption holds, no polynomial time adversary can selectively break our system with a challenge matrix of size $l^* \times n^*$, where $n^* \le q$.

Proof: Suppose we have an adversary A with non-negligible advantage $\varepsilon = adv_A$ in the selective security game against our construction and suppose it chooses a challenge matrix M^* with the dimension at most $q-1$ columns. In the security game, the adversary can query any secret keys that cannot be used for decryption in combination with any keys it can obtain from the corrupted AAs. With these constraints, the security game in multi-authority systems can be treated equally to the one in single authority systems. Similarly, we can build a simulator A' that deals the decisional q-parallel BDHE problem with non-negligible advantage. The detailed proof will be described in the Appendix of DAC-MACS.

Theorem 3. Our scheme is secure against the collusion attack.

Proof: In our scheme, each user in the system is finally assigned with a global unique identity uid , and all the secret keys issued to the same user from different AAs are associated with the unique identity of this user. Thus, it is impossible for two or more users to collude and decrypt the ciphertext. Moreover, due to the unique aid of each AA, all the attributes are distinguishable, even though some AAs may issue the same attribute. This can prevent the user from replacing the components of a secret key issued by an AA with those components from other secret keys issued by another AA.

Privacy-Preserving Guarantee Due to the decryption outsourcing, the server can get the users' secret keys. However, the server still cannot decrypt the ciphertext without the knowledge of the users' global secret keys. Moreover, the ciphertext update is done by using the proxy re-encryption method, thus the server does not need to decrypt the ciphertext.

5.2. Performance Analysis

In this paper, the sharing secret S is a point on additive cyclic group G_1 and the commitment Q_j is the value relative to bilinear-pairs. The verifiableness of the sharing secret can be implemented by the properties of bilinear-pairs without implementation of complex interaction proofs of participants or CAs, and numerous calculation. The communication efficiency was improved by the VSS scheme compared with other VSS schemes. The security of this VSS scheme is equivalent to the intractability of bilinear Diffie-Hellman assumption.

Compared with the scheme proposed in [1], the global certificate authorities only need to increase a round of computation cost and trivial storage cost to generate system parameters, the keys and unique identities for legal users as well as legal attribute authorities in our scheme. The other entities almost incur no computation overhead and storage overhead.

The storage overhead is one of the most significant issues of the access control scheme in cloud storage systems. Suppose $|p|$ is the element size in the G , G_T and $\square_p$. Let N_A and N_U denote the number of legal attribute authorities (AAs) and the number of legal users respectively. Let $n_{a,k}$ and $n_{a,k,uid}$ denote the total number of attributes managed by AA_k and the number of attributes assigned to the user with uid from AA_k respectively. Let n_c be the total number of ciphertexts stored on the cloud server, $n_{c,x}$ the number of ciphertexts contain the revoked attribute x , l the total number of attributes that appeared in the ciphertext. Suppose n_t is the threshold value of the verifiable secret sharing scheme proposed in this paper. We compare the storage overhead on each entity in our scheme with DAC-MACS scheme, showing in Table 1.

Table 1. Comparison of Storage Overhead

Entity	DAC-MACS	Our Scheme
CA/CAs	$3+3N_U+N_A$	$3+n_t(3N_U+N_A)$
AA_k	$(n_{a,k}+3)\|p\|$	$(n_{a,k}+3)\|p\|$
Owner	$(3N_A+1+\sum_{k=1}^{N_A} n_{a,k})\|p\|$	$(3N_A+1+\sum_{k=1}^{N_A} n_{a,k})\|p\|$
User	$(2N_A+1+\sum_{k=1}^{N_A} n_{a,k,uid})\|p\|$	$(2N_A+1+\sum_{k=1}^{N_A} n_{a,k,uid})\|p\|$
Server	$(3l+2)\|p\|$	$(3l+2)\|p\|$

Comparing to Yang's DAC-MACS scheme, we only increase two linear bilinear map computations and several finite cyclic group operations for each legal user or legal attribute authority added in the system. Therefore, the scheme in this paper can minimize the workload of data owners on the system operations. At the sacrifice of incurring linear computation overhead to certificate authorities, this scheme also improves the security for our scheme in the cloud.

6. Conclusion

This paper proposes an efficient and secure data access control scheme for multi-authority cloud storage systems. Taking advantage of Pedersen commitment as well as the properties of bilinear-pairs on elliptic curve and bilinear Diffie-Hellman problem, we propose a verifiable secret sharing scheme based on elliptic curve. It is also applied to implement a ciphertext-policy ABE to realize the decentralization of the certificate authority. Besides, it enables the verification whether there is any fraudulent conduct in certificate authorities. This proposed scheme guarantees that the cloud storage system properly launches and authenticates legal users as well as attribute authorities if and only if every CA successfully verifies the message sent by another CA using the latter CA's public commitment. This verifiable secret sharing scheme highly improves the security of the participative certificate authority at the sacrifice of a little cost. Through the security analysis and performance analysis, it is showed that this scheme is provably secure in the random oracle model and incurs little storage overhead, communication cost and computation cost. In the future, we will continue to explore a more efficient and secure CP-ABE scheme.

Acknowledgements

This work is supported by the NSFC (61232016, U1405254, 61173141, 61173142, 61173136, 61373133), 201301030, 2013DFG12860, BC2013012 and PAPD fund.

References

[1] K. Yang, X. Jia, K. Ren, and B. Zhang, "Dac-macs: Effective data access control for multi-authority cloud storage systems," in *INFOCOM, 2013 Proceedings IEEE*, **(2013)**, pp. 2895-2903.

[2] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, “Plutus: Scalable secure file sharing on untrusted storage,” in Proceedings of the 2nd USENIX Conference on File and Storage Technologies (FAST’03). USENIX, **(2003).**

[3] E.-J. Goh, H. Shacham, N. Modadugu, and D. Boneh, “Sirius: Securing remote untrusted storage,” in Proceedings of the Network and Distributed System Security Symposium (NDSS’03). The Internet Society, **(2003)**.

[4] D. Naor, M. Naor, and J. Lotspiech, “Revocation and tracing schemes for stateless receivers,” Electronic Colloquium on Computational Complexity (ECCC), no. 043, **(2002)**.

[5] E. Damiani, S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, “Key management for multi-user encrypted databases,” in Proceedings of the 2005 ACM Workshop On Storage Security And Survivability (StorageSS’05). ACM, **(2005)**, pp. 74–83.

[6] W. Wang, Z. Li, R. Owens, and B. K. Bhargava, “Secure and efficient access to outsourced data,” in Proceedings of the first ACM Cloud Computing Security Workshop (CCSW’09). ACM, **(2009)**, pp. 55–66.

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS’06). ACM, **(2006)**, pp. 89–98.

[8] L. Dignan. Cloud computing hasn’t gone fortune 500 yet, but it’s coming [EB/OL]. [**2011**-04]. http://blogs.zdnet.com/ BTL/? p=8199.

[9] C. Cachin, I, Keidar, A. Shraer. Trusting the cloud [J]. ACM SIGACT News, **(2009)**, 40(2) , pp. 81–86.

[10] A. Shamir, “Identity-based cryptosystems and signature schemes,” in Proceedings of the 4st Annual International Cryptology Conference: Advances in Cryptology - CRYPTO’84. Springer, **(1984)**, pp. 47–53.

[11] D. Boneh and M. K. Franklin, “Identity-based encryption from the weil pairing,” in Proceedings of the 21st Annual International Cryptology Conference: Advances in Cryptology - CRYPTO’01. Springer, **(2001)**, pp. 213–229.

[12] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology - EUROCRYPT’05. Springer, **(2005)**, pp. 457–473.

[13] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Security and Privacy, 2007. SP'07. IEEE Symposium on, **(2007)**, pp. 321-334.

[14] M. Chase, "Multi-authority attribute based encryption," in Theory of Cryptography, ed: Springer, **(2007)**, pp. 515-534.

[15] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in Proceedings of the 14th ACM conference on Computer and communications security, **(2007)**, pp. 456-465.

[16] Q. Tang and D. Ji, "Verifiable Attribute Based Encryption," IJ Network Security, vol. 10, pp. 114-120, **(2010)**.

[17] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the Decryption of ABE Ciphertexts," in USENIX Security Symposium, **(2011)**, pp. 3.

[18] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in Advances in Cryptology–EUROCRYPT 2011, ed: Springer, **(2011)**, pp. 568-588.

[19] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," Information Forensics and Security, IEEE Transactions on, vol. 8, **(2013)**, pp. 1343-1354.

[20] Z. Liu, Z. Cao, and D. S. Wong, "Blackbox traceable CP-ABE: how to catch people leaking their keys by selling decryption devices on ebay," in Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, **(2013)**, pp. 475-486.

[21] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 7, **(2011)**, pp. 1214–1221.

[22] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS'10). ACM, **(2010)**, pp. 261–270.

[23] S. Jahid, P. Mittal, and N. Borisov, "Easier: encryption-based access control in social networks with efficient revocation," in Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS'11). ACM, **(2011)**, pp. 411–415.

[24] S. M¨uller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," in Proceedings of the 11th International Conference on Information Security and Cryptology (ICISC'08). Springer, **(2008)**, pp. 20–36.

[25] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09). ACM, **(2009)**, pp. 121–130.

[26] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," Inf. Sci., vol. 180, no. 13, **(2010)**, pp. 2618–2632.

[27] J. Li, Q. Huang, X. Chen, S. S. M. Chow, D. S. Wong, and D. Xie, "Multi-authority ciphertext-policy attribute-based encryption with accountability," in Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS'11). ACM, **(2011)**, pp. 386–390.

[28] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," IEEE Transactions on Parallel and Distributed Systems, **(2012).**

[29] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'01) , **(2001)**, pp. 514-532.

[30] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in Advances in Cryptology—CRYPTO'91, **(1992)**, pp. 129-140.

[31] F. Paci, N. Shang, S. Kerr, K. Steuer Jr, J. Woo, and E. Bertino, "Privacy-preserving management of transactions' receipts for mobile environments," in Proceedings of the 8th Symposium on Identity and Trust on the Internet, **(2009)**, pp. 73-84.

[32] Z. Xia, X. Wang, X. Sun, and B. Wang. Steganalysis of least significant bit matching using multi-order differences. Security And Communication Networks, **(2014)**, 7(8): 1283-1291.

[33] K. Yang, X. Jia, K. Ren, R. Xie, and L. Huang, "Enabling efficient access control with dynamic policy updating for big data in the cloud," in INFOCOM, 2014 Proceedings IEEE, **(2014)**, pp. 2013-2021.

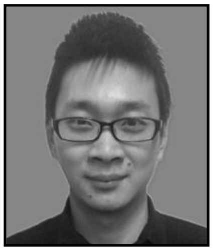
Authors

Xingming Sun, He received his BS in mathematics from Hunan Normal University, China, in 1984, MS in computing science from Dalian University of Science and Technology, China, in 1988, and PhD in computing science from Fudan University, China, in 2001. He is currently a professor in School of Computer & Software, Nanjing University of Information Science & Technology, China. His research interests include network and information security, digital watermarking.

Lin Xin, She received her BS in computer science and technology from Nanjing University of Information Science & Technology in 2013, China. She is currently pursuing his MS in computer science and technology at the College of Computer and Software, in Nanjing University of Information Science & Technology, China. Her research interest is cloud computing security.

Zhangjie Fu, He received his BS in education technology from Xinyang Normal University, China, in 2006, MS in education technology from the College of Physics and Microelectronics Science and PhD in computer science from the College of Computer, Hunan University, China, in 2008 and 2012. Currently, he works as an assistant professor in School of Computer and Software, Nanjing University of Information Science and Technology, China. His research interests include cloud computing, digital forensics, network and information security.

Liang-Ao Zhang, He received his BS in computer science and technology from Nanjing University of Information Science & Technology in 2013, China. He is currently pursuing his MS in computer science and technology at the College of Computer and Software, in Nanjing University of Information Science & Technology, China. His research interest is cloud computing security.



Jie Xi, He received his BS in computer science and technology from Nanjing University of Information Science & Technology in 2013, China. He is currently pursuing his MS in software engineering at the College of Computer and Software, in Nanjing University of Information Science & Technology, China. His research interest is cloud computing security.