

# Secure Data Group Sharing and Conditional Dissemination with Multi-Owner in Cloud Computing

Qinlong Huang, *Member, IEEE*, Yixian Yang, Wei Yue and Yue He

**Abstract**—With the rapid development of cloud services, huge volume of data is shared via cloud computing. Although cryptographic techniques have been utilized to provide data confidentiality in cloud computing, current mechanisms cannot enforce privacy concerns over ciphertext associated with multiple owners, which makes co-owners unable to appropriately control whether data disseminators can actually disseminate their data. In this paper, we propose a secure data group sharing and conditional dissemination scheme with multi-owner in cloud computing, in which data owner can share private data with a group of users via the cloud in a secure way, and data disseminator can disseminate the data to a new group of users if the attributes satisfy the access policies in the ciphertext. We further present a multiparty access control mechanism over the disseminated ciphertext, in which the data co-owners can append new access policies to the ciphertext due to their privacy preferences. Moreover, three policy aggregation strategies, including full permit, owner priority and majority permit, are provided to solve the privacy conflicts problem caused by different access policies. The security analysis and experimental results show our scheme is practical and efficient for secure data sharing with multi-owner in cloud computing.

**Index Terms**—Data sharing, cloud computing, conditional proxy re-encryption, attribute-based encryption, privacy conflict

## 1 INTRODUCTION

THE popularity of cloud computing is obtained from the benefits of rich storage resources and instant access [1]. It aggregates the resources of computing infrastructure, and then provides on-demand services over the Internet. Many famous companies are now providing public cloud services, such as Amazon, Google, Alibaba. These services allow individual users and enterprise users to upload data (e.g. photos, videos and documents) to cloud service provider (CSP), for the purpose of accessing the data at any time anywhere and sharing the data with others. In order to protect the privacy of users, most cloud services achieve access control by maintaining access control list (ACL). In this way, users can choose to either publish their data to anyone or grant access rights merely to their approved people. However, the security risks have raised concerns in people, due to the data is stored in plaintext form by the CSP. Once the data is posted to the CSP, it is out of the data owner's control [2]. Unfortunately, the CSP is usually a semi-trusted server which honestly follows the designated protocol, but might collect the users' data and even use them for benefits without users' consents. On the other hand, the data has tremendous usages by various data consumers to learn the behavior of users [3].

These security issues motivate the effective solutions to

protect data confidentiality. It is essential to adopt access control mechanisms to achieve secure data sharing in cloud computing [4]. Currently, cryptographic mechanisms such as attribute-based encryption (ABE) [5], identity-based broadcast encryption (IBBE) [6], and remote attestation [7] have been exploited to settle these security and privacy problems. ABE is one of the new cryptographic mechanisms used in cloud computing to reach secure and fine-grained data sharing [8]. It features a mechanism that enables an access control over encrypted data using access policies and ascribed attributes among decryption keys and ciphertexts. As long as the attribute set satisfies the access policy that the ciphertext can be decrypted. IBBE is another prevalent technique employed in cloud computing [9, 10], in which users could share their encrypted data with multiple receivers at one time and the public key of the receiver can be regarded as any valid strings, such as unique identity and email. In fact, IBBE can be seen as a special case of ABE for policies consisting of an OR gate. Compared to ABE in which the secret key and ciphertext are both correspond to a set of attributes, IBBE incurs low-cost key management and small constant policy sizes, which is more suitable for securely broadcasting data to specific receivers in cloud computing. Hence, by using identities, data owner can share data with a group of users in a secure and efficient manner, which motivates more users to share their private data via cloud.

Actually, these encryption techniques can prevent unauthorized entities (e.g. semi-trusted CSP and malicious users) from accessing the data, but it may not consider data dissemination in cloud computing. In the cloud col-

- Q. Huang, Y. Yang, W. Yue, and Y. He are with the School of Cyber-space Security, Beijing University of Posts and Telecommunications, Beijing, China, and also with National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing, China. E-mail: longsec@bupt.edu.cn, yx-yang@bupt.edu.cn, yuewei718@bupt.edu.cn, heyue@bupt.edu.cn.

laboration scenario such as Box [11] and OneDrive [12], the data disseminators (e.g. editor and collaborator) may share the documents with new users even those outside the organization. However, once the data is encrypted with the above techniques, data disseminators are not able to modify the ciphertext uploaded by data owners [13]. Proxy re-encryption (PRE) scheme [14] is employed to achieve secure data dissemination in cloud computing by delegating a re-encryption key associated with the new receivers to the CSP. However, the data disseminator can disseminate all of the data owner's data to others with this re-encryption key, which may not meet the practical requirement since the data owner may only permit the data disseminator to disseminate a particular document. A refined concept referred to as conditional PRE (CPRE) [15, 16] could address this issue, in which data owner can enforce re-encryption control over the initial ciphertexts and only the ciphertexts satisfying specific condition can be re-encrypted with corresponding re-encryption key. However, traditional CPRE schemes only support simple keyword conditions, so they cannot match complex situations in cloud computing well. In order to support expressive conditions rather than keywords, attribute-based CPRE is proposed [17], which deploys an access policy in the ciphertext. The re-encryption key is associated with a set of attributes, thus the proxy can re-encrypt the ciphertext only when the re-encryption key matches the access policy. In this way, data owner can customize fine-grained dissemination condition for the shared data. For example, data owner allows project managers in the organization to disseminate the progress report in OneDrive, while only permits executive directors in finance department to disseminate the project budget in OneDrive during a specific time period.

Besides the requirement of conditional data dissemination, multiparty access control problem for data sharing in cloud computing such as cloud collaboration and cloud-based social networks comes along [18, 19], which means the special authorization requirements from multiple associated users can be accommodated together to control the shared data. Consider an example where a co-authoring document or a co-photo in cloud computing with three users, Alice, Bob, and Carol. If Alice who is the data owner uploads this co-authoring document or co-photo to the CSP and tags both Bob and Carol as the co-owners. Alice can restrict this data to be disseminated to a certain group of users, while the co-owners Bob and Carol may have different privacy concerns about this data. It is a massive and serious privacy problem if applying the preference of only one party, which may cause such data to be shared with undesired receivers.

However, merging privacy preferences of data owner and multiple co-owners is not an easy task, due to privacy conflict is inevitable in multiparty authorization enforcement [20, 21]. Privacy conflict happens when the co-owners have opposite privacy policies, and it results in data being impossibly accessed with anyone [22]. To deal with this dilemma, multiparty access control mechanisms

(e.g. voting scheme) are further provided. However, all of them are based on plaintext data. In this paper, we propose an identity-based secure data group sharing and conditional dissemination scheme with multi-owner in cloud computing. To mitigate the problems mentioned above, we introduce a solution to achieve ciphertext group sharing among multiple users, and capture the core feature of multiparty authorization requirements. The contributions of our scheme are as follows:

(1) We achieve fine-grained conditional dissemination over the ciphertext in cloud computing with attribute-based CPRE. The ciphertext is firstly deployed with an initial access policy customized by data owner. Our proposed multiparty access control mechanism allows the data co-owners to append new access policies to the ciphertext due to their privacy preferences. Hence, the ciphertext can be re-encrypted by the data disseminator only if the attributes satisfy enough access policies.

(2) We provide three strategies including full permit, owner priority and majority permit to solve the privacy conflicts problem. Specially, in full permit strategy, data disseminator must satisfy all the access policies defined by data owner and co-owners. With the majority permit strategy, data owner can firstly choose a threshold value for data co-owners, and the ciphertext can be disseminated if and only if the sum of the access policies satisfied by data disseminator's attributes is greater than or equal to this fixed threshold.

(3) We prove the correctness of our scheme, and conduct experiments to evaluate the performance at each phase to indicate the effectiveness of our scheme.

This paper is structured as follows. We review related work in Section 2 and introduce the preliminaries in Section 3. We provide the system model and policy aggregation strategies in Section 4, and describe the proposed scheme in Section 5. We present the system analysis and experimental results in Section 6 and Section 7 respectively. Finally, we conclude this paper in Section 8.

## 2 RELATED WORK

A series of unaddressed security and privacy issues emerge as important research topics in cloud computing. To deal with these threats, appropriate encryption techniques should be utilized to guarantee data confidentiality. By utilizing the IBBE technique [23], Huang et al. [24], Patranabis et al. [25] and Liu et al. [9] proposed several private data sharing schemes in cloud computing. In these schemes, data owner outsources encrypted data to the CSP by defining a list of receivers, thus only the intended users in the list can get the decryption key and further decrypt the private data. ABE is another promising one-to-many cryptographic technique to realize data encryption and fine-grained access control in cloud computing [26, 27]. Specially, ciphertext-policy ABE (CP-ABE) is suited for access control in real world applications due to its expressiveness in describing the access policy of ciphertext [28]. Guo et al. [29] proposed a privacy-

preserving data dissemination scheme in mobile social networks based on CP-ABE. Teng et al. [30] proposed an efficient access control scheme with hierarchical CP-ABE to achieve privacy preservation in cloud storage systems. In the schemes of [31] and [32], ABE has been utilized to provide access control of medical documents when providing health services in cloud, so that health record can only be decrypted by authorized document requesters with corresponding attributes.

Secure data dissemination is another important security requirement for data storage in cloud computing. The identity-based PRE [33] is a basic encryption algorithm to reach secure data dissemination in cloud computing, with which the data disseminators could send their re-encryption keys to the semi-trusted proxy to transform data owner's ciphertext for new users [34]. Further, attribute-based PRE [17] has been employed in cloud computing by incorporating the ABE technique. The proxy can transform the ciphertext under an access policy into the one under another access policy with data disseminator's re-encryption key, and the users who satisfy the new access policy can access the plaintext. However, the above PRE schemes only allow data dissemination in an all-or-none manner. This issue is further addressed by CPRE scheme [35], in which the proxy can successfully re-encrypt the ciphertext only if the prescribed conditions are met. However, in earlier CPRE schemes [35, 36], the conditions are keywords only, which would limit the flexibility when enforcing complex delegations in cloud computing. Yang et al. [37] proposed an attribute-based CPRE scheme by deploying an access policy in a ciphertext generated by public-key encryption. The re-encryption key is generated by the secret key associated with a set of attributes, which allows the proxy to re-encrypt the ciphertext only when these attributes satisfy the access policy. Wang et al. [38] proposed a pre-authentication approach for sharing data in cloud, which achieves receiver's attribute authentication before the re-encryption operation.

The multiparty privacy control among co-owners is indispensable in cloud computing. Thomas et al. [20] showed how Facebook's privacy model can be adopted to achieve multiparty privacy. It allows all associated parties to specify exposure policies for the data, so that users can access the data if satisfying the exposure policies of owner and all the associated parties. Based on this multiparty privacy control model, Xu et al. [19] designed a mechanism to enable each user in a photo to participate in the decision of access control conditions of the photo. However, the above schemes may have privacy conflicts problem, which do not consider how users would actually achieve compromise [39]. To resolve the privacy conflicts among multiparty (negotiating users), Such et al. [40] proposed the first computational mechanism. The core idea is to estimate item sensitivity, relative importance and willingness for each conflicting negotiating users, and let the one who has less stringent privacy requirement compromise. Hu et al. [41] proposed a systematic

approach to enable privacy-preserving data sharing with multi-owner. This scheme introduces three strategies based on a voting mechanism to resolve the multiparty privacy conflicts. Unfortunately, this scheme only focuses on co-owners' access control over plaintext data, and ignores the data confidentiality towards semi-trusted CSP and malicious users.

### 3 PRELIMINARIES

#### 3.1 Bilinear Pairing

Let  $\mathbb{G}_0$  and  $\mathbb{G}_T$  be two multiplicative groups of prime order  $p$ . A bilinear map is a function  $e: \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_T$  with the following properties [42]:

- 1) Computability. There is an efficient algorithm to compute  $e(g, h) \in \mathbb{G}_T$ , for any  $g, h \in \mathbb{G}_0$ .
- 2) Bilinearity. For all  $g, h \in \mathbb{G}_0$  and  $a, b \in \mathbb{Z}_p$ , we have  $e(g^a, h^b) = e(g, h)^{ab}$ .
- 3) Non-degeneracy. If  $g$  is a generate of  $\mathbb{G}_0$ , then  $e(g, g)$  is also a generator of  $\mathbb{G}_T$ .

#### 3.2 Access Tree

Let  $T$  be a tree representing an access policy [43]. Each non-leaf node  $x$  of tree represents a threshold gate. Let  $num_x$  denote the number of children of a node  $x$ , and  $k_x$  represent its threshold value, then  $1 \leq k_x \leq num_x$ . The threshold gate is an AND gate if  $k_x = num_x$ , and an OR gate if  $k_x = 1$ . For each leaf node  $x$  of tree, we denote  $attr_x$  as an attribute associated with it and have  $k_x = 1$ . Each child node of node  $x$  is numbered from 1 to  $num_x$ . The function  $parent(x)$  represents a parent node of the node  $x$ , and  $index(x)$  returns the index of the node  $x$ .

Let  $T_x$  be a subtree rooted at the node  $x$  in the access tree. If the access tree  $T_x$  is satisfied by a set of attributes  $S$ , we denote it as  $T_x(S) = 1$ . For any node  $x$ ,  $T_x(S)$  is computed as follows. If  $x$  is a leaf node,  $T_x(S)$  returns 1 only if  $attr_x \in S$ . If  $x$  is a non-leaf node, it evaluates  $T_n(S)$  for all children  $n$  of  $x$ , and returns 1 if and only if at least  $k_x$  children return 1.

### 4 PROBLEM STATEMENT

#### 4.1 System Model

The system model consists of the following entities, as shown in Fig. 1. The notations used throughout this paper are presented in Table 1.

1) Trusted authority: The trusted authority is a fully trusted part that initializes the system public key, and generates private keys as well as attribute keys for users. For example, it can be acted by the administrator of the organization [18] or social security administration [44].

2) CSP: The CSP is a semi-trusted part that provides each user with a virtual space and convenient data storage service with the cloud infrastructure. It also appends access policies to the ciphertexts for data co-owners and

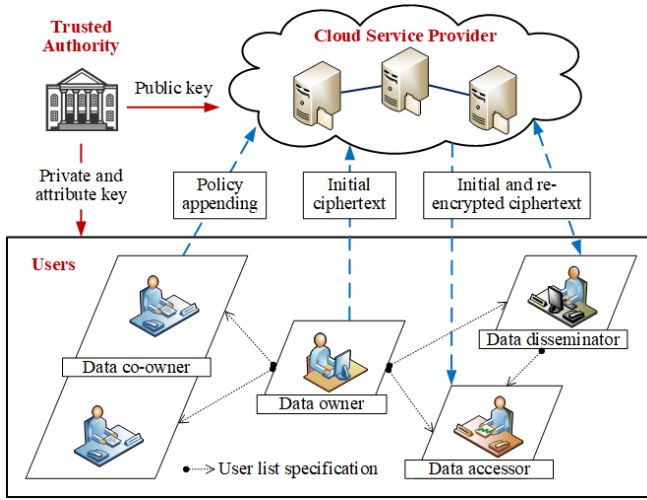


Fig. 1. System model of proposed scheme. The user role is divided into the following categories: data owner, data co-owner, data disseminator and data accessor.

generates re-encrypted ciphertexts for users.

3) User: We divide the user role into the following categories: data owner, data co-owner, data disseminator and data accessor. The data owner can choose a policy aggregation strategy and define an access policy to enforce dissemination conditions. Then he encrypts data for a set of receivers, and outsources the ciphertext to CSP for sharing and dissemination. The data co-owners tagged by data owner can append access policies to the encrypted data with CSP and generate the renewed ciphertext. The data disseminator can access the data and also generate the re-encryption key to disseminate data owner's data to others if he satisfies enough access policies in the ciphertext. The data accessor can decrypt the initial, renewed and re-encrypted ciphertext with her or his private key.

## 4.2 Policy Aggregation Strategies

In our scheme, data co-owners can renew the ciphertexts by appending their access policies as the dissemination conditions. As described in [41], we provide following strategies to fulfill the authorization requirements from

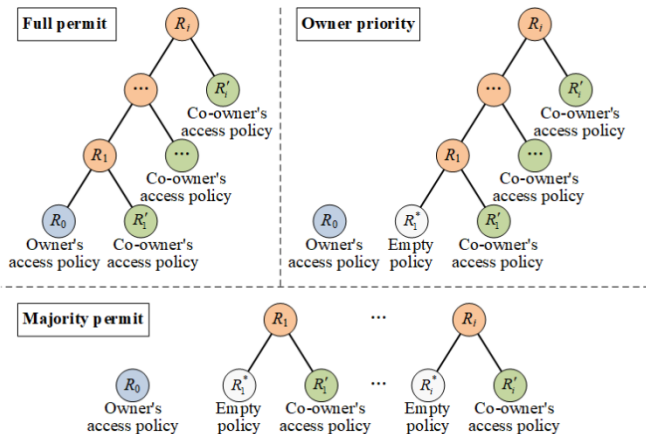


Fig. 2. Three policy aggregation strategies with multi-owner.

TABLE 1  
NOTATIONS

Symbols	Description
$MK, PK$	The master secret key and system public key
$SK$	The private key of user
$AK$	The attribute key of user
$M$	The data
$U$	The set of data accessors' identities
$W$	The set of data co-owners' identities
$DK$	The symmetric key
$CT_0$	The initial ciphertext
$T_0$	The access tree of $CT_0$
$CT_i$	The renew ciphertext generated by policy appending
$T'_{i+1}$	The access tree customized by data co-owner for $CT_i$
$TK_i$	The transformation key of data co-owner for $CT_i$
$T_i$	The access tree of $CT_i$
$U'$	The set of new accessors' identities
$RK$	The re-encryption key of data disseminator
$CT'_i$	The re-encrypted ciphertext

multi-owner, as shown in Fig. 2.

1) Full permit: All owners (including data owner and data co-owners) have the same right to decide the dissemination conditions of data. The data disseminator should satisfy all the access policies defined by these owners.

2) Owner priority: The data owner's decision has high priority, though he tags the co-owners. The data disseminator can disseminate the data only when he satisfies the access policy of data owner or all the access policies of data co-owners.

3) Majority permit: The data owner firstly chooses a threshold value, and the data can be disseminated if and only if the sum of access policies satisfied by disseminator's attributes is greater than or equal to this fixed threshold.

## 4.3 Security Definitions and Goals

We first assume that trusted authority is fully trusted by other entities and will not collude with any entities, which is also employed by related works [18, 31, 34, 36]. We then assume that CSP is semi-trusted, which will honestly execute the requests from the entities and may be curious to learn as much information about the stored data as possible. Besides, we assume that data owners are trusted, but some users will try to access the data beyond their privileges, even by colluding with other users and CSP. Further, we do not consider data version management, which means once a ciphertext is renewed, the users cannot obtain the previous ciphertext, and we assume that the ownership of data can be guaranteed by the ciphertext deduplication scheme [45]. Specifically, the security goals are summarized as follows.

1) Data confidentiality: The data should be well protected against the semi-trusted CSP and unauthorized users. The users who are not the receivers of a ciphertext defined by the data owner or data disseminator should not be able to access the plaintext.

2) Fine-grained dissemination conditions: The data owner and data co-owners can customize fine-grained and tree-based dissemination conditions for their data. The ciphertext can only be disseminated by the users who satisfy these conditions.

3) Continuous policy enforcement: The data owner's access policy is enforced in the initial ciphertext as well as the renewed ciphertext.

4) Collusion resistance: If each of the data disseminators' attributes cannot satisfy the access policies in the ciphertext individually with their own attributes, these users could not collude and decrypt this ciphertext.

## 5 PROPOSED SCHEME

### 5.1 System Setup

The trusted authority selects a bilinear map  $e: \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_T$ , where  $\mathbb{G}_0$  and  $\mathbb{G}_T$  are two multiplicative groups with prime order  $p$ . Then trusted authority chooses a security parameter  $\lambda \in \mathbb{Z}_p$ , a maximum number of receivers  $N$ , and randomly chooses  $g, h, u \in \mathbb{G}_0$  and  $\gamma, \beta \in \mathbb{Z}_p$ , cryptographic hash functions  $H_1: \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ ,  $H_2: \{0,1\}^* \rightarrow \mathbb{G}_0$ ,  $H_3: \mathbb{G}_T \rightarrow \mathbb{G}_0$  and  $H_4: \mathbb{G}_T \rightarrow \mathbb{Z}_p^*$ . Then it generates the master secret key  $MK = (g, \gamma, \beta)$ , and outputs the system public key

$$PK = (h, h^\gamma, \dots, h^{\gamma^N}, u, u^\gamma, \dots, u^{\gamma^N}, h^\beta, h^{\gamma/\beta}, u^\beta, g^\gamma, g^\beta, e(g, h), e(g, h)^\gamma) \quad (1)$$

### 5.2 Key Generation

The trusted authority generates the private key  $SK$  for the user with identity  $ID$ .

$$SK = g^{1/(\gamma + H_1(ID))} \quad (2)$$

The trusted authority generates the attribute key  $AK$  for data disseminator. It chooses a random  $\alpha \in \mathbb{Z}_p$ , and random  $r_j \in \mathbb{Z}_p$  for each attribute  $j \in S$ , where  $S$  is the attribute set. The  $AK$  is outputted as follows.

$$AK = (D_0 = g^{(\gamma + \alpha)/\beta}, \{D_j = g^\alpha H_2(j)^{r_j}, D'_j = h^{r_j}\}_{j \in S}) \quad (3)$$

### 5.3 Data Encryption

Let  $M$  be the shard data. The data owner chooses a set  $U$  of data accessors' identities, a set  $W$  of data co-owners' identities, where  $|U| \leq N$  and  $|W| \leq N$ . Then the data owner customizes a tree-based access policy, and chooses a random  $DK$  which is used to encrypt data  $M$  based on symmetric encryption algorithm  $SE$ .

For each access tree, the data owner chooses a polynomial  $p_x$  for each node  $x$ . We set the degree  $d_x$  of polynomial  $p_x$  to be one less than the threshold value  $k_x$ , that is  $d_x = k_x - 1$ . These polynomials are chosen in a top-down manner. For the root node  $R$ , data owner chooses a random  $secret$  and sets  $p_R(0) = secret$ , and chooses  $d_R$  other

points of  $p_R$  randomly to define it completely. For any other node  $x$ , it sets  $p_x(0) = p_{parent(x)}(index(x))$  and randomly chooses  $d_x$  other points to define  $p_x$  completely. Specially, the empty policy has only one child which can be satisfied by any data disseminator. Then data owner picks  $k, k', \mu, \lambda \in \mathbb{Z}_p$  randomly, computes  $b = \mu \parallel \lambda$ , and encrypts  $DK$  according to the policy aggregation strategy.

1) Full permit: The data owner defines an access tree  $T_0$  with root node  $R_0$ . Let  $Y_0$  be the set of leaf nodes in  $T_0$ . The data owner randomly chooses  $t_0 \in \mathbb{Z}_p$ , and sets  $p_{R_0}(0) = t_0$ , and outputs the initial ciphertext  $CT_0$ .

$$\begin{aligned} CT_0 &= (C_0 = SE_{DK}(M), C_1 = DK \cdot e(g, h)^k, \\ C_2 &= b \cdot H_4(e(g, h)^{k'}), C_3 = h^{\prod_{ID_i \in U} (\gamma + H_1(ID_i))}, \\ C_4 &= h^{\prod_{ID_i \in W} (\gamma + H_1(ID_i))}, C_5 = g^{-\gamma k}, C_6 = g^{-\gamma k'}, \\ C_{0,7} &= u^{\beta \mu t_0 + k \cdot \prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}}, C_{0,8} = h^{\beta \mu t_0}, C_{0,9} = g^{\beta \mu t_0}, \\ C_{0,10} &= \{\tilde{C}_y = h^{\mu p_y(0)}, \tilde{C}'_y = H_2(attr_y)^{\mu p_y(0)}\}_{y \in Y_0} \end{aligned} \quad (4)$$

2) Owner priority: The data owner defines an access tree  $T_0$  with root node  $R_0$  for himself, and an empty policy  $T_1^*$  with root node  $R_1^*$  for all data co-owners. Then the data owner chooses random  $f_0, t_0, s_0 \in \mathbb{Z}_p$ , sets  $p_{R_0}(0) = t_0$  and  $p_{R_1^*}(0) = s_0$ . Let  $X_0$  be the set of leaf nodes in  $T_1^*$ . Then the data owner outputs the initial ciphertext  $CT_0$ .

$$\begin{aligned} CT_0 &= (C_0, C_1, C_2, C_3, C_4, C_5, C_6, \\ \bar{C} &= u^{\beta \mu f_0 + k \cdot \prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}}, C_7 = u^{\mu(t_0 + f_0)}, C_8 = h^{\beta \mu t_0}, \\ C_9 &= g^{\beta \mu t_0}, C_{10} = \{\tilde{C}_y, \tilde{C}'_y\}_{y \in Y_0}, C_{0,7} = u^{\mu(s_0 + f_0 + \lambda)}, \\ C_{0,8} &= h^{\beta \mu s_0}, C_{0,9} = g^{\beta \mu s_0}, C_{0,10} = \{\tilde{C}_x, \tilde{C}'_x\}_{x \in X_0} \end{aligned} \quad (5)$$

3) Majority permit: The data owner defines an access tree  $T_0$  with root node  $R_0$  for himself and  $|W|$  empty policies for each data co-owner. For each access tree of data co-owner  $T_i^*$  where  $i > 0$ ,  $R_i^*$  is the root node,  $Y_i$  is the set of leaf nodes. For each access tree, data owner chooses a random  $t_i \in \mathbb{Z}_p$ , and sets  $p_{R_0}(0) = t_0$  and  $p_{R_i^*}(0) = t_i$ . The data owner chooses a threshold value  $t$  and a polynomial  $f$ , and sets the degree  $d = t - 1$ . Then data owner chooses a random  $f_0 \in \mathbb{Z}_p$  and sets  $f(0) = f_0$ , and randomly chooses  $d$  other points of the polynomial  $f$ . Finally, the initial ciphertext  $CT_0$  is outputted as follows.

$$\begin{aligned} CT_0 &= (C_0, C_1, C_2, C_3, C_4, C_5, C_6, \\ \bar{C} &= u^{\beta \mu f_0 + k \cdot \prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}}, C_{0,7} = u^{\mu(t_0 + f(1))}, C_{0,8} = h^{\beta \mu t_0}, \\ C_{0,9} &= g^{\beta \mu t_0}, C_{0,10} = \{\tilde{C}_y, \tilde{C}'_y\}_{y \in Y_0}, \{C_{i,7} = u^{\mu(t_i + f(i+1) + \lambda)}, \\ C_{i,8} &= h^{\beta \mu t_i}, C_{i,9} = g^{\beta \mu t_i}, C_{i,10} = \{\tilde{C}_y, \tilde{C}'_y\}_{y \in Y_i}\}_{1 \leq i \leq |W|} \end{aligned} \quad (6)$$

## 5.4 Co-owner Key Generation

The data co-owner can append her or his own access policy to the ciphertext  $CT_i$  (such as  $CT_0$ ). First, the data co-owner runs *DecryptIdentity* algorithm by inputting private key  $SK$ , identity  $ID$ , and the ciphertext. If  $ID \in W$ , data co-owner computes

$$\begin{aligned} I &= \text{DecryptIdentity}(SK, ID, W, C_6, C_4) \\ &= (e(C_6, h^{\Delta_{\gamma}(ID, W)})) \cdot e(SK, C_4) \cdot \frac{1}{\prod_{ID_i \in W \wedge ID_i \neq ID} H_1(ID_i)} \\ &= (e(g, h^{k' \cdot \prod_{ID_i \in W \wedge ID_i \neq ID} H_1(ID_i)})) \cdot \frac{1}{\prod_{ID_i \in W \wedge ID_i \neq ID} H_1(ID_i)} \\ &= e(g, h)^{k'} \end{aligned} \quad (7)$$

with  $h^{\Delta_{\gamma}(ID, W)} = h^{\gamma^{-1} \cdot (\prod_{ID_i \in W \wedge ID_i \neq ID} (\gamma + H_1(ID_i))) - \prod_{ID_i \in W \wedge ID_i \neq ID} H_1(ID_i)}$ . Then, the data co-owner recovers  $b = C_2/H_4(I)$ , and customizes a new access policy  $T'_{i+1}$ . It chooses a polynomial  $p_z$  for each node  $z$  in  $T'_{i+1}$ . For the root node  $R'_{i+1}$ , the data co-owner chooses a random  $v_i \in \mathbb{Z}_p$  and sets  $p_{R'_{i+1}}(0) = v_i$ . Let  $Z_i$  be the set of leaf nodes in  $T'_{i+1}$ . Then data co-owner computes  $K_{i,7} = u^{-\beta \mu v_i / 2}$  for full permit strategy, and  $K_{i,7} = u^{-\mu(v_i/2 + \lambda)}$  for majority permit strategy.

For owner priority strategy, the data co-owner computes as follows.

$$K_{i,7} = \begin{cases} u^{-\mu(v_0/2 + \lambda)} & i = 0 \\ u^{-\mu v_i / 2} & i > 0 \end{cases} \quad (8)$$

Then data co-owner sends transformation key  $TK_i = (K_{i,7}, K_{i,8} = h^{-\beta \mu v_i / 2}, K_{i,9} = g^{-\beta \mu v_i / 2}, K_{i,10} = \{\tilde{C}_z, \tilde{C}'_z\}_{z \in Z_i})$  to the CSP.

## 5.5 Policy Appending

When receiving  $TK_i$ , the CSP generates the new ciphertext from  $CT_i$  according to the policy aggregation strategy.

1) Full permit: The CSP first constructs a new access tree  $T_{i+1}$  with the root node  $R_{i+1}$ . Let  $Y_{i+1} = Y_i + Z_i$ ,  $t_{i+1} = t_i - v_i / 2^{i+1}$ . The CSP computes the following with  $TK_i$ .

$$\begin{aligned} C_{i+1,7} &= C_{i,7} \cdot (K_{i,7})^{1/2^i} = u^{\beta \mu t_i + k' \cdot \prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}} \cdot (u^{-\beta \mu v_i / 2})^{1/2^i} \\ C_{i+1,8} &= C_{i,8} \cdot (K_{i,8})^{1/2^i} = h^{\beta \mu t_i} \cdot (h^{-\beta \mu v_i / 2})^{1/2^i} \\ C_{i+1,9} &= C_{i,9} \cdot (K_{i,9})^{1/2^i} = g^{\beta \mu t_i} \cdot (g^{-\beta \mu v_i / 2})^{1/2^i} \end{aligned} \quad (9)$$

Then the CSP outputs a new ciphertext  $CT_{i+1}$ .

$$\begin{aligned} CT_{i+1} &= (C_0, C_1, C_2, C_3, C_4, C_5, C_6, \\ &C_{i+1,7} = u^{\beta \mu t_{i+1} + k' \cdot \prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}}, C_{i+1,8} = h^{\beta \mu t_{i+1}}, \\ &C_{i+1,9} = g^{\beta \mu t_{i+1}}, C_{i+1,10} = \{\tilde{C}_y, \tilde{C}'_y\}_{y \in Y_{i+1}}) \end{aligned} \quad (10)$$

2) Owner priority: The CSP constructs a new access tree  $T_{i+1}$  with the root node  $R_{i+1}$ . Let  $X_{i+1} = X_i + Z_i$ ,  $s_{i+1} = s_i - v_i / 2^{i+1}$ . The CSP computes the following with  $TK_i$ .

$$C_{i+1,7} = C_{i,7} \cdot (K_{i,7})^{1/2^i} = \begin{cases} u^{\mu(s_0 + f_0 + \lambda)} \cdot u^{-\mu(v_0/2 + \lambda)} & i = 0 \\ u^{\mu(s_i + f_0)} \cdot u^{-\mu v_i / 2^{i+1}} & i > 0 \end{cases} \quad (11)$$

Then the CSP computes the following with  $TK_i$ .

$$\begin{aligned} C_{i+1,8} &= C_{i,8} \cdot (K_{i,8})^{1/2^i} = h^{\beta \mu s_i} \cdot (h^{-\beta \mu v_i / 2})^{1/2^i} \\ C_{i+1,9} &= C_{i,9} \cdot (K_{i,9})^{1/2^i} = g^{\beta \mu s_i} \cdot (g^{-\beta \mu v_i / 2})^{1/2^i} \end{aligned} \quad (12)$$

Then the CSP outputs a new ciphertext  $CT_{i+1}$ .

$$\begin{aligned} CT_{i+1} &= (C_0, C_1, C_2, C_3, C_4, C_5, C_6, \bar{C}, \\ &C_7, C_8, C_9, C_{10}, C_{i+1,7} = u^{\mu(s_{i+1} + f_0)}, C_{i+1,8} = h^{\beta \mu s_{i+1}}, \\ &C_{i+1,9} = g^{\beta \mu s_{i+1}}, C_{i+1,10} = \{\tilde{C}_x, \tilde{C}'_x\}_{x \in X_{i+1}}) \end{aligned} \quad (13)$$

3) Majority permit: The CSP constructs a new access tree  $T_{i+1}$  with root node  $R_{i+1}$ . Let  $Y'_{i+1} = Y_{i+1} + Z_i$ ,  $t'_{i+1} = t_{i+1} - v_i / 2$ . Then the CSP computes the following with  $TK_i$ .

$$\begin{aligned} C'_{i+1,7} &= C_{i+1,7} \cdot K'_{i,7} = u^{\mu(t_{i+1} + f(i+2) + \lambda)} \cdot u^{-\mu(v_i/2 + \lambda)} \\ C'_{i+1,8} &= C_{i+1,8} \cdot K_{i,8} = h^{\beta \mu t'_{i+1}} \cdot h^{-\beta \mu v_i / 2} \\ C'_{i+1,9} &= C_{i+1,9} \cdot K_{i,9} = g^{\beta \mu t'_{i+1}} \cdot g^{-\beta \mu v_i / 2} \end{aligned} \quad (14)$$

The the CSP then outputs a new ciphertext  $CT_{i+1}$ .

$$\begin{aligned} CT_{i+1} &= (C_0, C_1, C_2, C_3, C_4, C_5, C_6, \bar{C}, \\ &C_{0,7}, C_{0,8}, C_{0,9}, C_{0,10}, \\ &\{C'_{i+1,7} = u^{\mu(t'_{i+1} + f(i+2))}, C'_{i+1,8} = h^{\beta \mu t'_{i+1}}, C'_{i+1,9} = g^{\beta \mu t'_{i+1}}, \\ &C'_{i+1,10} = \{\tilde{C}_y, \tilde{C}'_y\}_{y \in Y'_{i+1}}\}_{1 \leq i+1 \leq |W|}) \end{aligned} \quad (15)$$

## 5.6 Re-encryption Key Generation

The data disseminator with identity  $ID$  can also disseminate data owner's data to her or his friends via the CSP. The data disseminator chooses a set  $U'$  of new accessors' identities, randomly picks  $l, s \in \mathbb{Z}_p$ , and computes the following with the SK.

$$\begin{aligned} R_1 &= SK \cdot u^{s/H_1(ID)}, R_2 = h^{l \cdot \prod_{ID_i \in U'} (\gamma + H_1(ID_i))} \\ R_3 &= H_3(e(g, h)^l) \cdot h^s, R_4 = g^{-\gamma l}, R'_4 = h^{\beta s} \end{aligned} \quad (16)$$

Then the data disseminator computes the following with the AK.

$$\begin{aligned} R_5 &= D_0 \cdot u^s = g^{(\gamma + \alpha)/\beta} \cdot u^s, \\ R_6 &= \{\tilde{R}_j = D_j = g^\alpha H_2(j)^{t_j}, \tilde{R}'_j = D'_j = h^{t_j}\}_{j \in S} \end{aligned} \quad (17)$$

Finally, the data disseminator sends the re-encryption key  $RK = (R_1, R_2, R_3, R_4, R'_4, R_5, R_6)$  to the CSP.

## 5.7 Data Re-encryption

The CSP can assist data disseminator to re-encrypt the ciphertext  $CT_i$  with  $RK$ . The CSP first generates

$$\begin{aligned} I &= \text{DecryptIdentity}(R_1, ID, U, C_5, C_3) \\ &= (e(C_5, h^{\Delta_{\gamma}(ID, U)})) \cdot e(R_1, C_3) \cdot \frac{1}{\prod_{ID_i \in U \wedge ID_i \neq ID} H_1(ID_i)} \\ &= e(g, h)^k \cdot e(u^s, h^k) \cdot \frac{\prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}}{H_1(ID_i)} \end{aligned} \quad (18)$$

Then, the CSP computes

$$C'_1 = C_1/I = DK \cdot e(u^s, h^{-k})^{\prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}}. \quad (19)$$

Then, the CSP inputs a ciphertext  $CT_i$ ,  $RK$  described by a set of attributes  $S$ , and a node  $x$  from  $T_i$ , and runs the recursive algorithm *DecryptNode*. If  $x$  is a leaf node, then we let  $a = attr_x$  and define as follows. If  $a \in S$ , then

$$\begin{aligned} DecryptNode(CT_i, RK, x, T_i) &= \frac{e(\tilde{R}_i, \tilde{C}_x)}{e(\tilde{R}'_i, \tilde{C}'_x)} \\ &= \frac{e(g^{\alpha} H_2(a)^{r_i}, h^{\mu_{p_x}(0)})}{e(h^{r_i}, H_2(a)^{\mu_{p_x}(0)})} \cdot (20) \\ &= e(g, h)^{\alpha \mu_{p_x}(0)} \end{aligned}$$

If  $a \notin S$ , then we define  $DecryptNode(CT_i, RK, x, T_i) = \perp$ . If  $x$  is a non-leaf node, the algorithm *DecryptNode*( $CT_i, RK, x, T_i$ ) proceeds as follows: for all nodes  $n$  that are children of  $x$ , it calls *DecryptNode*( $CT_i, RK, n, T_i$ ) and stores the result as  $F_n$ . Let  $S_x$  be an arbitrary  $k_x$ -sized set of child nodes  $n$  such that  $F_n \neq \perp$ . If no such set exists, the algorithm returns  $\perp$ . Otherwise, we compute

$$\begin{aligned} F_x &= \prod_{n \in S_x} F_n^{\Delta_{j, S_x}(0)}, \text{ where } \begin{matrix} j = index(n) \\ S_x = \{index(n) : n \in S_x\} \end{matrix} \\ &= \prod_{n \in S_x} (e(g, h)^{\alpha \mu_{parent(n)}(index(n))})^{\Delta_{j, S_x}(0)} \\ &= \prod_{n \in S_x} e(g, h)^{\alpha \mu_{p_x}(j) \cdot \Delta_{j, S_x}(0)} \\ &= e(g, h)^{\alpha \mu_{p_x}(0)} \end{aligned} \quad (21)$$

where  $\Delta_{j, S_x}$  is Lagrange coefficient. If the access tree is satisfied by  $S$ , we set the result of entire computation as  $A$ , such that  $A = DecryptNode(CT_i, RK, R_i, T_i) = e(g, h)^{\alpha \mu_{R_i}(0)}$ . Then the CSP re-encrypts the ciphertext as follows according to the policy aggregation strategy.

1) Full permit: If the attributes  $S$  satisfy access tree  $T_i$ , CSP calculates  $e(g, h)^{\alpha \mu_{R_i}(0)} = e(g, h)^{\alpha \mu(2^i \cdot s_{i-1} - v_{i-1})} = e(g, h)^{\alpha \mu 2^i \cdot t_i}$  and computes

$$\begin{aligned} \tilde{C}' &= \frac{e(C_{i,8}, R_5)}{e(C_{i,9}, h^{\gamma/\beta}) \cdot (A)^{1/2}} \\ &= \frac{e(h^{\beta \mu_{t_i}}, g^{(\gamma+\alpha)/\beta} \cdot u^s)}{e(g^{\beta \mu_{t_i}}, h^{\gamma/\beta}) \cdot e(g, h)^{\alpha \mu 2^i \cdot t_i / 2^i}} = e(h^{\beta \mu_{t_i}}, u^s) \end{aligned} \quad (22)$$

Finally, the CSP generates the re-encrypted ciphertext  $CT'_i = (C'_0 = C_0, C'_1, C'_2 = R_2, C'_3 = R_3, C'_4 = R_4, \tilde{C} = C_{i,7}, \tilde{C}')$ .

2) Owner priority: If the attributes  $S$  satisfy access tree  $T_0$ , CSP calculates  $e(g, h)^{\alpha \mu_{R_0}(0)} = e(g, h)^{\alpha \mu_{t_0}}$  and computes

$$\begin{aligned} B_i &= \frac{e(C_8, R_5)}{e(C_9, h^{\gamma/\beta}) \cdot A} = \frac{e(h^{\beta \mu_{t_0}}, g^{(\gamma+\alpha)/\beta} \cdot u^s)}{e(g^{\beta \mu_{t_0}}, h^{\gamma/\beta}) \cdot e(g, h)^{\alpha \mu_{t_0}}} \\ &= e(h^{\beta \mu_{t_0}}, u^s) \\ \tilde{C}' &= \frac{e(C_7, R_4)}{B_i} = \frac{e(u^{\mu(t_0 + f_0)}, h^{\beta s})}{e(h^{\beta \mu_{t_0}}, u^s)} = e(h^{\beta \mu_{f_0}}, u^s) \end{aligned} \quad (23)$$

Otherwise, if the attributes  $S$  satisfy  $T_i$ , CSP calculates

$$e(g, h)^{\alpha \mu_{R_i}(0)} = e(g, h)^{\alpha \mu(2^i \cdot s_{i-1} - v_{i-1})} = e(g, h)^{\alpha \mu 2^i \cdot s_i} \text{ and computes}$$

$$\begin{aligned} B_i &= \frac{e(C_{i,8}, R_5)}{e(C_{i,9}, h^{\gamma/\beta}) \cdot (A)^{1/2}} = e(h^{\beta \mu_{s_i}}, u^s) \\ \tilde{C}' &= \frac{e(C_{i,7}, R_4)}{B_i} = \frac{e(u^{\mu(s_i + f_0)}, h^{\beta s})}{e(h^{\beta \mu_{s_i}}, u^s)} = e(h^{\beta \mu_{f_0}}, u^s) \end{aligned} \quad (24)$$

Finally, the CSP generates the re-encrypted ciphertext  $CT'_i = (C'_0 = C_0, C'_1, C'_2 = R_2, C'_3 = R_3, C'_4 = R_4, \tilde{C} = \tilde{C}, \tilde{C}')$ .

3) Majority permit: For each access tree  $T_i$ , the CSP can calculate  $e(g, h)^{\alpha \mu_{R_i}(0)} = e(g, h)^{\alpha \mu 2^i}$  and compute

$$\begin{aligned} B_i &= \frac{e(C'_{i,8}, R_5)}{e(C'_{i,9}, h^{\gamma/\beta}) \cdot (A)^{1/2}} = \frac{e(h^{\beta \mu_{t'_i}}, g^{(\gamma+\alpha)/\beta} \cdot u^s)}{e(g^{\beta \mu_{t'_i}}, h^{\gamma/\beta}) \cdot e(g, h)^{\alpha \mu 2^i / 2}} \\ &= e(h^{\beta \mu_{t'_i}}, u^s) \end{aligned} \quad (25)$$

Then the CSP computes

$$B'_i = \frac{e(C'_{i,7}, R_4)}{B_i} = \frac{e(u^{\mu(t'_i + f(i+1))}, h^{\beta s})}{e(h^{\beta \mu_{t'_i}}, u^s)} = e(h^{\beta \mu}, u^s)^{f(i+1)}. \quad (26)$$

Let  $\mathcal{T}$  be an arbitrary  $t$ -sized set of access trees  $T_i$  such that  $B'_i \neq \perp$ , and  $L = \{1, 2, \dots, |W| + 1\}$ . The CSP computes

$$\begin{aligned} \tilde{C}' &= \prod_{T_i \in \mathcal{T}} (B'_i)^{\Delta_{i+1, L}(0)} \\ &= \prod_{T_i \in \mathcal{T}} (e(h^{\beta \mu}, u^s)^{f(i+1)})^{\Delta_{i+1, L}(0)} \\ &= e(h^{\beta \mu}, u^s)^{f(0)} \end{aligned} \quad (27)$$

Finally, if the attributes  $S$  satisfy at least  $t$  access trees, the CSP outputs a re-encrypted ciphertext  $CT'_i = (C'_0 = C_0, C'_1, C'_2 = R_2, C'_3 = R_3, C'_4 = R_4, \tilde{C} = \tilde{C}, \tilde{C}')$ .

## 5.8 Data Decryption

1) If the ciphertext is an initial or renewed ciphertext  $CT_i$ , the data accessor can compute  $I = DecryptIdentity(SK, ID, U, C_5, C_3) = e(g, h)^k$  if her or his identity  $ID \in U$ . Then, data accessor computes  $DK = C_1/I$  and recovers  $M$  with the symmetric decryption algorithm.

2) If the ciphertext is a re-encrypted ciphertext  $CT'_i$ , the data accessor can compute  $I = DecryptIdentity(SK', ID', U', C'_4, C'_2) = e(g, h)^l$  if her or his identity  $ID' \in U'$ . Then, the data accessor can compute  $V = C'_3/H_3(I) = h^s$ . Moreover, the data accessor can generate  $Q = e(V, \tilde{C}) / \tilde{C}' =$

$e(h^s, u^{\prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}})$  under three policy aggregation strategies. Therefore, data accessor can decrypt  $DK = C'_1 \cdot Q$  and further get  $M$  using symmetric decryption algorithm.

## 6 SYSTEM ANALYSIS

### 6.1 Correctness

For any re-encrypted ciphertext, if the data accessor is an

intended receiver, the decryption can execute correctly.

(1) If the policy aggregation strategy is full permit, data accessor computes the follows.

$$Q = \frac{e(V, \tilde{C})}{\tilde{C}'} = \frac{e(h^s, u^{\beta\mu_{t_i} + k \cdot \prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}})}{e(h^{\beta\mu_{t_i}}, u^s)} = e(h^s, u^{k \cdot \prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}})$$

(2) If the policy aggregation strategy is owner priority, data accessor computes the follows.

$$Q = \frac{e(V, \tilde{C})}{\tilde{C}'} = \frac{e(h^s, u^{\beta\mu_{f_0} + k \cdot \prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}})}{e(h^{\beta\mu_{f_0}}, u^s)} = e(h^s, u^{k \cdot \prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}})$$

(3) If the policy aggregation strategy is majority permit, data accessor computes the follows.

$$Q = \frac{e(V, \tilde{C})}{\tilde{C}'} = \frac{e(h^s, u^{\beta\mu_{f_0} + k \cdot \prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}})}{e(h^{\beta\mu_{f_0}}, u^s)} = e(h^s, u^{k \cdot \prod_{ID_i \in U} \frac{\gamma + H_1(ID_i)}{H_1(ID_i)}})$$

Hence, data accessor can further generate DK with the result and then recover message M.

## 6.2 Security Analysis

**Definition 1:** The Decisional Bilinear Diffie-Hellman (DBDH) assumption is that no polynomial-time adversary  $\mathcal{A}$  is able to distinguish the following two tuples  $(g^a, g^b, g^c, e(g, g)^{abc})$  and  $(g^a, g^b, g^c, e(g, g)^r)$ , where  $a, b, c$  and  $r$  are randomly chosen.

**Theorem 1:** Our scheme is secure against chosen plaintext attacks under DBDH assumption.

The IBBE scheme [6] has been proven secure against the selective identity and chosen plaintext attack (IND-sID-CPA) in random oracle model. Let  $C$  be the challenger defined in the IND-sID-CPA security of IBBE scheme. We describe a security game among adversary  $A$ , adversary  $B$  and challenger  $C$ . The challenger  $C$  tests adversary  $B$ 's ability in breaking the IND-sID-CPA security of IBBE scheme, while adversary  $B$  serves as a challenger to test adversary  $A$ 's ability in breaking the IND-sID-CPA security of our scheme.

The adversary  $\mathcal{A}$  chooses a set  $U^*$  of challenge identities and a challenge access policy  $T^*$ . Let  $Adv_A^{DBDH}$  be the advantage of adversary  $\mathcal{A}$  to break the DBDH problem,  $Adv_A^{IBBE}$  be the advantage of adversary  $\mathcal{A}$  to break the IBBE scheme. Followed by the security game described in [36], suppose adversary  $A$  totally queries the re-encryption key  $q$  times and has the advantage  $Adv_A$  to break our scheme. Then we have that adversary  $B$  has the advantage  $Adv_B^{IBBE} = Adv_A(1 - q \cdot Adv_A^{DBDH} - q \cdot Adv_A^{IBBE})$  to break the IND-sID-CPA security of IBBE scheme. Since the IBBE scheme is IND-sID-CPA secure in random oracle model, we have that  $Adv_B^{IBBE}$  and  $Adv_A^{IBBE}$  are negligible. Besides,  $Adv_A^{DBDH}$  is also negligible since the DBDH assumption holds. Therefore,  $Adv_A$  must be negligible, which means our scheme is also IND-sID-CPA secure in random oracle model.

Next, we analyze that our scheme can satisfy the fol-

lowing security properties of data sharing and dissemination in cloud computing.

1) Data confidentiality: The data in the cloud is first encrypted with a random symmetric key, and the random key will be encrypted with a set of receivers' identities and access policies based on IBBE and CP-ABE. Hence, the data confidentiality can be protected against users whose identities are not in the set. Moreover, based on the secure CPRE scheme, the CSP cannot get any confidential information about the data during the dissemination phase of any strategy.

2) Fine-grained data dissemination: The symmetric key is protected with attribute-based CPRE mechanism as well, which allows flexibility in enforcing complex access conditions to data disseminators. The data owner and co-owners are able to customize expressive and flexible access policies supporting AND and OR gates to the ciphertext together according to their privacy preferences. Therefore, the CSP can re-encrypt the ciphertext only when the attributes of data disseminator satisfy enough access policies in the ciphertext.

3) Continuous policy enforcement: In full permit strategy, the data owner's access policy and data co-owners' access policies are aggregated by AND gate. Hence, the data owner's access policy is enforced in the renewed ciphertext, since the data disseminator who can disseminate the ciphertext must satisfy the data owner's access policy. In owner priority and majority permit strategies, the data disseminator can only disseminate the initial ciphertext by satisfying the data owner's access policy, since the component  $C_{0,7} = u^{\mu(s_0 + f_0 + \lambda)}$  and  $C_{i,7} = u^{\mu(t_i + f(i+1) + \lambda)}$  are obfuscated by the data owner with random secret  $\lambda$ . Moreover, only the authorized data co-owners can generate the valid transformation key, since the adversary cannot forge a valid transformation key without acquiring  $b$ . Once the ciphertext is renewed by data co-owner, the renewed ciphertext can be disseminated according to the chosen strategy, such as by satisfying all the data co-owners' access policies in owner priority strategy.

4) Collusion resistance: The malicious data disseminators may combine their attributes to disseminate the ciphertext. In our scheme, a random *secret* is embedded into the ciphertext, and colluders must recover this *secret* with their attribute keys. However, the attribute key of each disseminator is associated with random and unique  $\alpha$ . The data disseminators cannot collude to re-encrypt the ciphertext with their different attribute keys. If they collude with the semi-trusted CSP, the collusion attack also cannot take effect.

## 6.3 Functionality Comparisons

We first compare our scheme with several recent schemes, as shown in Table 2. Firstly, our scheme is advanced in fine-grained conditional dissemination as data owner and co-owners could enforce flexible access policies on the ciphertexts, while data owner only can enforce simple keyword conditions in Xu et al. [36]. Further, though Guo



TABLE 2  
FUNCTIONALITY COMPARISONS

Schemes	Data confidentiality	Multiple receivers	Secure dissemination	Re-encryption key generation	Conditional dissemination	Multiple access control	Privacy conflict
[29]	CP-ABE	Yes	Yes	-	Access policy	No	-
[36]	IBBE	Yes	Yes	Disseminator	Keyword	No	-
[40]	-	Yes	No	-	-	Yes	Concession evaluation
[41]	-	Yes	No	-	-	Yes	Voting
Our scheme	IBBE	Yes	Yes	Disseminator	Access policy	Yes	Policy aggregation strategies

et al. [29] achieved fine-grained conditional data dissemination based on ABE, it cannot support data group sharing which is the basic requirement in cloud computing. In our scheme, data disseminators can transform the encrypted data to a new group of users based on IBBE and attribute-based CPRE.

We also compare our scheme with Thomas et al. [20], Such et al. [40] and Hu et al. [41], which are the latest multiparty access control schemes. Thomas et al. [20] gives the definition and solution of multiparty access control, but it ignores the privacy conflicts which may happen when multiple users enforce their different privacy preferences on the shared data. Such et al. [40] and Hu et al. [41] solve the problem of privacy conflicts on plaintext based on concession evaluation mechanism and voting mechanism respectively, while our scheme supports multiparty access control on ciphertext and introduces three strategies of aggregating privacy preferences to solve the problem of privacy conflicts.

## 6.4 Performance Analysis

We next analyze the performance of our scheme. Generally, the costs of pairing and exponentiation operations dominate the major computation time, thus we ignore the multiplication, hash, symmetric encryption and decryption computation. Let  $N_c$  be the number of attributes in

TABLE 3  
COMPUTATION EFFICIENCY

Phase	Computation cost
Key generation	$(2 S +3)T_{exp0}$
Data encryption	Full permit: $2T_{exp1}+(2 U +2N_c+ W +8)T_{exp0}$ Owner priority: $2T_{exp1}+(3 U +2N_c+ W +12)T_{exp0}$ Majority permit: $2T_{exp1}+(2 U +2N_c+ W +4 W +6)T_{exp0}$
Co-owner key generation	$T_{exp1}+(2N_c+5)T_{exp0}+2T_{pair}$
Policy appending	Full permit: $3T_{exp0}$ Owner priority: $3T_{exp0}$ Majority permit: 0
Re-encryption key generation	$T_{exp1}+( U' +5)T_{exp0}$
Data re-encryption	Full permit: $(N_c+1)T_{exp1}+( U +3)T_{exp0}+(2N_c+4)T_{pair}$ Owner priority: $T_0 : (N_c+1)T_{exp1}+( U +2)T_{exp0}+(2N_c+4)T_{pair}$ $T_i : (N_c+1)T_{exp1}+( U +3)T_{exp0}+(2N_c+4)T_{pair}$ Majority permit: $(N_c+t+1)T_{exp1}+( U +3)T_{exp0}+(2N_c+5)T_{pair}$
Data decryption	Initial or renewed ciphertext: $T_{exp1}+( U +2)T_{exp0}+2T_{pair}$ Re-encrypted ciphertext: $T_{exp1}+( U' +2)T_{exp0}+3T_{pair}$

access policy,  $T_{pair}$  be the computation cost of a single pairing operation,  $T_{exp0}$  and  $T_{exp1}$  be the computation cost of an exponent operation on  $\mathbb{G}_0$  and  $\mathbb{G}_T$ . Table 3 shows the computation cost of each phase in our scheme.

First, data owner can choose one of three policy aggregation strategies to encrypt the data. The initial ciphertext is associated with an empty policy  $T_1^*$  and  $|W|$  number of empty policies  $T_1^*$  in strategies of owner priority and majority permit respectively. Thus, the corresponding computation cost are  $2T_{exp1}+(3|U|+2N_c+|W|+12)T_{exp0}$  and  $2T_{exp1}+(2|U|+2N_c+|W|+4|W|+6)T_{exp0}$  in these two strategies which are more than that in full permit strategy.

Then, data co-owners can renew the ciphertext by appending their access policies as the dissemination conditions into empty policy. The computation cost in full permit and owner priority strategies are both  $3T_{exp0}$ , while the computation cost in majority permit strategy is only three multiplications. When the data disseminator disseminates a ciphertext to other users, he must send the re-encryption key to the CSP. The CSP can re-encrypt ciphertext if he satisfies enough access policies in the ciphertext. According to different strategies adopted by data owner, the CSP spends different computation cost. In owner priority strategy, the data disseminator must satisfy either  $T_0$  customized by data owner or  $T_i$  customized by all the data co-owners, of which the computation cost are  $(N_c+1)T_{exp1}+(|U|+2)T_{exp0}+(2N_c+4)T_{pair}$  and  $(N_c+1)T_{exp1}+(|U|+3)T_{exp0}+(2N_c+4)T_{pair}$ . In majority permit strategy, the CSP would spend  $(N_c+t+1)T_{exp1}+(|U|+3)T_{exp0}+(2N_c+5)T_{pair}$  to re-encrypt ciphertext if data disseminator satisfies at least  $t$  access policy trees.

Finally, data accessor can decrypt initial or renewed ciphertext with her or his private key  $SK$  if he is an intended receiver in the set of  $U$ , and the computation cost on data accessor side is  $T_{exp1}+(|U|+2)T_{exp0}+2T_{pair}$ . If the data accessor is a member in the set of  $U'$ , he would have one extra pairing operation to decrypt the ciphertext.

## 7 EXPERIMENTAL RESULTS

In this section, we implement our scheme on a cloud server with a 2.53 GHz Intel Core 2 Duo CPU and 4 GB memory based on pairing-based cryptography library [46]. A pairing-friendly type-A 160-bit elliptic curve group based on the supersingular curve  $y^2 = x^3 + x$  over a 512-bit finite field is used, and the public parameters are chosen to provide 80 bits security level. We conduct vari-

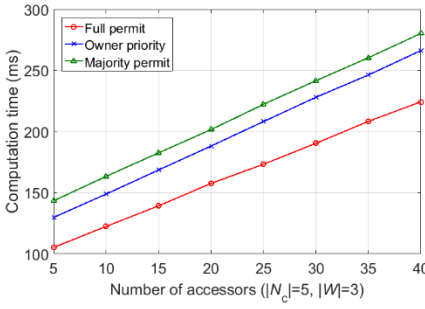


Fig. 3. Computation time versus users in encryption phase.

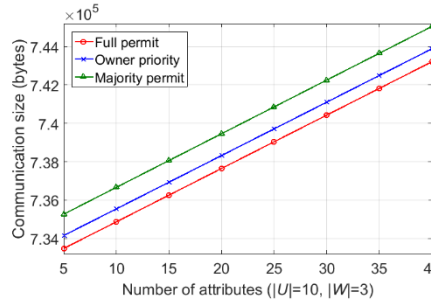


Fig. 4. Communication size versus attributes in encryption phase.

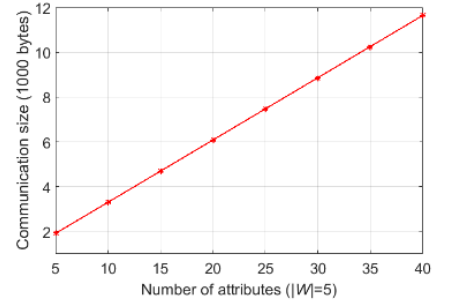


Fig. 5. Communication size versus attributes in co-owner key generation phase.

ous experiments and chooses the Advanced Encryption Standard (AES) as the symmetric encryption scheme. The experimental results are the mean of 100 trials.

In the encryption phase, data owner defines a set of identities and an access policy, and then uploads the encrypted data to the CSP. We utilize the computation time and communication size as the metric to measure complexity. The computation time is mainly related to two factors, that are number of accessors and attributes in the access policy. Fig. 3 shows the computation time of data encryption versus  $|U|$  under a fixed access policy with 5 attributes and 3 co-owners. Due to data owner should set up one and multiple empty policies for co-owners in owner priority strategy and majority permit strategy respectively, the computation cost of these two strategies is higher than that of full permit strategy. Fig. 4 compares the communication cost of data owner when he chooses each of three strategies. On the whole, ciphertext sizes in three strategies are all increasing linearly with  $N_c$ . More particularly, communication cost of majority permit strategy is the highest, and the communication cost of owner priority strategy is a little more than full permit strategy, since the number of shares of  $C_7, C_8, C_9, C_{10}$  in owner priority strategy is twice as much as that in full permit strategy. The number of shares in majority permit strategy is equal to the number of co-owners, that is 3 in Fig. 4.

In the co-owner key generation phase, the data co-owners define access policies according to their privacy concerns and generate the transformation key with private keys. We consider a common case where the number of co-owners is fixed to be 5, since three to five data co-

owners are very common for situations in real world. The communication cost in this phase is given in Fig. 5. We also measure the computation cost of policy appending, as shown in Fig. 6. In particular, the results show that the computation cost of each co-owner in each strategy to enforce her or his access policy on the ciphertext. It can be observed that the cost for policy appending is almost the same in full permit strategy and owner priority strategy, and the result in majority permit strategy is the lowest and almost constant in 0.18 ms.

Further, in order to evaluate the relationship between the computation cost of re-encryption and the number of attributes in the access policy in each strategy, we fix the number of accessors and co-owners to be 10 and 4 respectively, and we assume that the re-encryption operation is performed after all co-owners have appended their access policies. Fig. 7 shows the computation cost of re-encryption in each strategy versus the number of attributes. In the owner priority strategy, the ciphertext can be re-encrypted if the attributes satisfy the access tree  $T_0$  or  $T_i$ . In the majority permit strategy, we evaluate the computation costs of data re-encryption when the threshold  $t$  is selected as 1, 3 and 5. If the threshold  $t$  is 1, the re-encryption will success when the data disseminator satisfies any one of the access policies, and the computation time is a little more than that in owner priority strategy under access tree  $T_0$ . If the threshold  $t$  is 5, the data disseminator needs to satisfy all five access trees and compute the result using polynomial interpolation, which causes highest computation cost compared to full permit strategy and owner priority strategy.

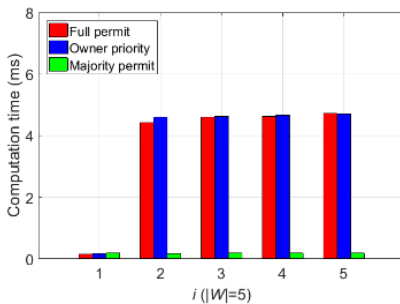


Fig. 6. Computation cost of three strategies in policy appending phase.

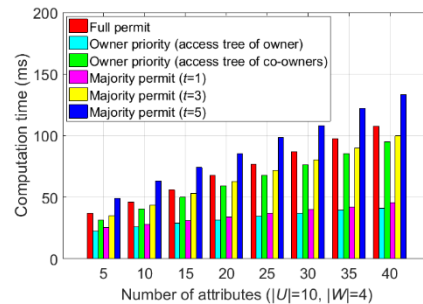


Fig. 7. Computation cost versus attributes in re-encryption phase.

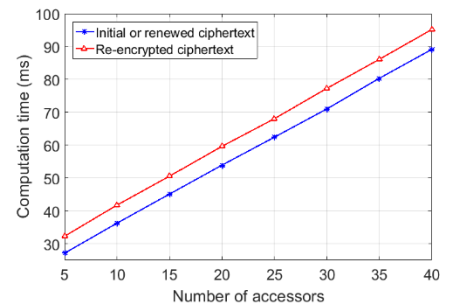


Fig. 8. Computation cost versus accessors in decryption phase.

Finally, Fig. 8 describes the computation time on accessor side when decrypting ciphertext versus the number of accessors. The computation time of decrypting a re-encrypted ciphertext is much higher than the time of decrypting an initial ciphertext. The reason is that data accessor needs to perform one more pairing operation and one more hash operation to decrypt the re-encrypted ciphertext.

The experimental results show that in full permit strategy, it takes about 122 ms to encrypt the shared data when there are 10 accessors, and the ciphertext size is only increased by 4145 bytes when the number of attributes is 10. In the policy appending phase, the communication cost for data co-owner is 3303 bytes which is mainly caused by the transformation key, and the maximum computation cost for the CSP is less than 5 ms in three strategies, even when the number of co-owners increases to 5. Therefore, our scheme is practical and efficient for data group sharing with multi-owner in cloud computing.

## 8 CONCLUSION

The data security and privacy is a concern for users in cloud computing. In particular, how to enforce privacy concerns of multiple owners and protect the data confidentiality becomes a challenge. In this paper, we present a secure data group sharing and conditional dissemination scheme with multi-owner in cloud computing. In our scheme, the data owner could encrypt her or his private data and share it with a group of data accessors at one time in a convenient way based on IBBE technique. Meanwhile, the data owner can specify fine-grained access policy to the ciphertext based on attribute-based CPRE, thus the ciphertext can only be re-encrypted by data disseminator whose attributes satisfy the access policy in the ciphertext. We further present a multiparty access control mechanism over the ciphertext, which allows the data co-owners to append their access policies to the ciphertext. Besides, we provide three policy aggregation strategies including full permit, owner priority and majority permit to solve the problem of privacy conflicts. In the future, we will enhance our scheme by supporting keyword search over the ciphertext [47, 48].

## ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant No. 61572080, National Key Research and Development Program of China under Grant No. 2016YFB0800605, Key Program of Joint Funds of the National Natural Science Foundation of China under Grant No. U1736212, and China Scholarship Council under Grant No. 201806475007.

## REFERENCES

- [1] Z. Yan, X. Li, M. Wang, and A. V. Vasilakos, "Flexible data access control based on trust and reputation in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 485-498, 2017.

- [2] B. Lang, J. Wang, and Y. Liu, "Achieving flexible and self-contained data protection in cloud computing," *IEEE Access*, vol. 5, pp. 1510-1523, 2017.
- [3] Q. Zhang, L. T. Yang, and Z. Chen, "Privacy preserving deep computation model on cloud for big data feature learning," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1351-1362, 2016.
- [4] H. Cui, X. Yi, and S. Nepal, "Achieving scalable access control over encrypted data for edge computing networks," *IEEE Access*, vol. 6, pp. 30049-30059, 2018.
- [5] K. Xue, W. Chen, W. Li, J. Hong, and P. Hong, "Combining data owner-side and cloud-side access control for encrypted cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2062-2074, 2018.
- [6] C. Delerablée, "Identity-based broadcast encryption with constant size ciphertexts and private keys," *Proc. International Conf. on the Theory and Application of Cryptology and Information Security (ASIACRYPT '2007)*, pp. 200-215, 2007.
- [7] N. Paladi, C. Gehrmann, and A. Michalas, "Providing user security guarantees in public infrastructure clouds," *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 405-419, 2017.
- [8] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute based encryption," *Proc. IEEE Symposium on Security and Privacy (SP '07)*, pp. 321-334, 2007.
- [9] L. Liu, Y. Zhang, and X. Li, "KeyD: secure key-deduplication with identity-based broadcast encryption," *IEEE Transactions on Cloud Computing*, 2018, <https://ieeexplore.ieee.org/document/8458136>.
- [10] Q. Huang, Y. Yang, and J. Fu, "Secure data group sharing and dissemination with attribute and time conditions in Public Clouds," *IEEE Transactions on Services Computing*, 2018, <https://ieeexplore.ieee.org/document/8395392>.
- [11] Box, "Understanding collaborator permission levels", <https://community.box.com/t5/Collaborate-By-Inviting-Others/Understanding-Collaborator-Permission-Levels/ta-p/144>.
- [12] Microsoft OneDrive, "Document collaboration and co-authoring", <https://support.office.com/en-us/article/document-collaboration-and-co-authoring-ee1509b4-1f6e-401e-b04a-782d26f564a4>.
- [13] H. He, R. Li, X. Dong, and Z. Zhang, "Secure, efficient and fine-grained data access control mechanism for P2P storage cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 4, pp. 471-484, 2014.
- [14] Z. Qin, H. Xiong, S. Wu, and J. Batamuliza, "A survey of proxy re-encryption for secure data sharing in cloud computing," *IEEE Transactions on Services Computing*, 2018, <https://ieeexplore.ieee.org/document/7448446>.
- [15] J. Son, D. Kim, R. Hussain, and H. Oh, "Conditional proxy re-encryption for secure big data group sharing in cloud environment," *Proc. of 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 541-546, 2014.
- [16] L. Jiang, and D. Guo "Dynamic encrypted data sharing scheme based on conditional proxy broadcast re-encryption for cloud storage," *IEEE Access*, vol. 5, pp. 13336 - 13345, 2017.
- [17] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, Y. Yu, and A. Yang, "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Generation Computer Systems*, vol. 52, pp. 95-108, 2015.
- [18] X. Li, Y. Zhang, B. Wang, and J. Yan, "Mona: secure multi-owner data sharing for dynamic groups in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1182 - 1191, 2013.
- [19] K. Xu, Y. Guo, L. Guo, Y. Fang, and X. Li, "My privacy my decision: control of photo sharing on online social networks," *IEEE Trans. on Dependable and Secure Computing*, vol. 14, no. 2, pp. 199-210, 2017.
- [20] K. Thomas, C. Grier, and D. M. Nicol, "UnFriendly: multi-party privacy risks in social networks," *Proc. International Symposium on Privacy Enhancing Technologies Symp. (PETs '2010)*, pp. 236-252, 2010.
- [21] L. Fang, L. Yin, Y. Guo, Z. Wang, and Fenzhua Li, "Resolving access conflicts: an auction-based incentive approach," *Proc. IEEE Military Communications Conference (MILCOM)*, pp. 1-6, 2018.

- [22] L. Xu, C. Jiang, N. He, Z. Han, and A. Benslimane, "Trust-based collaborative privacy management in online social networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 48-60, 2019.
- [23] C. Gentry and B. Waters, "Adaptive security in broadcast encryption systems (with short ciphertexts)," *Proc. 28th Ann. International Conf. on Advances in Cryptology: the Theory and Applications of Cryptographic (EUROCRYPT '09)*, pp. 171-188, 2009.
- [24] Q. Huang, W. Yue, Y. He, and Y. Yang, "Secure identity-based data sharing and profile matching for mobile healthcare social networks in cloud computing," *IEEE Access*, vol. 6, pp. 36584-36594, 2018.
- [25] S. Patranabis, Y. Shrivastava, and D. Mukhopadhyay, "Provably secure key-aggregate cryptosystems with broadcast aggregate keys for online data sharing on the cloud," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 891-904, 2017.
- [26] A. Sahai and B. Waters, "Fuzzy identity-based encryption," *Proc. 24th Ann. International Conf. on Theory and Applications of Cryptographic Techniques (EUROCRYPT '05)*, pp. 457-473, 2005.
- [27] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," *Proc. 13th ACM Conf. on Computer and Communications Security (CCS '06)*, pp. 89-98, 2006.
- [28] S. Wang, K. Liang, J. K. Liu, J. Chen, J. Yu, and W. Xie, "Attribute-based data sharing scheme revisited in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1661-1673, 2016.
- [29] L. Guo, C. Zhang, H. Yue, and Y. Fang, "A privacy-preserving social-assisted mobile content dissemination scheme in DTNs," *Proc. 32nd IEEE International Conf. on Computer Communications (INFOCOM '2013)*, pp. 2301-2309, 2013.
- [30] W. Teng, G. Yang, Y. Xiang, T. Zhang, and D. Wang, "Attribute-based access control with constant-size ciphertext in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 617-627, 2017.
- [31] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: efficient policy-hiding attribute-based access control," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2130-2145, 2018.
- [32] K. Seol, Y. Kim, E. Lee, Y. Seo, and D. Baik, "Privacy-preserving attribute-based access control model for XML-based electronic health record system," *IEEE Access*, vol. 6, pp. 9114-9128, 2018.
- [33] M. Green and G. Ateniese, "Identity-based proxy re-encryption," *Proc. 5th International Conf. on Applied Cryptography and Network Security (ACNS '07)*, pp. 288-306, 2007.
- [34] Y. Zhou, H. Deng, Q. Wu, B. Qin, J. Liu, and Y. Ding, "Identity-based proxy re-encryption version 2: Making mobile access easy in cloud," *Future Generation Computer Systems*, vol. 62, pp. 128-139, 2016.
- [35] J. Weng, R. H. Deng, X. Ding, C. K. Chu, and J. Lai, "Conditional proxy re-encryption secure against chosen-ciphertext attack," in *Proc. of 4th International Symposium on Information, Computer, and Communications Security (ASIACCS '09)*, pp. 322-332, 2009.
- [36] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin, "Conditional identity-based broadcast proxy re-encryption and its application to cloud email," *IEEE Trans. on Computers*, vol. 65, no. 1, pp. 66-79, 2016.
- [37] Y. Yang, H. Lu, J. Weng, Y. Zhang, and K. Sakurai, "Fine-grained conditional proxy re-encryption and application," *Proc. International Conf. on Provable Security (ProvSec '2014)*, pp. 206-222, 2014.
- [38] K. Wang, J. Yu, X. Liu and S. Guo, "A pre-authentication approach to proxy re-encryption in big data context," *IEEE Transactions on Big Data*, 2018, <https://ieeexplore.ieee.org/document/7921569>.
- [39] H. Hu, G. J. Ahn, and J. Jorgensen, "Detecting and resolving privacy conflicts for collaborative data sharing in online social networks," *Proc. 27th Ann. Computer Security Applications Conf. (ACSAC '11)*, pp. 103-112, 2011.
- [40] J. M. Such and N. Criado, "Resolving multi-party privacy conflicts in social media," *IEEE Trans. on Knowledge and Data Engine*, vol. 28, no. 7, pp. 1851-1863, 2016.
- [41] H. Hu, G. Ahn, and J. Jorgensen, "Multipart access control for online social networks: Model and mechanisms," *IEEE Trans. on Knowledge and Data Engine*, vol. 25, no. 7, pp. 1614-1627, 2013.
- [42] Q. Huang, Y. Yang, and M. Shen, "Secure and efficient data collaboration with hierarchical attribute-based encryption in cloud computing," *Future Generation Computer Systems*, vol. 72, pp. 239-249, 2017.
- [43] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Trans. on Knowledge and Data Eng.*, vol. 25, no. 10, pp. 2271-2282, 2013.
- [44] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1735-1744, 2014.
- [45] S. Jiang, T. Jiang, and L. Wang, "Secure and efficient cloud data deduplication with ownership management," *IEEE Transactions on Services Computing*, <https://ieeexplore.ieee.org/document/8100969>
- [46] B. Lynn. The pairing-based cryptography library. [Online]. Available: <http://crypto.stanford.edu/pbc/>, accessed March 1, 2018.
- [47] A. Michalas, "The lord of the shares: combining attribute-based encryption and searchable encryption for flexible data sharing," *Proc. 34th ACM/SIGAPP Symposium On Applied Computing (SAC)*, pp. 146-155, 2019.
- [48] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: a provably secure scheme under keyword guessing attack," *IEEE Transactions on Computers*, vol. 62, no. 11, pp. 2266-2277, 2013.



**Qinlong Huang** received the Ph.D. degree from the School of Computer, Beijing University of Posts and Telecommunications, China, in 2014. He is currently a lecturer in the School of Cyberspace Security, Beijing University of Posts and Telecommunications, and the associated director of National Engineering Laboratory for Disaster Backup and Recovery, China. He is the principal investigator of the project funded by National Natural Science Foundation of China. He was serving as reviewers for IEEE Transactions on Information Forensics & Security, IEEE Journal of Biomedical and Health Informatics, IEEE Access, IEEE GLOBECOM. His research interests include cloud computing security, social network security and IoT security.



**Yixian Yang** received the Ph.D. degree from Beijing University of Posts and Telecommunications, China, in 1988. He was a Changjiang Distinguished Professor of China in 1993, and was selected in National Science Fund for Distinguished Young Scholars of China in 1994. He is currently a professor in the School of Cyberspace Security, Beijing University of Posts and Telecommunications, China. He is the director of National Engineering Laboratory for Disaster Backup and Recovery, and Information Security Center in Beijing University of Posts and Telecommunications, China. He is the Fellow of China Institute of Communications, the Fellow of Chinese Association for Cryptologic Research, the Council Member of Chinese Institute of Electronics. He is the Editor-in-Chief of the Journal on Communications. He has published more than 300 journals and conference papers. His research interests include cryptography, information and network security.



**Wei Yue** is currently a master candidate in the School of Cyberspace Security, Beijing University of Posts and Telecommunications. Her research interests include cloud computing security.



**Yue He** is currently a master candidate in the School of Cyberspace Security, Beijing University of Posts and Telecommunications. His research interests include cloud computing security.