Graphical Abstract

Hybrid Proxy Re-encryption Scheme for Data Sharing in the Cloud

Xinyu Feng, Qingni Shen, Cong Li, Jisheng Dong, Yuejian Fang, Zhonghai Wu

Highlights

Hybrid Proxy Re-encryption Scheme for Data Sharing in the Cloud

Xinyu Feng, Qingni Shen, Cong Li, Jisheng Dong, Yuejian Fang, Zhonghai Wu

- New cryptographic primitive of hybrid proxy re-encryption.
- Practical data sharing scheme for cloud environment.
- Perfect fit for scenarios of data sharing from an individual to multiple people.
- More complete security model and security proof for CPA security.

Hybrid Proxy Re-encryption Scheme for Data Sharing in the Cloud

Xinyu Feng[a,c,d], Qingni Shen[a,c,d,*], Cong Li[b,c,d], Jisheng Dong[a,c,d], Yuejian Fang[a,c,d,*], Zhonghai Wu[a,b,c,d]

[a] *School of Software and Microelectronics, Peking University, Beijing, 102600, China*
[b] *School of Computer Science, Peking University, Beijing, 100871, China*
[c] *National Engineering Research Center for Software Engineering, Peking University, Beijing, 100871, China*
[d] *PKU-OCTA Laboratory for Blockchain and Privacy Computing, Peking University, Beijing, 100871, China*

Abstract

Due to the rapid development of cloud computing, massive personal data is stored and shared in the cloud. To ensure data security becomes a key issue for cloud computing. For the scenarios of data sharing from individual to multiple people, there has been a series of existing solutions, such as attribute-based proxy re-encryption (AB-PRE) and identity-based broadcast proxy re-encryption (IB-BPRE). However, these solutions either have high performance overhead or cannot achieve flexible data sharing. We propose Hybrid Proxy Re-encryption (HyPRE) to better cope with the above scenarios. Our scheme allows a semi-trusted proxy to transfer a ciphertext of identity-based encryption (IBE) to a ciphertext of attribute-based encryption (ABE) without revealing the underlying plaintext. Therefore, a data owner can encrypt data for an individual through IBE, then the receiver can further transform the ciphertext to a new ciphertext of ABE to share it to multiple receivers without decrypting it. Besides, we define the honest

This work was supported by the National Key R&D Program of China under Grant No. 2022YFB2703301 and the National Natural Science Foundation of China (No. 61672062, No. 61232005).

*Corresponding author

Email addresses: xyf@pku.edu.cn (Xinyu Feng), qingnishen@ss.pku.edu.cn (Qingni Shen), li.cong@pku.edu.cn (Cong Li), dongjisheng@pku.edu.cn (Jisheng Dong), fangyj@ss.pku.edu.cn (Yuejian Fang), wuzh@pku.edu.cn (Zhonghai Wu)

re-encryption attacks (HRA) security for our scheme to improve the incompleteness of the security under chosen plaintext attacks (CPA) in traditional proxy re-encryption (PRE) schemes. Experimental results demonstrate the practicality and efficiency of our HyPRE scheme.

Keywords: Hybrid Proxy Re-encryption, Attribute-based Encryption, HRA Security, Secure Data Sharing, Cloud Security

1. Introduction

With the rapid development of cloud computing and cloud storage technology, there are now various cloud platforms available to users for uploading, storing, and sharing data. These platforms, such as Dropbox and OneDrive, offer cloud storage services through the Internet. However, since users' private data is entrusted to third-party cloud platforms, ensuring the security and privacy of the data becomes crucial. In most data sharing scenarios, encryption serves as the fundamental step to guarantee data security. For instance, individual users can encrypt their data with their own identities and then upload the encrypted data to the cloud for storage or backup. When they wish to share the data with others, they can authorize the cloud platform to grant decryption permissions to the intended recipients.

There are several challenges for data storage and sharing in the cloud. Firstly, the data owner often lacks knowledge of the intended recipients during the data storage process. Therefore, the data is often encrypted using the data owner's identity (or public key) to ensure that only the data owner can decrypt the ciphertext. When sharing the data, however, the ciphertext must be transformed into a new ciphertext suitable for the specific target recipients. Since there may exist diverse categories of recipients, it becomes crucial to support fine-grained access control for the ciphertext. Various approaches exist for secure data storage and sharing. Identity-based encryption (IBE) allows a data owner to encrypt data using the recipient's identity while attribute-based encryption (ABE) enables data encryption based on access policies that encompass multiple individuals with distinct attributes. However, traditional IBE and ABE lack support for re-authorization of ciphertexts, making them unsuitable for data sharing within cloud storage services. On the other hand, Proxy re-encryption (PRE) facilitates the conversion of ciphertexts with different access rights. Extensions of PRE, such as identity-based proxy re-encryption (IB-PRE), attribute-based proxy re-encryption (AB-PRE), and Identity-based proxy re-encryption (IB-BPRE) enable the transformation of ciphertexts derived from IBE, ABE, or IBE to IBBE, respectively. Nonetheless, IB-PRE and IB-BPRE solely support re-encryption for an individual or a set of identities. AB-PRE necessitates the data owner's awareness of the target access policy during the initial encryption process before uploading the ciphertext. Additionally, employing ABE to encrypt data for an individual becomes redundant as it introduces more parameters, resulting in larger ciphertext and key sizes, along with increased computational costs.

From the perspective of provable security, schemes based on chosen ciphertext attack (CCA) security provide the most comprehensive security scenarios, but often come with significant performance compromises. Conversely, schemes based on chosen plaintext attack (CPA) security generally offer better performance, albeit with a specific degree of security loss. Traditional CPA secure PRE schemes allow the adversary to decrypt the corresponding plaintext by obtaining the re-encrypted ciphertext and the corresponding private key. However, the adversary is not allowed to access the re-encrypted ciphertexts and corresponding private keys simultaneously in a CPA security game for PRE due to the semi-trusted nature of the proxy. Thus, if the adversary obtains any of the re-encrypted ciphertexts, traditional CPA security for PRE cannot guarantee anything about the target PRE scheme. To address this limitation, the idea of honest re-encryption attacks (HRA) was proposed to improve the CPA security for traditional PRE [1]. HRA security divides the re-encrypted ciphertexts into corrupted and uncorrupted ones, with the adversary granted access only to the corrupted ciphertexts. However, the HRA security model is currently only defined for traditional PRE schemes, with no such definition available for Hybrid PRE schemes. Consequently, our proposed HyPRE scheme aims to strike a balance between security and performance by establishing an HRA-secure model.

1.1. Contribution

In this paper, we are motivated to propose a novel cryptographic primitive for efficient data storage and sharing in the cloud. The primary gap in facilitating secure data storage and sharing within the cloud lies in developing an efficient cryptographic primitive that enables data encryption for storage and supports the conversion from the original ciphertext to another ciphertext that allows for fine-grained access control. Thus, there is a need

Table 1: Comparison with related works

Schemes	Fine-grained sharing	Cross Domain	Security Model
[2]	×	×	CCA
[3]	×	×	CPA
[4]	×	✓	CPA
[5]	✓	×	CPA
[6]	×	✓	CPA
Ours	✓	✓	HRA

to address this gap in cloud data security by efficiently combining data encryption and access control mechanisms. Our contributions are summarized as follows. Table 1 shows the comparison of our HyPRE scheme with other related works. Our contributions can be summarized as follows.

1. We propose a novel cryptographic primitive called hybrid proxy re-encryption (HyPRE). Our HyPRE scheme is the first one to realize the transformation from a ciphertext under an identity to a new ciphertext under an access policy. Additionally, our HyPRE scheme supports a more expressive access structure and can efficiently handle a large attribute universe.

2. We propose a new definition of HRA security for HyPRE. We expand the adversary's ability by introducing the concept of HRA security and defining the honest and corrupted parties based on the challenging identity (ID) and access structure ($\mathbb{A}$) separately. Moreover, it is critical to prove the indistinguishability of re-encryption keys in the security proof of hybrid PRE schemes. However, in [6], the authors didn't give a formal proof for this point. To address these problems, in this work, we provide a formal proof by introducing a sequence of games played between the adversary and the simulator. Finally, we prove our HyPRE scheme selectively secure in the HRA model.

3. We conduct a comprehensive evaluation from both theoretical and experimental perspectives. We compare our scheme against state-of-the-art alternatives, highlighting the significant advantages of our approach. Specifically, we demonstrate that the time costs associated with

decryption and re-encryption key generation phases in the previous IB-BPRE scheme [4] grow exponentially with the number of identities, while our HyPRE scheme exhibits linear growth in these operations.

1.2. Related work

Cryptographic schemes provide multiple solutions for data security in cloud services, and there is a series of research works for data storage and sharing in the cloud [6, 7, 8, 9]. Two popular schemes for secure data storage and access control are identity-based encryption (IBE) and attribute-based encryption (ABE). The definitions for secure identity-based encryption schemes were given by Boneh and Franklin, who also proposed the first IBE scheme [10]. In 2005, Sahai and Waters introduced the first ABE scheme known as Fuzzy IBE (FIBE) [11]. Waters then proposed a new ciphertext policy ABE (CP-ABE) scheme, which is secure in the standard model [12]. Lewko et al. pointed out that in the ABE schemes developed earlier, the size of the universe or the attribute set was fixed after setup [13]. To address this limitation, they proposed the first large universe scheme. Subsequently, Rouselakis and Waters enhanced the efficiency of large universe ABE schemes by basing it on prime order groups [14].

The first PRE scheme was proposed by Blaze et al. in 1998 [15]. Liang et al. then introduced the attribute-based PRE (AB-PRE) primitive [16], which combined the concepts of ABE and PRE. In this primitive, a ciphertext generated under an access policy can be converted into another one with a new policy. Various AB-PRE schemes have been proposed afterwards [17, 16, 18, 5, 19]. Additionally, Xu et al. proposed an identity-based broadcast proxy re-encryption (IB-BPRE) scheme [20], which combines the notions of IBE and IBBE. Their scheme allows a ciphertext under one identity to be transformed into a ciphertext under a set of different identities. Another IB-BPRE scheme was proposed by Ge et al. in 2019 [4] to address the key revocation problem. However, these IB-BPRE schemes have limitations in fine-grained access control for different users compared to ABE schemes. Hence, the objective of this paper is to propose a new hybrid proxy re-encryption scheme that addresses the problem of data sharing from an individual to multiple people. Notably, to the best of our knowledge, there is currently no primitive available that can transform a ciphertext of an IBE scheme into a ciphertext of an ABE scheme.

The first PRE scheme was proposed by Blaze et al. in 1998 [15]. By combining the concept of ABE and PRE, Liang et al. proposed the attribute-based PRE (AB-PRE) primitive [16], where a ciphertext generated under an access policy can be converted into another one with a new policy. After that, a series of AB-PRE schemes was proposed [17, 16, 18, 5, 19]. Xu et al. proposed an identity-based broadcast proxy re-encryption (IB-BPRE) scheme [20] by combining the notion of IBE and identity-based broadcast encryption (IBBE). Through their scheme, a ciphertext under one identity can be transformed into a ciphertext under a set of different identities. In 2019, Ge et al. proposed another IB-BPRE scheme [4] to further solve the key revocation problem. However, the IB-BPRE schemes only supports discrete identities, which is very limited in fine-grained access control for different users compared with the ABE schemes. Therefore, in this paper, we focus on proposing a new hybrid proxy re-encryption scheme to better solve the problem of data sharing from an individual to multiple people. To the best of our knowledge, there is still no such primitive that can transform a ciphertext of an IBE scheme into a ciphertext of an ABE scheme.

To address high security requirements, Canetti and Hohenberger introduced the first chosen ciphertext attack (CCA) secure Proxy Re-Encryption (PRE) scheme [21], following the previously proposed CPA secure PRE schemes [15, 22, 23]. Subsequently, several works were published to enhance the CCA security of PRE [24, 25, 26, 27]. While CCA security provides greater robustness and flexibility, it is more efficient to construct PRE schemes under the standard chosen plaintext attack (CPA) security model. In 2019, Cohen introduced the notion of HRA security, which addresses scenarios where the delegatee may access the delegator's secret key through honestly re-encrypted ciphertext [1]. Susilo et al. formalized the definition of HRA-secure Key-Policy Attribute-Based Proxy Re-Encryption (KP-ABPRE) and proposed a construction in 2021 [28]. However, currently, there is no HRA-secure Hybrid PRE (HyPRE) model. Additionally, in the security proofs of hybrid PRE schemes, it is essential to prove the indistinguishability of re-encryption keys [6]. However, in [6], the security proof is incomplete as it lacks a formal proof of this property. Motivated by these gaps, this paper aims to formalize the definition of HRA security for HyPRE and provide a security proof for the proposed scheme in the HRA model.

1.3. Organization

The rest of this paper is organized as follows: In Section 2, we describe the system model and algorithms of our scheme, then in Section 3, we introduce some background information about bilinear maps, linear secret sharing

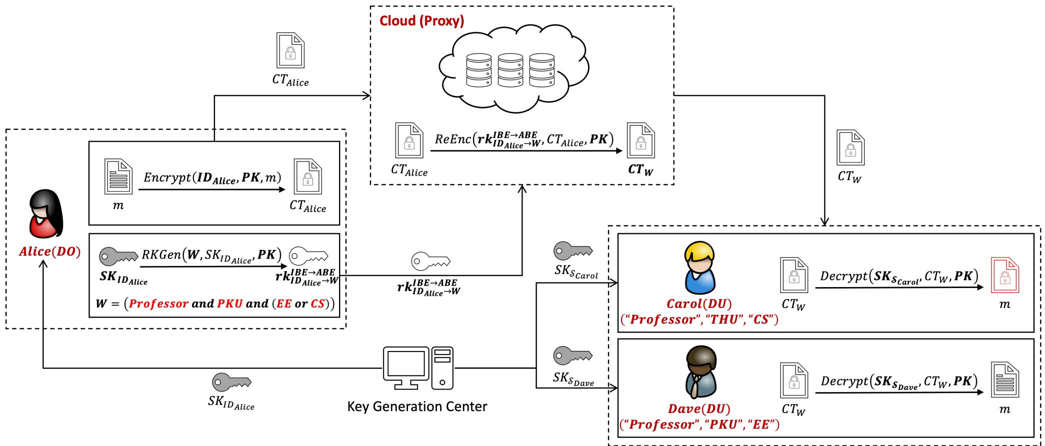


Figure 1: System model of our HyPRE scheme

schemes and the complexity assumption of our scheme. We propose the HRA security model of our HyPRE scheme in Section 4. Our AB-PRE construction is presented in Section 5. We give the security analysis of our HyPRE scheme in Section 6. We evaluate the efficiency of our proposed schemes theoretically and experimentally in Section 7. At last, we conclude the paper in Section 8.

2. System Model and Algorithms

2.1. System model

Figure 1 shows the system model of our HyPRE scheme, there are mainly four types of participants:

1. **Data Owner (DO)**: The data owner (Alice in Figure 1) encrypts its own data with an identity (ID) and uploads the ciphertexts to the cloud server (proxy) to store the original data on the cloud. When the DO wants to access the original data, it can download the uploaded ciphertexts and decrypt them using its own private key. In order to authorize the encrypted data to the data users, DO can generate re-encryption keys with its own private key and a specified access policy, and then upload them to the proxy.

2. **Cloud Server (Proxy)**: The cloud server (proxy) is responsible for providing various services including cloud storage, data sharing, and

maintaining the interface between the data owner and the data user. This server offers functionalities such as data uploading, downloading, and sharing. Furthermore, it facilitates the delegation of data by accepting re-encryption keys from the data owner. By utilizing these re-encryption keys, the cloud server transforms the ciphertext of Identity-Based Encryption (IBE) to another ciphertext of Attribute-Based Encryption (ABE) through the process of re-encryption.

3. **Data User (DU)**: The data users (DUs) (Carol and Dave in Figure 1) are the users who access the original data from the data owner. To decrypt the re-encrypted ciphertexts, DUs must possess private keys that have associated attributes satisfying the access policy of the ciphertext. Only then can they successfully decrypt the ciphertexts.

2.2. Algorithms

Our HyPRE scheme consists of the following algorithms:

$\mathsf{Setup}(1^\lambda) \to (pp, msk)$. The setup algorithm is executed by the key generation center (KGC) when the system starts up. It takes the security parameter λ as input, and outputs the public parameters pp and the master secret key msk.

$\mathsf{KeyGen}_{\mathsf{ID}}(pp, msk, \mathsf{ID}) \to sk_{\mathsf{ID}}$. The key generation algorithm associated with the original ciphertexts is executed by the KGC. It takes as input the public parameter pp, the master secret key msk and the user's identity ID, it outputs the corresponding secret key sk_{ID}. A user with sk_{ID} can decrypt the original ciphertexts related to its identity.

$\mathsf{KeyGen}_{\mathcal{S}}(pp, msk, \mathcal{S}) \to sk_{\mathcal{S}}$. The key generation algorithm associated with the re-encrypted ciphertexts is executed by the KGC. It takes as input the public parameters pp, the master secret key msk and a user's attribute set $\mathcal{S}$, it outputs the corresponding secret key $sk_{\mathcal{S}}$. A user with $sk_{\mathcal{S}}$ can decrypt the re-encrypted ciphertexts when its attributes satisfy the access structure.

$\mathsf{Encrypt}(pp, m, \mathsf{ID}) \to ct_{\mathsf{ID}}$. The encryption algorithm is executed by the data owner. It takes the public parameter pp, the user's identity ID and the plaintext message m as input and outputs the ciphertext ct_{ID}.

$\mathsf{RKGen}(pp, sk_{\mathsf{ID}}, \mathsf{W}) \rightarrow rk_{\mathsf{ID}\rightarrow\mathsf{W}}$. The re-encryption key generation algorithm is executed by the data owner or the data agent that owns the secret key. It takes as input the public parameter pp, the secret key sk_{ID} and the target access structure W and outputs the re-encryption key $rk_{\mathsf{ID}\rightarrow\mathsf{W}}$.

$\mathsf{ReEnc}(pp, rk_{\mathsf{ID}\rightarrow\mathsf{W}}, ct_{\mathsf{ID}}) \rightarrow ct_{\mathsf{W}}$. The re-encryption algorithm is executed by the proxy. It takes as input the public parameter pp the re-encryption key $rk_{\mathsf{ID}\rightarrow\mathsf{W}}$ and the ciphertext to be re-encrypted ct_{ID}. It outputs the re-encrypted ciphertext ct_{W}.

$\mathsf{Decrypt}_{\mathsf{ID}}(ct_{\mathsf{ID}}, sk_{\mathsf{ID}}) \rightarrow m$. The decryption algorithm for the original ciphertexts is executed by the user with an identity ID. It takes as input the ciphertext corresponding to the user's identity ct_{ID} and the user's secret key sk_{ID}, and outputs the plaintext message m.

$\mathsf{Decrypt}_{\mathcal{S}}(ct_{\mathsf{W}}, sk_{\mathcal{S}}) \rightarrow m$. The decryption algorithm for the re-encrypted ciphertexts is executed by the user with a set of attributes $\mathcal{S}$. It takes as input the ciphertext corresponding to the access policy ct_{W} and the user's secret key $sk_{\mathcal{S}}$, and outputs the plaintext message m.

3. Preliminaries

3.1. Bilinear maps

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

- Bilinearity: $\forall g, h \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}_p$, the equation $e(g^a, h^b) = e(g, h)^{ab}$ holds;
- Non-degeneracy: There exist $g, h \in \mathbb{G}$ such that $e(g, h) \neq 1$;
- Computability: $\forall g, h \in \mathbb{G}$, there is a way to compute $e(g, h)$ efficiently.

3.2. Linear secret sharing schemes (LSSS)

We first define p as the prime order and $\mathcal{U}$ as the universe of attributes. Let Π be a linear secret sharing scheme, for each access structure $\mathbb{A}$ over $\mathcal{U}$, there is a way to construct a matrix called the share-generating matrix (denoted as $M \in \mathbb{Z}_p^{l\times n}$). Plus a specific function ρ, we can map the attributes of $\mathcal{U}$ to the rows of M respectively. We denote this map as $\rho : [\ell] \rightarrow \mathcal{U}$.

Let s be the secret to be shared and $r_2, \cdots, r_n \in \mathbb{Z}_p$ be random elements. Based on Π, we can get ℓ shares of s by calculating $M \cdot \vec{v} \in \mathbb{Z}_p^{\ell \times 1}$, where $\vec{v} = (s, r_2, \cdots, r_n)^\top$. Each share represents an attribute (denoted as $\rho(j)$). Consequently, we consider the pair (M, ρ) to represent the policy of $\mathbb{A}$.

3.3. The modified BDHE (M-BDHE) Assumption

The challenger takes the security parameter as input and runs the group generation algorithm. It picks a random group element $g \in \mathbb{G}$ and $q+3$ random exponents $a, s, b, b_1, b_2, \cdots, b_q \in \mathbb{Z}_p$. Then it sends to the adversary the tuple $(p, \mathbb{G}, \mathbb{G}_T, e)$ which describes a group and all of the following terms:

$$
\begin{array}{ll}
g, g^a, g^b, g^s, g^{a^{2q}}, & \\
g^{as/b}, g^{a^2 s/b}, g^{(as)^2/b^2}, & \\
g^{a^i}, g^{b_j}, g^{asb_j/b}, g^{a^i b/b_j^2} & \forall (i,j) \in [q,q], \\
g^{a^i b_j/b_{j'}^2} & \forall (i,j,j') \in [2q,q,q], j \neq j', \\
g^{a^i/b_j} & \forall (i,j) \in [2q,q], i \neq q+1, \\
g^{a^i b_j}, g^{a^i/b_j^2} & \forall (i,j) \in [2q,q], \\
g^{a^i s/(b \cdot b_j^2)}, g^{a^i s b_j/b} & \forall (i,j) \in [q+1,q].
\end{array}
$$

The challenger firstly flips a random coin $b \in \{0,1\}$. If $b=0$, it answers the adversary with the term of $e(g,g)^{sa^{q+1}}$ and otherwise it returns a random term of $R \in \mathbb{G}_T$. Finally the adversary outputs a guess $b' \in \{0,1\}$ on the original bit b.

Definition 1. *We say that the M-BDHE assumption holds if all PPT attackers have at most a negligible advantage in λ in the above security game, where the advantage is defined as* $\mathsf{Adv} = |\Pr[b' = b] - 1/2|$.

4. HRA Security Model

In the selective setting of HRA security for PRE schemes, the adversary must choose the set of parties it corrupts before issuing any queries, while in the adaptive setting, the adversary is allowed to corrupt users at any time during the game [2]. In this work, we focus on the selective security of HyPRE scheme. In this section, we extend the idea of HRA security from [1] and propose a new selective security game for the HyPRE scheme.

In the selective setting of HRA security for HyPRE schemes, the target identity and policy are declared by the adversary before the query phases.

Init. The adversary $\mathcal{A}$ declares the challenge identity ID^* and the challenge access structure $\mathbb{A}^* = (M^*, \rho^*)$ which it is going to attack, and it sends them to the challenger $\mathcal{B}$. After that, $\mathcal{B}$ runs the $\mathsf{Setup}(1^\lambda)$ algorithm to generate the public parameters pp and the master secret key msk, and then $\mathcal{B}$ gives pp to $\mathcal{A}$. Additionally, $\mathcal{B}$ initializes an empty set $\mathcal{C}$ to store honestly generated ciphertexts which can be re-encrypted afterwards.

Phase 1. $\mathcal{A}$ is given access to the following five types of query oracles:

- $\mathcal{O}^{\mathsf{ID}}_{\mathsf{KeyGen}}$. For a query of this type with input ID, if $\mathsf{ID} = \mathsf{ID}^*$, $\mathcal{B}$ rejects this query and returns $\perp$ to $\mathcal{A}$, otherwise, $\mathcal{B}$ calls $sk_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}_{\mathsf{ID}}(pp, msk, \mathsf{ID})$ and sends sk_{ID} to $\mathcal{A}$.

- $\mathcal{O}^{\mathcal{S}}_{\mathsf{KeyGen}}$. For a query of this type with input S, if S satisfies $\mathbb{A}^*$, $\mathcal{B}$ rejects this query and returns $\perp$ to $\mathcal{A}$, otherwise, $\mathcal{B}$ calls $sk_S \leftarrow \mathsf{KeyGen}_{\mathcal{S}}(pp, msk, S)$ and sends sk_S to $\mathcal{A}$.

- $\mathcal{O}^{\mathsf{ID}\to\mathbb{A}}_{\mathsf{RKGen}}$. For a query of this type with input $(\mathsf{ID}, \mathbb{A})$, if $\mathsf{ID} = \mathsf{ID}^*$ and $\mathbb{A} \not\subset \mathbb{A}^*$, $\mathcal{B}$ rejects this query and returns $\perp$ to $\mathcal{A}$. If $\mathsf{ID} = \mathsf{ID}^*$ and $\mathbb{A} \subset \mathbb{A}^*$, $\mathcal{A}$ calls $sk_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}_{\mathsf{ID}}(pp, msk, \mathsf{ID})$ and $rk_{\mathsf{ID}\to\mathbb{A}} \leftarrow \mathsf{RKGen}(pp, sk_{\mathsf{ID}}, \mathbb{A})$, then sends $rk_{\mathsf{ID}\to\mathbb{A}}$ to $\mathcal{A}$. Note that for other cases where $\mathsf{ID} \neq \mathsf{ID}^*$, $\mathcal{A}$ can simply obtain sk_{ID} from $\mathcal{O}^{\mathsf{ID}}_{\mathsf{KeyGen}}$ to generate $rk_{\mathsf{ID}\to\mathbb{A}}$ by itself, so we can omit them in this query oracle.

- $\mathcal{O}^{\mathsf{ID}}_{\mathsf{Encrypt}}$. For a query of this type with input m, $\mathcal{B}$ runs $ct \leftarrow \mathsf{Encrypt}(pp, m, \mathsf{ID}^*)$, adds the value ct to the set $\mathcal{C}$ and returns it to $\mathcal{A}$.

- $\mathcal{O}^{\mathsf{ID}\to\mathbb{A}}_{\mathsf{ReEnc}}$. For a query of this type with the input $(rk_{\mathsf{ID}\to\mathbb{A}}, ct_{\mathsf{ID}})$, we first consider the case where $\mathsf{ID} = \mathsf{ID}^*$ and $\mathbb{A} \not\subset \mathbb{A}^*$. If $ct_{\mathsf{ID}} \notin \mathcal{C}$, $\mathcal{B}$ rejects this query and returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ runs $ct_{\mathbb{A}} \leftarrow \mathsf{ReEnc}(pp, rk_{\mathsf{ID}\to\mathbb{A}}, ct_{\mathsf{ID}})$ and returns $ct_{\mathbb{A}}$ to $\mathcal{A}$. For the case where $\mathsf{ID} = \mathsf{ID}^*$ and $\mathbb{A} \subset \mathbb{A}^*$, $\mathcal{A}$ can simply obtain $rk_{\mathsf{ID}\to\mathbb{A}}$ from the $\mathcal{O}^{\mathsf{ID}\to A}_{\mathsf{RKGen}}$ to generate $ct_{\mathbb{A}}$ by itself, so we can omit it in this query oracle. For the remaining cases where $\mathsf{ID} \neq \mathsf{ID}^*$, $\mathcal{A}$ can simply obtain sk_{ID} from $\mathcal{O}^{\mathsf{ID}}_{\mathsf{KeyGen}}$ to generate $rk_{\mathsf{ID}\to\mathbb{A}}$ and then $ct_{\mathbb{A}}$ by itself, so we can omit them in this query oracle.

Challenge. $\mathcal{A}$ submits two messages m_0, m_1 of the same length to $\mathcal{B}$. $\mathcal{B}$ flips a random coin $b \in \{0, 1\}$, then it runs $ct^* \leftarrow \mathsf{Encrypt}(pp, m_b, \mathsf{ID}^*)$, and sends ct^* to $\mathcal{A}$.

Phase 2. This phase is the same as **Phase 1**.

Guess. $\mathcal{A}$ outputs a guess b' of b and wins if $b' = b$.

HRA Security. Given a security parameter λ, a hybrid proxy re-encryption scheme is HRA secure if for all probabilistic polynomial time (PPT) adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$Adv_{hra}^{\mathcal{A}}(\lambda) < \frac{1}{2} + \mathsf{negl}(\lambda).$$

5. Construction

$\mathsf{Setup}(1^\lambda) \rightarrow (pp, msk)$. The setup algorithm first calls the group generation algorithm with a security parameter λ, then it picks random terms $g, u, h, w, v, f \in \mathbb{G}$ and $\alpha, \beta \in \mathbb{Z}_p$. It outputs

$$pp = (D, g, u, h, w, v, f, e(g, g)^\alpha, e(g, g)^\beta), msk = (\alpha, \beta).$$

$\mathsf{KeyGen}_{\mathsf{ID}}(pp, msk, \mathsf{ID}) \rightarrow sk_{\mathsf{ID}}$. The KGC picks a random exponent $r \in \mathbb{Z}_p$ and computes $K_0 = g^\alpha w^r, K_1 = (u^{\mathsf{ID}} h)^{-r}, K_2 = g^r$. The secret key is formed as

$$sk_{\mathsf{ID}} = (K_0, K_1, K_2).$$

$\mathsf{KeyGen}_{\mathcal{S}}(pp, msk, \mathcal{S}) \rightarrow sk_{\mathcal{S}}$. The KGC randomly chooses $r, \tilde{r}, r_1, r_2, \cdots, r_k \in \mathbb{Z}_p$ and it then computes $K_0 = g^\beta w^r, K_1 = g^r$, for every $i \in [k]$, it computes $K_{i,2} = g^{r_i}, K_{i,3} = (u^{A_i} h)^{r_i} v^{-r}$. The secret key for attribute list $\mathcal{S}$ is formed as

$$sk_{\mathcal{S}} = \left(K_0, K_1, \{K_{i,2}, K_{i,3}\}_{i \in [k]} \right).$$

$\mathsf{Encrypt}(pp, m, \mathsf{ID}) \rightarrow ct_{\mathsf{ID}}$. The data owner picks random elements $s, t \in \mathbb{Z}_p$ and computes $C = m \cdot e(g, g)^{\alpha s}, C_0 = g^s, C_1 = g^t, C_2 = (u^{\mathsf{ID}} h)^t w^{-s}, C_3 = f^s$. The ciphertext is formed as

$$ct_{\mathsf{ID}} = (C, C_0, C_1, C_2, C_3).$$

$\mathsf{RKGen}(pp, sk_{\mathsf{ID}}, \mathsf{W}) \rightarrow rk_{\mathsf{ID} \rightarrow \mathsf{W}}$. The data owner with secret key sk_{ID} takes the target access structure encoded in an LSSS policy $\mathsf{W} = (M, \rho)$, where $M \in \mathbb{Z}_p^{\ell \times n}$ and $\rho : [\ell] \rightarrow \mathbb{Z}_p$. Firstly, it picks $\vec{y} = (s', y_2, \cdots, y_n)^\top \leftarrow \mathbb{Z}_p^{n \times 1}$, where s' is the random secret to be shared among the shares. The vector of the shares is $\lambda' = (\lambda'_1, \lambda'_2, \cdots, \lambda'_\ell)^\top = M\vec{y}$. It randomly chooses exponents

$t', t'_1, t'_2, \cdots t'_\ell \in \mathbb{Z}_p$ and then computes $d_0 = K_0 \cdot f^{t'}, d_1 = K_1, d_2 = K_2$. For $i \in [\ell]$, it computes $d_{i,3} = w^{\lambda'_i} v^{t'_i}, d_{i,4} = (u^{\rho(i)} h)^{-t'_i}, d_{i,5} = g^{t'_i}$. Then it computes $d_6 = F(e(g,g)^{\beta s'}) \cdot g^{t'}, d_7 = g^{s'}$. The re-encryption key is formed as

$$rk_{\mathsf{ID} \to \mathsf{W}} = \left(d_0, d_1, d_2, \{d_{i,3}, d_{i,4}, d_{i,5}\}_{i \in [\ell]}, d_6, d_7\right).$$

$\mathsf{ReEnc}(pp, rk_{\mathsf{ID}\to\mathsf{W}}, ct_{\mathsf{ID}}) \to ct_{\mathsf{W}}$. On input the re-encryption key $rk_{\mathsf{ID}\to\mathsf{W}}$ and the ciphertext ct_{ID}, the re-encryption algorithm first computes $B = e(d_0, C_0) \cdot e(d_1, C_1) \cdot e(d_2, C_2)$, then it computes $C' = C/B$, the other parts of the re-encrypted ciphertext are $C'_0 = d_6, \{C'_{i,1} = d_{i,3}, C'_{i,2} = d_{i,4}, C'_{i,3} = d_{i,5}\}_{i \in [\ell]}, C'_4 = C_3 = f^s, C'_5 = d_7$. The re-encrypted ciphertext is formed as

$$ct' = \left(C', C'_0, \left\{C'_{i,1}, C'_{i,2}, C'_{i,3}\right\}_{i \in [\ell]}, C'_4, C'_5\right).$$

$\mathsf{Decrypt}_{\mathsf{ID}}(ct_{\mathsf{ID}}, sk_{\mathsf{ID}}) \to m$. For the original ciphertext, the decryption algorithm calculates

$$B = e(K_0, C_0) \cdot e(K_1, C_1) \cdot e(K_2, C_2),$$

303 and then it outputs $m = C/B$.

$\mathsf{Decrypt}_{\mathcal{S}}(ct_{\mathsf{W}}, sk_{\mathcal{S}}) \to m$. For re-encrypted ciphertexts, assume the set of rows in M is $\mathcal{I} = \{i : \rho(i) \in \mathcal{S}\}$, the decryptor first calculates the constants $\{w_i \in \mathbb{Z}_p\}_{i \in \mathcal{I}}$, such that $\sum_{i \in \mathcal{I}} w_i M_i = (1, 0, \cdots, 0)$, then it computes

$$\frac{e\left(C'_5, K_0\right)}{\prod_{i \in \mathcal{I}} \left(e\left(C'_{i,1}, K_1\right) \cdot e\left(C'_{i,2}, K_{j,2}\right) \cdot e\left(C'_{i,3}, K_{j,3}\right)\right)^{w_i}} = e(g,g)^{\beta s'}.$$

304 Finally, it computes $g^{t'} = C'_0 / F(e(g,g)^{\beta s'})$ and outputs $m = C' \cdot e(g^{t'}, C'_4)$.

305 ***Correctness.***

- For an original ciphertext ct_{ID}, if the ciphertext ct_{ID} and the secret key sk_{ID} are associated with the same identity ID, we have

$$\begin{aligned} B &= e\left(g^\alpha w^r, g^s\right) \cdot e\left(\left(u^{\mathsf{ID}} h\right)^{-r}, g^t\right) \cdot e\left(g^r, \left(u^{\mathsf{ID}} h\right)^t w^{-s}\right) \\ &= e(g,g)^{\alpha s}. \end{aligned}$$

306 Then we can get the plaintext $m = C/e(g,g)^{\alpha s}$.

- For a re-encrypted ciphertext ct_{W} and a secret key $sk_{\mathcal{S}}$, if $\mathcal{S}$ satisfies W, assume the set of rows in M is $\mathcal{I} = \{i : \rho(i) \in \mathcal{S}\}$, we can recover the secret s' by calculating $\sum_i w_i \lambda'_1 = s'$, then we have

$$\begin{aligned} B' &= e\left(g^{s'}, g^{\beta} w^{r}\right) / \prod_{i \in \mathcal{I}} e\left(w^{\lambda'_i} v^{t'_i}, g^{r}\right)^{w_i} \\ &\quad \cdot \prod_{i \in \mathcal{I}} \left(e\left(\left(u^{\rho(i)} h\right)^{-t'_i}, g^{r_i}\right) \cdot e\left(g^{t'_i}, \left(u^{A_i} h\right)^{r_i} v^{-r}\right)\right)^{w_i} \\ &= e(g,g)^{\beta s'}, \\ g^{t'} &= C'_0 / F\left(e(g,g)^{\beta s'}\right). \end{aligned}$$

Then we can get the plaintext $m = C' \cdot e(g^{t'}, f^{s})$.

6. Proof of security

Our security proof of the HyPRE scheme is based on the modified BDHE (M-BDHE) assumption, the security of the M-BDHE assumption is analyzed in Appendix A. Then we prove our HyPRE scheme selectively secure through the following theorem.

Theorem 1. *Assume that the M-BDHE assumption holds and the* RW13 *scheme [14] is CPA-secure. The HyPRE scheme is secure against the honest re-encryption attacks (HRA).*

The above theorem is proved by a sequence of games. We first define two types of re-encryption keys: the real re-encryption keys and the nominal re-encryption keys.

- **Real Re-encryption Keys**. For an identity ID, we first run the normal KeyGen algorithm to generate a normal secret key $sk_{\mathsf{ID}} = (K_0, K_1, K_2)$. Then for an access policy (M, ρ), we run the RKGen algorithm to generate a well-formed re-encryption key. The real re-encryption key is formed as

$$d_0 = K_0 \cdot f^{t''}, d_1 = K_1, d_2 = K_2, d_{i,3} = w^{\lambda'_i} v^{t'_i}, d_{i,4} = (u^{\rho(i)} h)^{-t'_i}, d_{i,5} = g^{t'_i},$$

$$d_6 = F(e(g,g)^{\beta s'}) \cdot g^{t''}, d_7 = g^{s'}.$$

- **Nominal Re-encryption Keys**. For the nominal re-encryption keys, they own the same values of the terms $d_1, d_2, d_{i,3}, d_{i,4}, d_{i,5}, d_7$ with the real re-encryption keys, we choose a random component $R \in \mathbb{G}$ and compute $d_6 = F(e(g,g)^{\beta s'}) \cdot R$. The nominal re-encryption key is formed as

$$d_0 = K_0 \cdot f^{t''}, d_1 = K_1, d_2 = K_2, d_{i,3} = w^{\lambda'_i} v^{t'_i}, d_{i,4} = (u^{\rho(i)} h)^{-t'_i}, d_{i,5} = g^{t'_i},$$

$$d_6 = F(e(g,g)^{\beta s'}) \cdot R, d_7 = g^{s'}.$$

As we can see, the real re-encryption keys are the well-formed ones in the original HyPRE scheme, and in the nominal re-encryption keys we replace the $g^{t''}$ part of the d_6 item with a random component R. As the items of $d_{i,3}, d_{i,4}, d_{i,5}, d_6, d_7$ construct a well-formed ciphertext of the $\mathsf{RW13}$ scheme, we can regard the difference between the real re-encryption keys and the nominal re-encryption keys as the difference between the ciphertext of message $g^{t''}$ and that of R under $\mathsf{RW13}$.

Then we give the definition of a series of games.

- Game_{real}. Game_{real} denotes the real security game defined in Section 4.
- Game_k. Let Q denotes the total number of re-encryption key queries from the adversary. In this game, the first k re-encryption keys are nominal and the remaining re-encryption keys are normal.
- Game_{final}. Game_{final} denotes the security game where all re-encryption keys are nominal.

Next, we prove Theorem 1 with the help of the following lemmas below.

Lemma 1. *For all PPT distinguishers $\mathcal{D}$, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $|\Pr[\mathcal{D}(\mathsf{Game}_{real}(k)) = 1] - \Pr[\mathcal{D}(\mathsf{Game}_{final}(k)) = 1]| \leq \mathsf{negl}(k)$

Proof. Assume there exists a PPT distinguisher $\mathcal{D}$ and a polynomial $\mathsf{poly}(\cdot)$ such that for infinitely many values of $k \in \mathbb{N}$, we have that $\mathcal{D}$ distinguishes between Game_{real} and Game_{final} with probability at least $1/\mathsf{poly}(k)$. Let $Q \in \mathsf{poly}(k)$ be the number of queries that $\mathcal{D}$ is allowed to ask to its oracle. For an index $t \in [0, Q]$, consider the hybrid game Game_t that answers the first t queries as in Game_{real} and all the subsequent queries as in game Game_{final}. Note that $\mathsf{Game}_{real} \equiv \mathsf{Game}_0$ and $\mathsf{Game}_{final} \equiv \mathsf{Game}_Q$.

By a standard hybrid argument, we have that there exists an index $t \in [0, Q]$ such that $\mathcal{D}$ tells apart Game_{t-1} and Game_t with non-negligible probability $1/Q \cdot 1/\mathsf{poly}(k)$. We build a PPT adversary $\mathcal{A}$ that (using distinguisher $\mathcal{D}$) breaks CPA security of $\mathsf{RW13}$ [14]. A formal description of $\mathcal{A}$ follows.

- The adversary $\mathcal{A}$ receives public parameters pp from the challenger, where $pp \leftarrow \mathsf{Setup}_{\mathsf{RW13}}(1^{\lambda})$.
- On input a collision query of from $\mathcal{D}$,
  - If $j \leq t-1$, for an identity ID, it first runs the KeyGen algorithm and generates the secret key $sk_{\mathsf{ID}} = (K_0, K_1, K_2)$, then it simulates random exponent t'' and runs the RKGen algorithm to calculate $d_0 = K_0 \cdot f^{t''}, d_1 = K_1, d_2 = K_2, d_{i,3} = w^{\lambda'_i} v^{t'_i}, d_{i,4} = (u^{\rho(i)} h)^{-t'_i}, d_{i,5} = g^{t'_i}, d_6 = F(e(g,g)^{\beta s'}) \cdot g^{t''}, d_7 = g^{s'}$. It returns the re-encryption key rk to $\mathcal{D}$ where
  $$rk = (d_0, d_1, d_2, d_{i,3}, d_{i,4}, d_{i,5}, d_6, d_7).$$
  - If $j = t$, for an identity ID, it first runs the KeyGen algorithm and generates the secret key $sk_{\mathsf{ID}} = (K_0, K_1, K_2)$, then it picks a random exponent t'' and calculate $d_0 = K_0 \cdot f^{t''}, d_1 = K_1, d_2 = K_2$. It picks a random component R, calculates $m_0 = g^{t''}, m_1 = R$, and submits m_0 and m_1 to the challenger. After receiving the challenging ciphertext ct_b, it constructs the re-encryption key, which is formed as $rk = (d_0, d_1, d_2, \{ct_b\}_{b \in \{0,1\}})$, and sends it to $\mathcal{D}$.
  - If $j \geq t$, for an identity ID, it first runs the KeyGen algorithm and generates the secret key $sk_{\mathsf{ID}} = \{K_0, K_1, K_2\}$. Then it picks a random exponent t'' and a random component R, and it runs the $\mathsf{Encrypt}_{rw13}$ algorithm to generate the $d_{i,3}, d_{i,4}, d_{i,5}, d_6, d_7$ part of the nominal re-encryption key
  $$\begin{aligned} ct' &= \mathsf{Encrypt}_{\mathsf{RW13}}(pp, R, (\mathsf{W}, \rho)) \\ &= \begin{pmatrix} d_{i,3} = w^{\lambda'_i} v^{t'_i}, d_{i,4} = \left(u^{\rho(i)} h\right)^{-t'_i}, d_{i,5} = g^{t'_i}, \\ d_6 = F(e(g,g)^{\beta s'}) \cdot R, d_7 = g^{s'} \end{pmatrix}. \end{aligned}$$
  Then it returns the nominal re-encryption key rk to $\mathcal{D}$, and rk is formed as
  $$rk = (d_0 = K_0 \cdot f^{t''}, d_1 = K_1, d_2 = K_2, d_{i,3}, d_{i,4}, d_{i,5}, d_6, d_7).$$

The only difference between Game_{t-1} and Game_t is on how the t-th re-encryption key is answered. In particular, in case the hidden bit b in the definition of CPA security of $\mathsf{RW13}$ equals zero, the adversary $\mathcal{A}$'s simulation produces exactly the same distribution as in Game_{t-1}, and otherwise $\mathcal{A}$'s simulation produces exactly the same distribution as in Game_t. Hence, $\mathcal{A}$ breaks CPA security with non-negligible advantage $1/Q \cdot 1/\mathsf{poly}(k)$, a contradiction. This concludes the proof. □

Lemma 2. *If the M-BDHE assumption holds, then all PPT adversaries with a challenge matrix of size $\ell \times n$, where $\ell, n \leq q$, have a negligible advantage in selectively breaking our scheme.*

Proof. In this proof, we assume that there exists a PPT adversary $\mathcal{A}$ with a challenge access policy $\mathbb{A}^* = (M^*, \rho^*)$ and a challenge identity ID^* that satisfies the restriction. And the adversary has a non negligible advantage in selectively breaking our scheme. We build a PPT simulator that attacks the M-BDHE assumption with a non negligible advantage.

Init. The adversary $\mathcal{A}$ declares a challenge identity ID^* and a challenge policy $\mathbb{A}^* = (M^*, \rho^*)$ and sends them to the simulator $\mathcal{B}$. Note that M^* is an $\ell \times n$ matrix where $\ell, n \leq q$ and $\rho^* : [\ell] \rightarrow \mathbb{Z}_p$.

Setup. The simulator implicitly sets the master key as $\alpha = a^{q+1} + \tilde{\alpha}$. It picks the random exponents $\tilde{v}, \tilde{u}, \tilde{h} \in \mathbb{Z}_p$, then it runs the Setup algorithm and set

$$\begin{aligned}
&g = g,\\
&u = g^{\tilde{u}} \cdot \prod_{(j,k)\in[\ell,n]} \left(g^{a^k/b_j^2}\right)^{M^*_{j,k}} \cdot g^{a^q},\\
&h = g^{\tilde{h}} \cdot \prod_{(j,k)\in[\ell,n]} \left(g^{a^k/b_j^2}\right)^{-\rho^*(j)M^*_{j,k}} \cdot (g^{a^q})^{-\mathsf{ID}^*} \cdot g^{as/b},\\
&w = g^a,\\
&v = g^{\tilde{v}} \cdot \prod_{(j,k)\in[\ell,n]} \left(g^{a^k/b_j}\right)^{M^*_{j,k}},\\
&e(g,g)^\alpha = e(g^a, g^{a^q}),\\
&e(g,g)^\beta = e(g^a, g^{a^q}) \cdot e(g,g)^{\tilde{\beta}}.
\end{aligned}$$

Then $\mathcal{B}$ sends the above public parameters to $\mathcal{A}$.

Phase 1. In this phase, the adversary $\mathcal{A}$ and the simulator $\mathcal{B}$ runs the following query oracles: $\mathcal{O}^{\mathsf{ID}}_{\mathsf{KeyGen}}$, $\mathcal{O}^{\mathcal{S}}_{\mathsf{KeyGen}}$, $\mathcal{O}^{\mathsf{ID}\to A}_{\mathsf{RKGen}}$, $\mathcal{O}^{\mathsf{ID}}_{\mathsf{Encrypt}}$ and $\mathcal{O}^{\mathsf{ID}\to\mathcal{S}}_{\mathsf{ReEnc}}$.

- $\mathcal{O}^{\mathsf{ID}}_{\mathsf{KeyGen}}$. In this oracle, $\mathcal{A}$ makes secret key queries to $\mathcal{B}$. For each query, the adversary $\mathcal{A}$ submits an attribute set $\mathcal{S}$ (which is not authorized for $\mathbb{A}^*$) to $\mathcal{B}$ and $\mathcal{B}$ generates the related secret key as follows. The simulator $\mathcal{B}$ generates a vector $\vec{w} = (w_1, w_2, \cdots, w_n) \in \mathbb{Z}_p^n$ such that for all $i \in [\ell]$ and $\rho^*(i) \in \mathcal{S}$, $w_1 = -1$ and $\langle M_i^*, \vec{w} \rangle = 0$. It picks $\tilde{r} \in \mathbb{Z}_p$ and implicitly sets $r = \tilde{r} + \sum_{i\in[n]} w_i a^{q+1-i}$, $t = \tilde{t} - a^q + \frac{as}{b\cdot(\mathsf{ID}-\mathsf{ID}^*)}$. Assume ID^* is honest and other IDs are corrupted. The adversary $\mathcal{A}$ makes queries on secret keys related to corrupted identities to simulator $\mathcal{B}$. Then $\mathcal{B}$ calculates the secret key terms K_0, K_1 and K_2 as

$$
\begin{aligned}
K_0 =& g^{\alpha} w^t = g^{a^{q+1}} \cdot (g^a)^{\tilde{t} - a^q + \frac{as}{b\cdot(\mathsf{ID}-\mathsf{ID}^*)}} = g^{a\tilde{t}} \cdot (g^{a^2 s/b})^{\frac{1}{\mathsf{ID}-\mathsf{ID}^*}},\\
K_1 =& \left(u^{\mathsf{ID}} h\right)^{-t} = \left(u^{\mathsf{ID}} h\right)^{-\tilde{t}} \cdot \left(u^{\mathsf{ID}} h\right)^{a^q - \frac{as}{b\cdot(\mathsf{ID}-\mathsf{ID}^*)}}\\
=& \left(u^{\mathsf{ID}} h\right)^{-\tilde{t}} \left(g^{\tilde{t}}/K_2\right)^{\tilde{u}\cdot\mathsf{ID}+\tilde{h}} \cdot \left(\prod_{(j,k)\in[l,n]} \left(g^{a^k/b_j^2}\right)^{M^*_{j,k}\cdot ID} \cdot g^{a^q \cdot \mathsf{ID}} \right)^{a^q - \frac{as}{b\cdot(\mathsf{ID}-\mathsf{ID}^*)}}\\
&\cdot \left(\prod_{(j,k)\in[l,n]} \left(g^{a^k/b_j^2}\right)^{-\rho^*(j) M^*_{j,k}} \cdot \left(g^{a^q}\right)^{-\mathsf{ID}^*} \cdot g^{as/b} \right)^{a^q - \frac{as}{b\cdot(\mathsf{ID}-\mathsf{ID}^*)}}\\
=& \left(\prod_{(j,k)\in[l,n]} \left(g^{a^{k+q}/b_j^2}\right)^{M^*_{j,k}\cdot\mathsf{ID}} \cdot g^{a^{2q}\cdot\mathsf{ID}} \right)\\
&\cdot \left(\prod_{(j,k)\in[l,n]} \left(g^{a^{k+1}s/(b\cdot b_j^2)}\right)^{M^*_{j,k}\cdot\mathsf{ID}} \cdot \left(g^{a^{q+1}s/b}\right)^{\mathsf{ID}} \right)^{-\frac{1}{\mathsf{ID}-\mathsf{ID}^*}}\\
&\cdot \left(\prod_{(j,k)\in[l,n]} \left(g^{a^{k+q}/b_j^2}\right)^{-\rho^*(j)\cdot M^*_{j,k}} \cdot \left(g^{a^2 q}\right)^{-\mathsf{ID}^*} \cdot g^{a^{q+1}s/b} \right)\\
&\cdot \left(\prod_{(j,k)\in[l,n]} \left(g^{a^{k+1}s/(b\cdot b_j^2)}\right)^{-\rho^*(j)\cdot M^*_{j,k}} \cdot \left(g^{a^{q+1}s/b}\right)^{-\mathsf{ID}^*} \cdot g^{(as)^2/b^2} \right)^{-\frac{1}{\mathsf{ID}-\mathsf{ID}^*}},\\
K_2 =& g^t = g^{\tilde{t} - q^q + \frac{as}{b\cdot(\mathsf{ID}-\mathsf{ID}^*)}} = g^{\tilde{t}} \cdot \left(g^{a^q}\right)^{-1} \cdot \left(g^{as/b}\right)^{\frac{1}{\mathsf{ID}-\mathsf{ID}^*}}.
\end{aligned}
$$

Then $\mathcal{B}$ returns $sk_{\mathsf{ID}} = (K_0, K_1, K_2)$ to $\mathcal{A}$.

- $\mathcal{O}^{\mathcal{S}}_{\mathsf{KeyGen}}$. Assume the users whose attributes satisfy $\mathbb{A}^*$ are honest, and the others are corrupted. $\mathcal{A}$ makes queries on secret keys related to corrupted attributes to simulator $\mathcal{B}$. If $\mathcal{S} \models \mathbb{A}^*$, $\mathcal{B}$ rejects the query and returns $\perp$ to $\mathcal{A}$, otherwise, $\mathcal{B}$ generates the secret key related to $\mathcal{S}$ as follows.

  $\mathcal{B}$ firstly calculates the common part v^{-r} for the terms of $K_{\tau,2}, K_{\tau,3}$ as

$$\begin{aligned}
& v^{-\tilde{r}} \cdot \left(g^{\tilde{v}} \prod_{(j,k)\in[\ell,n]} g^{a^k M^*_{j,k}/b_j} \right)^{-\sum_{i\in[n]} w_i a^{q+1-i}} \\
= {} & v^{-\tilde{r}} \cdot \prod_{i\in[n]} \left(g^{a^{q+1-i}} \right)^{-\tilde{v} w_i} \cdot \prod_{(i,j,k)\in[n,\ell,n]} g^{-w_i M^*_{j,k} a^{q+1+k-i}/b_j} \\
= {} & \underbrace{v^{-\tilde{r}} \cdot \prod_{i\in[n]} \left(g^{a^{q+1-i}} \right)^{-\tilde{v} w_i} \cdot \prod_{\substack{(i,j,k)\in[n,\ell,n] \\ i\neq k}} \left(g^{\frac{a^{q+1+k-i}}{b_j}} \right)^{-w_i M^*_{j,k}}}_{\Phi} \\
& \cdot \prod_{(i,j)\in[n,\ell]} g^{-w_i M^*_{j,i} a^{q+1}/b_j} \\
= {} & \Phi \cdot \prod_{\substack{j\in[\ell] \\ \rho^*(j)\notin\mathcal{S}}} g^{-\left\langle \vec{w}, \vec{M}^*_j \right\rangle a^{q+1}/b_j}.
\end{aligned}$$

  As the Φ part can be calculated by $\mathcal{B}$ through the assumption items. $\mathcal{B}$ only needs to cancel the other part by the $(u^{A_\tau} h)^{r_\tau}$ part. It implicitly sets

$$\begin{aligned}
r_\tau &= \tilde{r}_\tau + r \cdot \sum_{\substack{i'\in[\ell] \\ \rho^*(i')\notin\mathcal{S}}} \frac{b_{i'}}{A_\tau - \rho^*(i')} \\
&= \tilde{r}_\tau + \tilde{r} \cdot \sum_{\substack{i'\in[\ell] \\ \rho^*(i')\notin\mathcal{S}}} \frac{b_{i'}}{A_\tau - \rho^*(i')} + \sum_{\substack{(i,i')\in[n,\ell] \\ \rho^*(i')\notin\mathcal{S}}} \frac{w_i b_{i'} a^{q+1-i}}{A_\tau - \rho^*(i')}.
\end{aligned}$$

The K_0, K_1 part can be calculated as

$$K_0 = g^{\beta} w^{r} = g^{\tilde{\beta}+a^{q+1}} \left(g^{a}\right)^{\tilde{r}+\sum_{i\in[n]} w_i a^{q+1-i}} = g^{\tilde{\beta}} \cdot g^{a^{q+1}} \cdot g^{a\tilde{r}} \cdot \prod_{i\in[n]} g^{w_i a^{q+2-i}}$$

$$= g^{\tilde{\beta}} \cdot \left(g^{a}\right)^{\tilde{r}} \cdot \prod_{i=2}^{n} \left(g^{a^{q+2-i}}\right)^{w_i},$$

$$K_1 = g^{r} = g^{\tilde{r}} \cdot \prod_{i\in[n]} \left(g^{a^{q+1-i}}\right)^{w_i}.$$

For the $K_{\tau,3}$ part, $\mathcal{B}$ calculates $(u^{A_\tau} h)^{r_\tau}$ as

$$\begin{aligned}
&\left(u^{A_\tau} h\right)^{r_\tau} \\
&= \left(u^{A_\tau} h\right)^{\tilde{r}_\tau} \cdot \left(K_{\tau,2}/g^{\tilde{r}_\tau}\right)^{\tilde{u}A_\tau+\tilde{h}} \cdot \prod_{\substack{(i',j,k)\in[l,l,n]\\ \rho^*(i')\notin\mathcal{S}}} \left(g^{a^k \cdot b_{i'}/b_j^2}\right)^{\tilde{r}\cdot M^*_{j,k}\cdot \frac{A_\tau-\rho^*(j)}{A_\tau-\rho^*(i')}} \\
&\quad \cdot \prod_{\substack{(i,i',j,k)\in[n,l,l,n]\\ \rho^*(i')\notin\mathcal{S}}} \left(g^{a^{k+q-i-1} b_{i'}/b_j^2}\right)^{M^*_{j,k}\frac{w_i(A_\tau-\rho^*(j))}{A_\tau-\rho^*(i')}} \cdot \prod_{\substack{(j,k)\in[l,n]\\ \rho^*(j)\notin\mathcal{S}}} \left(g^{a^k/b_j}\right)^{\tilde{r}\cdot M^*_{j,k}} \\
&\quad \cdot \prod_{\substack{(i,j,k)\in[n,l,n]\\ \rho^*(j)\notin\mathcal{S}}} \left(g^{a^{k+q-i+1} b_i/b_j^2}\right)^{M^*_{j,k}\cdot w_i} \cdot \prod_{\substack{i'\in[l]\\ \rho^*(i')\notin\mathcal{S}}} \left(\left(g^{asb_{i'}/b}\right)^{\frac{\tilde{r}}{A_\tau-\rho^*(i')}} \cdot \left(g^{a^q b_{i'}}\right)^{\tilde{r}\cdot\frac{A_\tau-\mathsf{ID}^*}{A_\tau-\rho^*(i')}} \right) \\
&\quad \cdot \prod_{\substack{(i,i')\in[n,l]\\ \rho^*(i')\notin\mathcal{S}}} \left(\left(g^{a^{2q-i+1} b_{i'}}\right)^{\frac{w_i(A_\tau-\mathsf{ID}^*)}{A_\tau-\rho^*(i')}} \cdot \left(g^{a^{q-i+2} s b_{i'}/b}\right)^{\frac{w_i}{A_\tau-\rho^*(i')}} \right) \\
&= \Psi \cdot \prod_{\substack{j\in[\ell]\\ \rho^*(j)\notin\mathcal{S}}} g^{\left\langle \vec{w}, \vec{M}^*_j \right\rangle a^{q+1}/b_j} \cdot \prod_{\substack{(j,k)\in[l,n]\\ \rho^*(j)\notin\mathcal{S}}} \left(g^{a^k/b_j}\right)^{\tilde{r}\cdot M^*_{j,k}} \\
&\quad \cdot \prod_{\substack{(i,j,k)\in[n,l,n]\\ \rho^*(j)\notin\mathcal{S}}} \left(g^{a^{k+q-i+1} b_i/b_j^2}\right)^{M^*_{j,k}\cdot w_i} \cdot \prod_{\substack{i'\in[l]\\ \rho^*(i')\notin\mathcal{S}}} \left(\left(g^{asb_{i'}/b}\right)^{\frac{\tilde{r}}{A_\tau-\rho^*(i')}} \cdot \left(g^{a^q b_{i'}}\right)^{\tilde{r}\cdot\frac{A_\tau-\mathsf{ID}^*}{A_\tau-\rho^*(i')}} \right) \\
&\quad \cdot \prod_{\substack{(i,i')\in[n,l]\\ \rho^*(i')\notin\mathcal{S}}} \left(\left(g^{a^{2q-i+1} b_{i'}}\right)^{\frac{w_i(A_\tau-\mathsf{ID}^*)}{A_\tau-\rho^*(i')}} \cdot \left(g^{a^{q-i+2} s b_{i'}/b}\right)^{\frac{w_i}{A_\tau-\rho^*(i')}} \right) \\
&= \Omega \cdot \prod_{\substack{j\in[\ell]\\ \rho^*(j)\notin\mathcal{S}}} g^{\left\langle \vec{w}, \vec{M}^*_j \right\rangle a^{q+1}/b_j}.
\end{aligned}$$

Where we have the Ψ part and the Ω part as

$$\Psi = \left(u^{A_\tau} h\right)^{\tilde{r}_\tau} \cdot \left(K_{\tau,2}/g^{\tilde{r}_\tau}\right)^{\tilde{u}A_\tau + \tilde{h}} \cdot \prod_{\substack{(i',j,k)\in[l,l,n] \\ \rho^*(i')\notin\mathcal{S}}} \left(g^{a^k \cdot b_{i'}/b_j^2}\right)^{\tilde{r}\cdot M^*_{j,k} \cdot \frac{A_\tau - \rho^*(j)}{A_\tau - \rho^*(i')}}$$

$$\cdot \prod_{\substack{(i,i',j,k)\in[n,l,l,n] \\ i'\neq j \vee i\neq k, \rho^*(i')\notin\mathcal{S}}} \left(g^{a^{k+q-i-1} b_{i'}/b_j^2}\right)^{M^*_{j,k} \frac{w_i(A_\tau - \rho^*(j))}{A_\tau - \rho^*(i')}},$$

$$\Omega = \Psi \cdot \prod_{\substack{(j,k)\in[l,n] \\ \rho^*(j)\notin\mathcal{S}}} \left(g^{a^k/b_j}\right)^{\tilde{r}\cdot M^*_{j,k}} \cdot \prod_{\substack{(i,j,k)\in[n,l,n] \\ \rho^*(j)\notin\mathcal{S}}} \left(g^{a^{k+q-i+1} b_i/b_j^2}\right)^{M^*_{j,k}\cdot w_i}$$

$$\cdot \prod_{\substack{i'\in[l] \\ \rho^*(i')\notin\mathcal{S}}} \left(\left(g^{asb_{i'}/b}\right)^{\frac{\tilde{r}}{A_\tau - \rho^*(i')}} \cdot \left(g^{a^q b_{i'}}\right)^{\tilde{r}\cdot \frac{A_\tau - \mathsf{ID}^*}{A_\tau - \rho^*(i')}}\right)$$

$$\cdot \prod_{\substack{(i,i')\in[n,l] \\ \rho^*(i')\notin\mathcal{S}}} \left(\left(g^{a^{2q-i+1} b_{i'}}\right)^{\frac{w_i(A_\tau - \mathsf{ID}^*)}{A_\tau - \rho^*(i')}} \cdot \left(g^{a^{q-i+2} s b_{i'}/b}\right)^{\frac{w_i}{A_\tau - \rho^*(i')}}\right),$$

$$K_{\tau,2} = g^{r_\tau}$$

$$= g^{\tilde{r}_\tau} \cdot \prod_{\substack{(i')\in[\ell] \\ \rho^*(i')\notin\mathcal{S}}} \left(g^{b_{i'}}\right)^{\tilde{r}/(A_\tau - \rho^*(i'))} \cdot \prod_{\substack{(i,i')\in[n,l] \\ \rho^*(i')\notin\mathcal{S}}} \left(g^{b_{i'} a^{q+1-i}}\right)^{w_i/(A_\tau - \rho^*(i'))}.$$

Then Ψ and $K_{\tau,2}$ can be calculated using the suitable terms of the q-BDHE assumption [14], and Ω can be calculated with Ψ and the terms of the modified BDHE assumption. The second part of $(u^{A_\tau}h)^{r_\tau}$ cancels the second part of v^{-r}. In this way, all the terms of $K_0, K_1, \{K_{\tau,2}, K_{\tau,3}\}_{\tau\in[|\mathcal{S}|]}$ can be calculated by $\mathcal{B}$. Then $\mathcal{B}$ can calculate the proper secret keys and send them to $\mathcal{A}$.

- $\mathcal{O}^{\mathsf{ID}\to A}_{\mathsf{RKGen}}$. The adversary $\mathcal{A}$ makes re-encryption key queries to $\mathcal{B}$ by submitting an identity ID and an access policy $\mathbb{A} = (M, \rho)$ where M is an $\ell \times n$ matrix and $\ell, n \le q$. If $\mathsf{ID} = \mathsf{ID}^*$ and $\mathbb{A} \not\subseteq \mathbb{A}^*$, $\mathcal{B}$ rejects the query and returns $\perp$. For $\mathsf{ID} = \mathsf{ID}^*$ and $\mathbb{A} \subseteq \mathbb{A}^*$, $\mathcal{A}$ first calls $sk_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}_{\mathsf{ID}}(pp, msk, \mathsf{ID})$ to calculate the secret key sk_{ID}, then it picks a random exponent $t'' \in \mathbb{Z}_p$ a random component $R \in \mathbb{G}$ and calculates

$$d_0 = K_0 \cdot f^{t''}, d_1 = K_1, d_2 = K_2, d_{i,3} = w^{\lambda'_i} v^{t'_i}, d_{i,4} = \left(u^{\rho(i)} h\right)^{-t'_i}$$
$$d_{i,5} = g^{t'_i}, d_6 = F\left(e(g,g)^{\beta s'}\right) \cdot R, d_7 = g^{s'}.$$
$$rk_{\mathsf{ID}\to\mathbb{A}} = \left(d_0, d_1, \{d_{i,3}, d_{i,4}, d_{i,5}\}_{i\in[\ell]}, d_6, d_7\right).$$

Then for each query, the simulator sends $rk_{\mathsf{ID}\to\mathbb{A}}$ to $\mathcal{A}$.

- $\mathcal{O}^{\mathsf{ID}}_{\mathsf{Encrypt}}$. The simulator $\mathcal{B}$ maintains a set $\mathcal{C}$, for a query of message m from $\mathcal{A}$, it calculates $ct = \mathsf{Encrypt}(\mathsf{ID}^*, m_i)$ and adds ct to $\mathcal{C}$. Then $\mathcal{B}$ sends ct to $\mathcal{A}$.

- $\mathcal{O}^{\mathsf{ID}\to\mathcal{S}}_{\mathsf{ReEnc}}$. $\mathcal{A}$ makes re-encryption queries to $\mathcal{B}$. For a query with input $(rk_{\mathsf{ID}\to\mathbb{A}}, ct)$, if $ct \notin \mathcal{C}$, $\mathcal{B}$ rejects the query and returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ runs $ct' \leftarrow \mathsf{ReEnc}(pp, rk_{\mathsf{ID}\to\mathbb{A}}, ct)$ and returns ct' to $\mathcal{A}$.

Challenge. The adversary $\mathcal{A}$ sends two messages of equal length (m_0, m_1) to the simulator. Then the simulator flips a random coin $b \in \{0,1\}$ and constructs

$$C = m_b \cdot T \cdot e(g,g)^{\tilde{\alpha}s}, C_0 = g^s, C_3 = f^s.$$

where T is the challenge term and g^s is the corresponding term of the assumption. $\mathcal{B}$ sets implicitly $t = \tilde{t} + b$ and calculates:

$$\begin{aligned}
C_1 &= g^t = g^{\tilde{t}} \cdot g^b,\\
C_2 &= \left(u^{\mathsf{ID}^*} h\right)^t \cdot w^{-s}\\
&= \begin{pmatrix} \left(g^{\tilde{u}} \cdot \prod_{(j,k)\in[l,n]} \left(g^{a^k/b_j^2}\right)^{M^*_{j,k}} \cdot g^{a^q}\right)^{\mathsf{ID}^*} \cdot g^{\tilde{h}} \cdot \\ \prod_{(j,k)\in[l,n]} \left(g^{a^k/b_j^2}\right)^{-\rho^*(j) M^*_{j,k}} \cdot \left(g^{a^q}\right)^{-\mathsf{ID}^*} \cdot g^{as/b} \end{pmatrix}^{\tilde{t}+b} \cdot g^{-as}\\
&= g^{\tilde{u}\tilde{t}\cdot\mathsf{ID}^*} \cdot \left(g^b\right)^{\tilde{u}\cdot\mathsf{ID}^*} \cdot \left(g^{a^q}\right)^{-\tilde{t}\cdot\mathsf{ID}^*} \cdot \left(g^{a^q b}\right)^{-\mathsf{ID}^*} \cdot \left(g^{as/b}\right)^{\tilde{t}} \cdot g^{\tilde{h}\tilde{t}} \cdot \left(g^b\right)^{\tilde{h}}\\
&\quad \cdot \prod_{(j,k)\in[l,n]} \left(g^{a^k/b_j^2}\right)^{\tilde{t}\cdot M^*_{j,k}\cdot(\mathsf{ID}^* - \rho^*(j))} \cdot \prod_{(j,k)\in[l,n]} \left(g^{a^k b/b_j^2}\right)^{M^*_{j,k}\cdot(\mathsf{ID}^* - \rho^*(j))}.
\end{aligned}$$

Then $\mathcal{B}$ sends $ct_b = (C, C_0, C_1, C_2, C_3)$ to $\mathcal{A}$.

Phase 2. This phase is the same as Phase 1.

Guess. $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$. If $T = e(g,g)^{a^{q+1}s}$, $\mathcal{A}$ and $\mathcal{B}$ played a well-formed security game, on the other hand, if T is a random term in $\mathbb{G}_T$, then played a random game. The advantage of the adversary in breaking the security game is

$$\mathsf{Adv}_{\mathcal{A}}^{(2)} = |\Pr[b' = b] - 1/2|.$$

As a result, if $\mathcal{A}$ breaks the security game with a non negligible advantage, then $\mathcal{B}$ has a non negligible advantage in breaking the M-BDHE assumption. This concludes the proof of Lemma 2.

Through Lemma 1 and Lemma 2 we can see that Game_{real} and Game_{final} are indistinguishable from the view of the adversary and Game_{final} perfectly simulates the game between the adversary and the challenger, the advantage of the adversary in breaking the joint games in Lemma 1 and Lemma 2 is

$$\mathsf{Adv}_{\mathcal{A}} = \frac{|\Pr[b' = b] - 1/2|}{Q \cdot \mathsf{poly}(k)}.$$

In summary, if $\mathcal{A}$ breaks the security games defined in Lemma 1 and Lemma 2, then $\mathcal{B}$ has a non negligible advantage in breaking the M-BDHE assumption, and this proves Theorem 1.

7. Evaluation

As we focus on the scenarios of data sharing from an individual to multiple people, we mainly compare our HyPRE scheme with the state-of-the-art IB-BPRE scheme [4] and ABPRE scheme [29] for similar scenarios in KeyGen, RKGen, ReEnc and $\mathsf{Decrypt}$ algorithms.

7.1. Theoretical Analysis

We first compare our HyPRE scheme on the time complexity of different algorithms with [6] and [4]. As Table 2 shows, our HyPRE scheme achieves $\mathcal{O}(1) \cdot (e+m)$ encryption complexity, which is the same as the state-of-the-art IB-BPRE scheme [4] while [6] achieved $\mathcal{O}(\ell) \cdot (e+m)$, and our scheme achieves better encryption performance than scheme [6], which is related to the data owner. Scheme [6] achieves only $\mathcal{O}(1) \cdot (m+p)$ time complexity on the decryption of the re-encrypted ciphertexts, which is better than our scheme and the scheme of [4], this is because it mainly focuses on the decryption efficiency of authorized clients, which may be lightweight devices. In addition, our scheme has the similar performance for the $\mathsf{Encrypt}$, ReEnc and

Table 2: Computational Complexity Comparison with [6] and [4]

Schemes	Encrypt	RKGen	ReEnc	$\mathsf{Decrypt}_{\mathsf{Ori}}$	$\mathsf{Decrypt}_{\mathsf{Re}}$
[6]	$\mathcal{O}(\ell)\cdot(e+m)$	$\mathcal{O}(1)\cdot(e+m)$	$\mathcal{O}(\ell)\cdot(m+p)$	$\mathcal{O}(\ell)\cdot(m+p)$	$\mathcal{O}(1)\cdot(m+p)$
[4]	$\mathcal{O}(1)\cdot(e+m)$	$\mathcal{O}(2^k)\cdot e+\mathcal{O}(1)\cdot m$	$\mathcal{O}(1)\cdot(m+p)$	$\mathcal{O}(1)\cdot(m+p)$	$\mathcal{O}(2^k)\cdot e+\mathcal{O}(k)(m+p)$
Our Scheme	$\mathcal{O}(1)\cdot(e+m)$	$\mathcal{O}(\ell)\cdot(e+m)$	$\mathcal{O}(1)\cdot(m+p)$	$\mathcal{O}(1)\cdot(m+p)$	$\mathcal{O}(\ell)\cdot(m+p)$

- The symbol k indicates the size of the identity set in IBBE.
- The symbol ℓ indicates the size of the attribute set in ABE.
- The symbols m, e, p indicate one time of multiply operation, exponential operation and pairing operation respectively.
- $\mathsf{Decrypt}_{\mathsf{Ori}}$ indicates the decryption algorithm for the original ciphertexts.
- $\mathsf{Decrypt}_{\mathsf{Re}}$ incicates the decryption algorithm for the re-encrypted ciphertexts.

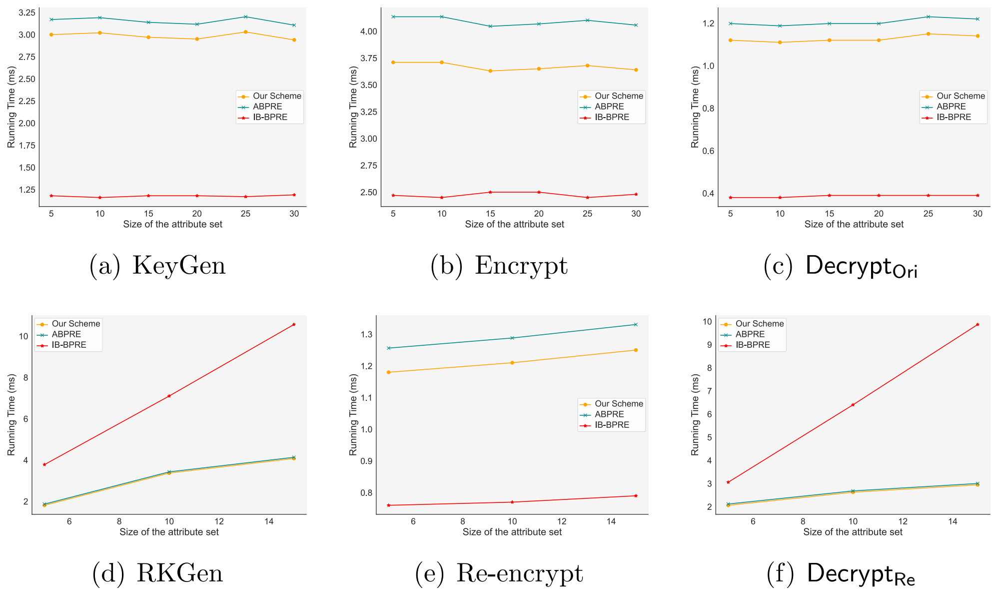
$\mathsf{Decrypt}_{\mathsf{Ori}}$ algorithms with [4], but our scheme is more expressive, because our scheme supports complex logical expression between attributes (access policy) while [4] only supports identity-based broadcast encryption (IBBE), which can be regarded as simple encryption of an identity set. It should be pointed out that the authors claimed their IB-BPRE scheme achieved the time complexity of $\mathcal{O}(|S|)$ ($|S|$ is the size of the identity set) on the RKGen, ReEnc and $\mathsf{Decrypt}_{\mathsf{Re}}$ algorithms. However, in the RKGen algorithm, the authors calculated the rk_5 part as $rk_5 = g^{s\cdot\prod_{id\in\mathcal{S}}(\alpha+H_1(id))}$. In this way, as **the data owner has no access to** α, which is the master key, it has to first calculate the $g^{\prod_{id\in\mathcal{S}}(\alpha+H_1(id))}$ part as:

$$
\begin{aligned}
& g^{\prod_{id\in\mathcal{S}}(\alpha+H_1(id))} \\
= \; & g^{(\alpha+H_1(id_1))\cdot\ldots\cdot\left(\alpha+H_1(id_{|S|})\right)} \\
= \; & g^{\alpha^k+\alpha^{k-1}\cdot\sum_{i\in[k]}H_1(id_i)}\cdot g^{\alpha^{k-2}\cdot\sum_{i,j\in[|S|],j>i}H_1(id_i)\cdot H_1(id_j)+\cdots+\prod_{i\in[|S|]}H_1(id_i)}.
\end{aligned}
$$

This takes the time complexity of $\mathcal{O}(2^{|S|})$, which is obviously impractical. Besides, similar calculations need to be performed in their $\mathsf{Decrypt}$ algorithm.

7.2. Implementation

In this paper, we evaluate the performance of our HyPRE scheme and the related schemes from two dimensions: time cost and storage/communication cost. To measure the time cost, we test the time cost of each algorithm when the number of attributes is set to be from 5 to 30 respectively. Our



(a) KeyGen (b) Encrypt (c) $\mathsf{Decrypt}_{\mathsf{Ori}}$

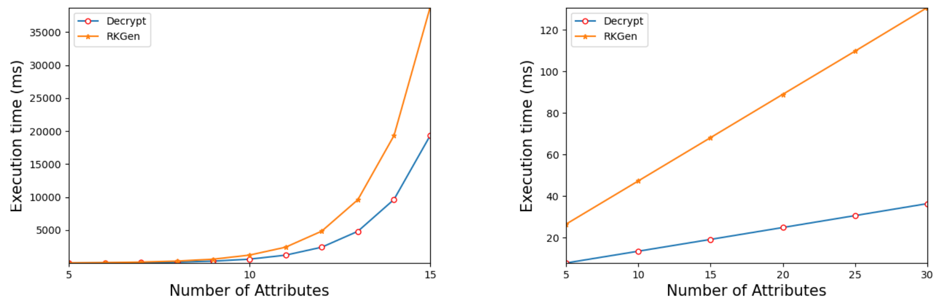
(d) RKGen (e) Re-encrypt (f) $\mathsf{Decrypt}_{\mathsf{Re}}$

Figure 2: Comparison of relations between execution time and number of attributes.

HyPRE scheme is implemented based on the PBC[1] library. The algorithms are built on Ubuntu 22.04 LTS Desktop with a 2.4 GHz Intel(R) Core(TM) i9-12900 CPU and 64 GB RAM. For the sake of accuracy, we execute each algorithm for 300 times and calculate the average running time. Note that as the Setup algorithm is only executed once when the system is initialized, the running time of the algorithm is not counted. To evaluate the storage/communication costs of the schemes, we test the length of the keys (private keys and re-encryption keys) and ciphertexts (original ciphertexts and re-encrypted ciphertexts) to show the storage/communication overhead of the related schemes.

Figure 2 shows the execution time of different algorithms of our scheme and the scheme of [4] separately. It can be seen that scheme [4] performs better in key generation and encryption phases, however, in re-encryption key generation and decryption of the re-encrypted ciphertext, as the number of identities increases, the time cost of their scheme increases exponentially, which has been discussed in the above theoretical analysis. On the other

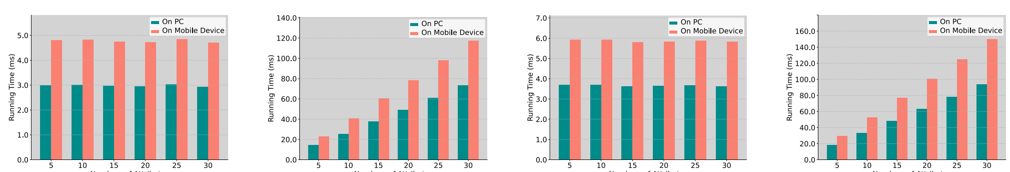
[1]https://crypto.stanford.edu/pbc/



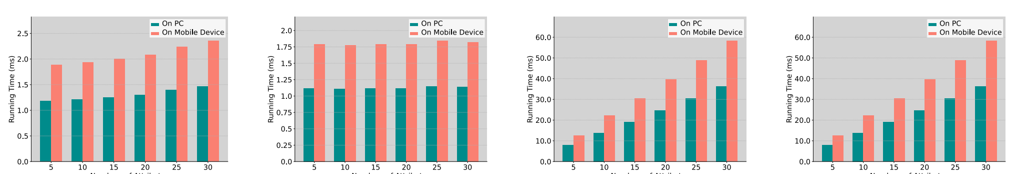
(a) IB-BPRE [4]　　(b) Our Work

Figure 3: Comparison of relations between execution time and number of attributes.

hand, as is shown in Figure 2, the time cost of the corresponding algorithms in our scheme increases linearly with the increase of attributes. Figure 3(a) and Figure 3(b) more directly reflect the growth of time cost of corresponding algorithms.



(a) Encrypt　　(b) RKGen　　(c) Decryption　　(d) Decryption



(e) Encrypt　　(f) RKGen　　(g) Decryption　　(h) Decryption

Figure 4: Comparison of execution time on PC and Mobile Device

In addition, we test our HyPRE scheme over different security parameters (i.e. different elliptic curve parameters). Figure 4 shows the comparison of the running time of each algorithms in HyPRE on "Curve P-256" and "Curve D224". The statistical security level of "Curve D224" is approximately 166

Table 3: Real-world testing of our proposed framework

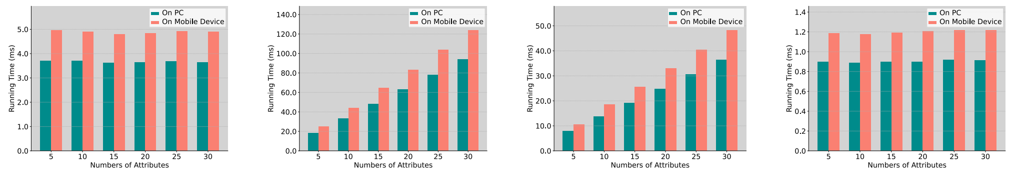
Data	Running Time (in milliseconds)				CT Length (bytes)	
	HyPRE.Enc	AES.Enc	HyPRE.Dec	AES.Dec	HyPRE.CT	AES.CT
2^{16}	2.40	2.17	1.13	0.03	640	65552
2^{18}	2.39	2.49	1.12	0.07	640	262160
2^{20}	2.41	3.21	1.13	0.27	640	1048592
2^{22}	2.40	4.45	1.13	1.06	640	4194320
2^{24}	2.38	13.32	1.13	4.20	640	16777232
2^{26}	2.40	49.14	1.13	16.85	640	67108880
2^{28}	2.39	191.66	1.13	66.37	640	268435472
2^{30}	2.39	761.55	1.13	265.331	640	1073741840

bits. "Curve P-256" or "secp256r1" in the NIST standard. The statistical security level of Curve P-256 is 256 bits.

Real-world experiment of HyPRE

We tested the computation costs and storage overheads of our framework on Amazon EC2, a real-world cloud service platform. To improve the efficiency of encryption over big data, we utilize the technique of digital envelop, that is, we encrypt the original data with symmetric encryption schemes such as AES and then encrypt the secret key of AES using the encryption algorithm of our HyPRE scheme. Table 3 shows the detailed results on the running time of different modules and the storage overheads of ciphertexts for different sizes of plaintext data, the tested data size varies from kilobytes (2^{16} bytes) to gigabytes (2^{30} bytes). The results shows that for a fixed-size access policy, with the increase of plaintext data, the running time of different algorithms of our HyPRE scheme remain stable. In real-world applications, when the amount of data is large, the time cost of our HyPRE scheme is negligible.

We also tested our HyPRE scheme on a mobile device with Dimensity 9000 Octa-core Max 3.05 GHz CPU and 12 GB RAM, and the operating system version is Android 13.0. As depicted in Figure 5, we compare the execution time of the $\mathsf{Encrypt}, \mathsf{RKGen}$ and $\mathsf{Decrypt}$ algorithms (in this part, we omit the other algorithms of HyPRE, because they are usually deployed



(a) Encrypt (b) RKGen (c) Decryption (d) Decryption

Figure 5: Comparison of execution time on PC and Mobile Device

on cloud servers), although the result shows a decrease of about 20% on computation efficiency on the mobile device compared with that on the PC, our HyPRE scheme still achieves a time cost of under 5 milliseconds for the encryption algorithm. What's more, the time costs of the RKGen and Decrypt for the original ciphertext performs linearly with the increase of the attribute numbers.

8. Conclusion

In this paper, we have constructed a novel hybrid proxy re-encryption (HyPRE) scheme which supports the transformation from ciphertexts of IBE to ciphertext of ABE via a semi-trusted proxy. Compared with related schemes, our HyPRE scheme better supports the scenario of data sharing from an individual to multiple people in the cloud environment. To overcome the incompleteness of traditional CPA security for PRE schemes, we extended the concept of honest re-encryption attacks (HRA) to HyPRE and prove our scheme is secure under HRA. We theoretically analyzed the performance of our scheme and related comparison schemes. Finally, we gave the implementation of the scheme and carried out experimental analysis to show that our HyPRE scheme is efficient compared with the related cryptographic schemes.

However, although we improved the incompleteness of the security proof in [6], we introduced an additional master secret key to our scheme, leading to more public parameters, which makes the proposed schemes a little bit redundant, and we regard the simplification of the scheme as a future work.

Table A.4: Matrix A and target vector $\tilde{A}^0$ for our M-BDHE assumption

Type	Terms	a	s	b	b_1	b_2	...	b_q
1	g	0	0	0	0	0	0	0
2	g^a	1	0	0	0	0	0	0
3	g^s	0	1	0	0	0	0	0
4	g^b	0	0	1	0	0	0	0
5	$g^{a^{2q}}$	$2q$	0	0	0	0	0	0
6	$g^{as/b}$	1	1	-1	0	0	0	0
7	$g^{a^2s/b}$	2	1	-1	0	0	0	0
8	$g^{(as)^2/b^2}$	2	2	-2	0	0	0	0
9	g^{a^i} $\forall i \in [q]$	i	0	0	0	0	0	0
10	g^{b_j} $\forall j \in [q]$	0	0	0	$[j:1]$			
11	$g^{asb_j/b}$ $\forall j \in [q]$	1	1	-1	$[j:1]$			
12	g^{a^ib/b_j^2} $\forall i \in [q], j \in [q]$	i	0	1	$[j:-2]$			
13	$g^{a^ib_j/b_{j'}^2}$ $\forall i \in [2q], j \in [q], j' \in [q]$ with $j \neq j'$	i	0	0	$[j:1, j':-2]$			
14	g^{a^i/b_j} $\forall i \in [2q], j \in [q]$ with $i \neq q+1$	i	0	0	$[j:-1]$			
15	$g^{a^ib_j}$ $\forall i \in [2q], j \in [q]$	i	0	0	$[j:1]$			
16	g^{a^i/b_j^2} $\forall i \in [2q], j \in [q]$	i	0	0	$[j:-2]$			
17	$g^{\frac{(as)^2b_i}{b_j}}$ $\forall\ (i,j) \in [q,q]$ with $i \neq j$	2	2	0	$[i:1, j:-1]$			
$\hat{A}^0$	$e(g,g)^{sa^{q+1}}$	$q+1$	1	0	0	0	0	0

Appendix A. Proof of the M-BDHE assumption

We utilize the proof technique in [14] to prove our M-BDHE assumption is secure in the generic group model.

Lemma 3. *The M-BDHE assumption is secure in the generic group model.*

Proof. We use the same notations as in [14], where we let $[i : x]$ and $[i : x, i' : y]$ denotes the row vectors in $\mathbb{Z}^{1\times q}$ with all components equal to 0, except the i-th component for the first vector and the i,i'-th positions for the second. The non zero elements are x for the first vector and x, y for the i, i'-th positions, respectively, of the second vector. Table A.4 shows a compact form of the matrix A where rows of similar type are shown in one line.

In this way, We only need to show that by adding any two rows of matrix A the row vector $\tilde{A}^0 = (q + 1, 1, 0, 0, \cdots, 0)$ can not be calculated. By inspecting Table A.4 we can easily see that we have to check only the rows of types 3, 6, 7 and 11, which have 1 in the s column.

The only rows that can be added to row 3 and give all zero's in the b_i columns are row 1, 2, 5 and 9. We cannot get the $q + 1$ component in the a

column through any of these rows. Rows of type 4 and type 12 can be added to rows of type 6, 7, and 11 to get zero in the b column, however, we cannot get the $q+1$ component for rows of type 4 and cannot get zeros in the b_i columns for rows of type 12. As a result, according to "*Corollary D.4*" in [14], the M-BDHE assumption is secure in the generic group model. □

References

[1] A. Cohen, What about bob? the inadequacy of cpa security for proxy reencryption, in: IACR International Workshop on Public Key Cryptography, Springer, Berlin Heidelberg, Beijing, China, 2019, pp. 287–316.

[2] G. Fuchsbauer, C. Kamath, K. Klein, K. Pietrzak, Adaptively secure proxy re-encryption, in: IACR International Workshop on Public Key Cryptography, Springer, Berlin Heidelberg, Beijing, China, 2019, pp. 317–346.

[3] H. Wang, Z. Cao, L. Wang, Multi-use and unidirectional identity-based proxy re-encryption schemes, Information Sciences 180 (20) (2010) 4042–4059.

[4] G. Chunpeng, Z. Liu, J. Xia, F. Liming, Revocable identity-based broadcast proxy re-encryption for data sharing in clouds, IEEE Transactions on Dependable and Secure Computing 18 (3) (2019) 1214–1226.

[5] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, Y. Yu, A. Yang, A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing, Future Generation Computer Systems 52 (2015) 95–108.

[6] H. Deng, Z. Qin, Q. Wu, Z. Guan, Y. Zhou, Flexible attribute-based proxy re-encryption for efficient data sharing, Information Sciences 511 (2020) 94–113.

[7] Q. Mei, M. Yang, J. Chen, L. Wang, H. Xiong, Expressive data sharing and self-controlled fine-grained data deletion in cloud-assisted iot, IEEE Transactions on Dependable and Secure Computing (2022).

[8] H. Song, F. Yin, X. Han, T. Luo, J. Li, Mpds-rca: Multi-level privacy-preserving data sharing for resisting collusion attacks based on an integration of cp-abe and ldp, Computers & Security 112 (2022) 102523.

[9] J. Liu, B. Zhao, J. Qin, X. Zhang, J. Ma, Multi-keyword ranked searchable encryption with the wildcard keyword for data sharing in cloud computing, The Computer Journal 66 (1) (2023) 184–196.

[10] D. Boneh, M. Franklin, Identity-based encryption from the weil pairing, in: Annual international cryptology conference, Springer, Berlin Heidelberg, Santa Barbara, California, USA, 2001, pp. 213–229.

[11] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: Annual international conference on the theory and applications of cryptographic techniques, Springer, Berlin Heidelbergv, Aarhus, Denmark, 2005, pp. 457–473.

[12] B. Waters, Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization, in: International Workshop on Public Key Cryptography, Springer, Berlin Heidelberg, Taormina, Italy, 2011, pp. 53–70.

[13] A. Lewko, B. Waters, Unbounded hibe and attribute-based encryption, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin Heidelberg, Tallinn, Estonia, 2011, pp. 547–567.

[14] Y. Rouselakis, B. Waters, New constructions and proof methods for large universe attribute-based encryption, Cryptology ePrint Archive, `https://eprint.iacr.org/2012/583` (2012).

[15] M. Blaze, G. Bleumer, M. Strauss, Divertible protocols and atomic proxy cryptography, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin Heidelberg, Espoo, Finland, 1998, pp. 127–144.

[16] X. Liang, Z. Cao, H. Lin, J. Shao, Attribute based proxy re-encryption with delegating capabilities, in: International Symposium on Information, Computer, and Communications Security, Association for Computing Machinery, New York, NY, USA, Sydney, Australia, 2009, pp. 276–286.

[17] S. Luo, J. Hu, Z. Chen, Ciphertext policy attribute-based proxy re-encryption, in: International Conference on Information and Commu-

nications Security, Springer, Berlin Heidelberg, Barcelona, Spain, 2010, pp. 401–415.

[18] K. Liang, L. Fang, D. S. Wong, W. Susilo, A ciphertext-policy attribute-based proxy re-encryption scheme for data sharing in public clouds, Concurrency and Computation: Practice and Experience 27 (8) (2015) 2004–2027.

[19] C. Ge, W. Susilo, L. Fang, J. Wang, Y. Shi, A cca-secure key-policy attribute-based proxy re-encryption in the adaptive corruption model for dropbox data sharing system, Designs, Codes and Cryptography 86 (11) (2018) 2587–2603.

[20] P. Xu, T. Jiao, Q. Wu, W. Wang, H. Jin, Conditional identity-based broadcast proxy re-encryption and its application to cloud email, IEEE Transactions on Computers 65 (1) (2015) 66–79.

[21] R. Canetti, S. Hohenberger, Chosen-ciphertext secure proxy re-encryption, in: Proceedings of the 14th ACM Conference on Computer and Communications Security, Association for Computing Machinery, New York, NY, USA, Alexandria, Virginia, USA, 2007, pp. 185–194.

[22] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE transactions on information theory 31 (4) (1985) 469–472.

[23] G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage, ACM Transactions on Information and System Security 9 (1) (2006) 1–30.

[24] B. Libert, D. Vergnaud, Unidirectional chosen-ciphertext secure proxy re-encryption, in: International Workshop on Public Key Cryptography, Springer, Berlin Heidelberg, Barcelona, Spain, 2008, pp. 360–379.

[25] J. Shao, P. Liu, Y. Zhou, Achieving key privacy without losing cca security in proxy re-encryption, Journal of Systems and Software 85 (3) (2012) 655–665.

[26] G. Hanaoka, Y. Kawai, N. Kunihiro, T. Matsuda, J. Weng, R. Zhang, Y. Zhao, Generic construction of chosen ciphertext secure proxy re-encryption, in: Cryptographers' Track at the RSA Conference, Springer, Berlin Heidelberg, San Francisco, CA, USA, 2012, pp. 349–364.

[27] T. Isshiki, M. H. Nguyen, K. Tanaka, Proxy re-encryption in a stronger security model extended from ct-rsa2012, in: Cryptographers' Track at the RSA Conference, Springer, Berlin Heidelberg, San Francisco, CA, USA, 2013, pp. 277–292.

[28] W. Susilo, P. Dutta, D. H. Duong, P. S. Roy, Lattice-based hra-secure attribute-based proxy re-encryption in standard model, in: European Symposium on Research in Computer Security, Springer, Berlin Heidelberg, Darmstadt, Germany, 2021, pp. 169–191.

[29] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, L. Fang, A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds, IEEE Transactions on Dependable and Secure Computing 19 (5) (2021) 2907–2919.