International Journal of Security and Its Applications
Vol.9, No.12 (2015), pp.223-236
http://dx.doi.org/10.14257/ijsia.2015.9.12.22

Multiparty Access Control of Ciphertext Sharing in Cloud-Based Online Social Networks

Huang Qinlong[1, 2], Ma Zhaofeng[1,2], Yang Yixian[1,2] and Niu Xinxin[1,2]

[1]*Information Security Center, Beijing University of Posts and Telecommunications, Beijing, China*
[2]*National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing, China*
longsec@bupt.edu.cn, mzf@bupt.edu.cn, yxyang@bupt.edu.cn, xxniu@bupt.edu.cn

Abstract

Although current online social networks (OSNs) schemes propose to encrypt data before sharing, the enforcement of access policies over encrypted data has become a challenging task, and the OSNs currently do not provide any mechanism to allow users to update access policies. In this paper, we propose a ciphertext sharing scheme in cloud-based OSNs, which allows the users to outsource encrypted data to the OSNs service provider for sharing. In order to meet the authorization requirement, we present a multiparty access control model based on ciphertext-policy attribute-based proxy re-encryption, which enables the access control of encrypted data associated with multiple users. On the basis of ciphertext-policy attribute-based encryption, the owners can customize the access policy of their own data. Based on proxy re-encryption, the disseminators such as friends and group members can further customize the access policy of the owners' data upon existing access policy. Besides, we achieve immediate user revocation based on secret sharing without issuing new attribute secret keys to unrevoked users. The security and performance analysis show that our proposed scheme is secure, efficient and practical.

Keywords: *Multiparty Access Control, Attribute-based Encryption, Proxy Re-Encryption, Data Sharing, Online Social Networks*

1. Introduction

The online social networks (OSNs) provide each user with a virtual space where the user can post data like photos and videos, and share with his friends and group members [1]. The popularity of OSNs is obtained from the convenience as well as efficiency of data dissemination and sharing based on the relationships built among users [2].

Nevertheless, the main obstacle of sharing data in OSNs is data security [3]. The users entrust their data to the OSNs service provider and customize their own access policies to share data with others. However, the OSNs service provider is usually a semi-trusted server who will honestly follow the designated protocol, but might collect the users' data and share with others. Hence, in order for OSNs to serve as a trustworthy channel for disseminating data, it is crucial to have an effective strategy to protect data security.

On the other hand, fine-grained access control is another key challenge in OSNs. The OSNs currently provide simple access control mechanisms allowing users to regular their data in OSNs by restricting data sharing to a specific set of trusted users. The OSNs often use user relationship and group membership to distinguish between trusted and untrusted users. However, most access control schemes in OSNs are achieved by requiring the user

ISSN: 1738-9976 IJSIA

to manually maintain the access control list (ACL) [4]. The user can only choose to either publish his data to all users or grant authorities merely to his approved friends by manually updating and maintaining the ACL, which is inflexible, and coarse-grained. Thus flexible and fine-grained access control is an important requirement in OSNs.

The requirements of data security and fine-grained access control are motivating the proposal of secure data sharing schemes in OSNs. These schemes enable the owner to encrypt the sensitive data before releasing them, and then share encrypted data by specifying access policy instead of maintaining a list of approved friends. Attribute-based encryption (ABE) is a new cryptographic technique adopted to realize scalable, flexible, and fine-grained access control in such schemes [5]. This protects data from unauthorized users and OSNs service provider. However, current attribute based schemes in OSNs suffer from several limitations on multiparty access control [1]. For example, Alice's friends or group members may want to further customize their own access policies of Alice's data based on Alice's access policies. Hence, it is essential to develop a flexible and efficient access control mechanism in OSNs, which allows multiple users to have different authorization for a single data.

In this paper, we propose a ciphertext sharing scheme in cloud-based OSNs which enables the protection of shared data associated with multiple users. In a summary, the main contributions of this paper are presented as follows:

(1) We allow users to outsource encrypted data to the semi-trusted OSNs service provider in cloud computing for sharing based on ciphertext-policy ABE (CP-ABE), which protects the security and privacy of the data, and enables the OSNs service provider to manage users' data without being able to access the plaintext.

(2) We present a multiparty access control model based on ciphertext-policy attribute-based proxy re-encryption (CP-ABPRE), which enables the access control of encrypted data associated with multiple users. Based on proxy re-encryption (PRE), the disseminators such as the owners' friends and group members can further customize the access policy of the encrypted data upon existing access policy.

(3) We achieve immediate user revocation based on secret sharing without issuing new attribute secret keys to unrevoked users.

This paper is structured as follows. We first review related work in Section 2. Then we introduce the preliminaries in Section 3, and provide the security model and system model of our scheme in Section 4. We provide a construction in Section 5, and user revocation is introduced in Section 6. We analyze the security and performance of our scheme in Section 7. Finally, we conclude this paper in Section 8.

2. Related Work

The OSN is a multi-user service which provides social interactions. Data security is important when sharing data to make friends. A normal solution to protect the data security is to encrypt data before outsourcing data to OSNs service provider.

Tran et al. suggested a framework that allows users of cloud-based social networks to share their private data in a secure manner based on PRE [6]. In this framework, the users use the same public key to encrypt data but different private key to decrypt data. When a user accesses the data, the encrypted data will be re-encrypted according to the user's private key before sending to the user. Dien et al. also proposed a framework for secure sharing data on cloud-based social networks [7]. In their framework, the data is encrypted by a dynamic one-time symmetric key before sending to the cloud, and the user encrypts the symmetric key with his own secret key and then sends the ciphertext to the proxy server. The proxy server re-encrypts the ciphertext for those who are accessing the data. Although these schemes based on PRE technique can protect data security, they do not provide fine-grained access control of the encrypted data.

Many efforts have been made to present different access control schemes in OSNs. Wang *et al.* provided an access control scheme based on users' relationship and resource's content [8]. If a requestor wants to gain the right of access resources, he needs to meet the requirement of users' relationship, and have a certain correlation with the resource's content. On the other hand, in order to meet the multiparty access control requirement, Hu et al. proposed an approach to enable the protection of shared data associated with multiple users in OSNs [1]. This scheme formulates an access control model to capture the essence of multiparty authorization requirements, along with a multiparty policy specification scheme and a policy enforcement mechanism. Meanwhile, this scheme provides a voting mechanism to deal with authorization and privacy conflicts. Hu et al. improved this scheme by presenting a logic representation that allows the operators to leverage the features of existing logic solvers to perform various analysis tasks [9]. However, these access control schemes in OSNs need to manually update and maintain the ACL, which is complex and inflexible.

ABE has been adopted in recent efforts concerning access control in OSNs. The notion of ABE was first introduced by Sahai and Waters [10]. In the ABE scheme, ciphertexts and users' decryption keys are associated with a set of attributes or an access policy over attributes, and a user is able to decrypt a ciphertext only if there is a match between his attributes and the access policy. Persona is a state-of-the-art design that proposes the use of ABE to enable fine-grained access control in OSNs [11]. A user can create groups by assigning different attributes and keys to his social contacts, and then encrypt data such that only particular users having the desired set of attributes can decrypt it. Jahid et al. proposed an architecture called EASiER that supports fine-grained access policies and dynamic group membership by using ABE [12]. This scheme creates a proxy that participates in the decryption process. Meanwhile, the proxy cannot decrypt ciphertext or provide access to previously revoked users. Guo et al. proposed a privacy-preserving content dissemination scheme in mobile OSNs based on ABE [13]. Shuai et al. proposed a novel access control mechanism called Masque employing ABE, as a hierarchical solution for interactive sharing of encrypted data in OSNs [4]. This scheme allows the OSNs service provider to manage users at a high level without being able to access their sensitive data. At the same time, this scheme enables users to customize their own access policy specifically based on CP-ABE. These above schemes protect data security and provide fine-grained access control in the OSNs. However, they do not support multiparty access control of encrypted data.

User revocation is an essential mechanism in OSNs. Liang et al. proposed an efficient and secure user revocation scheme based on ABE in mobile social networks [14]. The proposed scheme enables a trusted authority to efficiently revoke a specific user's data decryption capability in each time slot, and also disables malicious users from decrypting any data packet. However, this scheme requires the trusted authority to periodically control the user's data decryption capability, which is inefficient. Therefore, immediate user revocation is proposed in many practical attribute-based systems. Ibraimi et al. introduces a mediator which maintains a revocation list so as to implement immediate attributes revocation [15]. Yu et al. introduced the semi-trusted agent based on PRE to achieve user revocation [16]. The proxy agent re-encrypts the ciphertext by the latest proxy key, and refreshes all the private keys held by the legal users. Shuai et al. improved the Masque and achieved immediate user revocation by updating corresponding secret parameters related to the revoked attributes [17]. However, these immediate user revocation schemes will cause the key update operation of large numbers of users in OSNs. Using our scheme, user revocation can be achieved immediately without issuing new attribute secret keys.

3. Preliminaries

3.1. Bilinear Map

Let G_1 and G_2 be two cyclic groups of some large prime order p, where G_1 and G_2 are multiplicative groups. A bilinear map $\hat{e}: G_1 \times G_1 \to G_2$, satisfies the following properties:

(1) Computability. There is a polynomial time algorithm to compute $\hat{e}(u,v) \in G_2$, for any $u, v \in G_1$.

(2) Bilinearity. $\hat{e}(u^a, v^b) = \hat{e}(u,v)^{ab}$ for all $u, v \in G_1$ and all $a, b \in Z_p$.

(3) Non-degeneracy. The map does not send all pairs in $G_1 \times G_1$ to the identity in G_2.

3.2. Secret Sharing

A *t*-out-of-*n* threshold secret sharing scheme can split a secret *s* into *n* distinct shares, and a user must retrieve at least *t* of n shares to reconstruct the secret.

The Shamir's secret sharing scheme [18] creates a random polynomial *P* of degree *t* such that $P(0) = s$. Given any *t*+1 shares $P(x_0), \ldots, P(x_t)$, it is possible to recover $P(0)$ using Lagrange interpolation:

$$P(0) = \sum_{i=0}^{t} \lambda_i P(x_i), \text{ where } \lambda_i = \prod_{j \neq i} \frac{x_j}{(x_j - x_i)}$$

The Benaloh and Leichter's secret sharing scheme [19] transforms an access policy into an access tree *T* and sets the value of the root node of *T* to be *s*. Then recursively for each non-leaf node does the following:

(1) If the symbol is $\vee$, set the values of each child node to be *s*.

(2) If the symbol is $\wedge$, for each child node, except the last one, assign a random value s_i where $1 \leq s_i \leq p-1$, and to the last child node assign $s_t = s - \sum_{i=1}^{t-1} s_i \bmod p$.

Thus the user whose attributes satisfy the access policy *T* can recover the secret *s*.

3.3. Ciphertext-policy Attribute-Based Proxy Re-encryption

The CP-ABPRE extends traditional PRE by allowing a semi-trusted proxy to transform a ciphertext under an access policy to the one with the same plaintext under another access policy [20]. The users identified by attributes could designate the proxy who can re-encrypt a ciphertext related with a certain access policy to another one with a different access policy. The CP-ABPRE scheme is a tuple of six algorithms [21]:

(1) System setup. The algorithm takes as input the security parameter *K*, outputs the system public key *PK* which is distributed to all users, and the master secret key *MK* which is kept secret.

(2) Key generation. The algorithm takes as input the *MK* and the attribute set *S* identifying the user, and it outputs the *SK* associated with the attribute set.

(3) Encryption. The algorithm takes as input the *PK*, a message *M* and an access policy *T* over a list of attributes which specifies the attributes the user needs to possess in order to obtain *M*. The algorithm outputs the ciphertext *CT* associated with the *T*.

(4) Re-encryption key generation. The algorithm takes as input the *PK*, an access policy T_A, an access policy T_B, and the *SK*. The algorithm outputs a

unidirectional re-encryption key $RK_{A->B}$ if SK satisfies T_A, or an error symbol if SK do not satisfy T_A.

(5) Re-encryption. The algorithm takes as input the ciphertext CT_A associated with the T_A and the $RK_{A->B}$, and outputs the ciphertext CT_B associated with the T_B.

(6) Decryption. The algorithm takes as input the ciphertext CT and the SK, and output the message M if SK satisfies T, or an error symbol if SK does not satisfy T.

4. Overview of Proposed Scheme

4.1. Security Model

In this paper, we assume attribute authority to be a trusted authority which is responsible for issuing attribute secret keys to users. We also assume OSNs service provider in cloud computing to be semi-trusted and curious-but-honest. That is, it will honestly execute the tasks, but would like to disclose users' personal data. In addition, the users may try to access data either within or outside the scope of their access privileges, so malicious users may collude with each other to access data beyond their privileges.

4.2. System Model

The system model of the proposed scheme consists of the following entities, as shown in Figure 1.

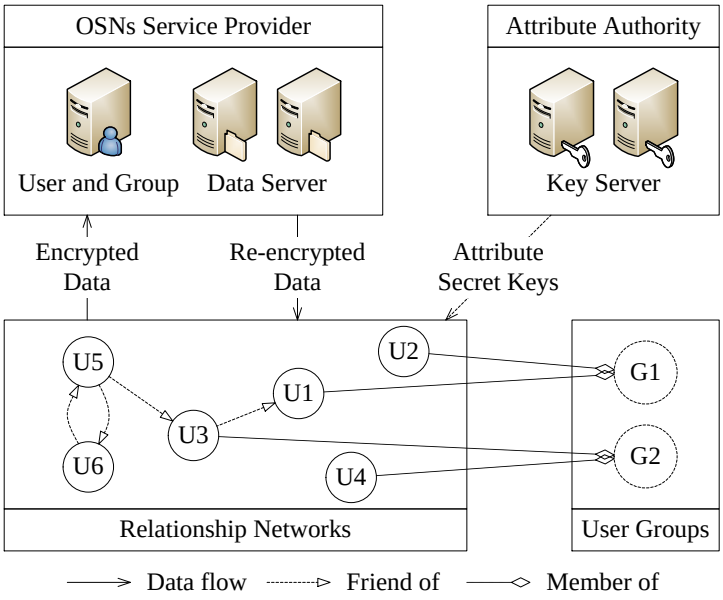

Figure 1. System Model of Proposed Scheme

Attribute authority. The attribute authority is a trusted third party, which sets up the system parameters (such as the PK and MK), and issues SK to the users corresponding to their attribute set.

OSNs service provider. The OSNs service provider is a semi-trusted server, which provides the OSN service and manages the users and groups. Besides, the OSNs service provider stores the encrypted data from the owners, and also re-encrypts the ciphertext in the data server with the disseminators' access policies when the disseminators share the data with other users.

User. The user can register with the OSNs service provider to generate his basic attributes, and can make friends with other users in OSNs. We define the roles of user as follows:

(1) Owner. The owner is the user who outsources data to his space in OSNs.

(2) Disseminator. The disseminator is the user who shares the other's data to his space in OSNs.

(3) Accessor. The accessor is the user who satisfies the access policy and accesses the owner's data in OSNs.

We define the proposed scheme by describing the following seven algorithms: *Setup*, *KeyGen*, *Encrypt*, *RKGen*, *ReEncrypt*, *Decrypt*. The OSNs service provider initiates the OSN service and the attribute authority specifies the system attributes in the *Setup* algorithm. The attribute authority generates the *SK* for the registered user by running the *KeyGen* algorithm. If an owner wants to publish his data, he will run *Encrypt* algorithm to encrypt the data with random *DK* using symmetric encryption algorithm and then encrypt the *DK* with customized access policy based on CP-ABE. If a disseminator wants to share an owner or other disseminator's data with his friends or group members by assigning new access policy, he will run the *RKGen* algorithm to generate the *RK* with the new access policy. Then the OSNs service provider will run *ReEncrypt* algorithm to re-encrypt the ciphertext with the *RK* for the accessor. If an accessor's attributes satisfy the access policy of encrypted data, he can run *Decrypt* algorithm to decrypt the *DK* and further decrypt the encrypted data.

5. Construction

In this section, we will provide a detailed construction for our proposed scheme.

1. *Setup*(*K*)

The algorithm selects a bilinear group G_1 of prime order *p* and generator *g*, and the bilinear map $\hat{e}:\mathrm{G}_1\times\mathrm{G}_1\to\mathrm{G}_2$. The attribute authority specifies the system attributes $\mathrm{A}=(a_1,a_2,...,a_n)$, and picks randomly $\alpha,\beta,\delta\in Z_p$. For each $a_i\in\mathrm{A}(1\le i\le n)$, the attribute authority chooses randomly $\chi_i\in Z_p(1\le i\le n)$. The algorithm also defines the function $H:\mathrm{G}_1\to\mathrm{G}_1$. The *PK* is published as:

$$PK=(g,\hat{e}(g,g)^{(\alpha+\beta)},g^{\delta},g^{1/\delta},\{g^{\chi_i}\}_{1\le i\le n},H).$$

The *MK* is kept secret by attribute authority and is constructed as:

$$MK=(\alpha,\beta,\delta,\{\chi_i\}_{1\le i\le n}).$$

The OSNs service provider generates random polynomial *P* of degree *t* (the maximum number of revoked users), and a random secret *s*.

2. *KeyGen*(*S*, *MK*)

The user can register with the OSNs service provider which generates a random user identity υ for the user. Moreover, the attribute authority assigns a set of attributes *S* to the user. The attribute authority takes as input the attribute set *S*, and chooses a random $\gamma\in Z_p$, and computes the *SK* which is constructed as:

$$SK=(D=g^{\alpha-\gamma},\{D_i=g^{(\gamma+\beta)/\chi_i}\}_{a_i\in S})$$

The OSNs service provider also generates the *UK* for the user with υ which is constructed as:

$$UK=(\upsilon,P(\upsilon))$$

Then the user keeps the *SK* and *UK* secret.

3. *Encrypt*(*T*, *M*, *PK*)

The user, for example Alice, can first encrypt her data with *DK* using symmetric encryption algorithm, such as Advanced Encryption Standard (AES):

$$E = SEnc(M, DK)$$

Then Alice can customize her own access policy T_A for her data by encrypting the *DK* based on CP-ABE. The algorithm chooses a random $\tau \in Z_p$ and assigns τ_i values to attributes in T_A. The τ_i values are shares of τ and are generated using Benaloh and Leichter secret sharing scheme. The resulted ciphertext *CT* is constructed as:

$$CT = (T_A, E, C = g^{\tau}, \tilde{C} = DK \cdot \hat{e}(g,g)^{(\alpha+\beta)\tau}, C_0 = g^{\delta\tau}, \{C_i = g^{\chi_i \tau_i}\}_{a_i \in T_A})$$

4. *RKGen*(*SK*, T_A, T_B, *PK*)

In order to customize the access policy when disseminating the data of an owner or a disseminator, the disseminator can generate the *RK* to re-encrypt the ciphertext associated with T_A to ciphertext associated with T_B, if SK_B matches the T_A.

We denote the attribute set of T_A as AS_A, and the attribute set of T_B as AS_B, where $AS_A \subseteq AS_B$. Let *AS* be the smallest attribute set which satisfies the T_A. The algorithm chooses randomly $\varphi, \varepsilon, \mu \in Z_p$, and assigns μ_i values to attributes in T_B. The μ_i values are shares of μ and are generated using Benaloh and Leichter secret sharing scheme. Then the algorithm computes the re-encryption key *RK* which consists of the following components:

$$R_1 = D \cdot g^{-\varepsilon} = g^{\alpha-\gamma-\varepsilon}$$

$$RK = (R_1, R_2 = g^{\varphi/\delta}, \{D_i = g^{(\gamma+\beta)/\chi_i}\}_{a_i \in AS}, T = g^{\mu}, \tilde{T} = g^{\varphi+\varepsilon} \cdot H(\hat{e}(g,g)^{(\alpha+\beta)\mu}),$$
$$T_0 = g^{\delta\mu}, \{C_i' = g^{\chi_i \mu_i}\}_{a_i \in T_B})$$

5. *ReEncrypt*(*RK*, *CT*)

When receiving the *RK* from the disseminator, the OSNs service provider will re-encrypt the ciphertext in the data server with *RK*.

The algorithm first computes the following:

$$I_1 = \prod_{a_i \in AS} \hat{e}(D_i, C_i) = \prod_{a_i \in AS} \hat{e}(g^{(\gamma+\beta)/\chi_i}, g^{\chi_i \tau_i}) = \hat{e}(g^{\gamma+\beta}, g^{\tau})$$

Then the algorithm computes the following:

$$I_2 = I_1 \cdot \hat{e}(C, R_1) = \hat{e}(g,g)^{(\gamma+\beta)\tau} \cdot \hat{e}(g^{\tau}, g^{\alpha-\gamma-\varepsilon}) = \hat{e}(g,g)^{(\alpha+\beta-\varepsilon)\tau}$$

Then the algorithm computes the following:

$$I_3 = \frac{\tilde{C}}{I_2} = \frac{DK \cdot \hat{e}(g,g)^{(\alpha+\beta)\tau}}{\hat{e}(g,g)^{(\alpha+\beta-\varepsilon)\tau}} = DK \cdot \hat{e}(g^{\tau}, g^{\varepsilon})$$

Then the algorithm computes the re-encrypted ciphertext *CT'*:

$$\tilde{C}' = I_3 \cdot \hat{e}(C_0, R_2) = DK \cdot \hat{e}(g^{\tau}, g^{\varepsilon}) \cdot \hat{e}(g^{\delta\tau}, g^{\varphi/\delta}) = DK \cdot \hat{e}(g,g)^{(\varphi+\varepsilon)\tau}$$

$$CT' = (T_B, E, C' = C, \tilde{C}', T, \tilde{T}, T_0, \{C_i'\}_{a_i \in T_B})$$

6. *Decrypt*(*SK*, *CT*)

In the OSNs, the accessor can conveniently view and share others' data such as photos and videos. Since the data is encrypted, only the accessor whose attributes satisfy the *T* can decrypt the ciphertext.

(1) If *CT* is an original ciphertext with access policy T_A and the accessor's attributes satisfy the T_A, the algorithm chooses the smallest attribute set *AS* which satisfies the T_A and computes:

$$Z_1 = \prod_{a_i \in AS} \hat{e}(D_i, C_i) = \prod_{a_i \in AS} \hat{e}(g^{(\gamma+\beta)/\chi_i}, g^{\chi_i \tau_i}) = \hat{e}(g^{\gamma+\beta}, g^{\tau})$$

Then the algorithm computes the following:

$$Z_2 = \hat{e}(D, C) \cdot Z_1 = \hat{e}(g^{\alpha-\gamma}, g^{\tau}) \cdot \hat{e}(g^{\gamma+\beta}, g^{\tau}) = \hat{e}(g, g)^{(\alpha+\beta)\tau}$$

Then the algorithm computes the following:

$$\frac{\tilde{C}}{Z_2} = \frac{DK \cdot \hat{e}(g, g)^{(\alpha+\beta)\tau}}{\hat{e}(g, g)^{(\alpha+\beta)\tau}} = DK$$

(2) If *CT* is a re-encrypted ciphertext with access policy T_B and the accessor's attributes satisfy the T_B, the algorithm chooses the smallest attribute set *AS* which satisfies the T_B and computes:

$$Z_1 = \prod_{a_i \in AS} \hat{e}(D_i, C_i') = \prod_{a_i \in AS} \hat{e}(g^{(\gamma+\beta)/\chi_i}, g^{\chi_i \mu_i}) = \hat{e}(g^{\gamma+\beta}, g^{\mu})$$

Then the algorithm computes the following:

$$Z_2 = \hat{e}(D, T) \cdot Z_1 = \hat{e}(g^{\alpha-\gamma}, g^{\mu}) \cdot \hat{e}(g^{\gamma+\beta}, g^{\mu}) = \hat{e}(g, g)^{(\alpha+\beta)\mu}$$

Then the algorithm computes the following:

$$Z_3 = \hat{e}(C', \frac{\tilde{T}}{H(Z_2)}) = \hat{e}(g^{\tau}, \frac{g^{\varphi+\varepsilon} \cdot H(\hat{e}(g, g)^{(\alpha+\beta)\mu})}{H(\hat{e}(g, g)^{(\alpha+\beta)\mu})}) = \hat{e}(g, g)^{(\varphi+\varepsilon)\tau}$$

Then the algorithm computes the following:

$$\frac{\tilde{C}'}{Z_3} = \frac{DK \cdot \hat{e}(g, g)^{(\varphi+\varepsilon)\tau}}{\hat{e}(g, g)^{(\varphi+\varepsilon)\tau}} = DK$$

Finally, the accessor can decrypt the ciphertext with the *DK*.

$$M = SDec(E, DK)$$

6. User Revocation

Our immediate user revocation method revokes certain users based on secret sharing without issuing new attribute secret keys to unrevoked users. The scheme includes the following two algorithms: *Update*, *Decrypt*. Whenever an owner or a disseminator wants to revoke some users, the OSNs service provider will update the ciphertext by running *Update* algorithm. The unrevoked users can run *Decrypt* to decrypt the ciphertext with their attribute secret keys, while the revoked users cannot.

1. *Update*(*UL*, *CT*)

If the owner or disseminator, for example Alice, wants to revoke some users, she will send the revoked user list *UL* to the OSNs service provider. The OSNs service provider then generates the *AK* with the revoked users' *UK*.

$$AK = \{\upsilon_i, P(\upsilon_i)\}_{1 \le i \le t}$$

Note: If the number of revoked users is less than *t*, the OSNs service provider generates randomly $(\upsilon_x, P(\upsilon_x))$ such that υ_x does not correspond to any user's identity. After generating the *AK*, the OSNs service provider will compute $\bar{C} = \hat{e}(g, g)^s$ and update the *CT* with the *AK*.

(1) If *CT* is an original ciphertext, the algorithm computes:

$$CT = (T_A, E, C, \tilde{C} \cdot \bar{C}, C_0, \{C_i\}_{a_i \in T_A}, AK)$$

(2) If *CT* is a re-encrypted ciphertext, the algorithm computes:

$$CT' = (T_B, E, C', \tilde{C}' \cdot \bar{C}, T, \tilde{T}, T_0, \{C_i'\}_{a_i \in T_B}, AK)$$

2. *Decrypt*(*SK*, *UK*, *CT*)

After the OSNs service provider updates the ciphertext, the user, for example Bob, can share Alice's data as follows. The algorithm first computes the λ with the *AK* and the user's *UK*.

$$\lambda_i = \frac{\upsilon_B}{(\upsilon_B - \upsilon_i)} \prod_{j \neq i} \frac{\upsilon_j}{(\upsilon_j - \upsilon_i)}, \forall i, j \in \{1,...,t\}$$

$$\lambda_B = \prod_{i=1}^{t} \frac{\upsilon_i}{(\upsilon_i - \upsilon_B)}$$

Then the algorithm recovers $P(0)$:

$$P(0) = \sum_{i=1}^{t} \lambda_i P(\upsilon_i) + \lambda_B P(\upsilon_B)$$

Thus if Bob is unrevoked, he can compute $P(0) = s$, and then recover the *DK*.

(1) If *CT* is an original ciphertext and Bob's attributes satisfy the access policy of *CT*, the algorithm computes:

$$DK = \frac{\tilde{C} \cdot \bar{C}}{Z_2 \cdot \hat{e}(g,g)^{P(0)}}$$

(2) If *CT* is a re-encrypted ciphertext and Bob's attributes satisfy the access policy of *CT*, the algorithm computes:

$$DK = \frac{\tilde{C}' \cdot \bar{C}}{Z_3 \cdot \hat{e}(g,g)^{P(0)}}$$

Finally Bob can decrypt the ciphertext with the *DK*.

7. Security and Performance Analysis

7.1. Security Analysis

(1) Data confidentiality. In our scheme, the owner encrypts his data before outsourcing. In the encryption phase, the owner encrypts his data with *DK* using symmetric encryption algorithm, and then encrypts the *DK* with access policy based on CP-ABE. The disseminator and accessor can decrypt the data if and only if they have a matching set of attributes. In the re-encryption phase, the OSNs service provider re-encrypts the ciphertext for the accessor with the *RK* generated by the disseminator, while the semi-trusted OSNs service provider cannot disclose the *DK*, which ensures the *DK* confidentiality.

(2) Unidirectionality. In the re-encryption phase, the OSNs service provider can re-encrypt ciphertext CT_A associated with T_A into ciphertext CT_B associated with T_B using the re-encryption key $RK_{A->B}$. But it cannot re-encrypt the CT_B into CT_A since $RK_{B->A}$ is impossible to compute.

(3) Re-encryption control. If OSNs service provider receives a re-encryption key $RK_{A->B}$ of ciphertext CT_A from the disseminator, it cannot forge a new re-encryption key $RK_{A->C}$ based on $RK_{A->B}$ to re-encrypt ciphertext CT_A to ciphertext CT_C. This is because that the key components μ_i in the $RK_{A->B}$ are assigned by the disseminator. Thus, the disseminator can determine whether the encrypted data can be re-encrypted, while the OSNs service provider can't forge any re-encryption key to re-encrypt the ciphertext arbitrarily.

(4) Collusion resistance. In the key generation phase, the attribute authority issues the attribute secret key *SK* of each user with a random γ. Thus even if two or more users collude sharing their attributes, the collusion attack will not take effect since the value $\hat{e}(g^{\gamma+\beta}, g^{\tau})$ cannot be recovered. Hence, the users cannot go through the access policy of

encrypted data by combining their *SK* to decrypt the data that they are not individually supposed to access.

(5) Master key security. In the re-encryption phase, the disseminator generates re-encryption key *RK* and sends it to OSNs service provider. However, the OSNs service provider cannot get key component *D* of disseminator from the *RK*, since it is protected by $g^{-\varepsilon}$. In the decryption phase, the accessor cannot recover $g^{-\varepsilon}$ though he can decrypt ciphertext successfully. Thus, the accessor cannot collude with OSNs service provider to obtain the disseminator's attribute secret key.

(6) Forward secrecy. In the user revocation phase, when a user comes to drop a set of attributes which satisfying the access policy in the ciphertext, the key component in the ciphertext is re-encrypted by the OSNs service provider with $\bar{C} = \hat{e}(g,g)^{s}$. Then the user cannot recover $P(0)$ and decrypt any ciphertext corresponding to the attributes after the revocation. In addition, even if the user has recovered $Z_2 = \hat{e}(g,g)^{(\alpha+\beta)\tau}$ before the revocation and stores it, he cannot generate the desired value $\tilde{C} \cdot \bar{C}$ to decrypt the subsequent ciphertext. Therefore, the forward secrecy of the shared data is guaranteed in our scheme.

7.2. Performance Analysis

We provide the performance analysis from both theoretical calculation and experimental evaluation. We evaluate the computation time of key generation, encryption, re-encryption, decryption and user revocation in our construction, and conduct the experiment on a virtual machine powered by Ubuntu 12.10 with 32-bit, Core i5 based on pairing-based cryptography library. We adopt the type A elliptic curve parameter where the group order is 160 bits. We use E_1 to denote the group exponentiation in $\mathbb{G}_1$, E_2 to denote the group exponentiation in $\mathbb{G}_2$, a to denote the number of attributes in S and n to denote the number of attributes in AS.

(1) Key generation. The attribute authority generates the attribute secret key for the user. The main computation overhead is a+1 exponentiations in $\mathbb{G}_1$.

(2) Encryption. The owner approximately needs $(n+2)E_1+E_2$ to encrypt a single data.

(3) Re-encryption. The disseminator first generates the re-encryption key with new access policy, which requires about n+2 exponentiations in $\mathbb{G}_1$ and one exponentiation in $\mathbb{G}_2$. Then the OSNs service provider re-encrypts the ciphertext with the re-encryption key, which needs n+2 pairing operations. Figure 2 shows the comparison of re-encryption time on the disseminator and OSNs service provider versus the number of attributes in access policy.

(4) Decryption. The decryption of original ciphertext needs about n+1 pairing operations, while the decryption of re-encrypted ciphertext requires one more pairing operation. Figure 3 shows the comparison of decryption time on the accessor versus the number of attributes in access policy.

(5) User revocation. The OSNs service provider needs one pairing operation to revoke certain users, while the accessor needs one more pairing operation to decrypt the updated ciphertext.

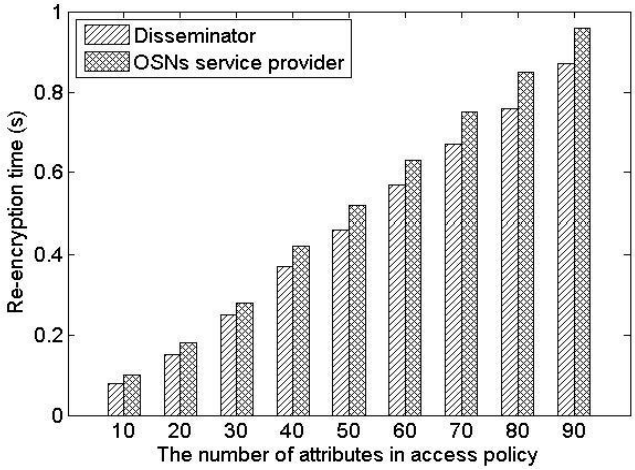

Figure 2. Re-encryption Time Evaluation of Proposed Scheme

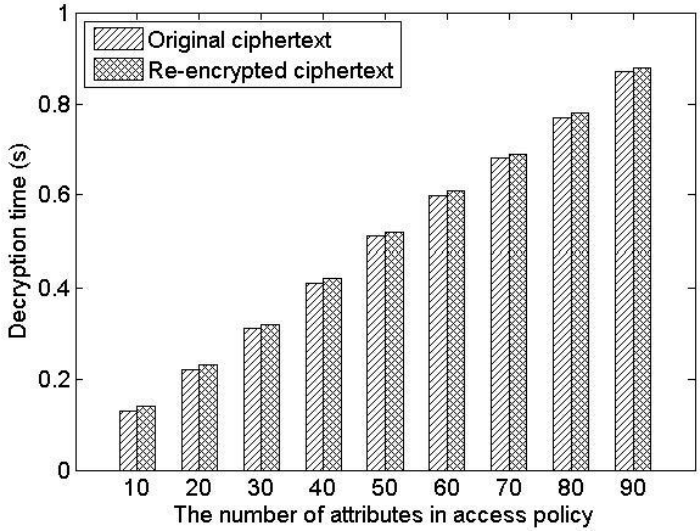

Figure 3. Decryption Time Evaluation of Proposed Scheme

7.3. Comparison

We compare our scheme with several access control schemes in OSNs, in terms of data confidentiality, access control, user revocation, multiple users. The comparison results are shown in Table 1. Compared with access control schemes based on ABE [11,12,13,17], our scheme encrypts the data with *DK* using symmetric encryption algorithm, and encrypts the *DK* based on CP-ABE, which is more efficient. Besides, our scheme supports the multiparty access control based on CP-ABPRE, and allows multiple controllers include the owner and the disseminator to customize access policies.

On the other hand, compared with current multiparty access control scheme [9], our scheme encrypts the data before sharing, which protects the data from the illegal users and semi-trusted OSNs service provider. Moreover, our scheme supports immediate user revocation without issuing new attribute secret keys to unrevoked users, which is more flexible and efficient in OSNs.

Table 1. Comparisons of Access Control Schemes in OSNs

Scheme	Data confidentiality	Access control	User revocation	Multiple users
Dien's scheme [7]	Proxy re-encryption	N/A	No	No
Guo's scheme [13]	Attribute-based encryption	Access policy	No	No
Baden's scheme [11]	Attribute-based encryption	Access policy	Yes, with key update	No
Shuai's scheme [17]	Attribute-based encryption	Access policy	Yes, with key update	No
Jahid's scheme [12]	Attribute-based encryption	Access policy	Yes, without key update	No
Hu's scheme [9]	N/A	ACL	N/A	Yes
Our scheme	Attribute-based encryption	Access policy	Yes, without key update	Yes

8. Conclusions

In this paper, we propose a ciphertext sharing scheme in cloud-based OSNs, which allows the users to outsource encrypted data to the OSNs service provider for sharing, and enables the OSNs service provider to manage users' data without being able to access the plaintext. We present a multiparty access control model based on CP-ABPRE, which enables the access control of encrypted data associated with multiple users. The owners can customize the access policy of their own data, while the disseminators can further customize the access policy of the owners' data. Besides, we also achieve immediate user revocation without issuing new attribute secret keys to unrevoked users.

We evaluated our scheme from aspects of security and performance, which leads to the conclusion that our scheme is secure, efficient and practical, because of not only the relatively low cost in computation but also its satisfaction with the need for multiparty access control of encrypted data.

Acknowledgements

This work has been supported by the National Natural Science Foundation of China under Grant No. 61272519, 61572080, the Youth Research and Innovation Program of Beijing University of Posts and Telecommunications under Grant No. 2015RC31.

References

[1] H. Hu and G. Ahn, "Multiparty authorization framework for data sharing in online social networks", Proceedings of 25th annual IFIP WG 11.3 conference on Data and applications security and privacy, **(2011),** pp. 29-43.
[2] N. P. Nguyen, G. Yan and M. T. Thai, "Analysis of misinformation containment in online social networks", Computer Networks, vol. 57, no. 10, **(2013)**, pp. 2133-2146.
[3] Y. Wu, Z. Wei and R.H. Deng, "Attribute-based access to scalable media in cloud-assisted content sharing networks", IEEE Transactions on Multimedia, vol. 15, no. 4, **(2013)**, pp. 778-788.
[4] H. Shuai and W. Zhu, "Masque: Access control for interactive sharing of encrypted data in social networks", Proceedings of 6th International Conference on Network and System Security, **(2012)**, pp. 503-515.
[5] J. Hur and D. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems", IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 7, **(2011)**, pp. 1214-1221.
[6] D. H. Tran, H. Nguyen, W. Zha and W. Ng, "Towards security in sharing data on cloud-based social networks", Proceedings of 8th International Conference on Information, Communications and Signal Processing, **(2011)**, pp. 1-5.
[7] N. M. Dien, J. Hwang and M. Yoo, "A new framework for secure sharing data on cloud-based social networks", Proceedings of 2012 International Conference on ICT Convergence, **(2012)**, pp. 333-335.

[8] F. Wang, H. Dong and Y. Liang, "An access control policy based on users' relationship and resource's content in online social networks", International Journal of Digital Content Technology and its Applications, vol. 6, no. 19, **(2012)**, pp. 322-329.
[9] H. Hu, G. Ahn and J. Jorgensen, "Multiparty access control for online social networks: model and mechanisms", IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 7, **(2013)**, pp. 1614-1627.
[10] A. Sahai and B. Waters, "Fuzzy identity based encryption", Proceedings of EUROCRYPT 2005, **(2005)**, pp. 457-473.
[11] B. Baden, A. Bender, N. Spring, B. Bhattacharjee and D. Starin, "Persona: An online social network with user-defined privacy", Proceedings of ACM SIGCOMM 2009 Conference on Data Communication, **(2009)**, pp. 135-146.
[12] S. Jahid, P. Mittal and N. Borisov, "EASiER: Encryption-based access control in social networks with efficient revocation", Proceedings of 6th International Symposium on Information, Computer and Communications Security, **(2011)**, pp. 411-415.
[13] L. Guo, C. Zhang, H. Yue and Y. Fang, "A privacy-preserving social-assisted mobile content dissemination scheme in DTNs", Proceedings of IEEE INFOCOM 2013, **(2013)**, pp. 2301-2309.
[14] X. Liang, X. Li, R. Lu, X. Lin and X. Shen, "An efficient and secure user revocation scheme in mobile social networks", Proceedings of 2011 IEEE Global Telecommunications Conference, **(2011)**, pp. 1-5.
[15] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application", Proceedings of 10th International Workshop on Information Security Applications, **(2009)**, pp. 309-323.
[16] S. Yu, C. Wang, K. Ren and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing", Proceedings of IEEE INFOCOM 2010, **(2010)**, pp. 1-9.
[17] H. Shuai, W. Zhu and X. Liu, "Publishing and sharing encrypted data with potential friends in online social networks", Security and Communication Networks, vol. 7, no. 2, **(2014)**, pp. 409-421.
[18] A. Shamir, "How to share a secret", Communications of the ACM, vol. 22, no. 11, **(1979)**, pp. 612-613.
[19] J. Benaloh and J. Leichter, "Generalized Secret Sharing and Monotone Functions", Proceedings of CRYPTO 88, **(1990)**, pp. 27-35.
[20] X. Liang, Z. Cao, H. Lin and J Shao, "Attribute based proxy re-encryption with delegating capabilities", Proceedings of 4th International Symposium on ACM Symposium on Information, Computer and Communications Security, **(2009)**, pp. 276-286.
[21] J. Zhao, D. Feng and Z. Zhang, "Attribute-based conditional proxy re-encryption with chosen-ciphertext security", Proceedings of 2010 IEEE Global Telecommunications Conference, **(2010)**, pp. 1-6.

Authors

Huang Qinlong, he received BS degree in information security from Yunnan University in 2009, PhD degree from Beijing University of Posts and Telecommunications (BUPT) in 2014. He is now a lecture in the School of Computer Science of BUPT. His research interest includes Cloud Computing Security and Digital Rights Management.

Ma Zhaofeng, he is an associate professor in the School of Computer Sicence, Beijing University of Posts and Telecommunications. He got the PhD degree from Xi'an Jiaotong University in 2004. His research interest includes Information Security and Network Security and Digital Rights Management

Yang Yixian, he received the BS degree in Applied Mathematics from Chengdu Institute of Telecommunication Engineering, China, in 1983, the MS degree and PhD degree from Beijing University of Posts and Telecommunications (BUPT), China, in 1986 and 1988, respectively. He is a professor of BUPT from 1992. He is also doctoral supervisor in school of computer science. His research interests are Information and Network Security, Cryptography, Chaos,

and Fuzzy Systems.

Niu Xinxin, he received the BS and MS degree from the Beijing University of Posts and Telecommunications (BUPT) in 1985 and 1988, and the PhD degree from the Department of Electronic Engineering of the Chinese University of Hong Kong. She is a professor and doctoral supervisor in School of Computer Science of BUPT. Her research areas include Information and Network Security, Information Hiding and Digital Watermark, Digital Content and Security.