






Attribute-Based Proxy Re-Encryption With Direct Revocation Mechanism for Data Sharing in Clouds

Chunpeng Ge , Member, IEEE, Willy Susilo , Fellow, IEEE, Zhe Liu , Senior Member, IEEE, Joonsang Baek , Senior Member, IEEE, Xiapu Luo , Member, IEEE, and Liming Fang , Member, IEEE

Abstract—Cloud computing, which provides adequate storage and computation capability, has been a prevalent information infrastructure. Secure data sharing is a basic demand when data was outsourced to a cloud server. Attribute-based proxy re-encryption has been a promising approach that allows secure encrypted data sharing on clouds. With attribute-based proxy re-encryption, a delegator can designate a set of shared users through issuing a re-encryption key which will be used by the cloud server to transform the delegator's encrypted data to the shared users'. However, the existing attribute-based proxy re-encryption schemes lack a mechanism of revoking users from the sharing set which is critical for data sharing systems. Therefore, in this article, we propose a concrete attribute-based proxy re-encryption with direct revocation mechanism (ABPRE-DR) for encrypted data sharing that enables the cloud server to directly revoke users from the original sharing set involved in the re-encryption key. We implemented the new schemes and evaluated its performance. The experimental results show that the proposed ABPRE-DR scheme is efficient and practical.

Index Terms—Attribute-based encryption, data sharing, cloud computing, revocation.

I. INTRODUCTION

CLOUD computing has been a dominant information infrastructure since it provides adequate storage and computation capability. With a cloud service, a user can escape from the

Manuscript received 11 January 2022; revised 23 March 2023; accepted 23 March 2023. Date of publication 11 April 2023; date of current version 14 March 2024. The work was supported in part by the Australian Research Council Discovery Projects under Grants DP200100144 and DP220100003, in part by the National Nature Foundation of China under Grants 62076125, 61672270, 61872181, in part by the Natural Science Foundation of Jiangsu Province under Grant BE2020106, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021A151012650, and in part by the Shenzhen Science and Technology Program under Grant JCYJ20210324134810028. (Corresponding authors: Willy Susilo; Liming Fang.)

Chunpeng Ge is with the Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR) & School of Software, Shandong University, Jinan 250000, China (e-mail: gechunpeng2022@163.com).

Willy Susilo and Joonsang Baek are with the Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: wsusilo@uow.edu.au; baek@uow.edu.au).

Zhe Liu is with the Zhejiang Lab., Zhejiang 311121, China (e-mail: zhe-liu@nuaa.edu.cn).

Xiapu Luo is with the The Hong Kong Polytechnic University, Hong Kong, China (e-mail: csxluo@comp.polyu.edu.hk).

Liming Fang is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China, and also with the University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: fangliming@nuaa.edu.cn).

Digital Object Identifier 10.1109/TDSC.2023.3265979

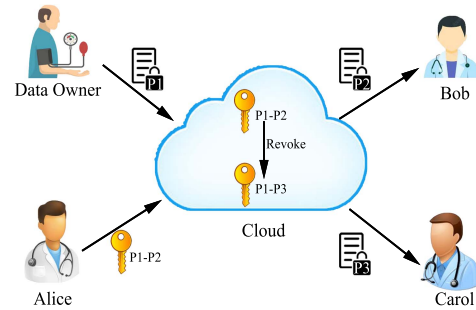


Fig. 1. Personal health record sharing system. The cloud server is able to convert a ciphertext under P_1 to a new ciphertext under P_2 with a re-encryption key $rk_{P_1 \rightarrow P_2}$. With direct revocation, the cloud server should be able to revoke some shared users directly from the key $rk_{P_1 \rightarrow P_2}$ without asking Alice to generate a new key $rk_{P_1 \rightarrow P_3}$.

burden of establishing local servers and data maintenance. However, data security and privacy issues arise since the cloud service is usually provided by third parties [1]. For instance, many female stars' private photos were disclosed when stored in the Apple cloud. To protect private data, users can encrypt them before uploading to the cloud server. Thus, only the ciphertexts are outsourced in the cloud server. Encryption is a fundamental approach to protect data confidentiality, and attribute-based encryption (ABE) [2] is a technique that can enable fine-grained access control on outsourced data. In a typical ABE setup, data owner is assigned with a list of attributes and uses the attributes to encrypt the data that will be uploaded to the cloud. The ABE operation should ensure the confidentiality of data so that the encrypted data can be only accessed by the designated users who have the access to the data. Though ABE provides fine-grained access control, it cannot support data sharing in the ciphertext form which is critical when collaboration is needed. Moreover, a revocation mechanism is a basic requirement for a data sharing in a collaboration system since many users may quite the collaboration.

We use the following personal health record system, shown in Fig. 1, to illustrate the aforementioned problem. A volunteer uploads his personal health record to an online disease research center for the scientists to conduct medical research. To prevent unauthorized users from learning his personal health data, the user encrypts his health data with an access policy P_1 : "Sydney" AND ("Research scientist" OR "Professor") AND "Brain" that indicates a brain research scientist or professor from Sydney can

access the health data, and then uploads this encrypted data to a cloud server. A recipient, named Alice, who satisfies P_1 may need to share this record with her colleagues with access policy P_2 : “Sydney” AND (“Research scientist” OR “Professor”) AND (“Brain” OR “Neuroscience”) due to collaboration needed. As the collaboration goes, Alice may only want to share health records with senior research scientists or professors. This means research scientists and professors with a junior title should be revoked from the original sharing policy P_2 . As a result, the sharing policy P_2 needs to be updated to a new sharing policy P_3 : “Sydney” AND (“Research scientist” OR “Professor”) AND (“Brain” OR “Neuroscience”) AND “Senior”.

A trivial solution to this problem is that Alice uses her private key to decrypt the data retrieved from the cloud server. After recovering the original health data through decryption, Alice encrypts the health data with the sharing policy P_2 and re-uploads the ciphertext to the cloud. Thus, anybody who satisfies the sharing policy P_2 can download and decrypt the ciphertext. Whenever a revocation occurs, all Alice needs to do is to encrypt the health record with the revoked policy (e.g., P_3). This approach, however, faces many limitations. First, it is not scalable. Alice in this example has to repeat the download-decryption-encryption process every time when the sharing policy changes. Such a process does not take the full advantage of cloud service which is supposed to do the heavy computation for the users. Moreover, during each sharing phase, the recipient Alice needs to stay online as her private key is involved in each decryption phase. Finally, local data maintenance can be cumbersome if Alice wants to share thousands of encrypted health records.

An alternative solution is to leverage the attribute-based proxy re-encryption (ABPRE) technique to address the above problems. In an ABPRE scheme, it is possible to generate a re-encryption key which can be used to transfer the ciphertext under the access policy P_1 to a different ciphertext under access policy P_2 . Later, this re-encryption key will be sent to the cloud and the cloud server can convert Alice’s ciphertext to a new one encrypted with the sharing policy. When a revocation occurs, the recipient Alice needs to generate a new re-encryption key for the remote cloud server to conduct exact transformation. Unfortunately, generating a new re-encryption key is also computation consuming which will be shown in Section IV-B. Moreover, Alice needs to be online similar to the first approach as her private key is needed for the re-encryption key generation phase. Such approach lacks the ability of revoking users from the original sharing policy directly.

A. Motivations

Although the existing ABE and ABPRE schemes can protect data confidentiality and enable data sharing for outsourced encrypted data, they do not scale well for the revocation issue. Hence, a secure attribute-based encryption scheme that ensures data confidentiality, data sharing as well as direct revocation is desired. In this paper, we aim to design an attribute-based encryption mechanism that

- 1) enables data sharing without involving Alice to conduct the decryption and then re-encryption operations;

- 2) achieves revoking shared users from the original sharing set directly;
- 3) does not introduce additional computation to Alice during the revocation phase.

Therefore, we introduce an attribute-based proxy re-encryption scheme with directly revocation scheme (ABPRE-DR) for cloud computing, whereby the cloud server can directly revoke shared users from the original sharing policy.

B. Related Work

Attribute-Based Encryption: In ABE [2], a user’s credential is viewed as an access policy or an attribute set and a ciphertext is generated with an attribute set of an access policy. A user is able to decrypt a ciphertext only when the attribute set satisfies the access policy. Generally speaking, ABE is categorized into key-policy ABE (KP-ABE) [3] and ciphertext-policy ABE (CP-ABE) [4]. In KP-ABE, a user’s private key is generated under an access policy and a ciphertext is encrypted with an attribute set. On the contrary, in the CP-ABE setting, an attribute set is used to generate a user’s secret and a ciphertext is encrypted with an access policy. As illustrated in [3], KP-ABE is suitable for the scenarios where the authority decides who are the authorized users, such as the pay-TV system [3]. CP-ABE, on the other hand fits well for the cloud sharing system as the user can appoint the access to the recipients [4]. Following their work, the work on ABE have addressed concerns on the efficiency [5], [6], [7], [8], [9], [10], [11], [12], security [13], [14], [15], [16], [17], anonymity [18], [19], [20], [21] and expressiveness [22]. At the meanwhile, Li et al. [23] proposed an attribute-based encryption scheme in which the encryption task was partially outsourced to a cloud server.

Revoking users is a vital issue in ABE scheme. Goyal et al. associated each attribute with a timestamp [3]. When a new time period comes, the authority center will issue a new private key for the unrevoked users. This approach, however, increases the burden of the authority center as it needs to issue private keys for all the unrevoked users in each time period. Moreover, the revocation in their scheme is lagging, which means that revocation can only be executed when each time period comes. Later, a new ABE [24] was claimed to support efficient revocation by assigning a unique identity to each user’s private key. By doing so the authority center knows to only issue private keys for the unrevoked users with the unique identity. At the meanwhile, Yu et al. [25] proposed a new revocation mechanism by re-encrypting the ciphertext that can only be decrypted by the unrevoked users. Following, there have been many schemes [26], [27], [28], [29], [30], [31] that deal with the revocation issues of ABE. After their work, Attrapadung et al. [32] proposed an attribute-based encryption scheme that supports unbounded dynamic predicate compositions in attributes. Unfortunately, these works cannot support the ciphertext sharing which is a basic requirement when collaboration is needed.

Attribute-Based Proxy Re-Encryption: Proxy re-encryption [33], [34] was introduced to include a proxy that can

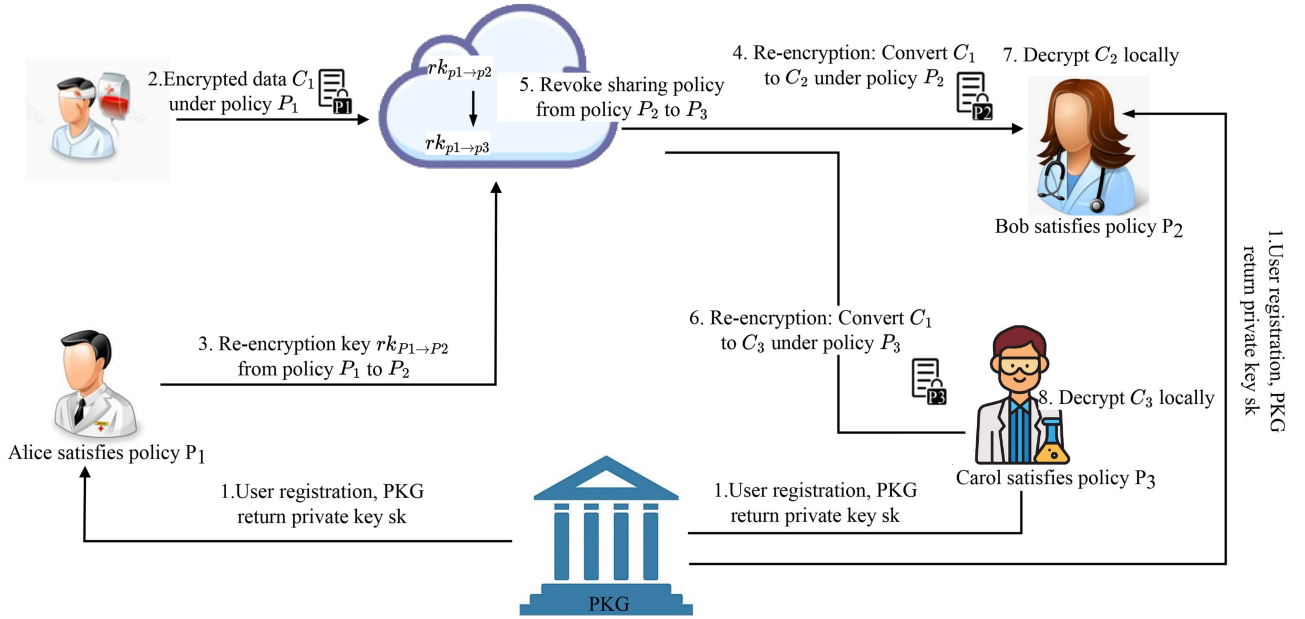


Fig. 2. System architecture.

convert a user's ciphertext to a new one of another user's with a re-encryption key. Proxy re-encryption allows a user to share its encrypted data with others without disclosing the plaintext to the proxy. Liang et al. [35] introduced this notion to the attribute-based setting and thus achieves a flexible description on the user's identity. Following their work, many ABPRE schemes were proposed to extend the expressiveness of the access policy [36], [37], [38] and enhance the security model [39], [40], [41]. Unfortunately, none of these ABPRE works consider revocation issue of sharing users which is critical for data sharing systems.

C. Our Contributions

We present an ABPRE-DR scheme. Our contributions in this paper are as follows:

- First, we formally present the definition of ABPRE-DR with the consideration on the revocation issue of attribute-based data sharing mechanism in cloud computing.
- We describe a concrete ABPRE-DR scheme and prove the confidentiality of the proposed scheme.
- We conduct an evaluation on our proposed scheme's performance to demonstrate its practicality and efficiency.

II. SYSTEM ARCHITECTURE AND DEFINITIONS

We first illustrate the system architecture and the work flow of the proposed attribute-based proxy re-encryption with direct revocation scheme in this section. Then, we describe the scheme's algorithms and security definitions afterwards. Note that we take the ciphertext-policy setting into consideration, however, our method is applicable in the key-policy setting.

A. System Architecture

A ciphertext-policy attribute-based proxy re-encryption with direct revocation (CP-ABPRE-DR) system includes the following entities (Fig. 2), the data owner, the original recipient (Alice), a cloud server and shared recipients (Bob and Carol). In addition, there is an authority center that initializes the system and issues private keys for users.

- The trusted authority (e.g., the PKG) generates and manages the security parameters and keys for the scheme, for example, the private keys for the participants.
- The data owner encrypts its personal data with an access policy P_1 and uploads the resulting ciphertext C_1 to the cloud server.
- The original recipient (Alice), who wants to share encrypted data that was originally sent to her with some other sharing users. Alice generates a re-encryption key $rk_{P_1 \rightarrow P_2}$ that enables the cloud to convert her ciphertext C_1 to shared users that satisfy P_2 .
- The cloud (e.g., AWS) stores the ciphertexts, and executes the re-encryption and revocation operations. The re-encryption algorithm converts an original ciphertext C_1 to a ciphertext C_2 under policy P_2 . The revocation operation revokes users that satisfy P_2 and not satisfy P_3 . Note that policy P_3 is more strict than P_2 .
- The shared recipients (Bob and Carol). They can use their own private key to decrypt the re-encrypted ciphertext as normal.

B. Threat Model

In our threat model, an adversary without a valid secret key may want to access the underlying data from the attribute-based ciphertext. The cloud server is semi-honest and thus it executes the protocols accordingly but is curious to obtain information as

much as possible. The authority center and the data sharer (i.e., Alice) are assumed fully trusted in our model.

C. Linear Secret Sharing Scheme

A Linear secret sharing scheme (LSSS) Π is linear over a set of parties (over Z_p) if:

- every party's share is a vector over Z_p .
- There exists an $l \times n$ matrix M , and a function π , where $\pi(j) \in \mathcal{P}$, $j \in \{1, \dots, l\}$. Let r be the secret to be shared, $r_2, \dots, r_n \in Z_p$ are random values from Z_p . Denote a column vector as $\mu = (r, r_2, \dots, r_n)$, then $M \cdot \mu$ is the vector of l shares of r according to Π . The share $(M \cdot \mu)_j$ belongs to party $\pi(j)$.

The linear reconstruction property of LSSS scheme is defined as follows. Let Att be any authorized attribute set. J is the set that $J = \{j : \pi(j) \in Att\} \subset \{1, \dots, l\}$. Then the vector $(1, 0, \dots, 0) \in Z_p^n$ is in the span of vectors $\{M_j\}$ where $j \in J$, i.e., there exist constants $\{\theta_j\}_{j \in J}$, such that $\sum_{j \in J} \theta_j \cdot M_j = (1, 0, \dots, 0)$, and $\sum_{j \in J} \theta_j M_j \cdot \mu = r$.

As illustrated in [42], an access policy $(\bar{M}, \bar{\pi})$ corresponds a boolean formula \bar{T} . We use $(\bar{M}, \bar{\pi})$ to denote the revocation access policy corresponds to \bar{T} . The revoked access policy \mathcal{T}' corresponding to (M', π') is $\mathcal{T}' = (\bar{T} \text{ AND } \bar{T})$.

D. Ciphertext-Policy Attribute-Based Proxy Re-Encryption With Direct Revocation (CP-ABPRE-DR)

Definition 1: A CP-ABPRE-DR scheme is composed of the following steps.

- *Setup*(λ, U): *Setup* is executed by the authority center to initialize the system. A security parameter λ along with the attribute universe U are the input to *Setup*. It outputs the master secret key msk and system public parameter PP .
- *KeyGen*(msk, Att): *KeyGen* is also executed by the authority party. This step outputs a secret key for the participant with attribute set Att . An attribute set Att and the master secret key msk are the input. The outputs include the generated secret key sk for Att .
- *Enc*($m, (M, \pi)$): This algorithm encrypts a message m with an access policy (M, π) and it outputs an original ciphertext CT .
- *ReKeyGen*($sk, (\bar{M}, \bar{\pi})$): This algorithm takes as input a private key sk of attribute set Att and an access policy $(\bar{M}, \bar{\pi})$, where $R(Att, (\bar{M}, \bar{\pi})) = 0$.¹ It outputs a re-encryption key rk .
- *ReEnc*(rk, CT): It takes in a re-encryption key rk and an original ciphertext CT , and outputs a re-encrypted ciphertext CT' .
- *Dec*($sk, CT/CT'$): The *Dec* algorithm uses a secret key sk for the attribute set Att to decrypt an original ciphertext CT or a re-encrypted ciphertext CT' and returns the plaintext m .
- *Revoke*($rk, (\bar{M}, \bar{\pi})$): The *Revoke* algorithm is executed by the cloud server to revoke users from the original sharing

policy by a newly generated re-encryption key. This step generates a new re-encryption key rk' that is used to transform an original ciphertext to a new ciphertext under revoked access policy (M', π') .²

In the above definition, there is the public parameter as part of the input for each algorithm and is omitted for simplicity.

E. Security Definitions

As illustrated in the threat model, we use the semantic security to describe data confidentiality. Semantic security indicates that an unauthorized user including the cloud server cannot reveal the plaintext from a ciphertext. In a CP-ABPRE-DR scheme, there two types of ciphertexts: the original ciphertext and the re-encrypted ciphertext. We use the Semantic-Or and Semantic-Re security games to describe the semantic security of the original ciphertext and the re-encrypted ciphertext, respectively. In our security model, we adopted the selective model, which means that the adversary needs to commit the challenge policy before the security game, as in [14].

Game Semantic-Or: A CP-ABPRE-DR scheme is semantic secure at original ciphertext in the selective model if the advantage of an adversary \mathcal{A} in the following game is negligible.

- **Init.** The attacker \mathcal{A} chooses (M^*, π^*) as the challenge access policy.
- **Setup.** In this step, the challenger \mathcal{C} generates public parameters and master secret key by executing the *Setup* algorithm. Then, \mathcal{C} passes public parameters to \mathcal{A} .
- **Query phase I.** \mathcal{A} queries:
 - 1) $\mathcal{O}_{sk}(Att)$: \mathcal{A} queries on attribute set Att , challenger \mathcal{C} executes $sk \leftarrow KeyGen(msk, Att)$ and returns sk to \mathcal{A} . \mathcal{O}_{sk} query indicates that the attacker maybe an inside attacker. He is a legal user and thus he obtains his own private key. His goal is to reveal other user's private data from their ciphertext.
 - 2) $\mathcal{O}_{rk}(Att, (\bar{M}, \bar{\pi}))$: \mathcal{A} makes a re-encryption key query on $(Att, (\bar{M}, \bar{\pi}))$, where $R(Att, (\bar{M}, \bar{\pi})) = 0$. \mathcal{C} executes $sk \leftarrow KeyGen(msk, Att)$ and $rk \leftarrow ReKeyGen(sk, (\bar{M}, \bar{\pi}))$, returns rk to \mathcal{A} .
 - 3) $\mathcal{O}_{re}(CT, Att, (\bar{M}, \bar{\pi}))$: \mathcal{A} makes a re-encryption query on $(CT, Att, (\bar{M}, \bar{\pi}))$, \mathcal{C} executes $sk \leftarrow KeyGen(msk, Att)$, $rk \leftarrow ReKeyGen(sk, (\bar{M}, \bar{\pi}))$ and $CT' \leftarrow ReEnc(CT, rk)$, returns CT' to \mathcal{A} .

During Query phase I, \mathcal{A} cannot make the following queries:

- 1) $\mathcal{O}_{sk}(Att)$ if $R(Att, (M^*, \pi^*)) = 1$;
- 2) $\mathcal{O}_{rk}(Att, (\bar{M}, \bar{\pi}))$ if $R(Att, (M^*, \pi^*)) = 1$ and \mathcal{A} has queried $\mathcal{O}_{sk}(Att)$ where $R(Att, (\bar{M}, \bar{\pi})) = 1$.
- **Challenge.** Two plaintext (m_0, m_1) with equal length are selected by \mathcal{A} and then \mathcal{A} sends them to the challenger \mathcal{C} . \mathcal{C} computes the challenge ciphertext $CT^* = Enc(m_\sigma, (M^*, \pi^*))$ where $\sigma \in \{0, 1\}$, and returns it to \mathcal{A} .

¹ $R(Att, (M, \pi)) = 0$ means the attribute set Att doesn't satisfy the access policy (M, π) , and $R(Att, (M, \pi)) = 1$ means Att satisfies (M, π) .

² Note that, (M', π') corresponds to an access boolean formula $\mathcal{T}' = (\bar{T} \text{ AND } \bar{T})$. In this manner, \bar{T} that corresponds to $(\bar{M}, \bar{\pi})$ was revoked from the original sharing policy \bar{T} .

- Query phase II. \mathcal{A} can queries as in Query phase I except that:
 - 1) $\mathcal{O}_{sk}(Att)$ if $R(Att, (M^*, \pi^*)) = 1$;
 - 2) $\mathcal{O}_{rk}(Att, (\bar{M}, \bar{\pi}))$ and $\mathcal{O}_{sk}(\bar{Att})$, if $R(Att, (M^*, \pi^*)) = 1$ and $R(\bar{Att}, (\bar{M}, \bar{\pi})) = 1$.
 - 3) $\mathcal{O}_{re}(CT^*, Att, (\bar{M}, \bar{\pi}))$ and $\mathcal{O}_{sk}(\bar{Att})$, if $R(Att, (M^*, \pi^*)) = 1$ and $R(\bar{Att}, (\bar{M}, \bar{\pi})) = 1$.
- Guess. \mathcal{A} outputs its guess σ' .

The adversary \mathcal{A} 's advantage to win the semantic security game is defined as

$$Adv_{\mathcal{A}}^{Sem-Or}(\lambda) = |Pr[\sigma' = \sigma] - 1/2|.$$

Game Semantic-Re: A CP-ABPRE-DR scheme is semantic secure at re-encrypted ciphertext in the selective model if the advantage of an adversary \mathcal{A} in the following game is negligible.

- Init. \mathcal{A} chooses (M^*, π^*) as the challenging access policy.
 - Setup. In this step, the challenger \mathcal{C} generates public parameters and master secret key by executing the *Setup* algorithm. Then, \mathcal{C} passes public parameters to \mathcal{A} .
 - Query phase I. \mathcal{A} queries:
 - 1) $\mathcal{O}_{sk}(Att)$: \mathcal{A} queries on attribute set Att , challenger \mathcal{C} executes $sk \leftarrow KeyGen(msk, Att)$ and returns sk to \mathcal{A} .
 - 2) $\mathcal{O}_{rk}(Att, (\bar{M}, \bar{\pi}))$: \mathcal{A} makes a re-encryption key query on $(Att, (\bar{M}, \bar{\pi}))$, where $R(Att, (\bar{M}, \bar{\pi})) = 0$. \mathcal{C} executes $sk \leftarrow KeyGen(msk, Att)$ and $rk \leftarrow ReKeyGen(sk, (\bar{M}, \bar{\pi}))$, returns rk to \mathcal{A} .
- During Query phase I, \mathcal{A} cannot make the $\mathcal{O}_{sk}(Att)$ query if $R(Att, (M^*, \pi^*)) = 1$.
- Challenge. Two plaintext (m_0, m_1) with equal length are selected by \mathcal{A} and then \mathcal{A} sends them to the challenger \mathcal{C} . \mathcal{C} computes the challenge ciphertext $CT^* = ReEnc(Enc(m_\sigma, (M, \pi)), rk)$, where $\sigma \in \{0, 1\}$, $rk = ReKeyGen(Att, (M^*, \pi^*))$, $R(Att, (M, \pi)) = 1$, and returns it to \mathcal{A} .
 - Query phase II. \mathcal{A} continues to make queries as in Query phase I except the query $\mathcal{O}_{sk}(Att)$ if $R(Att, (M^*, \pi^*)) = 1$;
 - Guess. \mathcal{A} outputs its guess σ' .

The adversary \mathcal{A} 's advantage to win the semantic security game is defined as

$$Adv_{\mathcal{A}}^{Sem-Re}(\lambda) = |Pr[\sigma' = \sigma] - 1/2|.$$

Note that, in the above two security game, the adversary can also issue a revocation query on $(rk, (\bar{M}, \bar{\pi}))$. The adversary can answer this query himself via the *Revoke*($rk, (\bar{M}, \bar{\pi})$) algorithm. Thus we omit it in the security model. Moreover, as there is no limitation on the re-encryption key query in the Semantic-Re game, anyone can answer the re-encryption query by issuing a re-encryption key query first to get the relevant re-encryption key and then answer the re-encryption query. Hence, we omit the re-encryption query in the Semantic-Re game.

Definition 1 (Semantic security). A CP-ABPRE-DR scheme is semantic secure if it is Semantic-Or secure and Semantic-Re secure.

III. PRELIMINARIES

A. Negligible Function

If for $\forall t > 0$, there exists a x_t such that $f(x) < 1/x^t$ for $\forall x > x_t$, then $f(x)$ is a negligible function.

B. Bilinear Pairing

A tuple (e, G, G_T, g, p) is a bilinear pairing if

- 1) $e(x^u, y^v) = e(x, y)^{uv}$ for all $x, y \in G$ and $u, v \in \mathbb{Z}_p^*$,
 - 2) $e(x, y) \neq I_{G_T}$, where I_{G_T} is the identical element in G_T ,
 - 3) $e(x, y)$ can be efficiently computed for all $x, y \in G$,
- where G and G_T are multiplicative cyclic groups with order p , g is a generator of group G .

C. Complex Assumption

Let (e, G, G_T, g, p) be a bilinear pairing and $a, r \in \mathbb{Z}_p, g \in G$ are randomly chosen. Given an instance $\vec{v} =$

$$g, g^r, g^a, \dots, g^{a^q}, g^{a^{q+2}}, \dots, g^{a^{2q}},$$

$$\forall j \in [1, q] \quad g^{r \cdot b_j}, g^{a/b_j}, \dots, g^{a^q/b_j}, g^{a^{q+2}/b_j}, \dots, g^{a^{2q}/b_j},$$

$$\forall j, t \in [1, q], t \neq j, \quad g^{a \cdot r \cdot b_t/b_j}, \dots, g^{a^q \cdot r \cdot b_t/b_j},$$

it is hard to distinguish $e(g, g)^{r \cdot a^{q+1}}$ from a random value $T \in G_T$. Formally, the advantage of a PPT algorithms \mathcal{B}

$$|Pr[\mathcal{B}(\vec{v}, e(g, g)^{r \cdot a^{q+1}}) = 1] - Pr[\mathcal{B}(\vec{v}, T) = 1]|.$$

is negligible. Note that, in the above equation, b_t/b_j stands for element b_t divided by b_j where $b_t, b_j \in \mathbb{Z}_p^*$.

The decisional q -parallel BDHE assumption holds if no polynomial probability time adversary can solve the above problem with a non-negligible advantage.

IV. PROPOSED CP-ABPRE-DR CONSTRUCTION

A. Challenges

Revoking sharing policy from the re-encryption key is a challenge work. Since the re-encryption key is generated with the original recipient's private key, the cloud server without the private key cannot generate a new re-encryption key directly. Moreover, if the recipient's private key is leaked to the cloud server, the cloud server can decrypt the ciphertext and thus data confidentiality cannot be protected.

To overcome this problem, we proposed the re-randomization method. In our construction, the original re-encryption key is re-randomized to a new re-encryption key that corresponds to the revoked sharing policy. Details are shown in the *Revoke* algorithm is Section IV-B.

B. Our CP-ABPRE-DR Construction

- *Setup*(λ, U): The authority center generates a bilinear pairing tuple (e, G, G_T, p) and chooses random value $\alpha, a \in \mathbb{Z}_p^*, g, f_1, \dots, f_U, Q \in G$ and a hash function $H: G_T \rightarrow \mathbb{Z}_p^*$. Sets the master secret key $msk = g^\alpha$ and public parameters $PP = (e, G, G_T, g, f_1, \dots, f_U, Q, g^a, e(g, g)^\alpha, H)$.

- *KeyGen*(*msk*, *Att*): Chooses a random value $s \in Z_p^*$, and computes
 $sk = (Att, sk_1 = g^{\alpha} g^{as}, sk_2 = g^s, \forall x \in Att, sk_x = f_x^s)$.
- *Enc*($m, (M, \pi)$): On input a message m and an access policy (M, π) , $M_{l \times n}$ is a matrix and π associates each row of M with an attribute in U . The algorithm selects a random vector $\vec{\mu} = (r, y_2, \dots, y_n) \in Z_p^{*n}$. For each row M_j of M , computes $\lambda_j = \vec{\mu} \cdot M_j, j \in [1, l]$. Randomly chooses $r_j \in Z_p$ for each $j \in [1, l]$. Then computes

$$C = m \cdot e(g, g)^{\alpha r}, \quad C_1 = g^r, \quad C_2 = Q^r,$$

$$C_{3,j} = g^{\alpha \lambda_j} f_{\pi(j)}^{-r_j}, \quad C_{4,j} = g^{r_j} \quad \forall j \in [1, l].$$

Outputs the ciphertext as $CT = ((M, \pi), C, C_1, C_2, \{C_{3,j}, C_{4,j}\}_{j \in [1, l]})$.

- *ReKeyGen*($sk, (\bar{M}, \bar{\pi})$): On input $sk = (Att, sk_1, sk_2, sk_x \ x \in Att)$ and an access policy $(\bar{M}, \bar{\pi})$, where \bar{M} is a $\bar{l} \times \bar{n}$ matrix. Randomly chooses $\chi \in G_T, \delta \in Z_p^*$ and computes

$$rk_1 = sk_1^{H(\chi)} \cdot Q^\delta, \quad rk_2 = g^\delta,$$

$$rk_3 = sk_2^{H(\chi)}, \quad rk_{4,x} = sk_x^{H(\chi)}, \quad \forall x \in Att.$$

Selects a random vector $\vec{\mu} = (\bar{r}, \bar{y}_2, \dots, \bar{y}_{\bar{n}}) \in Z_p^{\bar{n}}$. For each row \bar{M}_j of \bar{M} , computes $\bar{\lambda}_j = \vec{\mu} \cdot \bar{M}_j, j \in [1, \bar{l}]$. Randomly chooses $\bar{r}_j \in Z_p$ for each $j \in [1, \bar{l}]$. Then computes

$$rk_5 = \chi \cdot e(g, g)^{\alpha \bar{r}}, \quad rk_6 = g^{\bar{r}},$$

$$rk_{7,j} = g^{\alpha \bar{\lambda}_j} f_{\bar{\pi}(j)}^{-\bar{r}_j}, \quad rk_{8,j} = g^{\bar{r}_j} \quad \forall j \in [1, \bar{l}].$$

Output the re-encryption as key $rk = (rk_1, rk_2, rk_3, \{rk_{4,x}\}_{x \in Att}, rk_5, rk_6, \{rk_{7,j}, rk_{8,j}\}_{j \in [1, \bar{l}]}, (\bar{M}, \bar{\pi}))$.

- *ReEnc*(rk, CT): On input an original ciphertext $CT = ((M, \pi), C, C_1, C_2, \{C_{3,j}, C_{4,j}\}_{j \in [1, l]})$ and a re-encryption key $rk = (rk_1, rk_2, rk_3, \{rk_{4,x}\}_{x \in Att}, rk_5, rk_6, \{rk_{7,j}, rk_{8,j}\}_{j \in [1, \bar{l}]}, (\bar{M}, \bar{\pi}))$. If $R(Att, (M, \pi)) = 0$, outputs \perp . Otherwise, let $J = \{j : \pi(j) \in Att\} \subset \{1, \dots, l\}$, finds elements $\theta_j \in Z_p^*$, such that $\sum_{j \in J} \theta_j \cdot M_j = (1, 0, \dots, 0)$. Then, computes

$$C' = \frac{e(rk_1, C_1) \cdot e(rk_2, C_2)^{-1}}{\prod_{j \in J} (e(rk_3, C_{3,j}) \cdot e(rk_{4,\pi(j)}, C_{4,j}))^{\theta_j}}.$$

Outputs the re-encrypted ciphertext $CT' = (C, C', rk_5, rk_6, \{rk_{7,j}, rk_{8,j}\}_{j \in [1, \bar{l}]}, (\bar{M}, \bar{\pi}))$.

- *Dec*($sk, CT/CT'$): On input a private key $sk = (Att, sk_1, sk_2, sk_x \ x \in Att)$,
 1) If $CT = ((M, \pi), C, C_1, C_2, \{C_{3,j}, C_{4,j}\}_{j \in [1, l]})$ which means CT is an original ciphertext, checks whether $R(Att, (M, \pi)) = 1$. If not, outputs \perp . Otherwise, let $J = \{j : \pi(j) \in Att\} \subset \{1, \dots, l\}$, finds elements $\theta_j \in Z_p^*$, such that $\sum_{j \in J} \theta_j \cdot M_j = (1, 0, \dots, 0)$. Then, computes

$$m = C / \frac{e(sk_1, C_1)}{\prod_{j \in J} (e(sk_2, C_{3,j}) \cdot e(sk_{\pi(j)}, C_{4,j}))^{\theta_j}}. \quad (1)$$

- 2) If $CT' = (C, C', rk_5, rk_6, \{rk_{7,j}, rk_{8,j}\}_{j \in [1, \bar{l}]}, (\bar{M}, \bar{\pi}))$, which means CT' is a re-encrypted ciphertext, checks whether $R(Att, (\bar{M}, \bar{\pi})) = 1$. If not, outputs \perp . Otherwise, let $J = \{j : \bar{\pi}(j) \in Att\} \subset \{1, \dots, \bar{l}\}$, finds elements $\theta_j \in Z_p^*$, such that $\sum_{j \in J} \theta_j \cdot \bar{M}_j = (1, 0, \dots, 0)$. Then, computes

$$\chi = rk_5 / \frac{e(sk_1, rk_6)}{\prod_{j \in J} (e(sk_2, rk_{7,j}) \cdot e(sk_{\bar{\pi}(j)}, rk_{8,j}))^{\theta_j}}, \quad (2)$$

and

$$m = C / C'^{\frac{1}{H(\chi)}}.$$

- *Revoke*($rk, (\tilde{M}, \tilde{\pi})$): On input a re-encryption key $rk = (rk_1, rk_2, rk_3, \{rk_{4,x}\}_{x \in Att}, rk_5, rk_6, \{rk_{7,j}, rk_{8,j}\}_{j \in [1, \bar{l}]}, (\bar{M}, \bar{\pi}))$ and a revocation access policy $(\tilde{M}, \tilde{\pi})$, where \bar{M} and \tilde{M} are $\bar{l} \times \bar{n}$ and $\tilde{l} \times \tilde{n}$ matrixes, outputs a revoked re-encryption key for access policy (M', π') . Sets (M', π') as

$$M' = \left(\begin{array}{c|c} \bar{M} & -\mathbf{c}_1 \\ \hline \mathbf{0} & \tilde{M} \end{array} \right), \quad \pi'(j) = \begin{cases} \pi(j) & j \leq \bar{l} \\ \bar{\pi}(j - \bar{l}) & j > \bar{l} \end{cases}, \quad (3)$$

where \mathbf{c}_1 is the first column of \bar{M} . Note that M' is an $l' \times n'$ matrix, where $l' = \bar{l} + \tilde{l}, n' = \bar{n} + \tilde{n}$. Computes $rk_5' = rk_5, rk_6' = rk_6$,

$$\begin{cases} rk_{7,j}' = rk_{7,j}, & rk_{8,j}' = rk_{8,j} & j \in [1, \bar{l}] \\ rk_{7,j}' = 1_G, & rk_{8,j}' = 1_G & j \in [\bar{l} + 1, l'] \end{cases},$$

where 1_G is the identity element of group G .

Then selects a random vector $\vec{\mu}''' = (r''', y_2''', \dots, y_{n'}''') \in Z_p^{*n'}$. For each row M_j' of M' , computes $\lambda_j''' = \vec{\mu}''' \cdot M_j', j \in [1, l']$. Randomly chooses $r_j''' \in Z_p$ for each $j \in [1, l']$, and computes

$$rk_5''' = e(g, g)^{\alpha r'''}, \quad rk_6''' = g^{r'''},$$

$$rk_{7,j}''' = g^{\alpha \lambda_j'''} f_{\pi'(j)}^{-r_j'''}, \quad rk_{8,j}''' = g^{r_j'''} \quad \forall j \in [1, l'].$$

Then, computes

$$rk_5' = rk_5''' \cdot rk_5''', \quad rk_6' = rk_6''' \cdot rk_6''',$$

$$rk_{7,j}' = rk_{7,j}''' \cdot rk_{7,j}''', \quad rk_{8,j}' = rk_{8,j}''' \cdot rk_{8,j}''', \quad \forall j \in [1, l'].$$

Outputs the revoked re-encryption key as $rk' = (rk_1, rk_2, rk_3, \{rk_{4,x}\}_{x \in Att}, rk_5', rk_6', \{rk_{7,j}', rk_{8,j}'\}_{j \in [1, l']}, (M', \pi'))$.

Note that, since the elements rk_5'' and rk_6'' equal to the rk_5 and rk_6 , an adversary can distinguish whether a re-encryption key is a revoked re-encryption or an original one. To avoid this problem, we introduce random elements rk_5''' and rk_6''' to re-randomize rk_5'' and rk_6'' . Further, we use the same method to re-randomize $rk_{7,j}''$ and $rk_{8,j}''$.

Correctness. Next, we explain the correctness of the decryption algorithm and the revoked re-encryption key.

When CT is an original ciphertext, then (1) is as

$$\begin{aligned} C / \frac{e(sk_1, C_1)}{\prod_{j \in J} (e(sk_2, C_{3,j}) \cdot e(sk_{\pi(j)}, C_{4,j}))^{\theta_j}} \\ = m \cdot e(g, g)^{\alpha r} / \frac{e(g^{\alpha} g^{as}, g^r)}{\prod_{j \in J} (e(g^s, g^{a\lambda_j} f_{\pi(j)}^{-r_j}) \cdot e(f_{\pi(j)}^s, g^{r_j}))^{\theta_j}} \\ = m / \frac{e(g^{as}, g^r)}{e(g^s, g^a)^{\sum_{j \in J} \lambda_j \theta_j}} = m. \end{aligned}$$

When CT' is a re-encrypted ciphertext, then

$$\begin{aligned} C' &= \frac{e(rk_1, C_1) \cdot e(rk_2, C_2)^{-1}}{\prod_{j \in J} (e(rk_3, C_{3,j}) \cdot e(rk_{4,\pi(j)}, C_{4,j}))^{\theta_j}} \\ &= \frac{e(sk_1^{H(\chi)} \cdot Q^{\delta}, g^r) \cdot e(g^{\delta}, Q^r)^{-1}}{\prod_{j \in J} (e(sk_2^{H(\chi)}, g^{a\lambda_j} f_{\pi(j)}^{-r_j}) \cdot e(sk_{\pi(j)}^{H(\chi)}, g^{r_j}))^{\theta_j}} \\ &= e(g, g)^{\alpha r H(\chi)}. \end{aligned}$$

χ in (2) can be verified in the same way. Thus,

$$C / C'^{\frac{1}{H(\chi)}} = m \cdot e(g, g)^{\alpha r} / e(g, g)^{\alpha r H(\chi) \frac{1}{H(\chi)}} = m.$$

Now, we present the correctness of the revoked re-encryption key. We present that a revoked re-encryption key rk' generated in the *Revoke* algorithm is a valid re-encryption key for the revoked access policy (M', π') through the following two lemmas.

Lemma 1: If $(\bar{M}, \bar{\pi})$ and $(\tilde{M}, \tilde{\pi})$ described as above are valid access structures correspond to LSSS schemes, then (M', π') is also a valid access structure corresponds to an LSSS scheme.

Proof: Since $(\bar{M}, \bar{\pi})$ and $(\tilde{M}, \tilde{\pi})$ are valid access structures, there exist two vectors $\bar{\theta} = (\bar{\theta}_1, \dots, \bar{\theta}_l) \in Z_p^l$ and $\tilde{\theta} = (\tilde{\theta}_1, \dots, \tilde{\theta}_l) \in Z_p^l$ that satisfy $\sum_{j \in [1, l]} \bar{\theta}_j \cdot \bar{M}_j = (1, 0, \dots, 0) \in Z_p^{\bar{n}}$ and $\sum_{j \in [1, l]} \tilde{\theta}_j \cdot \tilde{M}_j = (1, 0, \dots, 0) \in Z_p^{\tilde{n}}$. Then, we can construct a vector $\theta' = (\theta'_1, \dots, \theta'_{l'}) = (\bar{\theta}_1, \dots, \bar{\theta}_l, \tilde{\theta}_1, \dots, \tilde{\theta}_l) \in Z_p^{l'}$ that satisfies $\sum_{j \in [1, l']} \theta'_j M'_j = (1, 0, \dots, 0)$. It can be verified as

$$\begin{aligned} \sum_{j \in [1, l']} \theta'_j M'_j \\ = \sum_{j \in [1, l]} \bar{\theta}_j M'_j + \sum_{j \in [1, l]} \tilde{\theta}_j M'_{l+j} \\ = \left(\overbrace{1, 0, \dots, 0}^{\bar{n}}, \overbrace{-1, 0, \dots, 0}^{\tilde{n}} \right) + \left(\overbrace{0, \dots, 0}^{\bar{n}}, \overbrace{1, \dots, 0}^{\tilde{n}} \right) \\ = (1, 0, \dots, 0). \end{aligned}$$

Lemma 2: If (M', π') described as in (3) is valid access structure corresponds to an LSSS scheme, then $(\bar{M}, \bar{\pi})$ and $(\tilde{M}, \tilde{\pi})$ as in (3) are valid access structures correspond to LSSS schemes.

Proof: Since (M', π') is a valid access structure, there exists a vector $\theta' = (\theta'_1, \dots, \theta'_{l'}) \in Z_p^{l'}$ that satisfies $\sum_{j \in [1, l']} \theta'_j M'_j = (1, 0, \dots, 0) \in Z_p^{n'}$. Denote c_1 as the first column of M' and

$\theta' = (\bar{\theta}, \tilde{\theta})$, where $\bar{\theta} \in Z_p^l$ and $\tilde{\theta} \in Z_p^l$.

$$\begin{aligned} \theta' \cdot M' &= (\bar{\theta}, \tilde{\theta}) \cdot \left(\begin{array}{c|c} \bar{M} & -c_1 \\ \hline \mathbf{0} & \tilde{M} \end{array} \right) \\ &= \left(\overbrace{\bar{\theta} \cdot \bar{M}}^{\bar{n}}, \overbrace{-e, 0, \dots, 0}^{\tilde{n}} \right) + \left(\overbrace{0, \dots, 0}^{\bar{n}}, \overbrace{\tilde{\theta} \cdot \tilde{M}}^{\tilde{n}} \right) \\ &= \left(\overbrace{1, 0, \dots, 0}^{\bar{n} + \tilde{n}} \right), \end{aligned}$$

where $e \in Z_p$ that equals the first element of $\bar{\theta} \cdot \bar{M}$.

Since $\bar{\theta} \cdot \bar{M} + (0, 0, \dots, 0) = (1, 0, \dots, 0) \in Z_p^{\bar{n}}$, we have $\bar{\theta} \cdot \bar{M} = (1, 0, \dots, 0) \in Z_p^{\bar{n}}$. Moreover, as $\tilde{\theta} \cdot \tilde{M} = (1, 0, \dots, 0) \in Z_p^{\tilde{n}}$ and $(-e, 0, \dots, 0) + \tilde{\theta} \cdot \tilde{M} = (0, 0, \dots, 0)$, then $e = 1$. Thus, $(\bar{M}, \bar{\pi})$ and $(\tilde{M}, \tilde{\pi})$ are valid access structures correspond to LSSS schemes.

Discussion: In our construction, the size of the matrix is linear with the number of revocations. The decryption time of the re-encrypted data and the size of the re-encryption key also increase linearly. This is because, in our scheme, we used a modular expansion method on the access structure. Since the access expands, it is hard to design other mechanism to reduce the decryption time of the re-encrypted data and the size of the re-encryption key.

C. Security Proof

Now, we prove the semantic security of our proposed CP-ABPRE-DR construction.

Theorem 1: The proposed CP-ABPRE-DR scheme is semantic secure under the q -parallel BDHE assumption.

Proof: According the definition of semantic security defined in Definition 1, we prove Theorem 1 through the following two lemmas, Lemmas 3 and 4.

Lemma 3: The proposed CP-ABPRE-DR scheme is Semantic-Or secure under the q -parallel BDHE assumption.

Proof: Suppose there exists PPT adversary \mathcal{A} that can break the Semantic-Or security of the proposed scheme with a non-negligible probability ϵ , we build a simulator \mathcal{B} that can solve the q -parallel BDHE assumption with an advantage ϵ . \mathcal{B} is given a q -parallel BDHE instance (\vec{v}, T) as illustrate in Section III-C, its goal is to decide whether T equals to $e(g, g)^{r \cdot \alpha^{q+1}}$ or T is randomly chosen from G_T .

Initially, \mathcal{B} maintains private key and re-encryption key lists which are initially empty.

- $List_{sk}$: stores a tuple of (Att, sk_{Att}) .
- $List_{rk}$ stores a tuple of $(Att, (\bar{M}, \bar{\pi}), rk, sign)$, where $sign \in \{0, 1\}$. $sign = 1$ indicates that rk is a valid re-encryption key, while $sign = 0$ indicates rk is a random one.
- Init. \mathcal{A} chooses a challenge access policy (M^*, π^*) , where M^* is a $l^* \times n^*$ matrix.
- Setup. \mathcal{B} chooses a random $\beta, \rho \in Z_p$ and sets $e(g, g)^{\alpha} = e(g^a, g^{a^q}) \cdot e(g, g)^{\beta}$, $Q = g^{\rho}$. This implicitly sets $\alpha =$

$a^{q+1} + \beta$ for some unknown α . For each $x \in U$, chooses a random value $t_x \in Z_p$. Denote $X = \{j : \pi^*(j) = x\}$, \mathcal{B} computes

$$f_x = g^{t_x} \prod_{j \in X} g^{aM_{j,1}^*/b_j} \cdot g^{a^2M_{j,2}^*/b_j} \dots g^{a^{n^*}M_{j,n}^*/b_j}.$$

If X is an empty set, then $f_x = g^{t_x}$. \mathcal{B} also chooses a hash function H . Finally, \mathcal{B} outputs the public parameter $PP = (e, G, G_T, g, f_1, \dots, f_U, Q, g^a, e(g, g)^\alpha, H)$.

• Query phase I. \mathcal{A} queries:

- 1) $\mathcal{O}_{sk}(Att)$: \mathcal{A} makes a private key query on attribute set Att . \mathcal{B} verifies that $R(Att, (M^*, \pi^*)) = 0$. If not, outputs \perp . Otherwise, \mathcal{B} first finds a vector $\vec{\theta} = (\theta_1, \dots, \theta_{n^*})$ such that $\theta_1 = -1$ and for all j where $\pi^*(j) \in Att$, then $\vec{\theta} \cdot M_j^* = 0$.³ \mathcal{B} chooses a random value $s' \in Z_p$ and computes

$$sk_2 = g^{s'} \prod_{j=1}^{n^*} \left(g^{a^{q+1-j}} \right)^{\theta_j} \triangleq g^s.$$

This implicitly sets $s = s' + \sum_{j=1}^{n^*} \theta_j \cdot a^{q+1-j}$.

Then, we have

$$\begin{aligned} sk_1 &= g^\alpha g^{as} \\ &= g^{a^{q+1} + \beta} \cdot g^{as' + \sum_{j=1}^{n^*} \theta_j \cdot a^{q+2-j}} \\ &= g^\beta (g^a)^{s'} \prod_{j=2}^{n^*} \left(g^{a^{q+2-j}} \right)^{\theta_j}, \end{aligned}$$

which can be easily computed.

For those $x \in Att$ and no j such that $\pi^*(j) = x$ exists, \mathcal{B} sets $sk_x = sk_2^{t_x}$.

For those $x \in Att$ and $X = \{j : \pi^*(j) = x\} \neq \emptyset$, as $\vec{\theta} \cdot M_j^* = 0$, then

$$\begin{aligned} sk_x &= f_x^s \\ &= \left(g^{t_x} \prod_{j \in X} \prod_{i=1}^{n^*} g^{a^i M_{j,i}^*/b_j} \right)^{s' + \sum_{k=1}^{n^*} \theta_k \cdot a^{q+1-k}} \\ &= sk_2^{t_x} \prod_{j \in X} \prod_{i=1}^{n^*} \left(g^{(a^i/b_j)/s'} \cdot \prod_{\substack{k=1 \\ k \neq i}}^{n^*} (g^{a^{q+1+i-k}/b_j})^{\theta_k} \right)^{M_{j,i}^*}, \end{aligned}$$

which can also be computed by the given parameters.

Thus, \mathcal{B} simulates the private key query and adds (Att, sk_{Att}) to $List_{sk}$.

- 2) $\mathcal{O}_{rk}(Att, (\overline{M}, \overline{\pi}))$: \mathcal{B} first checks that

- If $R(Att, (M^*, \pi^*) = 1)$ and there exists an entry (Att, sk_{Att}) , where $R(Att, (\overline{M}, \overline{\pi})) = 1$, in $List_{sk}$, outputs \perp .

- Else if $R(Att, (M^*, \pi^*) = 1)$ and no such an entry (Att, sk_{Att}) , where $R(Att, (\overline{M}, \overline{\pi})) = 1$, exists in $List_{sk}$, \mathcal{B} selects a random rk and adds $(Att, (\overline{M}, \overline{\pi}), rk, 0)$ to $List_{rk}$. Otherwise,
- \mathcal{B} first makes a query $\mathcal{O}_{sk}(Att)$ to get sk_{Att} , and then computes rk with sk_{Att} as in the *ReKeyGen* algorithm. Finally \mathcal{B} adds $(Att, (\overline{M}, \overline{\pi}), rk, 1)$ to $List_{rk}$.
- 3) $\mathcal{O}_{re}(CT, Att, (\overline{M}, \overline{\pi}))$: If $R(Att, (M^*, \pi^*)) = 1$ and exists an entry (Att, sk_{Att}) , where $R(Att, (\overline{M}, \overline{\pi})) = 1$, in $List_{sk}$, outputs \perp . Otherwise, if there exists an entry $(Att, (\overline{M}, \overline{\pi}), rk, sign)$ in $List_{rk}$, \mathcal{B} re-encrypts CT with rk . Otherwise, \mathcal{B} first makes a query $\mathcal{O}_{rk}(Att, (\overline{M}, \overline{\pi}))$ to get rk and then re-encrypts CT with rk .
- Challenge. Two plaintexts (m_0, m_1) with equal length are selected by \mathcal{A} and then \mathcal{A} sends them to \mathcal{B} . \mathcal{B} chooses a random $\sigma \in \{0, 1\}$ and computes

$$C = m_\sigma \cdot T \cdot e(g^r, g^\beta), \quad C_1 = g^r,$$

$$C_2 = (g^r)^\rho = Q^r.$$

\mathcal{B} randomly choose $y'_2, \dots, y'_{n^*} \in Z_p^{n^*-1}$ and $r_1, \dots, r'_l \in Z_p^l$. For $j \in [1, l^*]$, denote K_j as $k \neq j$ and $\pi^*(k) = \pi^*(j)$. \mathcal{B} implicitly set $\vec{\mu} = (r, ra + y'_2, ra^2 + y'_3, \dots, ra^{n^*-1} + y'_{n^*})$ and $\lambda_j = \vec{\mu} \cdot M_j^*$, $j \in [1, l^*]$.

Then, \mathcal{B} computes

$$C_{4,j} = g^{-rb_j} g^{-r'_j} \triangleq g^{r_j},$$

$$C_{3,j} = g^{a\lambda_j} f_{\pi^*(j)}^{-r_j}$$

$$= f_{\pi^*(j)}^{r'_j} \cdot (g^{r \cdot b_j})^{t_{\pi^*(j)}}$$

$$\cdot \left(\prod_{k \in K_j} \prod_{i=1}^{n^*} (g^{a^i r b_j / b_k})^{M_{k,i}^*} \right) \cdot \left(\prod_{i=2}^{n^*} (g^a)^{y'_i M_{j,i}^*} \right).$$

- Query phase II. \mathcal{A} can query as in Query phase I under the restrictions defined in the Semantic-Or security model.
- Guess. \mathcal{A} outputs its guess σ' . If $\sigma' = \sigma$, \mathcal{B} outputs 1, which indicates $T = e(g, g)^{r \cdot \alpha^{q+1}}$. Otherwise, \mathcal{B} outputs 0, which indicates that T is a random value in G_T .

Analysis. When $T = e(g, g)^{r \cdot \alpha^{q+1}}$, then $C = m_\sigma \cdot T \cdot e(g^r, g^\beta) = m \cdot e(g, g)^{r \cdot (a^{q+1} + \beta)} = m \cdot e(g, g)^{\alpha r}$, which is a perfect ciphertext. Then, $Pr[\mathcal{B}(\vec{v}, e(g, g)^{r \cdot \alpha^{q+1}}) = 1] = 1/2 + \epsilon$. When T is a random value in G_T , then $Pr[\mathcal{B}(\vec{v}, T) = 1] = 1/2$. Thus, $|Pr[\mathcal{B}(\vec{v}, e(g, g)^{r \cdot \alpha^{q+1}}) = 1] - Pr[\mathcal{B}(\vec{v}, T) = 1]| = |1/2 + \epsilon - 1/2| = \epsilon$, which means that \mathcal{B} can solve the q -parallel BDHE assumption with a non-negligible advantage ϵ .

Lemma 4: The proposed CP-ABPRE-DR scheme is Semantic-Re secure under the q -parallel BDHE assumption.

Proof: Suppose there exists PPT adversary \mathcal{A} that can break the Semantic-Or security of the proposed scheme with a non-negligible probability ϵ , we build a simulator \mathcal{B} that can solve the q -parallel BDHE assumption with an advantage ϵ . \mathcal{B} is given a q -parallel BDHE instance (\vec{v}, T) as illustrate in Section III-C,

³Note that, as illustrated in [4], if $R(Att, (M^*, \pi^*)) = 0$, such a vector $\vec{\theta}$ must exists and can be found in polynomial time.

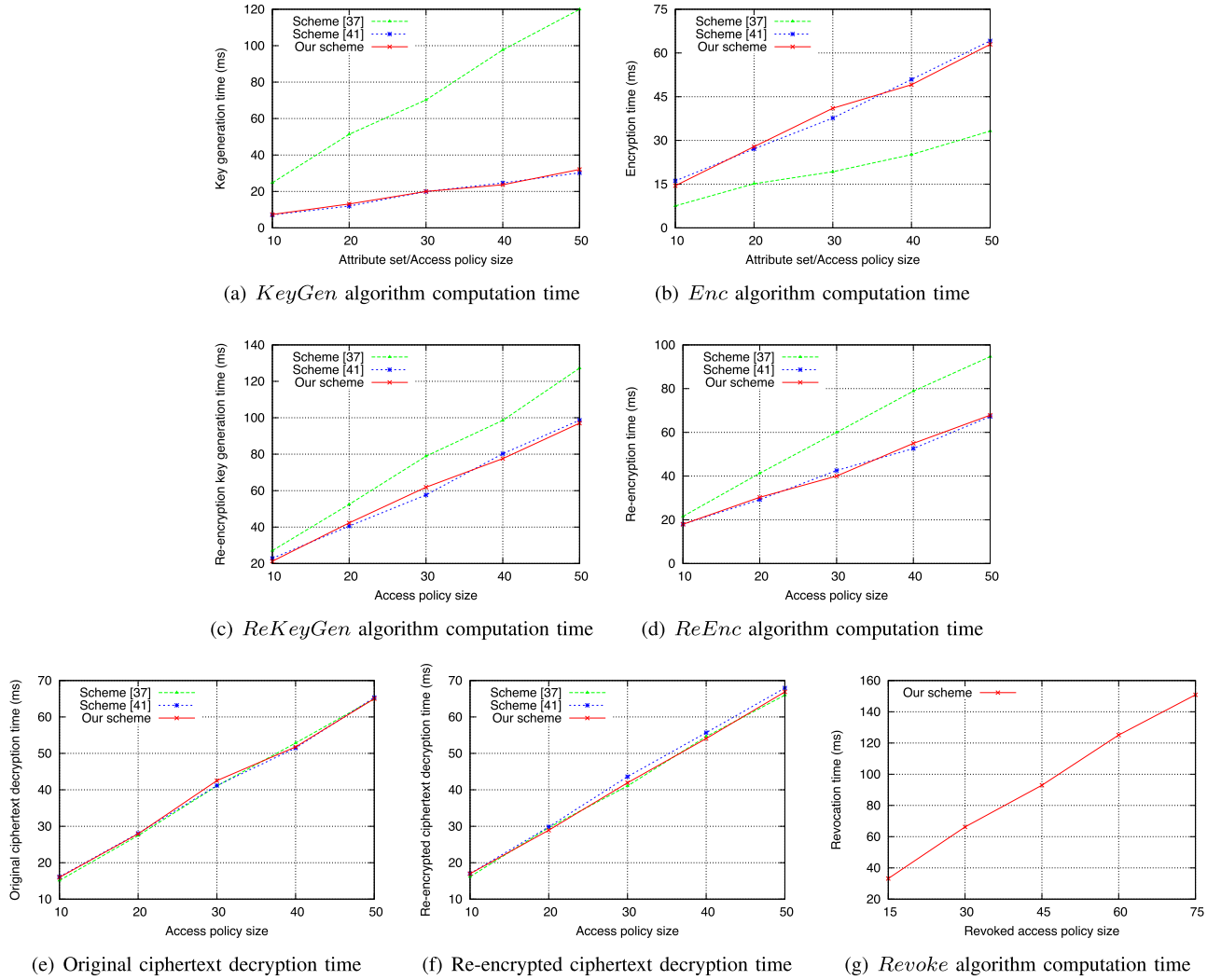


Fig. 3. Computation time of our proposed CP-ABPRE-DR scheme.

its goal is to decide whether T equals to $e(g, g)^{r \cdot \alpha^{q+1}}$ or T is randomly chosen from G_T .

The Initial, Setup and Query phase I are identical to that in the Lemma 3 proof.

- **Challenge.** Two plaintexts (m_0, m_1) with equal length are selected by \mathcal{A} and then \mathcal{A} sends them to \mathcal{B} . \mathcal{B} first choose an attribute set Att where $R(Att, (M^*, \pi^*)) = 0$, and generates a private key sk_{Att} and re-encryption key $rk = ReKeyGen(sk_{Att}, (M^*, \pi^*))$. Then \mathcal{B} chooses (M, π) where $R(Att, (M, \pi)) = 1$ and $CT = Enc(m_\sigma, (M, \pi))$ as in the Challenge phase in the proof of Lemma 3. Finally, \mathcal{B} computes $CT'^* = ReEnc(CT, rk)$, and returns CT'^* to \mathcal{A} .
- **Query phase II.** \mathcal{A} can query as in Query phase I under the restrictions defined in the Semantic-Re security model.
- **Guess.** \mathcal{A} outputs its guess σ' . If $\sigma' = \sigma$, \mathcal{B} outputs 1 that indicates $T = e(g, g)^{r \cdot \alpha^{q+1}}$. Otherwise, \mathcal{B} outputs 0 indicates that T is a random value in G_T .

Analysis. When $T = e(g, g)^{r \cdot \alpha^{q+1}}$, CT'^* is a perfect ciphertext. Then, $Pr[\mathcal{B}(\vec{v}, e(g, g)^{r \cdot \alpha^{q+1}}) = 1] = 1/2 + \epsilon$. When T is a random value in G_T , then $Pr[\mathcal{B}(\vec{v}, T) = 1] = 1/2$. Thus, we have $|Pr[\mathcal{B}(\vec{v}, e(g, g)^{r \cdot \alpha^{q+1}}) = 1] - Pr[\mathcal{B}(\vec{v}, T) = 1]| = |1/2 + \epsilon - 1/2| = \epsilon$, which means that \mathcal{B} can solve the q -parallel BDHE assumption with a non-negligible advantage ϵ .

D. Collision-Resistant and Non-Transferability

Collision Resistant: The collision resistant property ensures that the cloud server colluding with a shared recipient cannot reveal the original recipient's private key. In our scheme, the original recipient's private key sk_1 part is bound by a random element Q^δ as $sk_1^{H(\chi)} \cdot Q^\delta$. The cloud server colluding with a shared recipient can only reveal the element $H(\chi)$. They cannot recover the element Q^δ and thus cannot reveal the original recipient's private key sk_1 part. In this manner, our scheme achieves the collision resistant property.

Non-Transferability: The non-transferability implies that the cloud server colluding with a shared recipient cannot generate a new re-encryption key that can transform the ciphertext to a new shared recipient's ciphertext. In our scheme, the cloud server with a re-encryption rk colluding with a shared recipient can generate a new re-encryption key rk' that can transform the original recipient's ciphertext to a new shared recipient's ciphertext. However, as expressed in [34], the transferability is "mild" since the cloud server colluding with a recipient can always get the underlying plaintext and send it to a new shared recipient.

V. PERFORMANCE EVALUATION

Now we present the conducted evaluation of the introduced CP-ABPRE-DR scheme in the perspective of computation cost. And we list the comparison between our proposed scheme and two other attribute-based sharing schemes [37], [41] which achieve key-policy and ciphertext-policy attribute-based data sharing respectively.

To implement our scheme, we employed a Java Pairing-based cryptography package [43] which is based on the C library for the pairing-based cryptography [44]. We use the type A elliptic curve $Y^2 = X^3 + X$ and the group order is 160 b. The hash function *SHA*-256 is adopted as the hash function H of our scheme. Here is the hardware specification we used in this experiment: CPU: Intel i5-8520 U CPU @ 1.60GHZ 1.80GHZ, RAM: 8 G, and the laptop runs on Linux Mint 18.1.

Implementation: In our experiment, the universal attribute set size is set to be $|U| = 1000$. We set the attribute set in the private key generation and the access policy during the generation of the original ciphertext are sized varied from 10 to 50 with a step 10. In the re-encryption key generation phase, the sharing access policy $(\bar{M}, \bar{\pi})$ is also sized varied from 10 to 50 with a step 10. In the *Revoke* algorithm, the size of the revoking access policy $(\tilde{M}, \tilde{\pi})$ in the revocation is set from 5 to 25 with a step of 5. Thus, the size of revoked access policy varied from 15 to 75 with a step of 15. All the access policies are set as the *AND* gate of the selected attributes. Each experiment was execute 100 times to get an accurate average execution time.

Fig. 3 shows the comparison of the execution time of each algorithm in our scheme with [37], [41]. Fig. 3(a) and (b) show that the key generation time in our scheme and [41] is less than that in [37]. The encryption time in our scheme and [41] is more than that in [37]. This is because our scheme and [41] work in the ciphertext-policy setting which means the access policy is embedded in the encryption phase. While scheme [37] works in the key-policy setting and the access policy is embedded in the key generation phase. Moreover, the execution time of the *KeyGen* and *Enc* linearly increases as the size access policy grows. Fig. 3(c) and (d) show that the re-encryption key generation time and the actual encryption time in our scheme are almost the same as that in scheme [41] and less than that in scheme [37]. As shown in Fig. 3(e) and (f), the original ciphertext and re-encrypted ciphertext decryption time are almost the same in the three schemes. Since only our scheme achieves the revocation property, we implement the *Revake* algorithm

in our scheme and the execution time is shown in Fig. 3(g). It shows that the revocation time with a revoked access policy sized 75 take about only 150 ms. The implementation shows that the algorithms in the proposed CP-ABPRE-DR scheme are efficient and practical.

Note that, though our construction is designed in the ciphertext-policy setting, it can also be leveraged in the key-policy setting. When it is implemented in the key-policy setting, the private key for each user will bind with an access structure while the ciphertext will bind with an attribute set. In the key-policy setting, it also achieves the functionality of fine-grained expression and revocation. Moreover, the key-policy construction can also work in the proposed threat model and proved to be chosen ciphertext secure. Regarding efficiency, the size of the private key, ciphertext and re-encrypted ciphertext is also linear with the access policy and attribute set.

VI. CONCLUSION

In this paper, we have investigated on the revocation requirement for attribute-based data sharing systems and put forward a notion of ciphertext-policy attribute-based proxy re-encryption with direct revocation (CP-ABPRE-DR), which enables the cloud server to directly revoke a sharing policy from the original sharing policy. We presented a concrete CP-ABPRE-DR scheme and proved its semantic security. We also conducted an implementation and compared the execution time between our scheme and previous schemes to demonstrate the practicality of the proposed CP-ABPRE-DR scheme.

REFERENCES

- [1] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Comput.*, vol. 16, no. 1, pp. 69–73, Jan./Feb. 2012.
- [2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 2005, pp. 457–473.
- [3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.
- [4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, 2007, pp. 321–334.
- [5] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Proc. Int. Conf. Secur. Pract. Exp.*, Springer, 2009, pp. 13–23.
- [6] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *Proc. Int. Workshop Public Key Cryptogr.*, Springer, 2013, pp. 162–179.
- [7] N. Attrapadung, B. Libert, and E. De Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Proc. Int. Workshop Public Key Cryptogr.*, Springer, 2011, pp. 90–108.
- [8] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *Proc. Int. Workshop Public Key Cryptogr.*, Springer, 2010, pp. 19–34.
- [9] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. De Panafieu, and C. Ràfols, "Attribute-based encryption schemes with constant-size ciphertexts," *Theor. Comput. Sci.*, vol. 422, pp. 15–38, 2012.
- [10] C. Chen, Z. Zhang, and D. Feng, "Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost," in *Proc. Int. Conf. Provable Secur.*, Springer, 2011, pp. 84–101.
- [11] X. Chen, J. Li, X. Huang, J. Li, Y. Xiang, and D. S. Wong, "Secure outsourced attribute-based signatures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3285–3294, Dec. 2014.
- [12] Y. Zhang, X. Chen, J. Li, D. S. Wong, and H. Li, "Anonymous attribute-based encryption supporting efficient decryption test," in *Proc. 8th ACM SIGSAC Symp. Inf. Comput. Commun. Secur.*, 2013, pp. 511–516.

- [13] A. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in *Proc. Annu. Cryptol. Conf.*, Springer, 2012, pp. 180–198.
- [14] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," *Lecture Notes Comput. Sci.*, vol. 2008, pp. 321–334, 2011.
- [15] J. Chen and H. Wee, "Semi-adaptive attribute-based encryption and improved delegation for boolean formula," in *Proc. Int. Conf. Secur. Cryptogr. Netw.*, Springer, 2014, pp. 277–297.
- [16] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, Springer, 2010, pp. 62–91.
- [17] V. Koppula and B. Waters, "Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption," in *Proc. 39th Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, Springer, 2019, pp. 671–700.
- [18] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, Springer, 2008, pp. 111–129.
- [19] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 1, pp. 35–45, Jan. 2016.
- [20] H. Cui, R. H. Deng, G. Wu, and J. Lai, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures," in *Proc. Int. Conf. Provable Secur.*, Springer, 2016, pp. 19–38.
- [21] A. Kapadia, P. P. Tsang, and S. W. Smith, "Attribute-based publishing with hidden credentials and hidden policies," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, Citeseer, 2007, pp. 179–192.
- [22] Y. Yu, J. Shi, H. Li, Y. Li, X. Du, and M. Guizani, "Key-policy attribute-based encryption with keyword search in virtualized environments," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1242–1251, Jun. 2020.
- [23] J. Li, X. Huang, J. Li, X. Chen, and Y. Xiang, "Securely outsourcing attribute-based encryption with checkability," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2201–2210, Aug. 2014.
- [24] K. Liang, R. Lu, X. Lin, and X. S. Shen, "Ciphertext policy attribute based encryption with efficient revocation," *Tech. Rep.*, Univ. Waterloo, vol. 2, pp. 1–9, 2010.
- [25] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proc. 5th ACM Symp. Inf. Comput. Commun. Secur.*, 2010, pp. 261–270.
- [26] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 7, pp. 1214–1221, Jul. 2011.
- [27] X. Xie, H. Ma, J. Li, and X. Chen, "New ciphertext-policy attribute-based access control with efficient revocation," in *Proc. Inf. Commun. Technol.-EurAsia Conf.*, Springer, 2013, pp. 373–382.
- [28] Y. Jiang, W. Susilo, Y. Mu, and F. Guo, "Ciphertext-policy attribute-based encryption supporting access policy update and its extension with preserved attributes," *Int. J. Inf. Secur.*, vol. 17, no. 5, pp. 533–548, 2018.
- [29] J. Kim, W. Susilo, J. Baek, S. Nepal, and D. Liu, "Ciphertext-delegatable CP-ABE for a dynamic credential: A modular approach," in *Proc. Australas. Conf. Inf. Secur. Privacy*, Springer, 2019, pp. 3–20.
- [30] J. Wei, X. Chen, X. Huang, X. Hu, and W. Susilo, "RS-HABE: Revocable-storage and hierarchical attribute-based access scheme for secure sharing of e-health records in public cloud," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2301–2315, Sep./Oct. 2021.
- [31] D. Han, N. Pan, and K.-C. Li, "A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 316–327, Jan./Feb. 2022.
- [32] N. Attrapadung, "Unbounded dynamic predicate compositions in attribute-based encryption," in *Proc. 38th Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, Darmstadt, Germany, Springer, 2019, pp. 34–67.
- [33] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, Springer, 1998, pp. 127–144.
- [34] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, 2006.
- [35] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *Proc. Int. Symp. Inf. Comput. Commun. Secur.*, 2009, pp. 276–286.
- [36] K. Liang et al., "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Gener. Comput. Syst.*, vol. 52, pp. 95–108, 2015.
- [37] K. Liang and W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 9, pp. 1981–1992, Sep. 2015.
- [38] C. Ge, W. Susilo, J. Wang, Z. Huang, L. Fang, and Y. Ren, "A key-policy attribute-based proxy re-encryption without random oracles," *Comput. J.*, vol. 59, no. 7, pp. 970–982, Jul. 2016.
- [39] K. Liang, M. H. Au, W. Susilo, D. S. Wong, G. Yang, and Y. Yu, "An adaptively CCA-secure ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," in *Proc. Int. Conf. Inf. Secur. Pract. Exp.*, Springer, 2014, pp. 448–461.
- [40] C. Ge, W. Susilo, L. Fang, J. Wang, and Y. Shi, "A CCA-secure key-policy attribute-based proxy re-encryption in the adaptive corruption model for dropbox data sharing system," *Designs, Codes Cryptogr.*, vol. 86, no. 11, pp. 2587–2603, 2018.
- [41] C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and F. Liming, "Secure keyword search and data sharing mechanism for cloud computing," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 6, pp. 2787–2800, Nov./Dec. 2021.
- [42] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Proc. Annu. Cryptol. Conf.*, Springer, 2012, pp. 199–217.
- [43] Nik-U, "PBC package," 2015. [Online]. Available: <https://github.com/Nik-U/pbc>
- [44] B. Lynn et al., "PBC library," 2006. [Online]. Available: <http://crypto.stanford.edu/pbc>



Chunpeng Ge (Member, IEEE) received the PhD degree in computer science and technology from the Nanjing University of Aeronautics and Astronautics, in 2016. He was research fellow with the Singapore University of Technology and Design and the University of Wollongong. His current research interests include information security and privacy-preserving for cloud computing, blockchain, security and privacy of AI systems. He has published more than 60 papers in prestigious journal and conferences. He served as the editor of the journal of CSI and program committee for more than 30 international conferences.



Willy Susilo (Fellow, IEEE) received the PhD degree in computer science from the University of Wollongong, Australia. He is currently a distinguished professor, the head of the School of Computing and Information Technology and the director of the Institute of Cybersecurity and Cryptology, University of Wollongong. He has published more than 400 research papers in the area of cybersecurity and cryptology. His main research interests include cybersecurity, cryptography, and information security. He was a recipient of the Australian Research Council (ARC) future fellow by the ARC and the researcher of the Year Award by the University of Wollongong, in 2016. He is the editor-in-chief of the *Elsevier's Computer Standards and Interfaces* and *MDPI's Information* journals. He has served as a program committee member in dozens of international conferences. He is currently serving as an associate editor in several international journals, including *ACM Computing Survey* and the *International Journal of Information Security* (Springer). His work has been cited more than 24,000 times in Google Scholar.



Zhe Liu (Senior Member, IEEE) received the BS and MS degrees from Shandong University, China, in 2008 and 2011, respectively, and the PhD degree from the University of Luxembourg, Luxembourg, in 2015. He is a professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. His research interests include security, privacy and cryptography solutions for the Internet of Things. He has co-authored more than 80 research peer-reviewed journal and conference papers. He was a recipient of the prestigious

FNR Awards-Outstanding PhD Thesis Award, in 2016, ACM CHINA SIGSAC Rising Star Award, in 2017 as well as DAMO Academy Young Fellow, in 2019. He serves as program committee member in more than 60 international conferences, including program chairs in INSCRYPT 2019, CRYPTOIC 2019 and ACM CHINA SIGSAC 2020.



Xiapu Luo (Member, IEEE) is an associate professor with the Department of Computing, The Hong Kong Polytechnic University. His current research interests include mobile and IoT security and privacy, blockchain and smart contracts, network security and privacy, and software engineering. His work appeared in top conferences and journals, and he has received eight best paper awards, including ACM SIGSOFT Distinguished Paper Award, in ICSE' 21, Best Paper Award in INFOCOM' 18, Best Research Paper Award in ISSRE' 16, etc.



Joonsang Baek (Senior Member, IEEE) received the PhD degree from Monash University, Australia, in 2004. He is a senior lecturer with the School of Computer Science and Information Technology and a member of the Institute of Cybersecurity and Cryptology, University of Wollongong (UOW), Australia. He was a research scientist with the Institute for Infocomm Research, Singapore, and an professor with the Khalifa University of Science and Technology, United Arab Emirates. His PhD thesis was on security analysis of signcryption, and has received

great attention from the research community. He has published his work in numerous reputable journals and conference proceedings. His current research interests are in the field of applied cryptography and cybersecurity. He has also served as a program committee member and the chair for a number of renowned conferences on information security and cryptography.



Liming Fang (Member, IEEE) received the PhD degree in computer science from the Nanjing University of Aeronautics and Astronautics, in 2012, and has been a postdoctor in the information security from the City University of Hong Kong. He is the associate professor with the School of Computer Science, Nanjing University of Aeronautics and Astronautics. Now, he is a visiting scholar of the Department of Electrical and Computer Engineering New Jersey Institute of Technology. His current research interests include cryptography and information security. His recent

work has focused on the topics of public key encryption with keyword search, proxy re-encryption, identity-based encryption.