Secure Data Group Sharing and Dissemination with Attribute and Time Conditions in Public Cloud

Qinlong Huang, *Member, IEEE*, Yixiang Yang and Jingyi Fu

Abstract—Cloud computing has become increasingly popular among users and businesses around the world. Although cryptographic techniques can provide data protection for users in public cloud, several issues also remain problematic, such as secure data group dissemination and fine-grained access control of time-sensitive data. In this paper, we propose an identity-based data group sharing and dissemination scheme in public cloud, in which data owner could broadcast encrypted data to a group of receivers at one time by specifying these receivers' identities in a convenient and secure way. In order to achieve secure and flexible data group dissemination, we adopt attribute-based and timed-release conditional proxy re-encryption to guarantee that only data disseminators whose attributes satisfy the access policy of encrypted data can disseminate it to other groups after the releasing time by delegating a re-encryption key to cloud server. The re-encryption conditions are associated with attributes and releasing time, which allows data owner to enforce fine-grained and timed-release access control over disseminated ciphertexts. The theoretical analysis and experimental results show our proposed scheme makes a tradeoff between computational overhead and expressive dissemination conditions.

Index Terms—Data dissemination, attribute-based encryption, conditional proxy re-encryption, timed-release encryption, cloud computing

1 Introduction

CLOUD computing is regarded as such a computing paradigm in which resources in the computing infrastructure are provided as services over the Internet. The cloud computing benefits individual users and enterprises with convenient access, increased operational efficiencies and rich storage resources by combining a set of existing and new techniques from research areas such as service-oriented architectures and virtualization [1]. Although the great benefits brought by cloud computing are exciting for users, security problems may somehow impede its quick development. Currently, more and more users would outsource their data to cloud service provider (CSP) for sharing. However, the CSP which deprives data owners' direct control over their data is assumed to be honest-but-curious, that may prompt security concerns. These security matters existing in public cloud motivate the requirement to appropriately keep data confidential. Several schemes exploiting cryptographic mechanisms to settle the security problems have been proposed. In order to guarantee secure data group sharing, identity-based broadcast encryption (IBBE) scheme [2] is employed in public cloud. The data owners could broadcast their encrypted data to a group of receivers at one time and the public key of the user can be regarded as email, unique id and username [3]. Hence, by using an identity, data owner can share data with other group users in a convenient and secure manner. Attribute-based encryption (ABE) is one of new cryptographic mechanisms used in cloud to reach flexible and fine-grained secure data group sharing [4]. Especially, ciphertext-policy ABE (CP-ABE) allows data owners to encrypt data with an access policy such that only users whose attributes satisfy the access policy can decrypt the data [5].

Except for being able to allow users to share data with others in public cloud, there is another requirement of data dissemination [6]. For example, when Bob views a photo which is owned by Alice and hopes to share this photo with others, he can specify policies to authorize others to see this photo. In this case, Bob is a disseminator of the photo. The proxy re-encryption (PRE) scheme [7] in a manner could achieve efficient data dissemination in cloud by re-encrypting the ciphertext to other users [8]. However, it may not meet the requirements when data owner doesn't expect all the authorized users who can view his data to disseminate data or allow the disseminators to disseminate all of his data. For example, Alice authorizes Bob and Carol to access her data, but she only allows Bob to disseminate some specific photos or videos to his space. The conditional PRE (CPRE) scheme [9] could address this issue by allowing a user to generate a re-encryption key associated with a condition, and only the encrypted data meeting the condition can be re-encrypted [10]. However, conditions in traditional CPRE which are only keywords may not well match situations in cloud because data owner may have a large number of requirements for different disseminators to disseminate his different data, such as photos taken in home only for families to disseminate and travelling photos allowed to

• *Q. Huang, Y. Yang, J. Fu are with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: longsec@bupt.edu.cn, yxyang@bupt.edu.cn, fujingyi@bupt.edu.cn.*

be disseminated by friends. Thus, fine-grained conditions are inevitably needed in data group dissemination situation in public cloud [11].

Simultaneously, time-sensitive data such as a business plan and a tender, is a special data in cloud which requires time-based exposing [12]. It means that data owner may want different users to disseminate data after different time. For instance, data owner may share sensitive business plan with directors, and he hopes these directors only can disseminate business plan to managers at an early time and then to other employees at last. Simply, the directors can manually release data owner's time-sensitive data, which requires data owner to formulate encrypted data only can be disseminated by directors to managers, and then be disseminated to other employees when the corresponding time arrives. However, this solution forces the directors to repeatedly disseminate different versions of the same data, which brings unnecessary burden. From the perspective of cryptography, this goal of time-based exposing can be achieved by timed-release encryption (TRE) [13]. Currently, some TRE-based systems incorporate the concept of time into a combination of CP-ABE or PRE to support fine-grained and time-based data exposing, whereas these approaches are failure to meet the above scenario of data dissemination.

1.1 Our Contribution

In this paper, we propose a secure data group sharing and dissemination scheme with attribute and time conditions in public cloud. The main contributions of our scheme are as follows:

(1) We employ IBBE technique to achieve secure data group sharing in public cloud, which allows data owner to outsource encrypted data to semi-trusted cloud and share it with a group of receivers at one time. It is more convenient that email and username could be used as public keys for users.

(2) We design an access policy embedding releasing time and take the advantages of attribute-based CPRE, to achieve fine-grained and timed-release data group dissemination. The CSP can re-encrypt initial ciphertexts for data disseminator after the designate time if his attributes associated with the re-encryption key satisfy the access policy in the ciphertexts.

(3) We analyze the security of our proposed scheme, and conduct a detailed theoretical and experimental analysis. The results show that our scheme makes a tradeoff between computational overhead and expressive dissemination conditions, and performs significantly better in data group sharing and dissemination in public cloud.

1.2 Application Scenario

Benefits brought by the proposed scheme are evident especially in public cloud storage systems. Our scheme is suitable for the scenarios where data can be shared and disseminated with time conditions in a group of users. Let's consider an application scenario. Suppose that company A uses the cloud storage service, in which the proposed scheme is being utilized. Some employees in company A usually share some important time-sensitive data with different intended workmates, and these workmates can access the data stored in the cloud with sufficient authorization, but gain their disclosure privilege at different time points. Specially speaking, since the business plan of this company may contain some business secrets, executive officer shares this plan with directors to discuss and improve the business plan at an early time, while others cannot access this plan. Then, only executive director can gain disclosure privilege to disseminate the business plan in the cloud to managers of some relevant departments at a later time point, when they take responsibility for the plan execution. At last, all the directors can disseminate the business plan in the cloud to make other employees in the company to have the access privilege after specific secrecy period.

1.3 Organization

This paper is structured as follows: we introduce the related work and preliminaries in section 2 and 3, and provide the system and security model in section 4. The system definition and detailed construction are given in section 5 and 6. Then, we analyze the security and performance of our scheme in section 7. Finally, we conclude this paper in section 8.

2 Related Works

There have been numerous works on secure data group sharing and dissemination in public cloud based on various cryptographic primitives [14], such as PRE [15,16,17], broadcast encryption [18,19] and ABE [20,21]. Ulybyshev et al. [22] designed a solution that ensures privacy-preserving data sharing based on the role-based access control and cryptographic capabilities of client's browser. Popa et al. [23] proposed CryptDB based on order-preserving encryption and homomorphic encryption to guarantee data confidentiality of database in public cloud. Zhou et al. [24] proposed a secure data group sharing scheme based on IBBE algorithm, in which data owner can broadcast encrypted data to a group of users at the same time. In order to achieve data collaboration and dissemination, this scheme adopted the PRE technique to allow an authorized proxy to convert an IBBE ciphertext into an identity-based encryption (IBE) ciphertext. Hence, the intended receiver can decrypt the IBE ciphertext. However, this PRE scheme only allows the re-encryption procedure to be executed in an all-or-nothing manner, which means the proxy can either re-encrypt all the initial ciphertexts or none of them. The CPRE scheme could allow users to generate a re-encryption key associated with a condition and only the encrypted data meeting the condition can be re-encrypted [25,26]. Xu et al. [10] proposed a conditional identity-based broadcast PRE (CIBPRE) scheme to achieve secure data group dissemination in cloud email. The CIBPRE scheme adopted IBBE technique to allow a sender to encrypt a message to a group of receivers by specifying the receivers' identities, and the sender can delegate a re-encryption key to a proxy so that he can convert the initial ciphertext into a new one to a new group of intended receivers. The re-encryption key can be associated with a condition (email subject) such that only the matching ciphertexts can be re-encrypted. However, conditions used in this scheme are keywords only, which would limit the flexibility.

Zhao et al. [27] proposed a key-policy attribute-based CPRE scheme, which is the first to support fine-grained conditions more than keywords. Fang et al. [28] and Yang et al. [11] proposed ciphertext-policy attribute-based CPRE (ABCPRE) schemes which support fine-grained re-encryption conditions on public key encrypted ciphertext. To achieve many-to-many access control of data dissemination, Huang et al. [29] proposed PRECISE, which combines attribute-based CPRE with IBBE to support fine-grained re-encryption conditions on IBBE ciphertext. Liang et al. [30] proposed a secure and efficient ciphertext-policy attribute-based PRE scheme for cloud data sharing, which allows the proxy to convert a ciphertext under an access policy which is satisfied by the attributes of requestor to another ciphertext under a new access policy. However, these schemes do not support the scenario where the access privilege of data is required to be respectively released to different groups of users after different time points. Towards this challenge, Rivest et al. [13] proposed a practical TRE system, using a trusted time agent rather than data owner to uniformly release the access privilege at a specific time. Fan et al. [31] proposed a predicate encryption based secure data storage scheme called TR-CPBRE, by integrating TRE into data access control, in which the evaluator decrypts the ciphertexts that satisfy the predicate only after a specified time period. Liang et al. [32] introduced a timed-release conditional proxy broadcast re-encryption, which allows a delegator not only to delegate the keyword enforced decryption rights of an encrypted data to a set of specified recipients, but also to control when the decryption rights will be delegated. However, these schemes lack fine-grained access control.

To achieve the capacity of both fine-grained access control and time-based exposing in cloud, a direct method is to handle time factor as an attribute. Liu et al. [33] designed a three-layer time tree to express the access time associated with a data and the effective time periods associated with a user. In this scheme, only the users whose attributes satisfy the access structure and whose access rights are effective in the access time can recover corresponding data. Yang et al. [34] proposed a time-domain access control system, in which access control takes both user's attribute set and the access time into consideration. However, it introduces heavy extra overhead since the authority needs to generate update keys for all potential attributes each time to implement the time-related function, and the computational complexity increases with the amount of involved attributes. Another method is embedding the TRE mechanism into CP-ABE, Zhu et al. [35] combined three advanced cryptographic techniques, that are integer comparison, TRE and ABE to help the data owner achieve temporal data access control on files stored in cloud servers. Hong et al. [12] proposed a time and attribute factors combined access control scheme (TAFC) on time-sensitive data for public cloud, which only publishes one corresponding secret at each time to expose the related trapdoors, and satisfies the security requirements for time-sensitive data in public cloud. However, this scheme does not consider the data owner's requirement that his encrypted data may be re-encrypted for other group users after different releasing time.

3 PRELIMINARIES

3.1 Bilinear Pairing

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

1) Computability. There is an efficient algorithm to compute $e(g,h)\in\mathbb{G}_T$, for any $g,h\in\mathbb{G}$.

2) Bilinearity. For all $g,h\in\mathbb{G}$ 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 $\mathbb{G}$, then $e(g, g)$ is also a generator of $\mathbb{G}_T$.

3.2 Identity-Based Broadcast Encryption

The IBBE scheme allows data owner to encrypt a message only once for many receivers via the broadcast channel. The data owner does not hold any private information and the encryption is performed with a set of identities of the receivers, which can be seen as an extension of the IBE. The definition of IBBE is given below.

1) $Setup(1^\kappa, N)$: On input a security parameter κ and the maximal size N of a set of receivers for an encryption, this algorithm outputs a pair of public key PK and master secret key MK.

2) $KeyGen(PK, MK, ID)$: On input an identity ID, the public key PK and the master secret key MK, this key generation algorithm outputs a secret key SK for user ID.

3) $Enc(PK, U, M)$: On input a set U of identities, the public key PK and a message M, the algorithm outputs a ciphertext CT for U.

4) $Dec(PK, CT, SK, ID)$: On input the public key PK, the ciphertext CT and the secret key SK, the algorithm outputs the message M if $ID \in U$.

3.3 Ciphertext-Policy Attribute-Based Encryption

The CP-ABE is a cryptography prototype for one-to-many secure communication, in which the data owner shares data to the intended users by designating an access policy and encrypting the data under the access policy. In CP-ABE based approach, the access policy is expressed as a tree over a set of attributes and logic gates. Each user obtains the secret key from the authority based on the attributes. It consists of following algorithms [36].

1) $Setup(1^\kappa)$: The setup algorithm takes as input the security parameter κ and outputs a public key PK and a master secret key MK.

2) $KeyGen(PK, MK, S)$: The key generation algorithm takes as input the public key PK, the master secret key MK, a set S of attributes, and outputs an attribute key SK.

3) $Enc(PK, M, T)$: The encryption algorithm takes as input the public key PK, a message M and an access policy T, and outputs a ciphertext CT.

4) *Dec*(*PK*, *SK*, *CT*): The decryption algorithm takes as input the public key *PK*, an attribute key *SK*, a ciphertext *CT* with an access policy *T*. If $S \in T$, it outputs the message *M*.

3.4 Timed-Release Encryption

The TRE allows data owner to encrypt message for the purpose that intended users can decrypt it after a designated time [13,37]. It is a two-factor encryption scheme combining public key encryption (PKE) and time-dependent encryption. In order to recover message, a trusted agent is required to expose time token, which is kept confidential by the trusted agent until at an appointed time, thus even the intended user cannot get the plaintext of message before the designated releasing time.

1) *Encryption*. When encrypting message, the ciphertext is generated with the public key of the intended user and the designated releasing time *t*.

2) *Decryption*. At each time point *t*, the trusted agent releases a time token TK_t. The ciphertext holds the feature that only with corresponding user's secret key and time token TK_t, can a user correctly get the plaintext; otherwise, the user cannot conduct the decryption successfully.

4 System and Security Model

4.1 System Model

The primary goal of our scheme is to achieve fine-grained and timed-release data group dissemination. Fig. 1 shows the system model of our scheme, which consists of the following system entities.

- The central authority (CA) is a fully trusted authority running on trusted cloud platform with flexibility and scalability that manages and distributes public/secret keys in the system, including generates system parameters to initialize system and generates private keys and attribute keys with users' identity and attributes. In addition, it acts as a trusted time agent to publish time token at each pre-defined time.
- The CSP is a semi-trusted entity that has abundant storage capacity and computation power to provide data sharing services in public cloud. It is in charge of controlling the accesses from outside users to the stored data and providing corresponding services. When it receives the request of data re-encryption, it is responsible for generating a re-encrypted ciphertext with re-encryption key from data disseminator. Hence, CSP stores not only initial ciphertexts, but also re-encrypted ciphertexts.
- The data owner wishes to outsource the data into cloud for convenience of group sharing and dissemination. The data owner is in charge of encrypting data for a set of receivers. If the data owner has the requirement to limit his data to be disseminated by some specific people after some specific time, the data owner is able to define attribute-based and timed-release access policy, and enforce it on his own data by encrypting the data under the policy before outsourcing it.

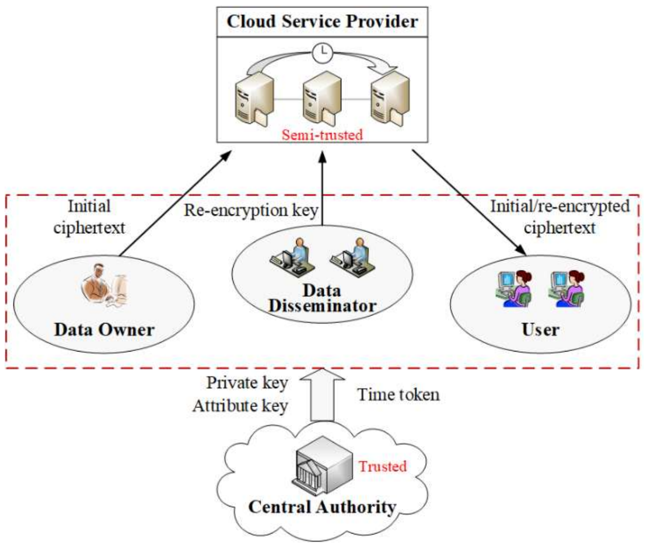

Fig. 1. System model

- The data disseminator is the person who wishes to share data owner's data with other people (e.g. his friends, family members, colleagues). For security and access control considerations, data disseminator must be one of intended receivers defined by the data owner, who could decrypt the initial ciphertexts. The data disseminator can generate re-encryption keys, and then send data re-encryption requests with these keys to the CSP to disseminate data owner's data to others. Only the attributes of data disseminator satisfy access policy and the pre-determined time arrives, data re-encryption request can be successfully executed by CSP.
- The user is the ciphertexts receiver who can access the outsourced data. The user is able to decrypt the initial and re-encrypted ciphertexts if he is the intended receiver defined by the data owners or data disseminators.

4.2 Security Model

In our scheme, we assume the CA running on the trusted cloud platform to be fully trusted, which means it would not be compromised by malicious attackers, or collude with other malicious entities. However, we assume the CSP is honest but curious, which means it executes the tasks and may collude to get unauthorized data. Specifically, security requirements cover the following aspects.

1) Data confidentiality. The unauthorized users who are not the intended receivers defined by data owner should be prevented from accessing the data. Additionally, unauthorized access from CSP which is not fully trusted, should also be prevented.

2) Re-encryption secrecy. The data disseminator whose attributes could not satisfy the access policy in ciphertexts alone, or who tries to disseminate the ciphertext before specified releasing time, should be prevented from disseminating the ciphertexts.

3) Flexible dissemination conditions. The data owner can custom fine-grained and timed-release conditions so that the data only can be disseminated by the users whose

attributes satisfy these conditions after the releasing time.

4) Collusion resistance. The unauthorized data disseminators cannot collude with each other to generate the re-encryption key, thus the re-encryption of ciphertext should not be successful.

5 System Definitions

5.1 Dissemination Conditions

1) Access policy. Let T denote an access tree, a logical representation of an access policy. Each non-leaf node x represents a threshold gate, described by its children and a threshold value. Let num_x denote the number of children of a node x, and k_x represent its threshold value. For each leaf node y, we have $k_y = 1$. Let $attr_y$ denote an attribute associated with leaf node y in the tree and $parent(z)$ represent a parent node of the node z in the tree. Each child node of the node x in the tree is labeled from 1 to num_x, and $index(x)$ returns such label associated with the node x. These index values are uniquely assigned to nodes in the access tree in an arbitrary manner. Let T_x be a subtree of T rooted at the node x.

If a set of attributes r satisfies access tree T_x, we denote it as $T_x(r) = 1$. We compute $T_x(r)$ recursively as follows. For a non-leaf node x, it evaluates $T_z(r)$ for all children z of node x. For non-leaf node x, $T_x(r)$ returns 1 if and only if at least k_x children return 1. For the leaf node y, it returns 1 if and only if $attr_y \in r$.

2) Time trapdoor. A time trapdoor is embedded in an access policy, such that the corresponding user's access permission is restricted by the status of trapdoor. We define two statuses for the time trapdoor: unexposed or exposed. A trapdoor is unexposed if intended users cannot access the corresponding secret through the trapdoor with their private keys, while it is exposed if the intended users can get the corresponding secret through this trapdoor. A trapdoor's status can be exposed with a relevant time token. In our scheme, a trapdoor is generated by the data owner when encrypting data, and a time token is generated and published by CA. The CSP transfers each trapdoor's status from unexposed to exposed after obtaining the corresponding time token.

We can turn to Fig. 2 for instance: The trapdoor TS_1 is related to a time t_1, and TS_2 is related to t_2. Users who satisfy "Planning Department $\wedge$ Senior $\wedge$ Manager" cannot get access privilege until the time token TK_2 is published.

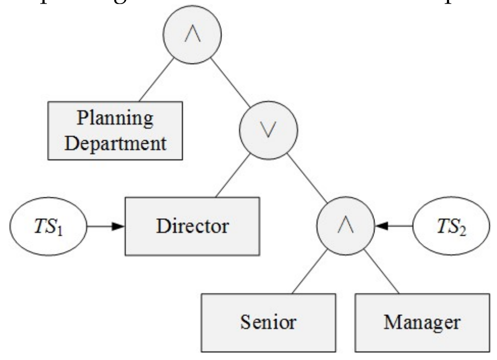

Fig. 2. Access policy

The users whose attributes are {Planning, Department, Director} satisfying "Planning Department $\wedge$ Director" can access data until TK_1 is published.

5.2 System Algorithms

Our proposed scheme consists of the following algorithms.

1) $Setup(1^\kappa, N)$: Given a security parameter κ and the maximal size of receiver set N, this algorithm outputs a system public key PK and a master secret key MK which is held by the CA and is used to generate time token and private key of user.

2) $KeyGen(PK, MK, ID, S)$: Given PK and MK, a user's identity ID and a set of attributes S, this algorithm outputs the private key of user SK, and the attribute key of user AK.

3) $Enc(PK, M, U, T, TS)$: Given PK, a message M, a set U of receivers' identities, an access policy T and time trapdoors TS embedded in T, this algorithm outputs an initial ciphertext CT.

4) $ReKeyGen(PK, ID, SK, AK, U')$: Given PK, a user's identity ID, the user's private key SK and attribute key AK, a set U' of receivers' identities, the data disseminator runs this algorithm to generate a re-encryption key RK which will be sent to CSP with the data re-encryption request.

5) $TokenGen(PK, MK, t)$: Given PK and MK, a time t, the CA runs this algorithm to output a time token TK_t which will be published publicly.

6) $ReEnc(PK, ID, RK, TK, CT)$: Given PK, a user's identity ID, a re-encryption key RK, a set TK of time token, an initial ciphertext CT, the CSP runs this algorithm to re-encrypt the initial ciphertext CT. If the attributes associated with RK satisfy access policy T in CT and the time trapdoors TS are exposed, this algorithm outputs a re-encrypted ciphertext CT'.

7) Dec-1(PK, ID, SK, CT): Given PK, a receiver's identity ID and private key SK, an initial ciphertext CT, this algorithm outputs a message M if ID is a member of the identity set in CT.

8) Dec-2(PK, ID, SK, CT'): Given PK, a receiver's identity ID and private key SK, a re-encrypted ciphertext CT', this algorithm outputs M if ID is a member of the identity set in CT'.

6 Construction

6.1 System Setup

$(PK, MK) \leftarrow Setup(1^\kappa, N)$. The CA runs $Setup$ algorithm to choose a security parameter $\kappa \in \mathbb{Z}_p$, the maximum number of receivers N, and select a bilinear map $e: \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where $\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative group with prime order p. Then CA randomly chooses $g, h, u \in \mathbb{G}$ and $\alpha, \beta, \gamma \in \mathbb{Z}_p^*$, chooses cryptographic hash functions H_1: $\{0,1\}^* \to \mathbb{Z}_p^*$, $H_2: \{0,1\}^* \to \mathbb{G}$, $H_3: \mathbb{G}_T \to \mathbb{G}$ and $H_4: \mathbb{G}_T \to \mathbb{Z}_p^*$, finally outputs a system public key $PK = (h, h^\alpha, ..., h^{\alpha^N}, h^\beta, h^\gamma, u^\beta, u^{\beta\alpha}, ..., u^{\beta\alpha^N}, e(g,h), e(g,h)^\alpha, g^\alpha, H_1, H_2, H_3, H_4)$ and a master secret key $MK=(g,u,\alpha,\beta,\gamma)$.

6.2 Key Generation

When a user registers on the CSP, the CA runs *KeyGen* algorithm to output the private key *SK* and attribute key *AK* for the user.

(*SK*, *AK*)←*KeyGen*(*PK*, *MK*, *ID*, *S*). The CA randomly picks $\lambda\in\mathbb{Z}_p$, and computes *SK* with the user's identity *ID* and *MK*.

$$SK=(K_0=g^{1/(\alpha+H_1(ID))},K_1=g^{1/(\alpha+H_1(ID))}\cdot u^{\beta\lambda/H_1(ID)},K_2=h^{\lambda})$$

For the data disseminator, the CA first chooses a random $r\in\mathbb{Z}_p$, and random $r_j\in\mathbb{Z}_p$ for each attribute $j\in S$, where *S* is the attribute set of data disseminator. Then, the CA outputs the attribute key *AK*.

$$AK=(D_0=g^{(\alpha+r)/\beta},D_0'=g^{(\alpha+r)/\beta}\cdot u^{\lambda},$$
$$\{D_j=g^rH_2(j)^{r_j},D_j'=h^{r_j}\}_{j\in S})$$

6.3 Data Encryption

In order to share data, the data owner runs *Enc* algorithm to encrypt data *M* with customized access policy *T* and time trapdoors *TS*, and then outsources the ciphertext to the CSP. For example, the access policy *T* can be expressed as "Planning Department$\wedge$(Director$\vee$(Senior$\wedge$Manager))", in which the subtree "Director" is associated with time t_1, and the subtree "Senior$\wedge$Manager" is associated with time t_2. The trapdoors are generated for these two time points.

CT←*Enc*(*PK*, *M*, *U*, *T*, *TS*). First, the data owner chooses a set U of receivers' identities (where $|U|\le N$), and a random $DK\in\mathbb{Z}_p$ which is used to encrypt the data *M* based on symmetric encryption algorithm *SE*. Then, the data owner chooses a polynomial p_x for each node *x* (including the leaves) in the access tree *T*. These polynomials are chosen in a top-down manner. For each node x in the tree, set the degree d_x of the polynomial p_x to be one less than the threshold value k_x of that node, that is $d_x=k_x-1$, and obtain three secret parameters as s_x^0, s_x^1 and s_x^{τ}, where s_x^{τ} is a time-related parameter.

Starting with the root node R, the algorithm chooses a random $s\in\mathbb{Z}_p$ and sets $p_R(0)=s_R^0=s$. Then, it chooses d_R other points of polynomial p_R randomly to define it completely. For each node x, the parameters are chosen as:

1) if x is linked to time, $s_x^{\tau}\in\mathbb{Z}_p$ and $s_x^{\tau}\cdot s_x^1=s_x^0$.

2) otherwise, $s_x^{\tau}=1$ and $s_x^1=s_x^0$.

For each non-leaf node x, it sets $p_x(0)=s_x^1$ and chooses d_x other points randomly to completely define p_x. For each x's child node y, it sets $s_y^0=p_x(index(y))$.

For each trapdoor TS_x related to a release time t_x, the data owner chooses a random $r_x^t\in\mathbb{Z}_p$ and computes $T_x=$ $(A_x=h^{r_x^t},B_x=s_x^{\tau}+H_4(e(H_2(t),h^{\gamma})^{r_x^t}))$.

Let Y be the set of leaf nodes in *T*, *Z* be the set of time trapdoors in *TS*, The data owner randomly picks $k\in\mathbb{Z}_p^*$ and outputs an initial ciphertext *CT*.

$$CT=(C_0=SE_{DK}(M),C_1=DK\cdot e(g,h)^k,$$
$$C_2=h^{k\cdot\prod_{ID_i\in U}(\alpha+H_1(ID_i))},C_3=u^{\beta(s+k\cdot\prod_{ID_i\in U}\frac{\alpha+H_1(ID_i)}{H_1(ID_i)})},$$
$$C_4=g^{-\alpha k},C_5=h^{\beta s},C_6=e(g,h)^{\alpha s},$$
$$C_7=\{C_y'=h^{s_y^1},C_y''=H_2(attr_y)^{s_y^1}\}_{y\in Y},C_8=\{A_z,B_z\}_{z\in Z})$$

6.4 Re-encryption Key Generation

When disseminating data owner's data, the data disseminator with identity *ID* runs *ReKeyGen* algorithm to generate the re-encryption key.

RK←*ReKeyGen*(*PK*, *ID*, *SK*, *AK*, *U′*). It first chooses a set U' of receivers' identities, randomly picks $k'\in\mathbb{Z}_p$, and computes the following with private key *SK*.

$$R_1=K_1,R_2=h^{k'\cdot\prod_{ID_i\in U'}(\alpha+H_1(ID_i))},R_3=H_3(e(g,h)^{k'})\cdot K_2,R_4=g^{-\alpha k'}$$

Then the data disseminator computes the following with attribute key *AK*.

$$R_5=D_0',R_6=\{R_j'=D_j,R_j''=D_j'\}_{j\in S}$$

Finally, the data disseminator outputs the re-encryption key $RK=(R_1,R_2,R_3,R_4,R_5,R_6)$.

6.5 Data Re-encryption

The CA acts as a trusted time agent to publish time token at each pre-defined time.

TK_t←*TokenGen*(*PK*, *MK*, *t*). The CA runs *TokenGen* algorithm to generate and publish a time token $TK_t=H_2(t)^{\gamma}$ at eatch time *t*.

With *RK* and time tokens, the CSP runs *ReEnc* algorithm to re-encrypt the initial ciphertext.

CT′←*ReEnc*(*PK*, *ID*, *RK*, *TK*, *CT*). First, the CSP queries all trapdoors associated to *t* from the access tree of *CT* and computes the exposed trapdoors as:

$$T_x'=B_x-H_4(e(TK_t,A_x))$$

If the procedure runs in a right manner, we have $s_x^{\tau}=T_x'$. Then the CSP runs *DecryptNode* algorithm which is a recursive algorithm. The algorithm *DecryptNode* (CT,RK,x) takes a ciphertext *CT*, the re-encryption key *RK* which is associated with a set of attributes *S*, and a node *x* from the access tree *T* as input. If the node x is a leaf node, then we let $z=attr_x$ and define as follows.

If $z\in S$ and no unexposed trapdoor is set upon *x*, then

$$DecryptNode(CT,RK,x)=\left(\frac{e(R_z',C_x')}{e(R_z'',C_x'')}\right)^{s_x^{\tau}}$$
$$=e(g,h)^{rs_x^1s_x^{\tau}}=e(g,h)^{rs_x^0}$$

If $z\notin S$ or T_x' is unexposed, then we define *DecryptNode* $(CT,RK,x)=\perp$.

When *x* is a non-leaf node, the algorithm *DecryptNode* (CT,RK,x) proceeds as follows: for all nodes n that are children of x, it calls $DecryptNode(CT,RK,n)$ and stores the output 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, then the node is not satisfied and the function returns $\perp$. Otherwise, it computes and returns the result.

$$F_x = \left(\prod_{n \in S_x} F_n^{\Delta_{j,S'_x}(0)} \right)^{s_x^r}, \text{ where } {}_{j=index(n)}^{S'_x=\{index(n):n\in S_x\}}$$
$$= e(g,h)^{rs_x^0}$$

If the access tree is satisfied by S, we set the result of entire evaluation for access tree T as A, such that $A = DecryptNode(CT,RK,R) = e(g,h)^{rs_R^0} = e(g,h)^{rs}$. Then the CSP computes

$$C'_5 = \frac{e(C_5,R_5)}{C_6 \cdot A} = \frac{e(h^{\beta s}, g^{(\alpha+r)/\beta} \cdot u^{\lambda})}{e(g,h)^{\alpha s} \cdot e(g,h)^{rs}} = e(h^{\beta s}, u^{\lambda}).$$

Then, the CSP computes

$$C'_1 = C_1 \cdot (e(C_4, h^{\Delta_\alpha(ID,U)}) \cdot e(R_1,C_2))^{\frac{-1}{\prod_{ID_i \in U \wedge ID_i \neq ID} H_1(ID_i)}},$$
$$= DK \cdot e(u^{\beta\lambda}, h^{-k})^{\prod_{ID_i \in U} \frac{\alpha+H_1(ID_i)}{H_1(ID_i)}}$$

where

$$\Delta_\alpha(ID,U)$$
$$= \alpha^{-1} \cdot (\prod_{ID_i \in U \wedge ID_i \neq ID} (\alpha + H_1(ID_i)) - \prod_{ID_i \in U \wedge ID_i \neq ID} H_1(ID_i)) \cdot$$

Finally, the CSP outputs a re-encrypted ciphertext.

$$CT' = (C'_0 = C_0, C'_1 = DK \cdot e(u^{\beta\lambda}, h^{-k})^{\prod_{ID_i \in U} \frac{\alpha+H_1(ID_i)}{H_1(ID_i)}},$$
$$C'_2 = R_2 = h^{k' \cdot \prod_{ID_i \in U'} (\alpha + H_1(ID_i))}, C'_3 = R_3 = H_3(e(g,h)^{k'}) \cdot h^{\lambda},$$
$$C'_4 = R_4 = g^{-\alpha k'}, C'_5 = e(h^{\beta s}, u^{\lambda}), C'_6 = C_3 = u^{\beta(s + k \cdot \prod_{ID_i \in U} \frac{\alpha+H_1(ID_i)}{H_1(ID_i)})})$$

6.6 Data Decryption

The user decrypts received ciphertext as follows.

1) *M*←*Dec*-1(*PK*, *ID*, *SK*, *CT*). For the initial ciphertext, the user with identity *ID* runs *Dec*-1 algorithm to decrypt. If $ID \in U$, the user computes

$$K = (e(C_4, h^{\Delta_\alpha(ID,U)}) \cdot e(K_0,C_2))^{\frac{1}{\prod_{ID_i \in U \wedge ID_i \neq ID} H_1(ID_i)}} = e(g,h)^k.$$

Then, the user computes the *DK* with *K*.

$$DK = \frac{C_1}{K} = \frac{DK \cdot e(g,h)^k}{e(g,h)^k}$$

Finally, the user recovers message *M* with *DK* using symmetric encryption algorithm.

2) *M*←*Dec*-2(*PK*, *ID*, *SK*, *CT'*). For the re-encrypted ciphertext, the user with identity ID' runs *Dec*-2 algorithm to decrypt. If $ID' \in U'$, the user computes

$$K' = (e(C'_4, h^{\Delta_\alpha(ID',U')}) \cdot e(K'_0,C'_2))^{\frac{1}{\prod_{ID_i \in U' \wedge ID_i \neq ID'} H_1(ID_i)}} = e(g,h)^{k'}.$$

Then, the user computes

$$Z = \frac{C'_3}{H_3(K')} = \frac{H_3(e(g,h)^{k'}) \cdot h^{\lambda}}{H_3(e(g,h)^{k'})} = h^{\lambda}.$$

Finally, the user computes *DK* and recovers message *M* with *DK* using symmetric encryption algorithm.

$$DK = \frac{C'_1 \cdot e(Z,C'_6)}{C'_5}$$
$$= \frac{DK \cdot e(u^{\beta\lambda}, h^{-k})^{\prod_{ID_i \in U} \frac{\alpha+H_1(ID_i)}{H_1(ID_i)}} \cdot e(h^{\lambda}, u^{\beta(s+k \cdot \prod_{ID_i \in U} \frac{\alpha+H_1(ID_i)}{H_1(ID_i)})})}{e(h^{\beta s}, u^{\lambda})}$$

7 Security and Performance Analysis

7.1 Security Analysis

We first analyze the security properties of our scheme as follows.

1) Data confidentiality. In our scheme, the data is first encrypted with a random symmetric key, and then this symmetric key is protected by identities and access tree. Since the symmetric encryption and IBBE scheme [2] are secure, the data confidentiality of the outsourced data can be guaranteed against users whose identities are not in the set of receivers' identities defined by data owner. Furthermore, the CSP cannot recover the value $A = e(g,h)^{rs}$ for data disseminators during the re-encryption phase, since the set of attributes cannot satisfy the access policy in the ciphertext or the time trapdoors are unexposed. Therefore, only the data disseminator with valid attributes that satisfy the access policy in the ciphertexts can re-encrypt the ciphertext after the releasing time.

2) Flexible data dissemination. Our scheme provides fine-grained and timed-release data dissemination which allows flexibility in specifying different access rights of individual data disseminators. In the encryption phase of our scheme, data owner is able to enforce expressive and flexible access policy to encrypt the message, then outsource the initial ciphertext to CSP. Moreover, time trapdoor can be embedded in each attribute in the access policy. Each time trapdoor's status can be transfered from unexposed to exposed only when CA publishes the corresponding time token. Therefore, the CSP can re-encrypt the ciphertexts only when the access policies in the ciphertexts are satisfied by the data disseminator's attributes and the time trapdoors are exposed. Specifically, the access policy of encrypted data defined in access tree supports complex operations including both AND and OR gate, which is able to represent any desired condition set.

3) Collusion resistance. We next show that our scheme provides collusion resistance against any collusion attacks. First, let us consider a set of unauthorized data disseminators whose attributes cannot satisfy the access policy individually. In the data encryption phase, the secret random number s must be embedded into the ciphertext. Thus, in order to re-encrypt the ciphertext, the colluders must recover $e(g,h)^{rs}$ with their attribute keys. But the attribute keys of each data disseminator are associated with the random r and r_j for each attribute, which are uniquely related to each data disseminator and make the combination of components in different attribute keys meaningless. Thus any corrupted data disseminators cannot collude to generate $e(g,h)^{rs}$ to re-encrypt ciphertext for

themselves. Second, considering another way of collusion attack, that is one data disseminator colludes his private key with another one's attribute key to generate *RK* to indiscriminately re-encrypt the ciphertext. However, both private key and attribute key of each user are associated with the same random λ which is uniquely related to each user. Specially, the data disseminator cannot forge a valid D_0' since *u* is a part of *MK*. This makes the collusion attack failed. Finally, let us consider an attack in which unauthorized data disseminators collude with a CSP. The collusion attack will not take effect since the CSP cannot get any useful key information in any phase.

We next prove that our scheme is secure against chosen plaintext attacks by describing a game among adversary $\mathcal{A}$ and challenger $\mathcal{C}$.

Initialization. The adversary $\mathcal{A}$ chooses a set U^* of challenge identities and a challenge access policy T^*. In addition, the challenger $\mathcal{C}$ prepares a table L_K to store the identities which's private keys have been queried by adversary $\mathcal{A}$.

Setup. The challenger runs the *Setup* algorithm of our scheme and gives the system public key to the adversary.

Phase 1. The adversary $\mathcal{A}$ is allowed to issue queries for a pair of private key and attribute key for an identity *ID* and a set of attributes *S*, a re-encryption key for a set of identities U', and declare an access time t_1. If $ID \in U'$, the challenger $\mathcal{C}$ responds $\perp$. The challenger $\mathcal{C}$ generates the *SK* and *AK* associated with *ID* and *S*, *RK* associated with U' and a series of time tokens that represent time points that are not later than t_1, and then gives them to the adversary $\mathcal{A}$. If $ID \in U^*$ and *S* satisfies T^* at the time point t_1, the challenger $\mathcal{C}$ responds $\perp$.

Challenge. The adversary $\mathcal{A}$ sends two challenge messages M_0 and M_1. The challenger $\mathcal{C}$ chooses a random coin $b \in \{0, 1\}$, and encrypts M_b with U^*, T^* and time trapdoors. If the time point *t* of each time trapdoor satisfies $t < t_1$, the challenger $\mathcal{C}$ generates T_x' to expose the trapdoor. Then the ciphertext is sent to the adversary $\mathcal{A}$.

Phase 2. This phase is the same with Phase 1. The adversary $\mathcal{A}$ declares a later access time t_2.

Guess. The adversary $\mathcal{A}$ outputs a guess b'.

We can find that adversary $\mathcal{A}$ can successfully attack the security of our scheme, except the re-encryption key query in the following cases:

Case 1: $ID \in U^*$, $U' \cap L_K \neq \varnothing$, $T^*(S) \neq 1$ and the releasing time has arrived.

Case 2: $ID \in U^*$, $U' \cap L_K \neq \varnothing$, $T^*(S) = 1$ and the releasing time has not arrived.

Case 3: $ID \in U^*$, $U' \cap L_K = \varnothing$, $T^*(S) \neq 1$ and the releasing time has arrived.

Case 4: $ID \in U^*$, $U' \cap L_K = \varnothing$, $T^*(S) = 1$ and the releasing time has not arrived.

Case 5: $ID \in U^*$, $U' \cap L_K = \varnothing$, $T^*(S) = 1$ and the releasing time has arrived.

In addition to these cases, the rest are those neither with satisfied attribute set, nor at the privilege releasing time, which cannot decrease the advantage of adversary $\mathcal{A}$. Let Adv_{DBDH} be the advantage to break the Decisional Bilinear Diffie-Hellman (DBDH) problem. As the proofs in [10] and [12], if the adversary $\mathcal{A}$ queries the re-encryption key *q* times and has the advantage Adv_A to break our scheme, then the advantage Adv_{IBBE} of adversary $\mathcal{A}$ to break the IBBE scheme [2] is no more than $Adv_A(1 - (2q + 1) \cdot Adv_{DBDH})$. Since the IBBE scheme [2] is secure against chosen plaintext attacks under the DBDH assumption, both Adv_{DBDH} and Adv_{IBBE} are negligible. So it demonstrates that Adv_A is also negligible. Therefore, our scheme is secure against chosen plaintext attacks if the DBDH assumption holds.

7.2 Performance Analysis

Firstly, we compare our scheme with related data access control schemes including ABCPRE [11], TAFC [12], CIBPRE [10], PRECISE [29] and TR-CPBRE [31] in terms of data confidentiality, secure data dissemination, conditional dissemination, and timed-release. The results are shown in Table 1.

Compared with ABCPRE [11] and TAFC [12] using PKE and ABE technique respectively, it can be seen that our scheme and other schemes all adopt IBBE to achieve data confidentiality and access control, in which the public key of the user can be represented by any valid string. Especially, since ABCPRE [11] cannot share a single data with a group of receivers at the same time, its scope of application may be limited. Compared with ABCPRE [11], CIBPRE [10] and PRECISE [29], a key benefit of our scheme, TAFC [12] and TR-CPBRE [31] for data sharing is time-sensitive data supporting. Unfortunately, TAFC [12] only supports timed-release data sharing, but ignores the requirement of data dissemination. Although TR-CPBRE [31] combines the CPRE with TRE to achieve time-based conditional data dissemination, it only supports simple keyword condition. Our scheme is advanced in data dissemination security as data owners could specify encrypted data to be disseminated by some certain users at different times by enforcing the fine-grained and timed-release access policy conditions.

TABLE 1
COMPARISON OF RELATED DATA ACCESS CONTROL SCHEMES

Schemes	Data confidentiality	Secure data dissemination	Conditional dissemination	Timed-release
ABCPRE [11]	PKE	Yes	Yes, access policy	No
TAFC [12]	ABE	No	No	TRE
CIBPRE [10]	IBBE	Yes	Yes, keyword	No
PRECISE [29]	IBBE	Yes	Yes, access policy	No
TR-CPBRE [31]	IBBE	Yes	Yes, keyword	TRE
Our scheme	IBBE	Yes	Yes, access policy	TRE

We then analyze the performance efficiency, including computation cost and communication cost of data encryption, re-encryption key generation, data re-encryption and initial ciphertext decryption in these schemes. Let T_p be the computation cost of a single pairing, T_e be the computation cost of an exponent operation in $\mathbb{G}$ or $\mathbb{G}_T$, N_u be the number of receivers in a ciphertext, N_c be the number of attributes in an access tree, N_t be the number of trapdoors in a ciphertext, and N_s be the number of attributes of data disseminator. We ignore the simple multiplication, hash, symmetric encryption and decryption operations.

Table 2 summarizes the computation cost of the above related schemes. It shows that our scheme is slightly more efficient than ABCPRE [11], since our scheme has the constant computation cost $3T_e$ of generating re-encryption key and has less growth rate of computation cost in data encryption phase if there is no time trapdoor in access policy (i.e. N_t=0). Further, we may notice that the computation cost of our scheme is seemingly more than that of CIBPRE [10] and PRECISE [29] in the data encryption phase, the reason is that our scheme sacrifices some computation cost to support fine-grained and timed-release access policy to control the data group dissemination. In fact, CIBPRE [10] takes a subject of the email as the input condition in data encryption phase, thus our scheme can be viewed as CIBPRE [10] when there is no time trapdoor and only one attribute in access policy. Similarly, our scheme can be viewed as PRECISE [29] if data owner does not specify time trapdoor in access policy. In these two situations, our scheme and these two schemes have almost the same computational overhead. Further, we compare our scheme with TAFC [12] and TR-CPBRE [31] which both support time-based exposing. The computation cost of initial ciphertext decryption in TAFC [12] is linear in N_c, and the computation costs of re-encryption key generation and initial ciphertext decryption in TR-CPBRE [31] are linear in N_t. However, our scheme is constant at $3T_e$ and $2T_p + T_e$ in these two phases. In addition, TR-CPBRE [31] can only support simple keyword based conditional re-encryption property, which is coarse-grained dissemination condition. Our scheme can be viewed as it when N_c is equal to 1. In this case, computation cost of TR-CPBRE [31] would increase more rapidly with N_t than our scheme in data encryption and re-encryption phase.

Table 3 compares the communication cost in terms of ciphertext size, re-encryption key size and re-encrypted ciphertext size. Here $|\mathbb{G}|$ and $|\mathbb{G}_T|$ are the bit size of an element of $\mathbb{G}$ and $\mathbb{G}_T$ respectively, $|\mathbb{Z}|$ is the bit size of an element of $\mathbb{Z}_p$. The ciphertext size implies the communication cost that data owner needs to send to CSP, and the re-encryption key size represents the communication cost that the data disseminator needs to send so that the initial ciphertext can be re-encrypted to new receivers after a specific time. The re-encrypted ciphertext size implies the communication cost that the new receivers would spend to get to decrypt.

Firstly, CIBPRE [10] only can enforce a single simple keyword condition and cannot achieve time-based exposing in data dissemination phase, so the ciphertext size and re-encryption key size in this scheme are both inevitable less than other schemes. Then, in order to compare with ABCPRE [11] and PRECISE [29] which both support fine-grained conditions, we suppose N_t is equal to 0, the ciphertext size in our scheme is $|\mathbb{G}_T|+(2N_c+5)|\mathbb{G}|$, which is obviously smaller than that in ABCPRE [11] and only one $|\mathbb{G}|$ more than PRECISE [29]. Though communication cost for re-encrypted ciphertext in ABCPRE [11] is only $2|\mathbb{G}_T|$, it cannot support one-to-many data dissemination compared with our scheme.

Compared with TAFC [12] which lacks of conditional

TABLE 2
COMPARISON OF COMPUTATION COST

Schemes	Encryption	Re-encryption key generation	Re-encryption	Initial ciphertext decryption
ABCPRE [11]	$(3N_c+2)T_e$	$(3N_s+2)T_e$	$3N_cT_p$	T_e
TAFC [12]	$N_tT_p+(2N_t+2N_c+2)T_e$	/	/	$(2N_c+1)T_p+N_cT_e$
CIBPRE [10]	$5T_e$	$6T_e$	$2T_p+3T_e$	$2T_p+T_e$
PRECISE [29]	$(2N_c+7)T_e$	$6T_e$	$(2N_c+3)T_p+2T_e$	$2T_p+T_e$
TR-CPBRE [31]	$N_tT_p+(3N_t+3)T_e$	$(2N_t+5)T_e$	$(4N_t+4)T_p$	$(2N_t+7)T_p+3T_e$
Our scheme	$N_tT_p+(2N_t+2N_c+6)T_e$	$3T_e$	$(N_t+2N_c+4)T_p+(N_c+2)T_e$	$2T_p+T_e$

TABLE 3
COMPARISON OF COMMUNICATION COST

Schemes	Ciphertext size	Re-encryption key size	Re-encrypted ciphertext size
ABCPRE [11]	$\|\mathbb{G}_T\|+3N_c\|\mathbb{G}\|$	$(2N_s+1)\|\mathbb{G}\|$	$2\|\mathbb{G}_T\|$
TAFC [12]	$\|\mathbb{G}_T\|+(2N_c+2N_t+1)\|\mathbb{G}\|+N_t\|\mathbb{Z}\|$	/	/
CIBPRE [10]	$\|\mathbb{G}_T\|+3\|\mathbb{G}\|$	$4\|\mathbb{G}\|$	$\|\mathbb{G}_T\|+4\|\mathbb{G}\|$
PRECISE [29]	$\|\mathbb{G}_T\|+(2N_c+4)\|\mathbb{G}\|$	$(2N_s+5)\|\mathbb{G}\|$	$\|\mathbb{G}_T\|+5\|\mathbb{G}\|$
TR-CPBRE [31]	$(2N_t+3)\|\mathbb{G}\|+(N_t+1)\|\mathbb{Z}\|$	$(2N_t+3)\|\mathbb{G}\|+\|\mathbb{Z}\|$	$N_t\|\mathbb{G}_T\|+(2N_t+4)\|\mathbb{G}\|+(N_t+2)\|\mathbb{Z}\|$
Our scheme	$\|\mathbb{G}_T\|+(2N_c+2N_t+5)\|\mathbb{G}\|+N_t\|\mathbb{Z}\|$	$(2N_s+5)\|\mathbb{G}\|$	$\|\mathbb{G}_T\|+4\|\mathbb{G}\|$

data dissemination, our scheme requires additional $4|\mathbb{G}|$ in data encryption. TR-CPBRE [31] regards a releasing time as an identity and a timed-release key as the secret key corresponding to the identity. Therefore, a trusted party needs to distribute time-associated secret key to each user at each time, meaning that the extra communication cost is linear to the number of users, and the size of re-encrypted ciphertext increases linearly with N_t. As opposed to TR-CPBRE [31], re-encrypted ciphertext size in our scheme is constant, and the size of re-encryption key only depends on N_s.

7.3 Experimental Evaluation

We conduct experiments on a Linux system with an Intel Core 2 Duo CPU at 2.53 GHz-processor and 4 GB memory. The experimental prototype is written in C language with the assistance of cpabe toolkit and pairing-based cryptography library [38]. We use a pairing-friendly type-A 160-bit elliptic curve group based on the supersingular curve over a 512-bit finite field. The Advanced Encryption Standard (AES) is chosen as the symmetric key encryption scheme. Then, we use randomly generated email addresses as the identities of data owner, data disseminator and user. The access policies are specified as access tree structure, and the time is system unified time format.

We first analyze the computation cost brought by supporting the function of timed-release. In our scheme, this function involves the computation complexity of two phases in our scheme, that are data encryption phase and data re-encryption phase. Table 4 summarizes the computational overheads of our scheme. In data encryption phase, we fix the number of attributes in access policy at 10 and the number of receivers at 5 (N_c = 10, N_u = 5), and vary the number of trapdoors N_t from 1 to 10. From Table 4, the computation time of data encryption is about 96 ms when N_t = 10, which is acceptable. In data re-encryption phase, we also fix the number of attributes in access policy at 10, the computation time of data re-encryption is about 34 ms when N_t = 10. More exactly, most of computational overhead is brought by fine-grained access control of data dissemination, while supporting timed-release data dissemination only brings the slight overhead, since CSP costs only one paring operation time to compute each exposed trapdoor. Further, considering data size and the number of receivers are also the influencing factors of computation cost, we then investigate the effect of these two factors on encryption time. The results are also shown in Table 4. When fixing N_c at 10 and N_t at 1, it is obvious that the computation cost of encrypting data grows very slowly with the number of the receivers. When adding 5 receivers, the average cost time is only increased by nearly 6 ms. Moreover, we fix the number of receivers at 5, while vary the size of data, and we find that the encryption time increases linearly with the scale of the data. The reason is that the data is first encrypted with a random symmetric key based on AES, and then this symmetric key is protected by identities and access tree, and most of increasing encryption time is brought by symmetric encryption algorithm of AES.

In order to give an intuitive evaluation of the performance of our scheme, we make a comparison with TAFC [12], which also supports attribute and time based access control, in terms of data encryption phase and decryption phase. Since TAFC [12] cannot support data dissemination, we make a further comparison with ABCPRE [11] and PRECISE [29] which achieve secure data dissemination based on attribute-based CPRE, to evaluate the computation cost and communication cost of re-encryption key generation phase and data re-encryption phase.

In the data encryption phase of our scheme, a data owner encrypts a file using a set of receiver identities and an access policy, and outsources the encrypted file to public cloud. To evaluate data encryption algorithm, we utilize computation time and communication size as the metric to measure complexity. Fig. 3 shows the computation time for encrypting a single data with 10 receivers versus the number of attributes in access policy. We can see that the computation cost of TAFC [12] is a little less than our scheme if N_t is equal to 3. When N_t is equal to 0, the computation cost of ABCPRE [11] grows at a faster pace than our scheme and the computation cost of PRECISE [29] is a little more than that of our scheme.

Fig. 4 compares the communication costs of data owner with TAFC [12], PRECISE [29] and ABCPRE [11]. In the data encryption phase, communication cost on the data owner side is mainly caused by ciphertext size. Viewed as a whole, ciphertext sizes are all increasing linearly with the number of attributes in access policy, and communication cost of our scheme is little more than TAFC [12]. More particularly, our scheme and TAFC [12] sacrifice some communication costs to support time-sensitive access control. When there is no time trapdoor in access policy, the communication cost of ABCPRE [11] grows faster with the increasing of the number of attributes in access policy than our scheme and TAFC [12]. Meanwhile,

TABLE 4
COMPUTATION COST OF ENCRYPTION AND RE-ENCRYPTION

Enc	N_u	5	10	15	20	25	30	35	40	45	50
	Cost (ms)	55.448	61.357	69.108	76.717	84.747	90.613	94.111	97.509	99.779	102.402
	File size	5 MB	10 MB	15 MB	20 MB	25 MB	30 MB	35 MB	40 MB	45 MB	50 MB
	Cost (ms)	59.684	70.233	79.711	100.638	116.667	128.220	142.636	156.111	169.978	184.983
	N_t	1	2	3	4	5	6	7	8	9	10
	Cost (ms)	55.670	60.031	63.438	66.549	71.849	78.718	84.455	89.042	91.792	95.910
ReEnc	N_t	1	2	3	4	5	6	7	8	9	10
	Cost (ms)	25.293	27.365	28.180	28.874	29.414	30.669	31.802	32.867	33.916	34.410

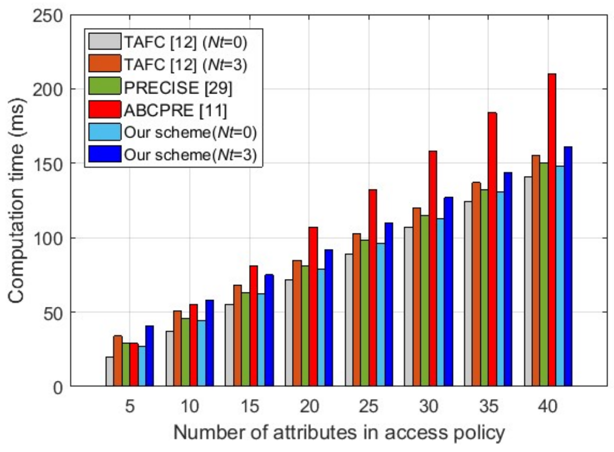

Fig. 3. Computation cost of encryption

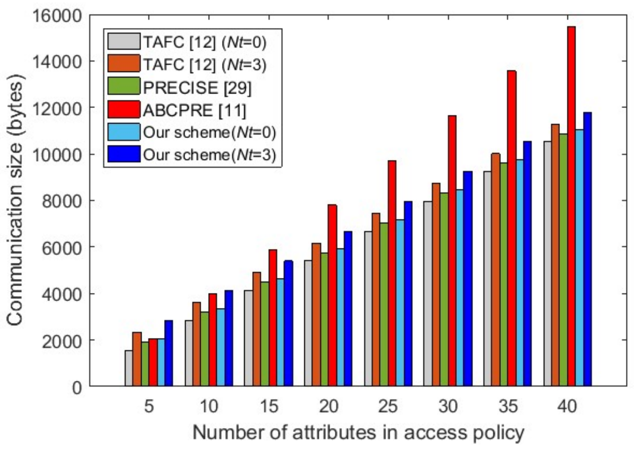

Fig. 4. Communication cost of encryption

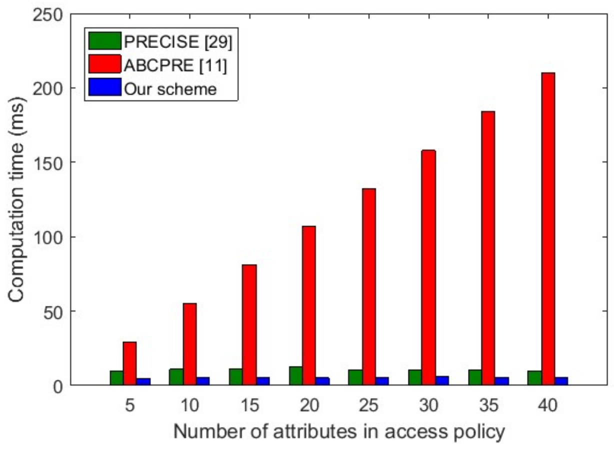

Fig. 5. Computation cost of re-encryption key generation

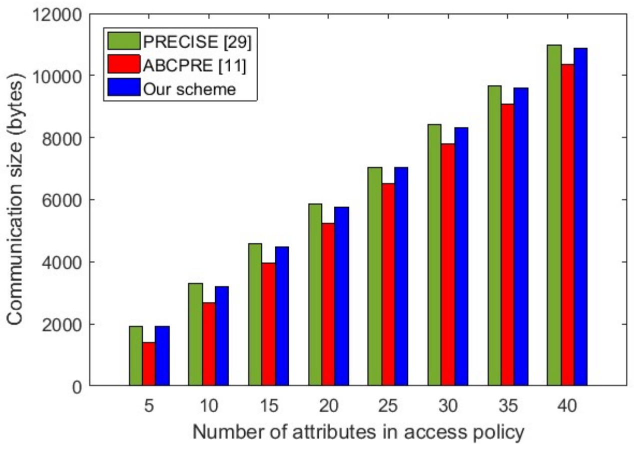

Fig. 6. Communication cost of re-encryption key generation

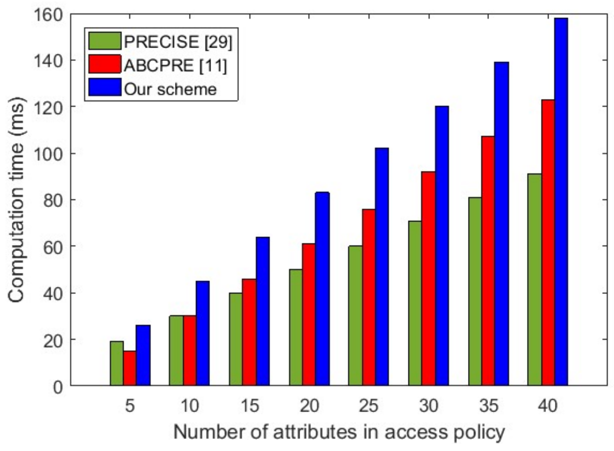

Fig. 7. Computation cost of re-encryption

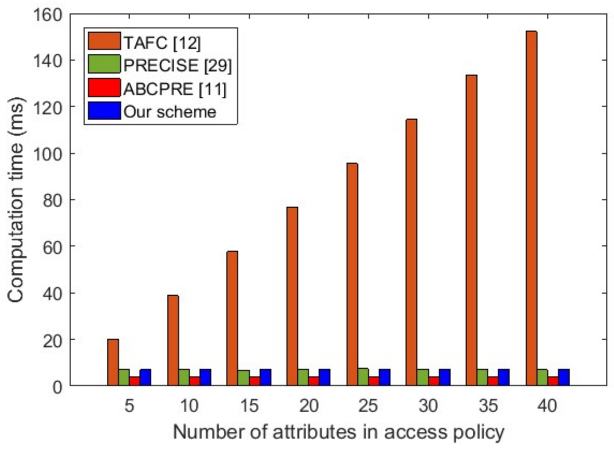

Fig. 8. Computation cost of decryption

the communication cost of our scheme is slightly more than that of PRECISE [29].

In data re-encryption phase, the data disseminator defines a set of new receivers, and then generates re-encryption key with private key. Fig. 5 and Fig. 6 describe the computation cost and communication cost in re-encryption key generation phase respectively. From Fig. 5, we can see that the computation cost of ABCPRE [11] is linearly growing with the number of data disseminator's attributes, while ours and PRECISE [29] are independence with it.

The reason is that the computation cost of data disseminator in our scheme and PRECISE [29] are only related with the number of receivers and have no relation to the number of attributes. Further, on the data disseminator side, the communication cost in this phase is given in Fig. 6. The communication cost in our scheme and PRECISE [29] are a little more than ABCPRE [11]. It costs 3419 bytes in our scheme when data disseminator transmits re-encryption key to CSP if he enforces 10 attributes. We also measure the computation cost of data re-encryption phase as shown in Fig. 7. From the experimental result, we can see the computation cost of our scheme grows faster than ABCPRE [11] and PRECISE [29] when N_t is equal to 0. The additional computational overhead of our scheme occurs by the function of fine-grained and timed-release conditional data dissemination that other two schemes cannot support simultaneously.

Fig. 8 illustrates the computation time on users by decrypting the initial ciphertext. The data decryption time of TAFC [12] is all increasing with the number of attributes in access policy, while ABCPRE [11], PRECISE [29] and our scheme remain constant on the contrary based on PKE and IBBE techniques.

Through the performance analysis on various aspects, we can conclude that our scheme can protect time-sensitive data with minor overheads on data disseminators and users, and makes a tradeoff to achieve fine-grained and timed-release access control for data group dissemination in public cloud.

8 Conclusion

In this paper, we propose a secure data group sharing and dissemination scheme in public cloud based on attribute-based and timed-release conditional identity-based broadcast PRE. Our scheme allows users to share data with a group of receivers by using identity such as email and username at one time, which would guarantee data sharing security and convenience in public cloud. Besides, with the usage of fine-grained and timed-release CPRE, our scheme allows data owners to custom access policies and time trapdoors in the ciphertext which could limit the dissemination conditions when outsourcing their data. The CSP will re-encrypt the ciphertext successfully only when the attributes of data disseminator associated with the re-encryption key satisfy access policy in the initial ciphertext and the time trapdoors in the initial ciphertext are exposed. We conduct our experiments with pairing-based cryptography library. The theoretical analysis and experiment results have shown the security and efficiency of our scheme.

Acknowledgment

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

References

[1] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69-73, 2012.

[2] C. Delerablée, "Identity-based Broadcast Encryption with Constant Size Ciphertexts and Private Keys," *Proc. the 13th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2007)*, pp. 200-215, 2007.

[3] F. Beato, S. Meul, and B. Preneel, "Practical Identity-based Private Sharing for Online Social Networks," *Computer Communications*, vol. 73, pp. 243-250, 2016.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy Attribute-based Encryption," *Proc. the 28th IEEE Symposium on Security and Privacy (S&P 2007)*, pp. 321-334, 2007.

[5] Z. Wan, J. Liu, and R. Deng, "HASBE: A Hierarchical Attribute-based Solution for Flexible and Scalable Access Control in Cloud Computing," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 743-754, 2012.

[6] 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, pp. 1614-1627, 2013.

[7] M. Blaze, G. Bleumer, and M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography," *Proc. Advances in Cryptology-EUROCRYPT 1998 (EUROCRYPT '98)*, pp.127-144, 1998.

[8] D. Tran, H. Nguyen, W. Zha, and W. Ng, "Towards Security in Sharing Data on Cloud-based Social Networks," *Proc. the 8th International Conference on Information, Communications and Signal Processing (ICICS2011)*, pp. 1-5, 2011.

[9] J. Weng, R. Deng, X. Ding, C. Chu, and J. Lai, "Conditional Proxy Re-Encryption Secure Against Chosen-ciphertext Attack," *Proc. the 4th International Symposium on ACM Symposium on Information, Computer and Communications Security (CCS 2009)*, pp. 322-332, 2009.

[10] 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 Transactions on Computers*, vol. 65, no. 1, pp. 66-79, 2016.

[11] Y. Yang, H. Lu, J. Weng, Y. Zhang, and K. Sakurai, "Fine-grained Conditional Proxy Re-encryption and Application," *Proc. the 8th International Conference on Provable Security (ProvSec 2014)*, pp. 206-222, 2014.

[12] J. Hong, K. Xue, W. Li, and Y. Xue, "TAFC: Time and Attribute Factors Combined Access Control on Time-Sensitive Data in Public Cloud," *Proc. 2015 IEEE Global Communications Conference (GLOBECOM 2015)*, pp. 1-6, 2015.

[13] R. Rivest, A. Shamir, and D. Wagner, "Time Lock Puzzles and Timed-release Crypto," *Massachusetts Institute of Technology, MA, USA*, 1996.

[14] J. Zhang, Z. Zhang, H. Guo, "Towards Secure Data Distribution Systems in Mobile Cloud Computing," *IEEE Transactions on Mobile Computing*, 2017, doi: 10.1109/TMC.2017.2687931

[15] 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*, 2016, doi: 10.1109/TSC.2016.2551238.

[16] K. Liang, M. H. Au, J. K. Liu, and W. Susilo, "A DFA-based Functional Proxy Re-Encryption Scheme for Secure Public Cloud Data Sharing," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1667-1680, 2014.

[17] M. Sepehri, S. Cimato, E. Damiani, and C. Yeuny, "Data Sharing on the Cloud: A Scalable Proxy-based Protocol for Privacy-preserving Queries," *Proc. 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2015)*, pp. 1357-1362, 2015.

[18] C. Chu, J. Weng, S. Chow, J. Zhou, and R. Deng, "Conditional Proxy Broadcast Re-encryption," *Proc. 14th Australasian Conference on Information Security and Privacy (ACISP 2009)*, pp. 327-342, 2009.

[19] W. Liu, J. Liu, and Q. Wu, "Practical Chosen-ciphertext Secure Hierarchical Identity-based Broadcast Encryption," *International Journal of Information Security*, vol. 15, no. 1, pp. 35-50, 2016.

[20] 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.

[21] J. Hur, "Improving Security and Efficiency in Attribute-Based Data Sharing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2271-2282, 2013.

[22] D. Ulybyshev, B. Bhargava, M. Villarreal-Vasquez, A. Al-salem, D. Steiner, L. Li, J. Kobes, H. Halpin, and R. Ranchal, "Privacy-Preserving Data Dissemination in Untrusted Cloud," *Proc. IEEE International Conference on Cloud Computing (CLOUD 2017)*, pp. 770-773, 2017.

[23] R. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Processing Queries on an Encrypted Database," *Communications of the ACM*, vol. 55, no. 9, pp. 103-111, 2012.

[24] Y. Zhou, H. Deng, Q. Wu, B. Qin, and J. Liu, "Identity-based Proxy Re-Encryption Version 2: Making Mobile Access Easy in Cloud," *Future Generation Computer Systems*, vol. 62, pp. 128-139, 2016.

[25] J. Shao, G. Wei, Y. Ling, and M. Xie, "Identity-based Conditional Proxy Re-encryption," *Proc. 2011 IEEE International Conference on Communications (ICC 2011)*, pp. 1-5, 2011.

[26] K. Liang, C. Chu, X. Tan, D. Wong, C. Tang, and J. Zhou, "Chosen-ciphertext Secure Multi-Hop Identity-based Conditional Proxy Re-encryption with Constant-size Ciphertexts," *Theoretical Computer Science*, vol. 539, pp. 87-105, 2014.

[27] J. Zhao, D. Feng, and Z. Zhang, "Attribute-based Conditional Proxy Re-encryption with Chosen-ciphertext Security," *Proc. 2010 IEEE Global Communications Conference (GLOBECOM 2010)*, pp. 1-6, 2010.

[28] L. Fang, W. Susilo, C. Ge, and J. Wang, "Interactive Conditional Proxy Re-encryption with Fine Grain Policy," *The Journal of Systems and Software*, vol. 84, pp. 2293-2302, 2011.

[29] Q. Huang, Y. Yang, and J. Fu, "PRECISE: Identity-based Private Data Sharing with Conditional Proxy Re-encryption in Online Social Networks," *Future Generation Computer Systems*, 2017, doi: 10.1016/j.future.2017.05.026

[30] K. Liang, M. Au, J. Liu, W. Susilo, D. 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. 2015, no. 52, pp. 95-108, 2015.

[31] C. Fan and S. Huang, "Timed-release Predicate Encryption and its Extensions in Cloud Computing," *Journal of Internet Technology*, vol. 15, no. 3, pp. 413-426, 2014.

[32] K. Liang, Q. Huang, R. Schlegel, D.S. Wong, and C. Tang, "A Conditional Proxy Broadcast Re-Encryption Scheme Supporting Timed-release," *Proc. International Conference on Information Security Practice and Experience (ISPEC 2013)*, pp. 132-146, 2013.

[33] Q. Liu, G. Wang, and J. Wu, "Time-based Proxy Re-Encryption Scheme for Secure Data Sharing in a Cloud Environment," *Information Sciences*, vol. 258, no. 3, pp. 355-370, 2014.

[34] K. Yang, Z. Liu, X. Jia, and X. Shen, "Time-domain Attribute-based Access Control for Cloud-based Video Content Sharing: A Cryptographic Approach," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 940-950, 2016.

[35] Y. Zhu, H. Hu, G. Ahn, D. Huang, and S. Wang, "Towards Temporal Access Control in Cloud Computing," *Proc. the 31st IEEE International Conference on Computer Communications (INFOCOM 2012)*, pp. 2576-2580, 2012.

[36] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," *Proc. the 29th IEEE International Conference on Computer Communications (INFOCOM 2010)*, pp. 1-9, 2010.

[37] K. Yuan, Z. Liu, C. Jia, J. Yang, and S. Lv, "Public Key Timed-release Searchable Encryption," *Proc. the 2013 Fourth International Emerging Intelligent Data and Web Technologies (EIDWT 13)*, pp. 241-248, 2013.

[38] B. Lynn, "The pairing-based cryptography library," http://crypto.stanford.edu/pbc/.

Qinlong Huang (M'17) received the Ph.D. degree from 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 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, ACM Computing Surveys, ACM Transactions on Multimedia Computing, Communications and Applications, IET Information Security, Security and Communication Networks. 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.

Jingyi Fu received the master degree from School of Computer Science, Beijing University of Posts and Telecommunications in 2015. Her research interest is cloud computing security and social network security.