IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 July 2022, accepted 15 August 2022, date of publication 18 August 2022, date of current version 25 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3200084

RESEARCH ARTICLE

Autonomous Path Identity-Based Broadcast Proxy Re-Encryption for Data Sharing in Clouds

HUIDAN HU, ZHENFU CAO, (Senior Member, IEEE), AND XIAOLEI DONG, (Member, IEEE)
Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China

Corresponding author: Xiaolei Dong (dongxiaolei@sei.ecnu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFA0712300, and in part by the National Natural Science Foundation of China under Grant 62132005 and Grant 62172162.

ABSTRACT Cloud computing with massive storage and computing capabilities has become widespread in actual applications. It is critical to ensure secure data sharing in cloud-based applications. Currently, numerous identity-based broadcast proxy re-encryption (IB-BPRE) schemes have been proposed to resolve the privacy issue. However, the existing IB-BPRE schemes cannot reach the transformation of the decryption right for outsourced encrypted data between the broadcast receiver sets (data user sets) delegated by the data owner (Alice) because it is difficult for the IB-BPRE to hold the character of multi-hop. Consequently, a new cryptographic primitive called autonomous path identity-based broadcast proxy re-encryption (APIB-BPRE) is presented to address the above issue. In an APIB-BPRE scheme, the delegator establishes an autonomous path involving preferred multiple broadcast receiver sets and the proxy can convert the decryption right for the broadcast receiver set into the decryption right for the next broadcast receiver set by the re-encryption key from the delegator. This solution is convenient and flexible for cloud users and utilizes the benefits of cloud computing. The evaluation and comparison indicate that our APIB-BPRE system is effective and practical.

INDEX TERMS Proxy re-encryption, broadcast encryption, cloud data sharing, autonomous path.

I. INTRODUCTION

Cloud computing has been widely used in data sharing because it is effective and flexible. However, there exist privacy issues (e.g., data confidentiality) when cloud computing is used for data sharing. Identity-based encryption (IBE) as an efficient approach is available to ensure data confidentiality in a cloud-based data sharing system because of simple public key infrastructure (PKI) [1], [2]. In a real-world scenario, the data owner would like to share outsourced encrypted data with the data users if he has no time to deal with encrypted sensitive data stored in the server cloud. For example, a data owner Alice with an identity id from the disease research unit wants to safely share the disease record m about volunteers with his n colleagues with identities $id_1, \ldots, id_n$, note that we denote a colleague set (a data user set) $S_1 = \{id_1, \ldots, id_n\}$. When IBE is applied in the above scene for achieving data confidentiality, Alice needs to perform the encryption algorithm Enc of IBE to generate the encrypted disease data c about the disease record m (note that $c = Enc(id, m)$) and upload the ciphertext to the cloud server.

Obviously, there are some shortages with identity-based encryption to ensure data confidentiality in outsourced data sharing. First, the data owner Alice needs to download the outsourced encrypted disease data c from the cloud server and decrypt the ciphertext c to obtain the data m, and re-set a ciphertext for every colleague. In other words, Alice has a high computing cost to share outsourced encrypted data with the data users because the number of ciphertexts shows a linear correlation with the size of data users. Second, Alice has to completely keep online for converting the decryption right for outsourced encrypted data c into the decryption right for outsourced encrypted data c_j because he needs to re-set the ciphertext $c_j = Enc(id_j, m)$ under identity id_j for each colleague id_j ($j = 1, \ldots, n$). Third, if all users in a data user set $S_1 = \{id_1, \ldots, id_n\}$ obtain the data m, Alice wants to transfer the decryption right for outsourced encrypted data from a data user set $S_1 = \{id_1, \ldots, id_n\}$ to another data user

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Asif.

set $S_2 = \{id'_1, \ldots, id'_n\}$ he trusts. In such a scenario, the traditional IBE guarantees data confidentiality but it is not flexible for the data owner to perform the transformation of decryption right between the data user sets delegated by the data owner.

Alternatively, it might be an idea to outsource the amount of computing overhead for Alice to the cloud server. That is, the cloud server needs to obtain Alice's private key so that it has ability to decrypt the encrypted disease data and re-set the ciphertext for each colleague. However, if the cloud server is an untrusted server, this solution cannot maintain data confidentiality. We did not expect the untrusted server to obtain the disease record about volunteers via Alice's private key because the disease data involves a lot of personal sensitive data, such as illness and allergies. Prior, Blaze *et al.* [3] introduced the concept of proxy re-encryption (PRE) that is a potential approach to dealing with outsourced encrypted data. In a PRE scheme, a proxy (e.g., a cloud server) can convert the decryption right for outsourced encrypted data between the users without exposing the underlying data to the cloud server. This approach uses the benefits of cloud computing because the cloud server undertakes heavy computation cost of re-setting ciphertexts.

Identity-Based Proxy Re-Encryption (IB-PRE): Green and Ateniese [4] presented identity-based PRE (IB-PRE) to simplify PKI since the concept of PRE was introduced. In an IB-PRE scheme, the proxy has the ability to convert the ciphertext under a delegator's identity into ciphertext under a delegatee's identity without obtaining any information about sensitive data. One may think that we can utilize the solution of IB-PRE to solve the drawbacks of IBE applied in cloud data sharing. Unfortunately, IB-PRE is still an inefficient approach for the data owner. For example, if IB-PRE is applied in the outsourced data sharing, Alice needs to set n re-encryption keys $rk_{id\to id_1}, \ldots, rk_{id\to id_n}$ for a data user set $S_1 = \{id_1, \ldots, id_n\}$ and secrectly send these re-encryption keys to the proxy during the process. It is flexible for the proxy to set the cipertexts for these data users via these re-encryption keys. Additionally, IB-PRE resolves the issue of complete online for the delegator by outsourcing the computation cost of re-setting ciphertexts to the proxy. However, IB-PRE is still an inefficient approach for the data owner because the size of re-encryption keys is equal to the number of delegatees. Therefore, IB-PRE is not suited to actual applications if there exist many delegatees.

Identity-Based Broadcast Proxy Re-Encryption (IB-BPRE): Chu *et al.* [5] introduced the concept of broadcast proxy re-encryption (BPRE) to solve the linear computing issue of the re-encryption key for the delegator. In a BPRE scheme, the proxy can convert the ciphertext for the delegator into the ciphertext for a broadcast receiver (delegatee) set. In the process, the delegator only generates a re-encryption key for multiple delegatees and the proxy (e.g., a cloud server) sets a re-encryption ciphertext for a broadcast receiver set without obtaining any information about sensitive data. Lately, Xu *et al.* [6] introduced the notion of identity-based BPRE (IB-BPRE) to take the identity of the user as his public key. Despite IB-BPRE solving the heavy computing issue of re-encryption keys for the delegator, the transformation of decryption rights between the broadcast receiver sets authorized by the delegator is still an issue in IB-BPRE schemes. Therefore, our challenge point is how to implement a cloud data sharing system to achieve the transformation of decryption rights for outsourced encrypted data from a data user set $S_1 = \{id_1, \ldots, id_n\}$ to another data user set $S_2 = \{id'_1, \ldots, id'_n\}$, where sets S_1 and S_2 are chosen by the data owner.

A. MOTIVATION

The existing IB-BPRE schemes are effective in addressing the issues of IBE applied in the outsourced data sharing system, but they cannot solve the issue of autonomous path multi-hop. In other words, the existing IB-BPRE cannot achieve the transformation of decryption rights between the broadcast receiver sets delegated by the delegator. However, autonomous path multi-hop is very critical in IB-BPRE since we can perform flexible data sharing according to the data owner's wishes. Consequently, this motivates us to discover an autonomous path identity-based broadcast proxy re-encryption (APIB-BPRE) as a new cryptographic mechanism that supports to easily achieve an autonomous path multi-hop in IB-BPRE. More specifically, in an APIB-BPRE scheme, the delegator designates a delegation path involving preferred broadcast receiver sets. The delegation path comprises multiple broadcast delegatee sets, if all receivers of a broadcast receiver set in the path complete the decryption, the proxy automatically transforms decryption rights to the next broadcast receiver set in the path. By the method, the delegator guarantees that the decryption right is carried out among these broadcast receiver sets he trusts.

Imagine a data owner Alice from the disease reseach unit holds the diseases data m about volunteers. If Alice is too busy to deal with the disease data m, he may share the outsourced encrypted data with a data user set $S_1 = \{id_1, id_2, id_3\}$. Meanwhile, if all users in S_1 gain the disease data, decryption rights will be automatically delegated to next set of data users $S_2 = \{id'_1, id'_2, id'_3\}$ choosen by Alice. Our APIB-BPRE is suitable to the above cloud data sharing system, the data owner Alice encrypts his sensitive data as $c = Enc(id, m)$ and sets an autonomous path $\text{Pa} = (id = S_0, S_1, S_2)$, and then uploads c and Pa to the cloud server. The proxy can transform the ciphertext c for Alice into the ciphertext c_1 for a data user set S_1 by the re-encryption key $rk_{id\to S_1}$ from Alice, and convert the ciphertext c_1 for a data user set S_1 into the ciphertext c_2 for a data user set S_2 via the re-encryption key $rk_{S_1\to S_2}$ from Alice. The idea of our APIB-BPRE for data sharing in clouds is shown in Figure.1. With this motivation in mind, we designed APIB-BPRE, in which the proxy can achieve the transformation of decryption right for the

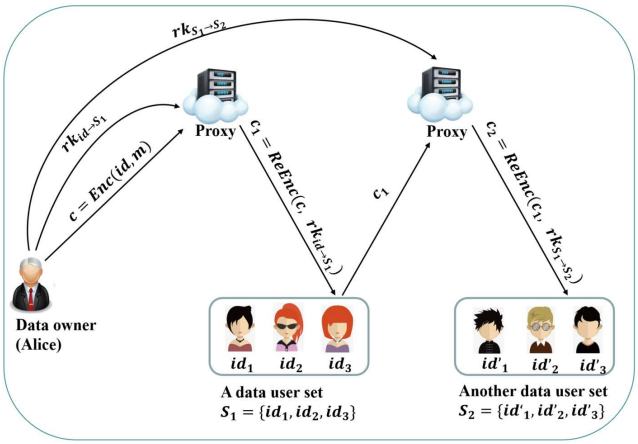

FIGURE 1. APIB-BPRE in a cloud data sharing system.

encrypted data between the broadcast receiver sets delegated by the delegator.

B. RELATED WORKS

Blaze *et al.* presented the concept of PRE and classified it into single-hop and multi-hop according to the permitted times of transformation [3]. In a multi-hop PRE scheme, the proxy can convert the ciphertext from Alice to Bob, from Bob to Carol and so on. In a single-hop PRE scheme, the proxy only transforms the ciphertext under Alice into the ciphertext under Bob. Since Blaze *et al.* proposed the concept of PRE, numerous works [4], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20] with different properties have been designed to meet kinds of actual demands. In traditional multi-hop PRE schemes, the delegator cannot dominate the selection of all delegatees with the decryption right for the encrypted data, he only chooses the first delegatee. For example, the proxy converts the decryption right from a delegator Alice to a delegatee Bob, and from a delegatee Bob to the delegatee Carol. In the process, Alice only chooses the first delegatee Bob, but the delegatee Carol is authorized by the delegatee Bob. It indicates that the delegator has no right to control all delegatees he trusts when decryption rights have been transformed from a delegatee to another delegatee. It is desirable for the delegator that he is able to control the decryption rights for encrypted files among the authorized delegatees in actual application demands. This ensures that the encrypted data can be decrypted by his authorized delegatees. Recently, Cao *et al.* [21] proposed an autonomous path PRE (AP-PRE) as a new cryptographic primitive to resolve the above issue. This approach has better fine-grained access control for encrypted data because AP-PRE has the property of autonomous path multi-hop. Put simply, autonomous path multi-hop in AP-PRE means that the delegator sets an autonomous delegation path Pa including multiple delegatees and the proxy can transform the ciphertext for the delegatee in Pa into the ciphertext for the next delegatee in Pa via the re-encryption key from the delegator.

Berkovits [22] introduced the concept of broadcast encryption (BE) that a sender broadcasts encrypted data to a broadcast receiver set and each receiver in the broadcast receiver set can decrypt the encrypted data via his private key. However, the user outside of the broadcast receiver set cannot get any information about the sensitive data. Since Fiat and Naor [23] gave the formal definitions about broadcast encryption and its security model, various BE works [24], [25] have been designed to increase efficiency. Broadcast proxy re-encryption (BPRE) is another interesting research field that the proxy can convert the decryption right for a delegator into the decryption right for a broadcast receiver (delegatee) set [5]. After that, Xu *et al.* [6] proposed a conditional IB-BPRE with constant re-encrypted ciphertext. Such a construction is significantly adapt to the cloud email system. After this work, Sun *et al.* [26] designed an IB-BPRE with CCA secure that is also sultable for the cloud computing environment application (e.g., cloud data sharing). Lately, Ge *et al.* [27] proposed an IB-BPRE with a revocation function that the proxy can revoke decryption rights for left delegatees. Unfortunately, none of these works addressed the property of autonomous multi-hop to IB-BPRE.

C. OUR CONTRIBUTIONS

In this work, we adopted the autonomous path multi-hop mechanism proposed for AP-PRE [21] to address the autonomous path multi-hop for IB-BPRE. One may think that this exists a direct connection between the autonomous path multi-hop for AP-PRE [21] and IB-BPRE. However, there are technical difficulties in applying the solution of autonomous path multi-hop showed in work [21] to the IB-BPRE scheme because there is a one-to-one correspondence between the re-encryption key and the delegatee in work [21]. That is, a delegator cannot set a re-encryption key for a broadcast receiver set by executing a re-encryption key generation algorithm. One might think that a possible attempt is to address the character of the autonomous path to the multi-hop IB-BPRE. Nevertheless, the existing IB-BPRE schemes do not have the character of multi-hop, mainly because it is a challenging task to set a re-encryption key $rk_{S_1 \to S_2}$ from a broadcast receiver set S_1 to another broadcast receiver set S_2. Therefore, reaching an autonomous path multi-hop for IB-BPRE is a challenging task.

This paper presents a new mechanism called autonomous path identity-based broadcast proxy re-encryption to guarantee the function of autonomous path multi-hop in IB-BPRE. Our APIB-BPRE allows the proxy to convert the decryption right for outsourced encrypted data from the data user set S_1 to the next data user set S_2, where S_1 and S_2 are delegated by the data owner. We give the formal definitions of our APIB-BPRE and its security model. Meanwhile, we give the concrete construction for our APIB-BPRE and prove its security in the decision n-BDHE problem. Additionally, the evaluation and comparison indicate that APIB-BPRE is efficient and practical.

D. ORGANIZATION

In Section *II*, we give the definitions of bilinear paring and hard problem assumption. Then, we define our APIB-BPRE and give the security model in Section *III*. In Section *IV*, we present a concrete construction of APIB-BPRE. Section *V* proves that our scheme is semantic security. In Section *VI*, The evaluation and comparison indicate that our scheme is efficient. Finally, we give a conclusion in Section *VII*.

II. PRELIMINARIES

We give the definition of the bilinear pairing and state the complex assumption needed for security proof.

A. BILINEAR PAIRING

Let $\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative cyclic groups of prime order q, and g is a generation of $\mathbb{G}$. A bilinear pairing is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following three properties [1], [28]:

- Bilinearity. For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q^*$, we have $e\left(u^a, v^b\right) = e\left(u, v\right)^{ab}$.
- Non-degeneracy. The map is not degenerate, i.e., $e\left(g, g\right) \neq 1$.
- Computability. There exists an efficient algorithm to compute the map e.

B. COMPLEX ASSUMPTION

The security of our APIB-BPRE scheme is based on the following assumption.

Assumption (decision n-bilinear Diffie-Hellman Exponent assumption (decision n-BDHE) [29]). Let $\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative cyclic groups of prime order q, and g is a generation of $\mathbb{G}$. The decision n-BDHE assumption is stated as follows: given a vector $\vec{y}_{g,\alpha,n} = \left(h, g, g_1, g_2, \cdots, g_n, g_{n+2}, \cdots, g_{2n}\right) \in \mathbb{G}^{2n+1}$ and an element $Z \in \mathbb{G}_T$ as input, decide whether Z is equal to $e\left(g_{n+1}, h\right)$. Note that we use g_i to denote $g_i = g^{\alpha^i} \in \mathbb{G}$ $(i = 1, \cdots, n, n+2, \cdots, 2n)$, an algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ with advantage ϵ in solving the decision n-BDHE problem in $\mathbb{G}$ if

$$\left|\Pr\left[\mathcal{B}\left(\vec{y}_{g,\alpha,n}, e\left(g_{n+1}, h\right)\right) = 0\right] - \Pr\left[\mathcal{B}\left(\vec{y}_{g,\alpha,n}, Z\right) = 0\right]\right| \geq \epsilon,$$

where the probability is the choice of random generation g and random h in $\mathbb{G}$, the choice of random α in $\mathbb{Z}_q^*$, the choice of random Z in $\mathbb{G}_T$, and the random bits consumed by $\mathcal{A}$.

Definition 1: The decision (t, ϵ, n)-BDHE assumption holds in $\mathbb{G}$ if any probabilistic polynomial time ($\mathcal{PPT}$) algorithm with an negligible advantage ϵ in solving the decision n-BDHE problem in $\mathbb{G}$.

III. DEFINITION AND SECURITY MODEL

We define our APIB-BPRE and the security model.

A. AUTONOMOUS PATH IDENTITY-BASED BROADCAST PROXY RE-ENCRYPTION (APIB-BPRE)

An APIB-BPRE refers to three types of entries: the delegator, the proxy, and the delegatee (receiver). In an APIB-BPRE system, the delegator id is able to choose multiple broadcast receiver sets $S_1, \ldots, S_m$ he trusts and generates a path $\text{Pa} = (id = S_0, S_1, \ldots, S_m)$ involving m preferred broadcast receiver sets (note that we denote id as $id = S_0$). To simplify the discussion, we suppose that each broadcast receiver set S_μ includes k receivers, where $S_\mu = \left\{id_{\mu_1}, \ldots, id_{\mu_k}\right\}$, for $\mu = 1, \ldots, m$. Meanwhile, the delegator uploads the ciphertext about his sensitive data to the proxy and sends the re-encryption key $rk_{\mu-1\to\mu}$ to the corresponding proxy through a secure channel for $\mu = 1, \ldots, m$. After obtaining the re-encryption key $rk_{\mu-1\to\mu}$ from the delegator, the corresponding proxy converts the ciphertext under broadcast receiver set $S_{\mu-1}$ into the ciphertext under the next broadcast receiver set S_μ without revealing sensitive data. In this way, we can gain the property of multi-hop from $S_{\mu-1}$ to S_μ in the autonomous path Pa for identity-based broadcast proxy re-encryption. The definition of APIB-BPRE is illustrated as follows.

Definition 2 (APIB-BPRE): An autonomous path identity-based broadcast proxy re-encryption scheme consists of the following algorithms:

- Setup $\left(1^\lambda, n\right) \to (msk, mpk)$. A trusted party key generation center (KGC) runs the setup algorithm *Setup* to generate the master public/secret keys. On input a security parameter 1^λ, and the maximum number of receivers n in one encryption. It outputs the master public key mpk and the master secret key msk.
- Extract $(msk, id) \to (sk_{id})$. KGC runs the key extraction algorithm *Extract* to set the private key. The algorithm inputs the master secret key msk and an identity id for the user (delegator or delegatee). It outputs a private key sk_{id}.
- CreatPath $(mpk, id) \to$ (Pa). The delegator id runs the path creation algorithm *CreatPath* to generate an autonomous path. It inputs the master public key mpk, and the identity id and outputs an autonomous path Pa of length m. The autonomous path $\text{Pa} = (id = S_0, S_1, \ldots, S_m)$ is a sequence of ordered m different broadcast receiver sets, where id is denoted to be S_0 and $S_\mu = \left\{id_{\mu_1}, \cdots, id_{\mu_k}\right\}$ is a set of broadcast receivers with identities id_{μ_j}, for $1 \leq \mu \leq m$, $k \leq n$. Note that, we implicitly assume that the size of each broadcast receiver set is k in order to simplify the discussion. Meanwhile, we denote a set S_μ in path Pa by $S_\mu \in \text{Pa}$ and denote that the length of Pa is equal to the number of broadcast receiver sets.
- RKeyGen $(mpk, id, \text{Pa}) \to (rk)$. The delegator id performs the re-encryption key generation algorithm *RKeyGen* to set the re-encryption key. It inputs the master public key mpk, identity id, and an autonomous path Pa created by the delegator id. It outputs the re-encryption key $rk = \left\{rk_{\mu-1\to\mu}\right\}_{\mu=1,\ldots,m}$. Note that the proxy can convert the ciphertext under $S_{\mu-1}$ into ciphertext under S_μ in the autonomous path Pa via the re-encryption key $rk_{\mu-1\to\mu}$.

- Enc $(mpk, id, m) \rightarrow c_0$. The delegator id runs the encryption algorithm Enc to set the ciphertext. It inputs the master public key mpk, the identity id, and a message m from the message space $\mathcal{M}$ and outputs the ciphertext c_0. For simplicity, we call c_0 the original ciphertext.
- ReEnc $\left(\text{Pa}, S_{\mu-1}, S_\mu, r\ k_{\mu-1\rightarrow\mu}, c_{\mu-1}\right) \rightarrow c_\mu$, where $1 \leq \mu < m$. The proxy performs the re-encryption algorithm $ReEnc$ to convert the ciphertext under $S_{\mu-1}$ into ciphertext under S_μ. On input an autonomous path Pa, two broadcast receiver sets $S_{\mu-1}$ and S_μ, a re-encryption key $rk_{\mu-1\rightarrow\mu}$, and a ciphertext $c_{\mu-1}$ under the broadcast receiver set $S_{\mu-1}$. It first checks whether $\left(S_{\mu-1}, S_\mu\right) \in$ Pa and outputs "$\perp$" if not. Otherwise, the algorithm outputs the re-encrypted ciphertext c_μ for the set of broadcast receivers S_μ. For simplicity, we denote call c_μ the re-encryption ciphertext.
- Dec $\left(mpk, c_0/c_\mu, s\ k_{id}\right) \rightarrow (m, \perp)$, where $\mu = 1, \ldots, m$. The delegator (delegatee) runs the decryption algorithm Dec to recover the message. It inputs the master public key mpk, the original ciphertext c_0 (re-encryption ciphertext c_μ), and a private key sk_{id} and outputs the message $m \in \mathcal{M}$, or an error symbol $\perp$.

Correctness: Our APIB-BPRE is correct, if for autonomous path Pa set by the delegator id, the following equations hold for any $m \in \mathcal{M}$:

$$\begin{aligned}&\text{Dec}\left(mpk, \text{Enc}\left(mpk, id, m\right), sk_{id}\right) = m,\\ &id \notin S_\mu,\ 1 \leq \mu \leq m;\\ &\text{Dec}\left(mpk, c_\mu, sk_{id}\right) = m,\ id \in S_\mu,\ 1 \leq \mu \leq m;\end{aligned}$$

where for any $\mu,\ 1 \leq \mu \leq m$,

$$\text{ReEnc}\left(\text{Pa}, S_{\mu-1}, S_\mu, r\ k_{\mu-1\rightarrow\mu}, c_{\mu-1}\right) \rightarrow c_\mu.$$

B. SECURITY MODEL FOR APIB-BPRE

We consider the security of APIB-BPRE in chosen plaintext attack model for the original ciphertext and the re-encryption ciphertext, respectively. We use the following two indistinguishable games between a $\mathcal{PPT}$ adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ to define the security for the original ciphertext and the re-encryption ciphertext separately.

Game 1. We define the following indistinguishable game of our APIB-BPRE scheme for the original ciphertext in the chosen plaintext attack model. The adversary $\mathcal{A}$ and the challenger $\mathcal{C}$ perform the following indistinguishable game:

- **Init**. $\mathcal{A}$ chooses an identity id^* as a challenging identity.
- **Setup.** $\mathcal{C}$ generates the master key public mpk and the master secret key msk via executing the setup algorithm $Setup$. It outputs mpk to $\mathcal{A}$.
- **Query phase 1.** $\mathcal{A}$ makes key extraction query $\mathcal{O}_{sk}(mpk, i\ d)$. It inputs an identity id and the master public key mpk, if $id = id^*$, $\mathcal{C}$ ouputs an error symbol $\perp$; otherwise, $\mathcal{C}$ generates the private key sk_{id} by running the key extraction algorithm $Extract$ and returns sk_{id} to $\mathcal{A}$.
- **Challenge**. After receiving two messages $m_0, m_1 \in \mathcal{M}$, $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$ and sets the challenging ciphertext c_0^*. It returns c_0^* to the adversary $\mathcal{A}$.
- **Query phase 2**. $\mathcal{A}$ continues making key extraction query and $\mathcal{C}$ responds to the query like as in the query phase 1.
- **Guess**. $\mathcal{A}$ outputs the guess b'. The adversary $\mathcal{A}$ wins if $b' = b$.

  Let $Adv_{\mathcal{A}}^{IND\text{-}CPA-Or}(\lambda)$ denote the advantage that $\mathcal{A}$ wins the above indistinguishable game in chosen plaintext attack model for the original ciphertext (IND-CPA-Or), where $Adv_{\mathcal{A}}^{IND\text{-}CPA-Or}(\lambda) = |\Pr\left[b' = b\right] - 1/2|$.

Definition 3: Our APIB-BPRE scheme is (t, q_{sk}, ϵ)- CPA secure at the original ciphertext if for any $\mathcal{PPT}$ adversary $\mathcal{A}$ who makes at most q_{sk} key extraction queries, we have $\text{AdvBr}_{\mathcal{A}}^{\text{IND-CPA-Or}}(\lambda) \leq \epsilon$.

Game 2: We define the following indistinguishable game of our APIB-BPRE scheme for the re-encryption ciphertext in chosen plaintext attack model. The adversary $\mathcal{A}$ and the challenger $\mathcal{C}$ perform the following indistinguishable game:

- **Init**. $\mathcal{A}$ outputs the challenging broadcast receiver set $S_\mu^* = \left\{id_{\mu_1}^*, \ldots, i\ d_{\mu_k}^*\right\}$ for any μ, where $1 \leq \mu \leq m$, $k \leq n$.
- **Setup**. $\mathcal{C}$ generates the master public key mpk and the master secret key msk via running the setup algorithm $Setup$ and returns mpk to $\mathcal{A}$.
- **Query phase 1**. $\mathcal{A}$ makes the following queries:

  a) Key extraction query $\mathcal{O}_{sk}(mpk, id)$. It inputs an identity id and the master public key mpk, if $id \in S_\mu^*$, $\mathcal{C}$ returns an error symbol $\perp$; otherwise $\mathcal{C}$ generates the private key sk_{id} via executing the key extraction algorithm $Extract$ and returns sk_{id} to $\mathcal{A}$.

  b) Path creation query $\mathcal{O}_{cp}(mpk, id)$. On input the master public key mpk and an identity id, $\mathcal{C}$ generates a path $\text{Pa} = (id = S_0, S_1, \ldots, S_m)$ via running the path creation algorithm $GreatPath$ and returns Pa to $\mathcal{A}$.

  c) Re-encryption key generation query $\mathcal{O}_{rk}\left(mpk, id, \text{Pa}, S_{\mu-1}, S_\mu\right)$. On input the master public key mpk, an identity id, broadcast receiver sets $S_{\mu-1}$ and S_μ, where $\left(S_{\mu-1}, S_\mu\right) \in$ Pa. $\mathcal{C}$ retrieves $rk_{\mu-1\rightarrow\mu}$ from rk via running the re-encryption key generation algorithm $RKeyGen$ and returns $rk_{\mu-1\rightarrow\mu}$ to $\mathcal{A}$.
- **Challenge**. After receiving two messages $m_0, m_1 \in \mathcal{M}$, $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$ and sets the challenging ciphertext c_μ^*. It returns c_μ^* to the adversary $\mathcal{A}$.
- **Query phase 2**. $\mathcal{A}$ continues making key extraction, path creation, and re-encryption key queries and $\mathcal{C}$ responds to these queries like as in the query phase 1.
- **Guess**. $\mathcal{A}$ outputs the guess b'. The adversary $\mathcal{A}$ wins if $b' = b$.

  Let $Adv_{\mathcal{A}}^{IND\text{-}CPA-Re}(\lambda)$ denote the advantage that $\mathcal{A}$ wins the above indistinguishable game in chosen plaintext attack model for the re-encryption ciphertext (IND-CPA-Re), where $Adv_{\mathcal{A}}^{IND\text{-}CPA-Re}(\lambda) = |\Pr\left[b' = b\right] - 1/2|$.

TABLE 1. Summary of notations.

Notation	Description
mpk	the master public key
msk	the master secret key
$\mathrm{Pa} = (id = S_0, S_1, \ldots, S_m)$	an autonomous path delegated by delegator id
$S_\mu = \{id_{\mu_1}, id_{\mu_2}, \ldots, id_{\mu_k}\}$, for $\mu = 1, 2, \ldots, m$	the set of broadcast receivers/delegatees
$K = \{1, 2, \ldots, k\}$	an index set about S_μ, where k is the size of S_μ
$rk_{\mu-1\to\mu}, \mu = 1, 2, \ldots, m$	the re-encryption key from $S_{\mu-1}$ to S_μ
sk_{id}	the private key for the user id
c_0	the original ciphertext
c_μ, for $\mu = 1, 2, \ldots, m$	the re-encryption ciphertext

Remark 1: The adversary $\mathcal{A}$ does not need to make the re-encryption query because there is to be no limitation on making re-encryption key query.

Definition 4: Our APIB-BPRE scheme is $(t, q_{sk}, q_{cp}, q_{rk}, \epsilon)$- CPA secure at re-encryption ciphertext if for any $\mathcal{PPT}$ adversary $\mathcal{A}$ who makes at most q_{sk} key extraction queries, q_{cp} path creation queries, and q_{rk} re-encryption key queries, we have $Adv_{\mathcal{A}}^{IND\text{-}CPA-Re}(\lambda) \leq \epsilon$.

Definition 5: Our APIB-BPRE scheme is semantic security (CPA secure), if $Adv_{\mathcal{A}}^{IND\text{-}CPA-Or}(\lambda) \leq \epsilon$ and $Adv_{\mathcal{A}}^{IND\text{-}CPA-Re}(\lambda) \leq \epsilon$.

IV. PROPOSED APIB-BPRE SCHEME

This section presents a concrete construction of APIB-BPRE. For ease of reference, Table 1 summary improtant notations.

A. TECHNICAL OVERVIEW

The autonomous path multi-hop is a significant property in PRE schemes that the proxy can transform decryption rights between the delegatees delegated by the delegator. However, it is a difficult for IB-BPRE schemes to support autonomous path multi-hop. We proposed an autonomous path identity-based broadcast proxy re-encryption to realize the autonomous path multi-hop in IB-BPRE. In our scheme, the delegator id sets an autonomous delegation path $\mathrm{Pa} = (id = S_0, S_1, \cdots, S_m)$ including m broadcast receiver sets S_j $(j = 1, \ldots, m)$ and the proxy can transform the ciphertext for a broadcast receiver set $S_{\mu-1}$ into the ciphertext for S_μ via the re-encryption key $rk_{\mu-1\to\mu}$ from the delegator id, for $\mu = 1, \ldots, m$. Here we simply describe the technical method of our APIB-BPRE. Suppose that the ciphertext $c_{\mu-1}$ for a broadcast receiver set $S_{\mu-1}$ consists of three elements $c_{\mu-1,1} = h^{t_{\mu-1}}$, $c_{\mu-1,2} = e(h, h_{n+1})^{t_{\mu-1}}$, and $c_{\mu-1,3} = (v \cdot \prod_{j\in K} h_{n+1-j})^{T_{\mu-1}} \cdot \prod_{j\in K} H(id_{\mu-1_j})^{\alpha^{n+1-j}}$. If the proxy needs to convert the ciphertext $c_{\mu-1}$ for $S_{\mu-1}$ into ciphertext c_μ for S_μ, we can view the ciphertext c_μ for S_μ as $c_{\mu,1} = c_{\mu-1,1} \cdot rk_{(\mu-1\to\mu)_1}$, $c_{\mu,2} = c_{\mu-1,2} \cdot rk_{(\mu-1\to\mu)_2}$ and $c_{\mu,3} = rk_{(\mu-1\to\mu)_3}$ via the re-encryption key $rk_{\mu-1\to\mu} = \left(rk_{(\mu-1\to\mu)_1}, rk_{(\mu-1\to\mu)_2}, rk_{(\mu-1\to\mu)_3}\right)$, where random $t_{\mu-1}$, $T_{\mu-1}$ in $\mathbb{Z}_q^*$.

B. CONSTRUCTION

Generally, an APIB-BPRE scheme consists of the following algorithms.

- $\mathrm{Setup}(1^\lambda, \mathrm{n})$. To set the master public key mpk and the master secret key msk, it generates a bilinear pairing group $\mathbb{PG} = (q, g, \mathbb{G}, \mathbb{G}_T, e)$. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear pairing, $\mathbb{G}$ and $\mathbb{G}_T$ are multiplicative groups with the same prime order q, g be a generation of group $\mathbb{G}$. The algorithm selects random $\alpha, s, r \in \mathbb{Z}_q^*$ and computes $h = g^s$, $\hat{h} = h^s$, $v = h^r$ $g_n = g^{\alpha^n}$, $h_i = h^{\alpha^i}$ for $i = 1, \ldots, n, n+2, \ldots, 2n$, and $d_i = (h_i)^r$ for $i = 1, \ldots, n$. Nextly, it selects a cryptographic hash function $H : \{0,1\}^* \to \mathbb{G}$. The master public key is $mpk = (\mathbb{PG}, h, \hat{h}, v, g_n, H, \{h_i\}_{i=1,\ldots,n,n+2,\ldots,2n}, \{d_i\}_{i=1,\ldots,n})$ and the master secret key is $msk = (s, \alpha)$. Note that it sends the secret key α to the delegator via the secure channel.
- $\mathrm{Extract}(mpk, id)$. To generate the private key for the user id, if the user id is the delegator, it sets private key $sk_{id} = H(id)^s$; otherwise, it sets private key $sk_{id} = H(id)^{s\alpha}$, where the user id is the delegatee.
- $\mathrm{CreatPath}(mpk, id)$. To set an autonomous path for the delegator id, it chooses m broadcast receiver sets $S_1, \ldots, S_m$ and generates an autonomous path $\mathrm{Pa} = (id = S_0, S_1, \cdots, S_m)$ of length m. Note that the broadcast receiver set $S_\mu = \{id_{\mu_1}, id_{\mu_2}, \ldots, id_{\mu_k}\}$ is a set of ordered k different receivers, for $\mu = 1, 2, \cdots, m$ and $k \leq n$.
- $\mathrm{RKeyGen}(mpk, \mathrm{Pa})$. To generate the re-encryption key $rk = \{rk_{\mu-1\to\mu}\}_{\mu=1,\ldots,m}$ for any broadcast receiver set S_μ in an autonomous path Pa delegated by the delegator id, it randomly chooses $t_0, t_\mu \in \mathbb{Z}_q^*$ and sets $rk_{(\mu-1\to\mu)_1} = h^{t_\mu}$, $rk_{(\mu-1\to\mu)_2} = e(h, h_{n+1})^{t_\mu}$, and $rk_{(\mu-1\to\mu)_3} = (v \cdot \prod_{j\in K} h_{n+1-j})^{T_\mu} \cdot \prod_{j\in K} H(id_{\mu_j})^{\alpha^{n+1-j}}$, where $T_\mu = t_0 + \cdots + t_\mu$. Finally, it sets $rk_{\mu-1\to\mu} = \left(rk_{(\mu-1\to\mu)_1}, rk_{(\mu-1\to\mu)_2}, rk_{(\mu-1\to\mu)_3}\right)$ and returns the re-encryption key $rk = \{rk_{\mu-1\to\mu}\}_{\mu=1,\ldots,m}$ to the corresponding proxy. Note that $e(h, h_{n+1})$ be constructed as $e(h_1, h_n)$.
- $\mathrm{Enc}(mpk, id)$. To encrypt a message $m \in \mathcal{M}$ under id, the delegator computes

$$c_{0,1} = h^{t_0}, c_{0,2} = m \cdot e(h, h_{n+1})^{t_0},$$

and

$$c_{0,3} = e(h, h_{n+1})^{t_0} \cdot e\left(\hat{h}, H(id)\right)^{t_0}.$$

Finally, it returns the ciphertext as $c_0 = \left(c_{0,1}, c_{0,2}, c_{0,3}\right)$.

- $\mathrm{ReEnc}(\mathrm{Pa}, S_{\mu-1}, S_\mu, rk_{\mu-1\to\mu}, c_{\mu-1})$, where $\mathrm{Pa} = (id = S_0, S_1, \cdots, S_m)$ designed by the delegator id and $S_\mu = \{id_{\mu_1}, \ldots, id_{\mu_k}\}$ for $1 \le \mu \le m$. To convert a ciphertext under the broadcast receiver set $S_{\mu-1}$ into a ciphertext under next broadcast receiver set S_μ, it first checks whether $(S_{\mu-1}, S_\mu) \in \mathrm{Pa}$, and outputs $'' \perp$ "if not. Otherwise, the proxy has the ciphertext $c_{\mu-1} = (c_{\mu-1,1}, c_{\mu-1,2}, c_{\mu-1,3})$ and the re-encryption key $rk_{\mu-1\to\mu} = (rk_{(\mu-1\to\mu)_1}, rk_{(\mu-1\to\mu)_2}, rk_{(\mu-1\to\mu)_3})$. The proxy computes the ciphertext c_μ as $(c_{\mu,1}, c_{\mu,2}, c_{\mu,3})$, where $c_{\mu,1} = c_{\mu-1,1} \cdot rk_{(\mu-1\to\mu)_1}$, $c_{\mu,2} = c_{\mu-1,2} \cdot rk_{(\mu-1\to\mu)_2}$ and $c_{\mu,3} = rk_{(\mu-1\to\mu)_3}$. Note that, we have $c_{\mu,1} = h^{T_\mu}$, $c_{\mu,2} = m \cdot e(h, h_{n+1})^{T_\mu}$, and $c_{\mu,3} = (v \cdot \prod_{j\in K} h_{n+1-j})^{T_\mu} \cdot \prod_{j\in K} H(id_{\mu_j})^{\alpha^{n+1-j}}$.
- $\mathrm{Dec}(mpk, c_0/c_\mu, s\ k_{id}) \to (m, \perp)$. To decrypt the original ciphertext c_0, the delegator id has the original ciphertext c_0 as $(c_{0,1}, c_{0,2}, c_{0,3})$. It computes $X_0 = (c_{0,3}/e(c_{0,1}, sk_{id}))$ and $m = (c_{0,2}/X_0)$. To decrypt the re-encryption ciphertext c_μ, the delegatee id_{μ_j} in S_μ has the re-encryption ciphertext c_μ as $(c_{\mu,1}, c_{\mu,2}, c_{\mu,3})$. For any $1 \le \mu \le m, j \in K$, the delegatee id_{μ_j} computes
  - $X^1_{\mu_j} = e(h_j, c_{\mu,3})$,
  - $X^2_{\mu_j} = e(c_{\mu,1}, d_j \cdot \prod_{k\in K, k\neq j} h_{n+1-k+j})$,
  - $X^3_{\mu_j} = \prod_{k\in K, k\neq j} e(h_{n+1-k+j}, H(id_{\mu_k}))$,
  - $X^4_{\mu_j} = (X^1_{\mu_j}/X^2_{\mu_j} \cdot X^3_{\mu_j})$,
  - $X^5_{\mu_j} = (X^4_{\mu_j}/e(g_n, sk_{id_{\mu_j}}))$.

  Finally, the delegatee id_{μ_j} outputs $m = (c_{\mu,2}/X^5_{\mu_j})$.

Correctness: Here we explore the correctness of the original ciphertext c_0 and the re-encryption ciphertext c_μ in our APIB-BPRE scheme.

1) For an original ciphertext $c_0 = (c_{0,1}, c_{0,2}, c_{0,3})$, the delegator id computes

$$X_0 = \frac{c_{0,3}}{e(c_{0,1}, sk_{id_i})} = e(h, h_{n+1})^{t_0},$$

and decrypts $m = (c_{0,2}/X_0) = (m \cdot e(h, h_{n+1})^{t_0}/e(h, h_{n+1})^{t_0}) = m$. The decryption is obviously correct.

2) For the re-encryption ciphertext $c_\mu = (c_{\mu,1}, c_{\mu,2}, c_{\mu,3})$, we have $c_{\mu,1} = h^{T_\mu}$, $c_{\mu,2} = m \cdot e(h, h_{n+1})^{T_\mu}$, and $c_{\mu,3} = (v \cdot \prod_{k\in K} h_{n+1-k})^{T_\mu} \cdot \prod_{k\in K} H(id_{\mu_k})^{\alpha^{n+1-k}}$. The delegatee id_{μ_j} in the set S_μ computes

$$\begin{aligned}
X^1_{\mu_j} &= e\left(h_j, c_{\mu,3}\right) \\
&= e(h_j, (v \cdot \prod_{k\in K} h_{n+1-k})^{T_\mu} \cdot \prod_{k\in K} H(id_{\mu_k})^{\alpha^{n+1-k}}) \\
&= e(h_j, (v \cdot \prod_{k\in K} h_{n+1-k})^{T_\mu}) \\
&\quad \cdot e(h_j, \prod_{k\in K} H(id_{\mu_k})^{\alpha^{n+1-k}}) \\
&= e\,(h, h)^{T_\mu\left(r\alpha^j + \sum_{k\in K} \alpha^{n+1-k+j}\right)} \\
&\quad \cdot e(h, \prod_{k\in K} H(id_{\mu_k})^{\alpha^{n+1-k+j}}),
\end{aligned}$$

then, the delegatee id_{μ_j} computes $X^2_{\mu_j}$, $X^3_{\mu_j}$, $X^4_{\mu_j}$, and $X^5_{\mu_j}$. We have

$$\begin{aligned}
X^2_{\mu_j} &= e(c_{\mu,1}, d_j \cdot \prod_{k\in K, k\neq j} h_{n+1-k+j}) \\
&= e(h^{T_\mu}, h^{r\alpha^j} \cdot \prod_{k\in K, k\neq j} h_{n+1-k+j}) \\
&= e\,(h, h)^{T_\mu\left(r\alpha^j + \sum_{k\in K, k\neq j} \alpha^{n+1-k+j}\right)},
\end{aligned}$$

and

$$\begin{aligned}
X^3_{\mu_j} &= \prod_{k\in K, k\neq j} e\left(h_{n+1-k+j}, H(id_{\mu_k})\right) \\
&= e(h, \prod_{k\in K, k\neq j} H(id_{\mu_k})^{\alpha^{n+1-k+j}}),
\end{aligned}$$

and

$$X^4_{\mu_j} = \frac{X^1_{\mu_j}}{X^2_{\mu_j} \cdot X^3_{\mu_j}} = e\,(h, h_{n+1})^{T_\mu} \cdot e\left(h_{n+1}, H(id_{\mu_j})\right),$$

and

$$\begin{aligned}
X^5_{\mu_j} &= \frac{X^4_{\mu_j}}{e\left(g_n, sk_{id_{\mu_j}}\right)} \\
&= \frac{e\left(h, h_{n+1}\right)^{T_\mu} \cdot e\left(h_{n+1}, H(id_{\mu_j})\right)}{e\left(g_n, H(id_{\mu_j})^{s\alpha}\right)} \\
&= e\,(h, h_{n+1})^{T_\mu}.
\end{aligned}$$

Finally, the delegatee id_{μ_j} computes m, where

$$m = \frac{c_{\mu,2}}{X^5_{\mu_j}} = \frac{m \cdot e(h, h_{n+1})^{T_\mu}}{e(h, h_{n+1})^{T_\mu}} = m$$

The decryption for re-encryption ciphertext is obviously correct.

V. SECURITY PROOF

This section proves that our APIB-BPRE system is the semantic security (CPA secure) by Theorem 1 and Theorem 2.

Theorem 1: Our APIB-BPRE scheme is CPA secure for the original ciphertext under the decision n-BDHE assumption in $\mathbb{G}$ without random oracle.

Proof 1: We suppose that there is a $\mathcal{PPT}$ adversary $\mathcal{A}$ with advantage ϵ in breaking the IND-CPA-Or security of our APIB-BPRE scheme in time t. We construct a simulator $\mathcal{B}$ to solve the decision n-BDHE assumption with the advantage ϵ' in time t'. $\mathcal{B}$ is given the decision n-BDHE instance $(h', h, h^\alpha, \ldots, h^{\alpha^n}, h^{\alpha^{n+2}}, \ldots, h^{\alpha^{2n}}, Z)$, where we denote $h = g^s$ and $\vec{y}_{\alpha,n,h} = (h', h, h^\alpha, \ldots, h^{\alpha^n}, h^{\alpha^{n+2}}, \ldots, h^{\alpha^{2n}})$. $\mathcal{B}$'s task is to decide whether $Z \stackrel{?}{=} e\left(h', h_{n+1}\right)$. $\mathcal{B}$ needs to maintains an initially empty table T_{sk} that is a private key table used to record tuples (id, sk_{id}). The simulator $\mathcal{B}$ interacts with $\mathcal{A}$, and works as follows:

- **Init**. $\mathcal{B}$ gains a challenging identity id^* from the adversary $\mathcal{A}$.

- **Setup**. To generate the master public key $mpk = (\mathbb{PG}, h, \hat{h}, v, g_n, H, \{h_i\}_{i=1,\ldots,n,n+2,\ldots,2n}, \{d_i\}_{i=1,\ldots,n})$. Firstly, $\mathcal{B}$ generates a bilinear pairing group $\mathbb{PG} = \{q, g, \mathbb{G}, \mathbb{G}_T, e\}$. Next, $\mathcal{B}$ randomly chooses $r \in Z_q^*$ and sets $\hat{h} = h^s$, $v = h^r$, $g_n = h_n^{s^{-1}}$, and $d_i = (h_i)^r$ for $i = 1, \ldots, n$, where the elements $h, \{h_i\}_{i=1,\ldots,n,n+2,\ldots,2n}$ are from the problem instance. Finally, $\mathcal{B}$ selects a secure hash function $H : \{0,1\}^* \to \mathbb{G}$ and returns the master public key mpk to $\mathcal{A}$. Note that the distribution of the master public key is identified as the distribution of the real world from the view of adversary $\mathcal{A}$, because these parameters r and s are uniforms and random distributions.
- **Query phase 1**. $\mathcal{A}$ makes key extraction query for id in this phase. If $id = id^*$, $\mathcal{B}$ outputs $\perp$; otherwise $\mathcal{B}$ searches T_{sk},
  - if T_{sk} includes $(id, s\, k_{id})$, returns sk_{id}.
  - Otherwise, $\mathcal{B}$ computes $sk_{id} = H(id)^s$ and returns sk_{id}.
  Finally, $\mathcal{B}$ adds $(id, s\, k_{id})$ to T_{sk}.
- **Challenge**. After receiving two messages $m_0, m_1 \in \mathcal{M}$, $\mathcal{B}$ randomly chooses $b \in \{0,1\}$ and sets the challenging ciphertext c_0^* as

$$c_0^* = \left(h', m_b \cdot Z, Z \cdot e\left(\left(h'\right)^s, H(id^*)\right)\right).$$

  Let $h' = h^{t_0}$, if $Z = (h, h_{n+1})^{t_0}$, we have

$$c_0^* = (h^{t_0}, m_b \cdot (h, h_{n+1})^{t_0}, \\ e\,(h, h_{n+1})^{t_0} \cdot e(\hat{h}, H(id^*))^{t_0})$$

  Therefore, c_0^* is a correct challenging ciphertext to encrypt message m_b for id^*.
- **Query phase 2**. $\mathcal{A}$ continues to issue the key extraction query and $\mathcal{B}$ responds to the query like as in query phase 1.
- **Guess**. $\mathcal{A}$ outputs a guess b' of b. If $b' = b$, $\mathcal{B}$ returns 0 to indicate $Z = e(h', h_{n+1})$; otherwise, it returns 1 to indicate that Z is random in $\mathbb{G}_T$.

This completes the simulation and the solution. $\mathcal{B}$ has the advantage ϵ' in solving the decision n-BDHE assumption in time t'. We here analyze the advantage ϵ' and time t'. If $Z \neq e\left(h', h_{n+1}\right)$, we have $\Pr\left[\mathcal{B}(\vec{y}_{\alpha,n,h}, Z) = 0 \mid Z \neq e(h', h_{n+1})\right] = (1/2)$ (indicating that B's view is independent of b). If $Z = e(h', h_{n+1})$, we have $\Pr\left[\mathcal{B}(\vec{y}_{\alpha,n,h}, Z) = 0 \mid Z = e(h', h_{n+1})\right] = (1/2) + (\epsilon/2)$ (indicating that $\mathcal{B}$'s output is dependent on $\mathcal{A}$'s output). Thus, $\mathcal{B}'$'s advantage in solving the decision n-BDHE assumption is $\epsilon' = |\Pr[\mathcal{B}(\vec{y}_{\alpha,n,h}, Z) = 0 \mid Z = e(h_{n+1})] - \Pr\left[\mathcal{B}(h', \vec{y}_{\alpha,n,h}, Z) = 0 \mid Z \neq e(h', h_{n+1})\right]| = |(1/2) + (\epsilon/2) - (1/2)| = (\epsilon/2)$. We denote the time cost of the simulation $T_s = \mathcal{O}(q_{sk})$, where key extraction queries mainly dominate the time cost of the simulation T_s. Thus, $\mathcal{B}$ will solve the decision n-BDHE assumption with time $t' = t + T_s$.

Theorem 2: The proposed APIB-BPRE scheme is is CPA secure for the re-encryption ciphertext under the decision n-BDHE assumption in $\mathbb{G}$ with the random oracle model.

Proof 2: We suppose that there is a $\mathcal{PPT}$ adversary $\mathcal{A}$ with the advantage ϵ in breaking the IND-CPA-Re security of our APIB-BPRE scheme in time t. We construct a simulator $\mathcal{B}$ to solve the decision n-BDHE assumption with the advantage ϵ' in time t'. $\mathcal{B}$ is given the decision n-BDHE instance $(h', h, h^{\alpha}, \ldots, h^{\alpha^n}, h^{\alpha^{n+2}}, \ldots, h^{\alpha^{2n}}, Z)$, where we denote $h = g^s$ and $\vec{y}_{\alpha,n,h} = (h', h, h^{\alpha}, \ldots, h^{\alpha^n}, h^{\alpha^{n+2}}, \ldots, h^{\alpha^{2n}})$. $\mathcal{B}$'s task is to decide whether $Z \stackrel{?}{=} e\left(h', h_{n+1}\right)$. $\mathcal{B}$ maintains private key table T_{sk}, re-encryption key table T_{rk}, and autonomous path table T_P. These tables are initially empty. Let T_{sk} record tuples (id, sk_{id}), T_{rk} record tuples $(id, S_{\mu-1}, S_{\mu}, rk_{\mu-1\to\mu})$, and T_P record tuples $\left(id, \mathrm{Pa} = \left(\ldots, S_{\mu-1}, S_{\mu}, \ldots\right)\right)$. The simulator $\mathcal{B}$ interacts with $\mathcal{A}$, and works as follows:

- **Init**. The adversary $\mathcal{A}$ outputs a challenging broadcast receiver set $S_{\mu}^* = \left\{id_1^*, \ldots, i\, d_k^*\right\}$, for any μ, $\mu = 1, \ldots, m$ and $k \leq n$.
- **Setup**. To generate the master public key $mpk = (\mathbb{PG}, h, \hat{h}, v, g_n, H, \{h_i\}_{i=1,\ldots,n,n+2,\ldots,2n}, \{d_i\}_{i=1,\ldots,n})$. Firstly, $\mathcal{B}$ generates a bilinear pairing group $\mathbb{PG} = \{q, g, \mathbb{G}, \mathbb{G}_T, e\}$. Next, $\mathcal{B}$ selects a secure hash function $H : \{0,1\}^* \to \mathbb{G}$ as the random oracle. Finally, $\mathcal{B}$ randomly chooses $r, u \in Z_q^*$ and sets $\hat{h} = h^s$, $g_n = (h_n)^{s^{-1}}$, $v = h^u(\prod_{j\in K} h_{n+1-j})$, and $d_i = (h_i)^r$ (note that the elements $h, \{h_i\}_{i=1,\ldots,n,n+2,\ldots,2n}$ are from the problem instance). It returns the master public key mpk to $\mathcal{A}$. Note that since these parameters r and u are uniforms and random distributions, the master public key is an identical distribution as the real construction from the view of adversary $\mathcal{A}$.
- H-**Query**. In this phase, $\mathcal{A}$ issues the hash query. $\mathcal{B}$ needs to maintain a hash table T_H that is initially empty and used to record queries and responses. For a query on id, $\mathcal{B}$ chooses random $x_{id} \in \mathbb{Z}_q^*$ and sets as

$$H\,(id) = h^{x_{id}}.$$

  $\mathcal{B}$ responds to the query on id with $H\,(id)$ and adds tuples $(id, x_{id}, H\,(id))$ to T_H.
- **Query phase 1**. $\mathcal{A}$ makes the following queries.
  a) Key extraction query $O_{sk}(mpk, id)$. $\mathcal{A}$ makes key extraction query on id. If $id \in S_{\mu}^*$, $\mathcal{B}$ outputs $\perp$; otherwise $\mathcal{B}$ searches T_{sk},
  - if T_{sk} includes $(id, s\, k_{id})$, returns sk_{id}.
  - Otherwise, $\mathcal{B}$ first makes hash query on id and gets x_{id}.
  Then, $\mathcal{B}$ computes

$$sk_{id} = (h_1)^{x_{id}\cdot s} = H(id)^{s\alpha}.$$

  Finally, $\mathcal{B}$ adds $(id, s\, k_{id})$ to T_{sk}.
  b) Path creation query $\mathcal{O}_{cp}(mpk, id)$. $\mathcal{A}$ makes path creation query for id, $\mathcal{B}$ generates a path $\mathrm{Pa} = (id = S_0, S_1, \ldots, S_m)$ for id via running the path creation algorithm *GreatPath* and returns the path Pa to $\mathcal{A}$.
  c) Re-encryption key generation query $\mathcal{O}_{rk}\left(id, S_{\mu-1}, S_{\mu-1}\right)$. To query the re-encryption key $rk_{\mu-1\to\mu}$ for id. The simulator $\mathcal{B}$ first checks whether T_p includes a path

Pa $= (\ldots, S_{\mu-1}, S_\mu, \ldots)$ for user id. If not, $\mathcal{B}$ returns $\perp$; otherwise $\mathcal{B}$ searches T_{rk},
- if T_{rk} includes $(id, S_{\mu-1}, S_\mu, rk_{\mu-1\to\mu})$, returns $rk_{\mu-1\to\mu}$.
- Otherwise, $\mathcal{B}$ sets re-encryption key $rk_{\mu-1\to\mu} = (rk_{(\mu-1\to\mu)_1}, rk_{(\mu-1\to\mu)_2}, rk_{(\mu-1\to\mu)_3})$. $\mathcal{B}$ chooses $t_\mu, T_\mu \in \mathbb{Z}_q^*$ and computes

$$rk_{(\mu-1\to\mu)_1} = h^{t_\mu}, rk_{(\mu-1\to\mu)_2} = e(h, h_{n+1})^{t_u},$$

and

$$\begin{aligned} rk_{(\mu-1\to\mu)_3} &= \Big(v \cdot \prod_{j\in K} h_{n+1-j}\Big)^{T_\mu} \cdot \prod_{j\in K} \big(h_{n+1-j}\big)^{x_{id_{\mu_j}}} \\ &= \Big(v \cdot \prod_{j\in K} h_{n+1-j}\Big)^{T_\mu} \cdot H(id_{\mu_j})^{\alpha^{n+1-j}} \end{aligned}$$

Therefore, the re-encryption key $rk_{\mu-1\to\mu}$ is a valid re-encryption key.

- **Challenge**. After receiving two messages $m_0, m_1 \in \mathcal{M}$, $\mathcal{B}$ randomly chooses $b \in \{0, 1\}$. We write $h' = h^{T_\mu^*}$ for some unknown $T_\mu^* \in \mathbb{Z}_q^*$. $\mathcal{B}$ sets the challenging ciphertext c_μ^* as

$$c_{\mu,1}^* = h' = h^{T_\mu^*}, c_{\mu,2}^* = m_b \cdot Z.$$

If $Z = e\,(h, h_{n+1})^{T_\mu^*}$, we have $c_{\mu,2}^* = m \cdot e\,(h, h_{n+1})^{T_\mu^*}$ and

$$\begin{aligned} c_{\mu,3}^* &= \big(h'\big)^u \cdot \prod_{j\in K} \big(h_{n+1-j}\big)^{x_{id_{\mu_j}^*}} \\ &= (h^u \cdot (\prod_{j\in K} h_{n+1-j})^{-1} \cdot (\prod_{j\in K} h_{n+1-j}))^{T_\mu^*} \\ &\quad \cdot H(id_{\mu_j}^*)^{\alpha^{n+1-j}} \\ &= (v \cdot \prod_{j\in K} h_{n+1-j})^{T_\mu^*} \cdot H(id_{\mu_j}^*)^{\alpha^{n+1-j}}. \end{aligned}$$

Therefore, c_μ^* is a correct challenging ciphertext to encrypt message m_b for id.

- **Query phase 2**. $\mathcal{A}$ continues making private key, path creation, and re-encryption key queries and $\mathcal{B}$ responds to these queries like as in the query phase 1.
- **Guess**. $\mathcal{A}$ outputs a guess b' of b. If $b' = b$, $\mathcal{B}$ returns 0 to indicate $Z = e(h', h_{n+1})$; otherwise, it returns 1 to indicate that Z is random in $\mathbb{G}_T$.

This completes the simulation and the solution. $\mathcal{B}$ has the advantage ϵ' in solving the decision n-BDHE assumption in time t'. We here analyze the advantage ϵ' and time t'. If $Z \neq e\big(h', h_{n+1}\big)$, we have $\Pr\big[\mathcal{B}(\vec{y}_{\alpha,n,h}, Z) = 0 \mid Z \neq e(h', h_{n+1})\big] = (1/2)$ (indicating that B's view is independent of b). If $Z = e(h', h_{n+1})$, we have $\Pr\big[\mathcal{B}(\vec{y}_{\alpha,n,h}, Z) = 0 \mid Z = e(h', h_{n+1})\big] = (1/2) + (\epsilon/2)$ (indicating that $\mathcal{B}$'s output is dependent on $\mathcal{A}$'s output). Thus, $\mathcal{B}$'s advantage in solving the decision n-BDHE assumption is $\epsilon' = |\Pr[\mathcal{B}(\vec{y}_{\alpha,n,h}, Z) = 0 \mid Z = e(h', h_{n+1})\big] - \Pr\big[\mathcal{B}(\vec{y}_{\alpha,n,h}, Z) = 0 \mid Z \neq e(h', h_{n+1})\big]| = |(1/2) + (\epsilon/2) - (1/2)| = (\epsilon/2)$. We denote denote the time cost of the simulation $T_s = \mathcal{O}\left(q_{sk} + q_{rk} + q_{cp} + q_H\right)$, where private key generation, re-encryption key generation, path creation, hash fuction queries mainly dominate the time cost of the simulation T_s. Thus, $\mathcal{B}$ will solve the decision n-BDHE assumption with time $t' = t + T_s$.

VI. EVALUATION AND COMPARISON ANALYSIS

We first define the notations used in Table 2. Let k denote the size of each broadcast receiver set. Notations t_p and t_e denote the times consumed for a pairing operation, and a modular exponentiation in $\mathbb{G}$ or $\mathbb{G}_T$, separately. Notations Dec(Or) and Dec(Re) denote the decryption execution for the original ciphertext and the re-encryption ciphertext, respectively. Here, we omit the computing time of addition, multiplication, and hash function operations because these operations are much less modular exponentiation and pairing operations. As shown in Table 2, the computation overhead of our APIB-BPRE scheme in each algorithm is compared to other works [6], [21], [26].

- Extract. In the key extraction algorithm, KGC in works [6], [21], [26] and our APIB-BPRE only excutes a modular exponentiation operation to generate the private key for each user. However, broadcast proxy re-encryption schemes [6] and [26] cannot realize the property of autonomous path multi-hop, and the work [21] has no the character of broadcast encryption.
- Enc. Our APIB-BPRE and work [21] has lower computing cost to set the original ciphertext. Nevertheless, the delegator in works [6] and [26] has to undertake the amount of computing overhead in the encryption phase. For example, the delegator in work [26] needs to undertake $\mathcal{O}(k)$ modular exponentiation operations and a pairing operation for setting ciphertext.
- RKeyGen. Table 2 shows that schemes [6], [26] and our APIB-BPRE have lower computation overhead to generate the re-encryption key. However, the delegator in work [21] needs abundant computing overhead to set the re-encryption key because each receiver in the broadcast receiver set needs one re-encryption key.
- ReEnc. In this phase, our APIB-BPRE has no modular exponentiation and paring operations. In fact, only a few lightweight multiplication calculations are required in APIB-BPRE. On the contrary, the related works [6], [21], and [26] need to perform a large number of modular exponentiation and pairing operations to set ren-encryption ciphertext.
- Dec(Or). In the decryption algorithm for the original ciphertext, the delegator in APIB-BPRE and work [21] only executes a pairing operation to decrypt the original ciphertext. However, there are heavy computing overhead in works [6] and [26].
- Dec(Re). Table 2 shows that our APIB-BPRE has less computing cost to execute the decryption algorithm for the re-encryption ciphertext compared with IB-BPRE

TABLE 2. Computation Overhead Comparison.

Schemes	Extract	Enc	RKeyGen	ReEnc	Dec(Or)	Dec(Re)
[21]	t_e	$t_e + t_p$	$\mathcal{O}(3k)t_e + \mathcal{O}(k)t_p$	$\mathcal{O}(k)t_p$	t_p	$t_e + 2t_p$
[6]	t_e	$\mathcal{O}(k)t_e$	$\mathcal{O}(k)t_e$	$\mathcal{O}(k)t_e + 2t_p$	$\mathcal{O}(k)t_e + 2t_p$	$\mathcal{O}(k)t_e + 3t_p$
[26]	t_e	$\mathcal{O}(k)t_e + t_p$	$\mathcal{O}(k)t_e + t_p$	$\mathcal{O}(k)t_e + 8t_p$	$\mathcal{O}(k)t_e + 8t_p$	$\mathcal{O}(k)t_e + 7t_p$
Ours	t_e	$2t_p$	$\mathcal{O}(k)t_e + t_p$	$0t_e + 0t_p$	t_p	$\mathcal{O}(k)t_p$

schemes [6] and [26]. While our APIB-BPRE is less efficinet comparied with work [21] in the decryption algorithm for the re-encryption ciphertext. However, it cannot support the broadcast encryption functionality.

The comparison results displayed in Table 2 clearly show that our APIB-BPRE has the least computation overhead compared to related works.

VII. CONCLUSION

This paper designed an autonomous path broadcast proxy re-encryption as a new cryptographic primitive to support flexible data sharing in clouds. We formally define autonomous path identity-based broadcast proxy re-encryption and its security model, and demonstrate that our APIB-BPRE is CPA secure in the decision n-BDHE problem. More importantly, through performance analysis, our APIB-BPRE system is efficient and practical. In addition, our APIB-BPRE must be a multi-hop IB-BPRE, so that our APIB-BPRE system can provide much better fine-grained access control to delegation broadcast receiver sets than the traditional IB-BPRE employed in a cloud environment. It motivates researchers to design other APIB-BPRE schemes to support many interesting applications.

REFERENCES

[1] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. 21st Annu. Int. Cryptol. Conf. (CRYPTO)*, New York, NY, USA, 2001, pp. 213–229.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, Copenhagen, Denmark, 2005, pp. 457–473.

[3] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, Helsinki, Finland, 1998, pp. 127–144.

[4] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Proc. 5th Int. Conf. Appl. Cryptogr. Netw. Security*, Beijing, China, 2007, pp. 288–306.

[5] C. K. Chu, J. Weng, S. S. Chow, J. Zhou, and R. H. Deng, "Conditional proxy broadcast re-encryption," in *Proc. 14th Australas. Conf. Inf. Secur. Privacy*, Canberra, QLD, Australia, 2009, pp. 327–342.

[6] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin, "Conditional identity-based broadcast proxy re-encryption and its application to cloud email," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 66–79, Jan. 2016.

[7] 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.

[8] Q. Tang, P. Hartel, and W. Jonker, "Inter-domain identity-based proxy re-encryption," in *Proc. 4th Int. Conf. Inf. Secur. Cryptol.*, Beijing, China, 2008, pp. 332–347.

[9] T. Matsuo, "Proxy re-encryption systems for identity-based encryption," in *Proc. 1st Int. Conf. Pairing-Based Cryptogr.*, Tokyo, Japan, 2007, pp. 247–267.

[10] K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing," in *Proc. 19th Eur. Symp. Res. Comput. Security*, Warsaw, Poland, 2014, pp. 257–272.

[11] K. Liang, W. Susilo, and J. K. Liu, "Privacy-preserving ciphertext multi-sharing control for big data storage," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 8, pp. 1578–1589, Aug. 2015.

[12] A. Paul, V. Srinivasavaradhan, S. S. D. Selvi, and C. P. Rangan, "A CCA-secure collusion-resistant identity-based proxy re-encryption scheme," in *Proc. 12th Int. Conf. Provable Security*, Seoul, South Korea, 2018, pp. 111–128.

[13] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments," *Int. J. Netw. Secur.*, vol. 16, no. 1, pp. 1–13, 2014.

[14] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilitie," in *Proc. ACM Symp. Inf., Comput. Commun. Security*, Canberra, QLD, Australia, 2009, pp. 276–286.

[15] K. Li, Y. Zhang, and H. Ma, "Key policy attribute-based proxy re-encryption with matrix access structure," in *Proc. 5th Int. Conf. Intell. Netw. Collaborative Syst.*, Sep. 2013, pp. 46–50.

[16] K. Li, J. Wang, Y. Zhang, and H. Ma, "Key policy attribute-based proxy re-encryption and RCCA secure scheme," *J. Internet Services Inf. Secur.*, vol. 4, no. 2, pp. 70–82, 2014.

[17] 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, pp. 970–982, Nov. 2015.

[18] 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," *Des., Codes Cryptogr.*, vol. 86, no. 11, pp. 2587–2603, Nov. 2018.

[19] A. Paul, S. S. D. Selvi, and C. P. Rangan, "Efficient attribute-based proxy re-encryption with constant size ciphertexts," in *Proc. 21st Int. Conf. Cryptol. India*, 2020, pp. 644–665.

[20] S. Maiti and S. Misra, "P2B: Privacy preserving identity-based broadcast proxy re-encryption," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5610–5617, May 2020.

[21] Z. Cao, H. Wang, and Y. Zhao, "AP-PRE: Autonomous path proxy re-encryption and its applications," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 5, pp. 833–842, Sep. 2019.

[22] S. Berkovits, "How to broadcast A secret," in *Proc. Workshop Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, London, U.K., 1991, pp. 535–541.

[23] A. Fiat and M. Naor, "Broadcast encryption," in *Proc. 13th Annu. Int. Cryptol. Conf. (CRYPTO)*, New York, NY, USA, 1993, pp. 480–491.

[24] S. Agrawal and S. Yamada, "Optimal broadcast encryption from pairings and LWE," in *Proc. 39th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, Zagreb, Croatia, 2020, pp. 13–43.

[25] I. Kim, S. O. Hwang, W. Susilo, J. Baek, and J. Kim, "Efficient anonymous multi-group broadcast encryption," in *Proc. 18th Int. Conf. Appl. Cryptogr. Netw. Secur.*, Rome, Italy, 2020, pp. 251–270.

[26] M. Sun, C. Ge, L. Fang, and J. Wang, "A proxy broadcast re-encryption for cloud data sharing," *Multimedia Tools Appl.*, vol. 77, no. 9, pp. 10455–10469, May 2018.

[27] C. Ge, Z. Liu, J. Xia, and L. Fang, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1214–1226, May 2021.

[28] D. Boneh and X. Boyen, "Efficient selective identity-based encryption without random oracles," *J. Cryptol.*, vol. 24, no. 4, pp. 659–693, 2011.

[29] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proc. 25th Annu. Int. Cryptol. Conf. (CRYPTO)*, New York, NY, USA, 2005, pp. 258–275.

HUIDAN HU received the master's degree from the College of Mathematics and Informatics, Fujian Normal University, in 2019. She is currently pursuing the Ph.D. degree with the Department of Cryptography and Cyber Security, School of Software Engineering, East China Normal University. Her research interests include secret sharing, proxy re-encryption, and applied cryptography.

ZHENFU CAO (Senior Member, IEEE) is currently a Distinguished Professor with East China Normal University, China. Since 1981, he has been published over 400 academic papers in journals or conferences. His research interests include cryptography, number theory, and information security. He has received a number of awards, including the Ying-Tung Fok Young Teacher Award, in 1989, the National Outstanding Youth Fund of China, in 2002, and the Special Allowance by the State Council, in 2005. He was a co-recipient of the 2007 IEEE International Conference on Communications Computer Award, in 2007.

XIAOLEI DONG (Member, IEEE) is currently a Distinguished Professor with East China Normal University. She hosts a lot of research projects supported by the National Basic Research Program of China (973 Program), the National Natural Science Foundation of China, and the Special Funds on Information Security of the National Development and Reform Commission. Her research interests include cryptography, number theory, and trusted computing.