This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2899300, IEEE Transactions on Dependable and Secure Computing

Revocable Identity-Based Broadcast Proxy Re-encryption for Data Sharing in Clouds

Chunpeng Ge, *Member, IEEE,* Zhe Liu*, *Senior Member, IEEE,* Jinyue Xia, *Member, IEEE,* and Liming Fang, *Member, IEEE*

***Abstract*—Cloud computing has become prevalent due to its nature of massive storage and vast computing capabilities. Ensuring a secure data sharing is critical to cloud applications. Recently, a number of identity-based broadcast proxy re-encryption (IB-BPRE) schemes have been proposed to resolve the problem. However, the IB-BPRE requires a cloud user (Alice) who wants to share data with a bunch of other users (e.g. colleagues) to participate the group shared key renewal process because Alice's private key is a prerequisite for shared key generation. This, however, does not leverage the benefit of cloud computing and causes the inconvenience for cloud users. Therefore, a novel security notion named revocable identity-based broadcast proxy re-encryption (RIB-BPRE) is presented to address the issue of key revocation in this work. In a RIB-BPRE scheme, a proxy can revoke a set of delegates, designated by the delegator, from the re-encryption key. The performance evaluation reveals that the proposed scheme is efficient and practical.**

***Index Terms*—Proxy Re-Encryption, Cloud Data Sharing, Broadcast Encryption, Revocation.**

I. Introduction

CLOUD computing has become a solution for data maintenance due to its flexibility and effectiveness. However, cloud computing has been suffering from security and privacy challenges. Encryption can be a straightforward approach to ensure data confidentiality and Identity-based encryption (IBE) is one of the promising representative secure mechanisms because it has a concise public key infrastructure [1]–[3]. When storing the identity-based encrypted data to the cloud, the data owner would like to share the data with others in particular scenarios. For example, a set of volunteers upload their genome data to the cloud in a genome record cloud system for the scientists to collaboratively conduct medical research [4]. If IBE is adopted into such a medical system, the genome data should be encrypted before uploading to the cloud as $Enc(m, id)$, where m is the genome data and id is the recipient's identity. A researcher Alice with the identity id from the genome research institute may want to share the volunteer's genome data with a list of her colleagues with identities $id_1, \cdots, id_n$ in the same research group.

However, there are quite a few potential flaws of IBE in above example. First, the user Alice has to download the encrypted genome data $Enc(m, id)$ which has been sent to her, then decrypts it and further re-encrypts m with identities $id_1, \cdots, id_n$ for each colleague she wants to share respectively. If some of Alice's colleagues leave to another research group, then Alice needs to revoke these identities from the sharing list because the genome data should not be available for an unauthorized staff. In such a scenario, although the traditional identity-based encryption can guarantee the confidentiality of data, it is lacking of the flexibility of data sharing. Second, IBE does not scale well in the above scenario. In order to share the genome data with peers, Alice needs to download all the ciphertext that contains the genome data, decrypt them and then re-encrypt the records with identities in the sharing list. Such a process brings a lot of extra burden to Alice as the number of ciphertexts grows because Alice sends a ciphertext to each identity on the sharing list which leads the communication cost linear to the size of sharing group. Moreover, downloading data from the cloud yields a new problem for data maintenance. So this solution of IBE does not embrace the advantages of cloud computing either. Third, Alice should remain to be available at each time when there is a change to group since her private key is required for shared secret generation.

Alternatively, one may think that Alice can delegate the cloud server to process the decryption and re-encryption work for her. In order for the cloud to do the task on behalf of her, Alice has to save her private key in the cloud and expose it to the cloud server. In this manner, however, the cloud has to be fully trusted by Alice as it gains Alice's secret key. Otherwise, it would be a disaster if the cloud is disclosed. For example, the leakage of personal genome data will seriously damage the privacy of the volunteer since it contains many sensitive personal information, such as allergies, vaccinations and illness.

Therefore, the challenge is how to implement a medical research system to support the researchers to share the extremely sensitive genome data among them without disclosing any private information from volunteers. It is desirable to find a new identity-based mechanism that supports to easily share outsourced encrypted data. In prior, the concept of proxy re-encryption came out to enable sharing outsourced encrypted data between users without revealing the underlying plaintext to the cloud server. So it could be a potential approach to address our research question as embedding proxy re-encryption into cloud also leverages the benefit of cloud computing — not only is the data saved on the cloud but the cloud server also can play a role as a proxy to do complex re-encryption computations.

* Zhe Liu is the corresponding author.

Chunpeng Ge, Zhe Liu and Liming Fang are with College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, NO.29 Yudao Street, Nanjing, China. E-mail: {gecp, zhe.liu, fangliming}@nuaa.edu.cn.

Jinyue Xia is with the affiliation of IBM, 3039 E Cornwallis Rd, Research Triangle Park, NC 27709, USA. E-mail: jinyue.xia@ibm.com.

Identity-Based Proxy Re-Encryption (IB-PRE). Proxy re-encryption was proposed to enable a semi-trust proxy to convert a ciphertext with one's identity to a new ciphertext under a different identity [5]. Later on, the notion of IB-PRE [6] was introduced to simplify PKI (Public Key Infrastructure) since the user's identity can be considered as a replacement of the public key in an IB-PRE scheme. One might think the IB-PRE can be a trivial solution to partially address the IBE drawback described in above application in a cloud environment. For example, Alice, with identity id, can generate a re-encryption key $rk_{id \to id_i}, \cdots, rk_{id \to id_n}$ for each colleague in the delegation list $S = \{id_1, id_2, \cdots, id_n\}$ then she forwards these re-encryption keys to the cloud server. As soon as the server receives the keys, it has the flexibility to re-encrypt the ciphertext for each delegatee accordingly. Moreover, with IB-PRE, it is convenient to revoke the individual's re-encryption by simply removing the user from the delegation/revocation list. However, similar to IBE, this solution is very inefficient as Alice is required to compute a re-encryption key for every delegate, in which the number of re-encryption keys is linear to the total counts of delegatees ($O(n)$). Consequently, IB-PRE will not scale well if a huge number of delegatees exist in the group.

Identity-Based Broadcast Proxy Re-Encryption (IB-BPRE). The notion of broadcast proxy re-encryption (BPRE) [7] has been proposed to eliminate the linear computation for re-encryption key generation. Doing so can also resolve the heavy computation issue of IBE. Instead of generating re-encryption key for every single delegatee in the group, a proxy (e.g. a cloud server) only needs to have a broadcast re-encryption key in a BPRE scheme to transform a delegator's ciphertext to a set of delegatees' ciphertext without revealing plaintext to the proxy. Since then, some researchers introduced the notion of identity-based broadcast proxy re-encryption where the user's identity is used as its public key [8]. Despite the potential heavy communication of re-encryption key is resolved by IB-BPRE, key revocation problem still exists in IB-BPRE. Some may argue that Alice can generate a new broadcast re-encryption key as soon as each revocation occurs. As we pointed out earlier, this brings inconvenience to the user Alice since she has to show and present her private key to produce the broadcast re-encryption key. Such a process violates the original intention of cloud computing which is leaving the heavy computing task to the cloud not the user. Moreover, if re-encryption key is leaked in existing IB-BPRE schemes, anybody who obtained the key can re-encrypt the ciphertext. Hence, Alice needs to establish a secure channel to transmit the re-encryption key for each re-encryption key update.

A. *Motivation*

Although existing IB-PRE and IB-BPRE schemes can practically address the drawbacks of IBE in cloud data sharing, they are not suitable to solve the problem of revocation. However, revocation is very important since we should protect the volunteers' genome data from unauthorized users. This, therefore, motivates us to discover a new identity-based

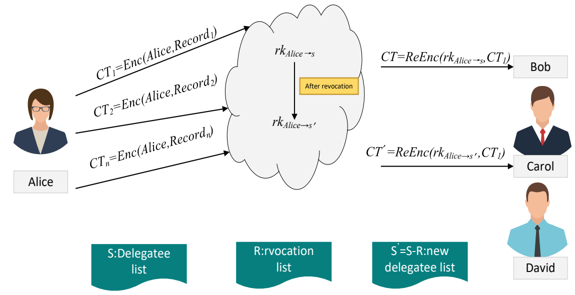


Fig. 1: RIB-BPRE in a genome research system

mechanism that supports to easily share outsourced encrypted data and sharing revocations. More specifically, an IB-BPRE scheme should have the ability of sharing revocation so that it provides a flexible revocation mechanism to allow the user to revoke any party in case some of her peers leave the research group. Imagine a research scientist Alice conducts research with her peers and she wants to share a set of genome data (e.g. $R_1, R_2, \cdots, R_n$) from a variety of volunteers, IB-BPRE should support the following features:

- Alice encrypts each volunteer's genome data under her identity and sends the encrypted genome data to the proxy — the cloud server. She also maintains a list S of delegatees (her colleagues). For the proxy to run re-encryption, Alice computes a re-encryption key (rk_S) and shared rk_S with the cloud server. If the list S does not change, the proxy is able to re-encrypt the encrypted data from Alice as normal. Once the delegatees receive the data, they can decrypt it by their own private keys.
- One day, one of Alice's research fellows, Bob, decides to quit the job. Thus, the system must revoke Bob's access to the data because he is no longer an authorized staff. Then Alice can create a revocation list (R), update her delegatee list ($S' = S - R$) and notify the proxy there is a change to delegatee list. In this case, the proxy can re-generate the re-encryption key ($rk_{S'}$) without knowing Alice's private key, which is the beauty of our RIB-BPRE scheme.

Figure 1 illustrates the idea of a RIB-BPRE system for medical research. In such a system, the user Alice herself maintains the delegatee revocation list. With the motivation in mind, we present a novel security notion — revocable identity-based broadcast proxy re-encryption (RIB-BPRE). In the RIB-BPRE scheme, the proxy can revoke a set of delegates, designated by the delegator, from the re-encryption key.

B. *Related Work*

The primitive of broadcast encryption was first pointed out by Berkovits [9] to enable a sender to broadcast a ciphertext to a set of users and each user from the recipient list is able

TABLE I: Functionality, Security and Technique Comparison with [8], [25], [31].

Schemes	Broadcast?	Revocable?	Collusion Resistant?	Technique to achieve Revocability[1]
Proposed IB-PRE scheme [25]	×	×	×	combine id with T[2]
Proposed IB-BPRE scheme [8]	✓	×	✓	combine id with T
Proposed IB-BPRE scheme [31]	✓	×	✓	combine id with T
Our Scheme	✓	✓	✓	re-randomization

[1]In the previous schemes [8], [25], [31], the revocability is not achieved, the term "Technique to achieve Revocability" represents the possible techniques that can be leveraged to realize revocability.
[2]The term id and T represent a user's identity and a separate time period T.

to decrypt the ciphertext. Fiat and Naor [10] formalized the definition and security model for broadcast encryption. After that, many broadcast encryption schemes were proposed to improve the efficiency [11]–[13]. Sakai and Furukawa [14] presented the notion of identity-based broadcast encryption (IBBE), in which an user's identity is considered as the public key in an identity based broadcast encryption. Delerablee [15] proposed an IBBE scheme with the ciphertext that has a constant size. While IBE offers the convenience on key management, it suffices a limitation of revoking user's identity. Boneh and Franklin [1] gave a seminal solution. In their scheme, the user's public key is replaced by an actual identity id and a separate time period T. Boldyreva, Goyal and Kumar [16] reduced the revocation cost from linear to logarithmic. Recently, Susilo et al. [17] presented an IBBE with a new idea for revocation that supports to directly revoke recipients from the original recipient list. Further, many attribute-based encryption (ABE) were proposed to enable the expression of identity [3], [18]–[21].

An notion of proxy re-encryption was proposed to delegate the decryption correctly [5]. Many schemes were proposed to deal with the functionality, efficiency, and security model [22]–[24]. Green and Ateniese [6] applied identity-based encryption to proxy re-encryption in an identity-based proxy re-encryption scheme. Subsequently, lots of IB-PRE schemes [25]–[30] were proposed mainly to focus on the functionality, efficiency and security. Another interesting research thread is BRPE. For instance, Chu et al. [7] proposed a broadcast proxy re-encryption scheme that enables a proxy to transform Alice's ciphertext to a set of delegates. Following their work, Xu et al. [8] and Sun et al. [31] proposed IB-BPRE schemes in which both their private key and ciphertext have a constant size. Unfortunately, none of these work addressed the re-encryption key revocation issue.

Table I summarizes the comparison of our proposed scheme with previous IB-PRE scheme [25] and IB-BPRE schemes [8], [31] in the aspects of scheme functionality, security analysis and the technique. It reveals that only our scheme supports both the broadcast re-encryption and revocation functionality compared with the other schemes [8], [25], [31]. Meanwhile, our scheme achieves the collusion resistant property as well.

C. Our Contribution

In this work, we adopted the revocation mechanism (recipient revocable) proposed for IBBE [17] to address key revocation issue for IB-BPRE. Although the approach sounds straightforward, there are technical difficulties to apply recipient revocation notion to IB-BPRE because we found the method is vulnerable to the collusion attack. A recipient colludes with the proxy can reveal the delegator's private key. Details can be found in Appendix **A**. Other than this recipient revocable method, one possible attempt is the approach proposed in [8]. In their scheme, another more N elements should be added in the public key for randomness. When generating a re-encryption key, a user Alice introduced a polynomial for variable μ with degree less than N to randomize her private key. Thus, a delegatee colluding with the proxy can not reveal Alice's private key. However, their scheme can not achieve the revocation functionality. Therefore, achieving an revocable identity-based broadcast proxy re-encryption scheme is a challenging work.

In this paper, we introduce an identity-based broadcast proxy re-encryption mechanism with revocation on delegated recipients. Our notion allows the sharing functionality on encrypted cloud data and revocation on delegated recipients. We present a concrete RIB-BPRE construction and prove it is semantic secure in the random model. Additionally, the evaluation demonstrates that our scheme is efficient and practical in terms of performance.

II. Definitions

This section describes the definition of revocable identity-based broadcast proxy re-encryption and the semantic security model.

A. RIB-BPRE

A RIB-BPRE system consists of a delegator, a proxy and a set of delegatees. In the system, the delegator outsources his encrypted data to the proxy. The delegator first generates a broadcast re-encryption key which will be used to transform his ciphertext to the delegatees' ciphertext. Then he sends the re-encryption key to the proxy so that the proxy can re-encrypt the delegator's ciphertext on his behalf. Whenever the delegator wants to revoke a set of identities R from the sharing list S, he needs to update the revocation list and sync the new sharing list with the proxy. After receiving the updated list, the proxy computes a new re-encryption key. We present the definition of RIB-BPRE as follows.

Definition 1 (RIB-BPRE). A revocable identity-based broadcast proxy re-encryption scheme consists of the following algorithms:

- $Setup(\lambda, N) \rightarrow (mpk, msk)$: The $Setup$ algorithm is run by a trusted party, on input a security parameter λ and the maximum number N of receivers in one encryption. Outputs the master public parameters mpk and a master secret key msk.
- $Extract(msk, id) \rightarrow sk_{id}$: The $Extract$ algorithm is run by the trusted party to generate a private key for each identity. It takes as input the master secret key msk, and an identity id, outputs a private key sk_{id}.
- $Enc(id, M) \rightarrow C$: The encryption algorithm Enc is run by anyone who encrypts the message with the delegator's identity. It takes as input a message M, an identity id, outputs the original ciphertext C that can be further re-encrypted.
- $RKeyGen(id, sk_{id}, S, k) \rightarrow rk$: The $RKeyGen$ algorithm is run by the delegator to generate a re-encryption key. It takes as input an identity id, private key sk_{id}, a set of delegates' identities $S = \{id_1, \cdots, id_n\}$ and a maximum revocation number k, where $id \notin S$ and $k \leq n \leq N$. Outputs the re-encryption key rk. The re-encryption key rk can be used to convert an original ciphertext C under id to a new broadcast ciphertext CT under S.
- $Revoke(rk, S, R) \rightarrow rk'$: The $Revoke$ algorithm is run by a proxy to generate a new re-encryption key that revokes identities from the sharing list. It takes as input a re-encryption key rk for identity set S, a revocation identity set R, where $R \subseteq S$ and $|R| \leq K$. Outputs a new re-encryption key rk'. The re-encryption key rk' can be used to convert an original ciphertext C under id to a new ciphertext CT under $S - R$.
- $ReEnc(C, rk) \rightarrow CT/\perp$: The re-encryption algorithm $ReEnc$ is run the proxy to transform the delegator's ciphertext to the delegatees' ciphertext. It takes as input an original ciphertext C, a re-encryption key rk, outputs the re-encrypted ciphertext CT or an error symbol $\perp$.
- $Dec(sk_{id}, C/CT) \rightarrow m/\perp$: The Dec algorithm is run by the delegator (or a delegatee) to decrypt the original ciphertext (or re-encrypted ciphertext). It takes as input a private key sk_{id}, an original/re-encrypted ciphertext C/CT. Outputs the plaintext M if the ciphertext is a valid ciphertext or an error symbol $\perp$ otherwise.

Note that, we omit the master public parameter mpk as other algorithms' input for the simplicity.

Consistency: The consistency of a RIB-BPRE scheme means any correctly generated ciphertext can be decrypted by a valid private key. Formally, for an message M, $(mpk, msk) \leftarrow Setup(\lambda, N)$, $sk_{id} \leftarrow KeyGen(msk, id)$, $rk \leftarrow RKeyGen(id, sk_{id}, S, k)$, $rk' \leftarrow Revoke(rk, S, R)$ we have

$$Dec(sk_{id}, Enc(id, M)) = M;$$
$$Dec(sk_{id'}, ReEnc(Enc(id, M), rk)) = M;$$
$$Dec(sk_{id''}, ReEnc(Enc(id, M), rk')) = M;$$

where $id \notin S$, $id' \in S$ and $id'' \in S - R$.

B. Security Model for RIB-BPRE

The security model for RIB-BPRE considers the semantic security for the original and re-encrypted ciphertext. We consider two security games between an adversary and a challenger, which guard the semantic security of the original ciphertext and re-encrypted ciphertext respectively.

Definition 2 (IND-CPA-Or). A RIB-BPRE scheme is indistinguishable chosen plaintext secure at original ciphertext (IND-CPA-Or) if no probability polynomial time (PPT) adversary $\mathcal{A}$ can win the following game with a non-negligible advantage.

1) **Init.** The adversary $\mathcal{A}$ outputs a challenge identity id^*.
2) **Setup.** The challenger $\mathcal{C}$ performs $Setup(\lambda, N)$ to get the public parameter mpk and the master secret key msk. Returns mpk to the adversary $\mathcal{A}$.
3) **Phase I.** The adversary $\mathcal{A}$ makes the following queries:
   a) Key extraction query $\mathcal{O}_{sk}(id)$: On input an identity id, if $id = id^*$, the challenger $\mathcal{C}$ aborts and returns an error symbol $\perp$. Otherwise the challenger $\mathcal{C}$ runs algorithm $KeyGen(msk, id)$ to obtain the private key sk_{id}. Returns sk_{id} to the adversary $\mathcal{A}$.
   b) Re-encryption key query $\mathcal{O}_{rk}(id, S, k)$: On input an identity id, an identity set S and the maximum revocation number k, where $id \notin S$, the challenger $\mathcal{C}$ runs $sk_{id} \leftarrow KeyGen(msk, id)$ and $rk \leftarrow RKeyGen(id, sk_{id}, S, k)$. Returns rk to the adversary $\mathcal{A}$. The restriction is that $\mathcal{A}$ can not make $\mathcal{O}_{rk}(id, S, k)$ query if $id = id^*$ and $\mathcal{A}$ has made a $\mathcal{O}_{sk}(id)$ query for $id \in S$.
   c) Re-encryption query $\mathcal{O}_{re}(C, id, S, k)$: On input an original ciphertext C under identity id, an identity set S and the maximum revocation number k, where $id \notin S$. Returns the re-encryption result $CT = ReEnc(C, RKeyGen(id, S, k))$ to the adversary $\mathcal{A}$.
4) **Challenge.** Once $\mathcal{A}$ decides Phase I is over, it outputs two equal length messages (M_0, M_1). Challenger $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$ and sets the challenge ciphertext to be $C^* = Enc(id^*, M_b)$. Finally returns the challenge ciphertext C^* to $\mathcal{A}$.
5) **Phase II.** $\mathcal{A}$ continues making queries as in the query phase I. Note that, $\mathcal{A}$ can not make $\mathcal{O}_{re}(C^*, id^*, S, k)$ query if $\mathcal{A}$ has made a $\mathcal{O}_{sk}(id)$ query for $id \in S$.
6) **Guess.** $\mathcal{A}$ outputs the guess b'. The adversary wins if $b' = b$.

The above adversary $\mathcal{A}$ is referred as an IND-CPA-Or adversary. Its advantage is defined as

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

Definition 3 (IND-CPA-Re). A PIB-BPRE scheme is indistinguishable chosen plaintext secure at re-encrypted ciphertext (IND-CPA-Re) if no PPT adversary $\mathcal{A}$ can win the following game with a non-negligible advantage.

1) **Init.** The adversary $\mathcal{A}$ outputs the non revoked challenge identity set $S^* = \{id_1^*, \cdots, id_{s^*}^*\}$, where $s^* \leq n$.
2) **Setup.** The challenger $\mathcal{C}$ performs $Setup(\lambda, N)$ to get the public parameter mpk and the master secret key msk. Returns mpk to the adversary $\mathcal{A}$.
3) **Phase I.** The adversary $\mathcal{A}$ makes the following queries:
   a) Key extraction query $\mathcal{O}_{sk}(id)$: On input an identity id, if $id \in S^*$, the challenger $\mathcal{C}$ aborts and returns an error symbol $\perp$. Otherwise the challenger $\mathcal{C}$ runs algorithm $KeyGen(msk, id)$ to obtain the private key sk_{id}. Returns sk_{id} to the adversary $\mathcal{A}$.
   b) Re-encryption key query $\mathcal{O}_{rk}(id, S, k)$: On input an identity id, an identity set S and the maximum revocation number k, where $id \notin S$, the challenger runs $\mathcal{C}$ runs $sk_{id} \leftarrow KeyGen(msk, id)$ and $rk \leftarrow RKeyGen(id, sk_{id}, S, k)$. Returns rk to the adversary $\mathcal{A}$.
4) **Challenge.** Once $\mathcal{A}$ decides Phase I is over, it outputs two equal length messages (M_0, M_1), a revocation identity set R^*.
   Once $\mathcal{A}$ decides Phase I is over, it outputs two equal length messages (M_0, M_1), a revocation identity set R^*. Let S^* denote the non-revoked identity set, so we have $S = S^* + R^*$ and $R^* \bigcap S^* = \emptyset$. Let $rk = RKeyGen(id, sk_{id}, S, k)$, where id is a random identity and $id \notin S$. At this phase, there are two types of challenge ciphertexts generated for S^*. First, when there is no revocation happened ($R^* = \emptyset$, $S^* = S$ and $rk^* = rk$), the challenge ciphertext is computed with rk^*. Second, if $R^* \neq \emptyset$, $rk^* = Revoke(rk, S, R^*)$, the challenge ciphertext is computed for non-revoked identity set S^*. Specifically, the Challenger $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$ and sets the challenge ciphertext CT^* as:
   Case 1: $R^* = \emptyset$. Let $rk^* = rk$,
   $$CT^* = ReEnc(Enc(id, M_b), rk^*).$$
   Case 2: $R^* \neq \emptyset$. Let $rk^* = Revoke(rk, S, R^*)$,
   $$CT^* = ReEnc(Enc(id, M_b), rk^*).$$
   Finally $\mathcal{C}$ returns the challenge ciphertext CT^* to $\mathcal{A}$.
5) **Phase II.** $\mathcal{A}$ continues making queries as in the query phase I. Note that, $\mathcal{A}$ can not make query $\mathcal{O}_{sk}(id)$ query for $id \in S^*$.
6) **Guess.** $\mathcal{A}$ outputs the guess b'. The adversary wins if $b' = b$.

The above adversary $\mathcal{A}$ is referred as an IND-CPA-Re adversary. Its advantage is defined as

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

Remarks: (1) During the IND-CPA-Re security game, there is no restriction for the re-encryption key query. Thus, the re-encryption query becomes unnecessary as the adversary can get any re-encryption key and re-encrypt the ciphertext itself. (2) During the IND-CPA-Re security game, case 1 means the challenge ciphertext is generated by a revoked re-encryption key, and case 2 means it is generated by a non-revoked re-encryption key. This fits the revocation property which indicates that the revoked re-encryption key can convert an original ciphertext to non-revoked identity set.

Definition 4 (IND-CPA). A RIB-BPRE scheme is said to be semantic secure IND-CPA, if $Adv_{\mathcal{A}}^{IND-CPA-Or}(\lambda)$ and $Adv_{\mathcal{A}}^{IND-CPA-Re}(\lambda)$ are negligible.

III. PRELIMINARIES

A. Negligible Function

A function $f : N \longrightarrow R$ is said to be negligible if for all positive integer $c \in N$ there exists a $n_c \in N$ such that $f(n) < n^{-c}$ for all $n > n_c$.

B. Bilinear Map

Let G and G_T be two multiplicative cyclic groups with the same prime order p, and g be a generator of G. A bilinear pairing is a map $e : G \times G \longrightarrow G_T$ if the following properties [1], [32] hold:

1) $e(g^a, h^b) = e(g, h)^{ab}$ for all $a, b \xleftarrow{R} Z_p^*$ and $g, h \in G$;
2) $e(g, g) \neq 1$.
3) $e(g, h)$ can be computed in polynomial time for all $g, h \in G$.

C. (f,g,F)-GDDHE Assumption

The (f,g,F)-GDDHE assumption [17] is defined as follows. Let (G, G_T, e, p) be a bilinear map group system, f, g be two co-prime polynomials with pairwise distinct roots of t and n. Let $g_0 \in G$ and $\mu_0 \in G$. The (f,g,F)-GDDHE assumption means, given a vector $\vec{y}$ as

$$g_0, \quad g_0^{\alpha}, \quad \cdots, \quad g_0^{\alpha^{2n}}, \quad g_0^{r\cdot g(\alpha)},$$

$$\mu_0, \quad \mu_0^{\alpha}, \quad \cdots, \quad \mu_0^{\alpha^{t-1}},$$

$$\mu_0^{\alpha\cdot f(\alpha)}, \quad \mu_0^{\alpha^2\cdot f(\alpha)}, \quad \cdots, \quad \mu_0^{\alpha^n\cdot f(\alpha)},$$

$$\mu_0^{r\cdot\alpha\cdot f(\alpha)}, \quad \mu_0^{r\cdot\alpha^2\cdot f(\alpha)}, \quad \cdots, \quad \mu_0^{r\cdot\alpha^n\cdot f(\alpha)},$$

and $T \in G_T$, no PPT adversary $\mathcal{A}$ can decide whether $T = e(g_0, \mu_0)^{r\cdot f(\alpha)}$ or an random element in G_T with un-negligible advantage.

Formally, for each PPT adversary $\mathcal{A}$, the following probability is negligible:

$$|\ Pr[\mathcal{A}(\vec{y}, e(g_0, \mu_0)^{r\cdot f(\alpha)})] - Pr[\mathcal{A}(\vec{y}, T)]\ |,$$

where T is a random element in G_T.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2899300, IEEE Transactions on Dependable and Secure Computing

D. Target Collision Resistant Hash Function

A hash function H is target collision resistant (TCR) [33] if given a random element x from the domain, there is no PPT adversary $\mathcal{A}$ can find an element y from the domain such that $y \neq x$ and $H(y) = H(x)$ with a negligible probability.

Formally, given an element x, for each PPT adversary $\mathcal{A}$, the following probability is negligible:

$$Pr[H(y) = H(x) \wedge y \neq x | x, y \in D]$$

where D is the domain of hash function H.

Note that, one of the three properties of cryptographic hash functions (e.g. SHA-1 or SHA-2) is target collision resistance. In our scheme, we need a target collision resistant hash function. So, for simplicity, we can just use SHA-1 or SHA-2 for our hash function.

IV. Proposed RIB-BPRE Scheme

In this section, we present our concrete construction for our scheme and further give the security proof of the proposed scheme.

A. Technical Overview

Unlike the approach commonly used in previous IB-PRE schemes [6], [25]–[27] by adding the timestamp T to an identity ID to keep the re-encryption key refresh, we propose a revocable key sharing mechanism without a setting a time period T. In our scheme, when there is a revocation request, the only thing needs to happen is to update the sharing list. Here we briefly illustrate the technical overview of our approach. Suppose the original re-encryption key for the identity set $\{id_1, \cdots, id_n\}$ comprises the group element $g^{r(\alpha+id_1)\cdots(\alpha+id_n)}$ where the random value r is in Z_p^*, if an identity id_k needs to be revoked from the sharing list, we can view the above element as $g^{r^*(\alpha+id_1)\cdots(\alpha+id_{k-1})(\alpha+id_{k+1})\cdots(\alpha+id_n)}$ where the new random number $r^* = r(\alpha + id_k)$. As a result, the corresponding re-encryption key can be viewed as a new re-encryption key for the identity set $\{id_1, \cdots, id_{k-1}, id_{k+1}, \cdots, id_n\}$ where id_k has been revoked.

B. Construction

Our proposed RIB-BPRE scheme consists of the following algorithms.

1) $Setup(\lambda, N)$: Let λ be the security parameter, N be the maximum number of receivers in one encryption, and (p, g, G, G_T, e) be the bilinear map parameters. Randomly choose $\alpha \in Z_p$, $g, \mu, Q \in G$, computes $g_1 = g^{\alpha}, g_2 = g^{\alpha^2}, \cdots, g_N = g^{\alpha^N}$, $\mu_1 = \mu^{\alpha}, \mu_2 = \mu^{\alpha^2}, \cdots, \mu_N = \mu^{\alpha^N}$ and $\nu = e(g, \mu)$. Choose two target collision resistant hash functions: $H_1 : \{0,1\}^* \rightarrow Z_p$, $H_2 : G_T \rightarrow G$.
The hash function H_1 can be implemented by a standard hash function (e.g. SHA-2) and H_2 can be computed from the output of H_1. Specifically, given an input $x \in G_T$, we first convert x to a string type and takes the string as the input for the hash function $H_1 : \{0,1\}^* \rightarrow Z_p$. With H_1's output z, where z is an element in Z_p, we can compute the hash function H_2's output as $y = g^z \in G$. The master public key mpk is $mpk = (G, G_T, e, p, g, g_1, \cdots, g_N, \mu_1, \cdots, \mu_N, \nu, Q, H_1, H_2)$. The master secret key msk is $msk = (\alpha, \mu)$,
2) $Extract(msk, id)$: On input an identity $id \in \{0,1\}^*$, the private key sk_{id} is computed as
$$sk_{id} = \mu^{1/(\alpha+H_1(id))}.$$
3) $Enc(id, M)$: To encrypt a message M under an identity id. Randomly choose $r \in Z_p$ and compute
$$C_M = M \cdot \nu^r, \quad C_0 = g^{r(\alpha+(H_1(id)))}, \quad C_1 = Q^r$$
Output the ciphertext $C = (C_M, C_0, C_1)$.
4) $RKeyGen(id, sk_{id}, S, k)$: Choose random elements $t, s \in Z_p$ and $\sigma \in G_T$. Compute
$$rk_1 = sk_{id} \cdot Q^t, \quad rk_2 = g^{\alpha t},$$
$$rk_3 = g^{tH_1(id)} \cdot H_2(\sigma),$$
$$rk_4 = e(g, \mu)^s \cdot \sigma, \quad rk_5 = g^{s \cdot \prod_{id \in S}(\alpha+H_1(id))},$$
for $i \in \{1, 2, \cdots, k+1\}$, $rk_{6,i} = \mu_i^s$.
Output the re-encryption key as
$$rk = (rk_1, rk_2, rk_3, rk_4, rk_5, (rk_{6,i})_{i \in \{1,2,\cdots,k+1\}}).$$
5) $Revoke(rk, S, R)$: On input a re-encryption key rk for S and a revocation set $R = \{id'_1, \cdots, id'_l\} \subseteq S$, the algorithm computes as follows.
   a) Denote a polynomial $F(x)$ in x as
   $$F(x) = \frac{1}{\prod_{id' \in R} H_1(id')} \prod_{id' \in R} (x + H_1(id')) = f_l x^l + f_{l-1} x^{l-1} + \cdots + f_1 x + f_0,$$
   where $f_0 = 1$.
   b) Compute
   $$rk'_4 = rk_4 \cdot e(g, \prod_{i=1}^{l} rk_{6,i}^{f_i}).$$
   c) Compute
   $$rk'_5 = rk_5^{\frac{1}{\prod_{id' \in R} H_1(id')}}.$$
   d) Compute
   $$rk'_6 = \prod_{i=1}^{l+1} rk_{6,i}^{f_{i-1}}.$$
   Output the revoked re-encryption key $rk = (rk_1, rk_2, rk_3, rk'_4, rk'_5, rk'_6)$ for $S' = S - R$.
6) $ReEnc(C, rk)$: On input an original ciphertext $C = (C_M, C_0, C_1)$ and a re-encryption key $rk = (rk_1, rk_2, rk_3, rk_4, rk_5, (rk_{6,i})_{i \in \{1,2,\cdots,k+1\}})$ for S or a revoked re-encryption key $rk = (rk_1, rk_2, rk_3, rk'_4, rk'_5, rk'_6)$ for $S' = S - R$. Compute
$$C'_M = C_M \cdot e(rk_1, C_0)^{-1} \cdot (rk_2, C_1),$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2899300, IEEE Transactions on Dependable and Secure Computing

$$C'_1 = C_1,$$

$$C'_2 = rk_3.$$

a) If $rk = (rk_1, rk_2, rk_3, rk_4, rk_5, (rk_{6,i})_{i\in\{1,2,\cdots,k+1\}})$, compute

$$C'_3 = rk_4, \quad C'_4 = rk_5, \quad C'_5 = rk_{6,1}.$$

b) If $rk = (rk_1, rk_2, rk_3, rk'_4, rk'_5, rk'_6)$, compute

$$C'_3 = rk'_4, \quad C'_4 = rk'_5, \quad C'_5 = rk'_6.$$

Output the re-encrypted ciphertext

$$CT = (C'_M, C'_1, C'_2, C'_3, C'_4, C'_5).$$

7) $Dec(sk_{id}, C/CT)$:
(a) C is an original ciphertext. Compute

$$M = C_M \cdot e(sk_{id}, C_0)^{-1}.$$

(b) CT is a re-encrypted ciphertext. Computes

$$T = \left(e({C'_5}^{-1}, g^{\rho_{i,S}(\alpha)}) \cdot e(sk_{id}, C'_4)\right)^{\frac{1}{\prod_{j=1,j\neq i}^{S} H_1(id_j)}}$$

where

$$\rho_{i,S}(\alpha) = \frac{1}{\alpha}\cdot\left(\prod_{j=1,j\neq i}^{S}(\alpha + H_1(id_j)) - \prod_{j=1,j\neq i}^{S} H_1(id_j)\right).$$

Then computes

$$\sigma = C'_3 \cdot T^{-1}, \quad g^{tH_1(id)} = C'_2 \cdot H_2(\sigma)^{-1}.$$

Finally, computes $M = C'_M \cdot e(C'_1, g^{tH_1(id)})$.

Consistency. We now explain the consistency of our proposed scheme:

1) For an original ciphertext, in the Dec algorithm we have

$$\begin{aligned} & C_M \cdot e(sk_{id}, C_0)^{-1} \\ =& M \cdot e(g,\mu)^r \cdot e\left(\mu^{1/(\alpha+H_1(id))}, g^{r(\alpha+(H_1(id)))}\right)^{-1} \\ =& M \end{aligned}$$

Thus, the consistency of an original ciphertext can be verified.

2) For a re-encrypted ciphertext, we first observe that:

$$\begin{aligned} rk'_4 &= rk_4 \cdot e(g, \prod_{i=1}^{l} rk_{6,i}^{f_i}) \\ &= \sigma \cdot e(g,\mu)^s \cdot e(g, \prod_{i=1}^{l} \mu^{\alpha^i \cdot s \cdot f_i}) \\ &= \sigma \cdot e(g,\mu)^{s\cdot\sum_{i=0}^{l} f_i \alpha^i} \\ &= \sigma \cdot e(g,\mu)^{sF(\alpha)} \\ &\triangleq \sigma \cdot e(g,\mu)^{s'}. \end{aligned}$$

Note, in the last equation s' is denote as $s' = sF(\alpha)$.

$$\begin{aligned} rk'_5 &= rk_5^{\frac{1}{\prod_{id'\in R} H_1(id')}} \\ &= g^{\frac{s\cdot\prod_{id\in S}(\alpha+H_1(id))}{\prod_{id'\in R} H_1(id')}} \\ &= g^{s\cdot\prod_{id\in S'}(\alpha+H_1(id))\cdot F(\alpha)} \\ &= g^{s'\cdot\prod_{id\in S'}(\alpha+H_1(id))}. \end{aligned}$$

Further,

$$\begin{aligned} rk'_6 &= \prod_{i=1}^{l+1} rk_{6,i}^{f_{i-1}} \\ &= \prod_{i=1}^{l+1} \mu^{\alpha^i \cdot s \cdot f_{i-1}} \\ &= \mu^{\alpha\cdot s\sum_{i=0}^{l} f_i \alpha^i} \\ &= \mu_1^{s'}. \end{aligned}$$

From the above equations, we can see that $rk_4, rk_5, rk_{6,1}$ and rk'_4, rk'_5, rk'_6 are of the same form for identity set S and S'. Next, we will verify the consistency of re-encrypted ciphertext. First, we have

$$\begin{aligned} T &= \left(e({C'_5}^{-1}, g^{\rho_{i,S}(\alpha)}) \cdot e(sk_{id}, C'_4)\right)^{\frac{1}{\prod_{j=1,j\neq i}^{S} H_1(id_j)}} \\ &= e(g,\mu)^s \end{aligned}$$

We can correctly compute $g^{tH_1(id)}$ as the decryption algorithm. For C'_M, we can compute

$$C'_M = M \cdot e(Q^t, g^{-rH_1(id)}) = M \cdot e(Q^r, g^{-tH_1(id)}).$$

Further, in the Dec algorithm for a re-encrypted ciphertext, M is correctly computed.

Thus, the consistency of a re-encrypted ciphertext are verified.

C. Security Proof

In this subsection, we prove the semantic security for our RIB-BPRE scheme. The proof is as follows.

Theorem 1. Our proposed RIB-BPRE scheme is IND-CPA secure under the $(f,g,F)-GDDHE$ assumption in the random oracle model assuming H_1, H_2 are TCR hash functions.

Lemma 1. The proposed RIB-BPRE scheme is IND-CPA-Or secure under the $(f,g,F)-GDDHE$ assumption in the random oracle model assuming H_1, H_2 are TCR hash functions.

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ that can break the IND-CPA-Or security of our scheme with advantage ϵ and time $time$. We built a simulator $\mathcal{B}$ which can solve the $(f,g,F)-GDDHE$ assumption with advantage ϵ' and time $time'$ that we will explain later. Assume n is the maximum number of identities include in one encryption, and t is the total number of extract queries, hash queries, re-encryption key queries and re-encryption queries issued by an adversary. $\mathcal{B}$ is given a $(f, g, F) - GDDHE$ instance as:

$$g_0, \quad g_0^{\alpha}, \quad \cdots, \quad g_0^{\alpha^{2n}}, \quad g_0^{r\cdot g(\alpha)},$$
$$\mu_0, \quad \mu_0^{\alpha}, \quad \cdots, \quad \mu_0^{\alpha^{t-1}},$$
$$\mu_0^{\alpha\cdot f(\alpha)}, \quad \mu_0^{\alpha^2\cdot f(\alpha)}, \quad \cdots, \quad \mu_0^{\alpha^n\cdot f(\alpha)},$$
$$\mu_0^{r\cdot\alpha\cdot f(\alpha)}, \quad \mu_0^{r\cdot\alpha^2\cdot f(\alpha)}, \quad \cdots, \quad \mu_0^{r\cdot\alpha^n\cdot f(\alpha)},$$

as well as an element $T \in G_T$, where $f(\alpha)$ and $g(\alpha)$ are two coprime polynomials in α. $\mathcal{B}$'s task is to decide whether $T \stackrel{?}{=} e(g_0, \mu_0)^{r\cdot f(\alpha)}$. We first define some notations as:

- $f(x) = \prod_{i=1}^{t}(x + \lambda_i), \qquad g(x) = \prod_{i=t+1}^{t+n}(x + \lambda_i),$
- $f_i(x) = \frac{f(x)}{x+\lambda_i}$ for $i \in [1, t]$,
- $g_i(x) = \frac{g(x)}{x+\lambda_i}$ for $i \in [t + 1, t + n]$,

The simulator $\mathcal{B}$ maintains the follows records which are initially empty.

- sk^{list}: Records tuples (id, d_{id}).
- rk^{list}: Records tuples $(id, S, k, rk, flag)$, where $flag \in \{0, 1\}$, $flag = 1$ denotes rk is a valid re-encryption key and $flag = 0$ denotes rk is a random value.

The simulator $\mathcal{B}$ works by interacting with $\mathcal{A}$ as follows:

1) Init. The adversary $\mathcal{A}$ outputs a challenge identity id^*.
2) Setup. Simulator $\mathcal{B}$ implicitly sets $\mu = \mu_0^{f(\alpha)}$ and

$$\mu_i = \mu_0^{\alpha^i f(\alpha)} = \mu^{\alpha^i}, i \in [1, n],$$
$$g = g_0^{\prod_{i=t+2}^{t+n}(\alpha+\lambda_i)},$$
$$\nu = e(g_0, \mu_0)^{f(\alpha)\cdot\prod_{i=t+2}^{t+n}(\alpha+\lambda_i)} = e(g, \mu).$$

$\mathcal{B}$ chooses a random value τ and sets $Q = \mu_0^{\tau\cdot\alpha\cdot f(\alpha)}$. The random oracles H_1 and H_2 are controlled as follows.

- H_1 queries: $\mathcal{B}$ maintains entry (id^*, λ_{t+1}) and $\{(*, \lambda_i)\}_{i=1}^{t}$ in H_1^{list} where $*$ denotes an empty entry. When $\mathcal{A}$ queries (id_i) to random oracle H_1, $\mathcal{B}$ searches an entry (id_i, λ_i) in H_1^{list} and returns the corresponding λ_i. If no such entry exists, $\mathcal{B}$ sets $H_1(id_i) = \lambda_i$, and adds (id_i, λ_i) to H_1^{list}.
- H_2 queries: When $\mathcal{A}$ queries (σ) to random oracle H_2, $\mathcal{B}$ searches an entry (σ, η) in H_2^{list} and returns the corresponding η. If no such entry exists, $\mathcal{B}$ chooses a random $\eta \in G$, and adds (σ, η) to H_2^{list}.

$\mathcal{B}$ outputs the public parameters as $mpk = (G, G_T, e, p, g, g_1, \cdots, g_n, \mu_1, \cdots, \mu_n, \nu, Q, H_1, H_2)$.

3) Phase I.
   a) $\mathcal{O}_{sk}(id_i)$: $\mathcal{A}$ issues key extract queries for id_i. If $id_i = id^*$, $\mathcal{B}$ aborts and outputs $\perp$. Otherwise $\mathcal{B}$ searches sk^{list},
      - if (id_i, sk_{id_i}) exists, returns sk_{id_i}.
      - Otherwise, $\mathcal{B}$ first queries id_i to H_1 and gets λ_i. Further, $\mathcal{B}$ computes

$$sk_{id_i} = \mu_0^{f_i(\alpha)} = \mu^{\frac{1}{\alpha+H_1(id_i)}}.$$

      Finally, $\mathcal{B}$ adds (id_i, sk_{id_i}) to sk^{list}.
   b) $\mathcal{O}_{rk}(id, S, k)$: If $id = id^*$ and $\mathcal{A}$ has made a $\mathcal{O}_{sk}(id')$ query for $id' \in S$, $\mathcal{B}$ aborts and outputs $\perp$. Otherwise $\mathcal{B}$ searches rk^{list}, if $(id, S, k, rk, *)$ exists, where $*$ is the wildcard, returns rk as the result. Otherwise proceeds,
      - If $id = id^*$ and there is no tuple $(id', sk_{id'})$ in sk^{list}, where $id' \in S$, $\mathcal{B}$ chooses random values for each element of rk. Adds $(id, S, k, rk, 0)$ to rk^{list} list.
      - Otherwise, $\mathcal{B}$ first queries $\mathcal{O}_{sk}(id)$ to get sk_{id} and then generates rk using sk_{id} via $RKeyGen$ algorithm. Adds (id, sk_{id}) and $(id, S, k, rk, 1)$ to sk^{list} and rk^{list} respectively.
   c) $\mathcal{O}_{re}(C, id, S, k)$: If there is a tuple $(id, S, k, rk, flag)$ in rk^{list}, re-encrypts C with rk. Otherwise, $\mathcal{B}$ first issues $\mathcal{O}_{rk}(id, S, k)$ to get the corresponding re-encryption key rk. Finally, $\mathcal{B}$ re-encrypts C with rk and adds $(id, S, k, rk, flag)$ to rk^{list}.
4) Challenge. Once $\mathcal{A}$ decides that Phase I is over, it outputs two equal length message (M_0, M_1). $\mathcal{B}$ chooses a random bit $b \in \{0, 1\}$ and constructs

$$C_M^* = M_b \cdot T^{\prod_{t+2}^{t+n}\lambda_i} \cdot e(g_0^{\varphi(\alpha)}, \mu_0^{r\alpha\cdot f(\alpha)}),$$

where $\varphi(\alpha) = \frac{1}{\alpha}\left(\prod_{i=t+2}^{t+n}(\alpha + \lambda_i) - \prod_{i=t+2}^{t+n}\lambda_i\right)$.

$\mathcal{B}$ then computes

$$C_0^* = g_0^{r\cdot g(\alpha)},$$

and

$$C_1^* = (\mu_0^{r\cdot\alpha\cdot f(\alpha)})^{\tau}.$$

$\mathcal{B}$ output the challenge ciphertext

$$C^* = (C_M^*, C_0^*, C_1^*).$$

Note that, by this setting, if $T = e(g_0, \mu_0)^{r\cdot f(\alpha)}$, we have

$$\begin{aligned} C_M^* &= M_b \cdot T^{\prod_{t+2}^{t+n}\lambda_i} \cdot e(g_0^{\varphi(\alpha)}, \mu_0^{r\alpha\cdot f(\alpha)}) \\ &= M_b \cdot e(g_0, \mu_0)^{r\cdot f(\alpha)\cdot\prod_{t+2}^{t+n}\lambda_i} \cdot e(g_0^{\varphi(\alpha)}, \mu_0^{r\alpha\cdot f(\alpha)}) \\ &= M_b \cdot e(g_0^{\prod_{i=t+2}^{t+n}(\alpha+\lambda_i)}, \mu_0^{f(\alpha)})^r \\ &= M_b \cdot e(g, \mu)^r, \end{aligned}$$

$$\begin{aligned} C_0^* &= g_0^{r\cdot g(\alpha)} \\ &= g_0^{r\cdot\prod_{i=t+1}^{t+n}(\alpha+\lambda_i)} \\ &= g_0^{r\cdot(\alpha+\lambda_{t+1})\cdot\prod_{i=t+2}^{t+n}(\alpha+\lambda_i)} \\ &= g^{r\cdot(\alpha+H_1(id^*))}, \end{aligned}$$

and

$$\begin{aligned} C_1^* &= (\mu_0^{r\cdot\alpha\cdot f(\alpha)})^{\tau} \\ &= (\mu_0^{\tau\cdot\alpha\cdot f(\alpha)})^{r} \\ &= Q^r. \end{aligned}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2899300, IEEE Transactions on Dependable and Secure Computing

is a valid challenge ciphertext. If T is a random value in G_T, the challenge ciphertext C^* is independent of b in the adversary's view.

5) Phase II. $\mathcal{A}$ continues making queries as in the query phase I except the restrictions described in the IND-CPA-Or game.
6) **Guess.** $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$. If $b' = b$, $\mathcal{B}$ outputs 1 to guess $T = e(g_0, \mu_0)^{r \cdot f(\alpha)}$; otherwise $\mathcal{B}$ outputs 0 to guess that T is a random element in G_T.

This completes the simulation. We here analyze the probability and time of the simulator $\mathcal{B}$ to solve the $(f,g,F)-GDDHE$ assumption. If $T \neq e(g_0,\mu_0)^{r\cdot f(\alpha)}$, the view of adversary $\mathcal{A}$ is independent of b, that means $Pr[\mathcal{B}(\vec{y},T) = 1|T \neq e(g_0,\mu_0)^{r\cdot f(\alpha)}] = \frac{1}{2}$. If $T = e(g_0,\mu_0)^{r\cdot f(\alpha)}$, the simulator $\mathcal{B}$'s output is dependent on $\mathcal{A}$'s output. More specifically, $Pr[\mathcal{B}(\vec{y},T) = 1|T = e(g_0,\mu_0)^{r\cdot f(\alpha)}] = \frac{1}{2} + \epsilon$. Thus, the simulator $\mathcal{B}$'s advantage of solve the $(f,g,F)-GDDHE$ assumption is $\epsilon' = |Pr[\mathcal{B}(\vec{y},T) = 1|T = e(g_0,\mu_0)^{r\cdot f(\alpha)}] - Pr[\mathcal{B}(\vec{y},T) = 1|T \neq e(g_0,\mu_0)^{r\cdot f(\alpha)}]| = |\frac{1}{2} + \epsilon - \frac{1}{2}| = \epsilon$. The running time of $\mathcal{B}$ is bound by $time' \leqslant time + \mathcal{O}(t(t_{H_1} + t_{H_2} + t_e + t_p))$ where t is the total number of queries that can be made by the adversary, t_{H_1} is the running time of H_1 hash function, t_{H_2} is the running time of H_2 hash function, t_e is the running time of exponentiation in group G and G_T, and t_p is the running time of a pairing.

This completes the proof of Lemma 1.

Lemma 2. The proposed RIB-BPRE scheme is IND-CPA-Re secure under the $(f,g,F)-GDDHE$ assumption in the random oracle model assuming H_1, H_2 are TCR hash functions.

Proof. Suppose there is a PPT adversary $\mathcal{A}$ that can break the IND-CPA-Re security of our scheme with advantage ϵ and time $time$. We built a simulator $\mathcal{B}$ which can solve the $(f,g,F)-GDDHE$ assumption with advantage ϵ' and time $time'$ that we will explain later. Assume n is the maximum number of identities include in one encryption, and t is the total number of extract queries, hash queries and re-encryption key queries issued by an adversary. $\mathcal{B}$ is given a $(f,g,F)-GDDHE$ instance as:

$$g_0, \quad g_0^{\alpha}, \quad \cdots, \quad g_0^{\alpha^{2n}}, \quad g_0^{r\cdot g(\alpha)},$$

$$\mu_0, \quad \mu_0^{\alpha}, \quad \cdots, \quad \mu_0^{\alpha^{t-1}},$$

$$\mu_0^{\alpha\cdot f(\alpha)}, \quad \mu_0^{\alpha^2\cdot f(\alpha)}, \quad \cdots, \quad \mu_0^{\alpha^n\cdot f(\alpha)},$$

$$\mu_0^{r\cdot\alpha\cdot f(\alpha)}, \quad \mu_0^{r\cdot\alpha^2\cdot f(\alpha)}, \quad \cdots, \quad \mu_0^{r\cdot\alpha^n\cdot f(\alpha)},$$

as well as an element $T \in G_T$, where $f(\alpha)$ and $g(\alpha)$ are two coprime polynomials in α. $\mathcal{B}$'s task is to decide whether $T \stackrel{?}{=} e(g_0,\mu_0)^{r\cdot f(\alpha)}$. We first define some notations as:

- $f(x) = \prod_{i=1}^{t}(x+\lambda_i), \qquad g(x) = \prod_{i=t+1}^{t+n}(x+\lambda_i),$
- $f_i(x) = \frac{f(x)}{x+\lambda_i}$ for $i \in [1,t]$,
- $g_i(x) = \frac{g(x)}{x+\lambda_i}$ for $i \in [t+1, t+n]$,

The simulator $\mathcal{B}$ maintains the follows records which are initially empty.

- sk^{list}: Records tuples (id, d_{id}).
- rk^{list}: Records tuples $(id, S, k, rk, flag)$, where $flag \in \{0,1\}$, $flag = 1$ denotes rk is a valid re-encryption key and $flag = 0$ denotes rk is a random value.

The simulator $\mathcal{B}$ works by interacting with $\mathcal{A}$ as follows:

1) Init. The adversary $\mathcal{A}$ outputs a challenge identity set $S^* = \{id_1^*, \cdots, id_{s^*}^*\}$, where $s^* \leq n$.
2) Setup. Simulator $\mathcal{B}$ implicitly sets $\mu = \mu_0^{f(\alpha)}$ and

$$\mu_i = \mu_0^{\alpha^i f(\alpha)} = \mu^{\alpha^i}, i \in [1,n],$$

$$g = g_0^{\prod_{i=t+1+s^*}^{t+n}(\alpha+\lambda_i)},$$

$$\nu = e(g_0,\mu_0)^{f(\alpha)\cdot\prod_{i=t+1+s^*}^{t+n}(\alpha+\lambda_i)} = e(g,\mu).$$

$\mathcal{B}$ chooses a random value $Q \in G$. The random oracles H_1 and H_2 are controlled as follows.

- H_1 queries: $\mathcal{B}$ maintains entry $\{(id_i^*, \lambda_i)\}_{i=t+1}^{t+s^*}$ and $\{(*, \lambda_i)\}_{i=1}^{t}$ in H_1^{list} where $*$ denotes an empty entry. When $\mathcal{A}$ queries (id_i) to random oracle H_1, $\mathcal{B}$ searches an entry (id_i, λ_i) in H_1^{list} and returns the corresponding λ_i. If no such entry exists, $\mathcal{B}$ sets $H_1(id_i) = \lambda_i$, and adds (id_i, λ_i) to H_1^{list}.
- H_2 queries: When $\mathcal{A}$ queries (σ) to random oracle H_2, $\mathcal{B}$ searches an entry (σ, η) in H_2^{list} and returns the corresponding η. If no such entry exists, $\mathcal{B}$ chooses a random $\eta \in G$, and adds (σ, η) to H_2^{list}.

$\mathcal{B}$ outputs the public parameters as $mpk = (G, G_T, e, p, g, g_1, \cdots, g_n, \mu_1, \cdots, \mu_n, \nu, Q, H_1, H_2)$.

3) Phase I.
   a) $\mathcal{O}_{sk}(id_i)$: $\mathcal{A}$ issues key extract queries for id_i. If $id_i \in S^*$, $\mathcal{B}$ aborts and outputs $\perp$. Otherwise $\mathcal{B}$ searches sk^{list},
      - if (id_i, sk_{id_i}) exists, returns sk_{id_i}.
      - Otherwise, $\mathcal{B}$ first queries id_i to H_1 and gets λ_i. Further, $\mathcal{B}$ computes

$$sk_{id_i} = \mu_0^{f_i(\alpha)} = \mu^{\frac{1}{\alpha+H_1(id_i)}}.$$

      Finally, $\mathcal{B}$ adds (id_i, sk_{id_i}) to sk^{list}.
   b) $\mathcal{O}_{rk}(id, S, k)$: $\mathcal{B}$ searches rk^{list}, if $(id, S, k, rk, *)$ exists, where $*$ is the wildcard, returns rk as the result. Otherwise proceeds,
      - If $id = id^*$, $\mathcal{B}$ chooses random values for each element of rk. Adds $(id, S, k, rk, 0)$ to rk^{list} list.
      - Otherwise, $\mathcal{B}$ first queries $\mathcal{O}_{sk}(id)$ to get sk_{id} and then generates rk using sk_{id} via $RKeyGen$ algorithm. Adds (id, sk_{id}) and $(id, S, k, rk, 1)$ to sk^{list} and rk^{list} respectively.
4) Challenge. Once $\mathcal{A}$ decides that Phase I is over, it outputs two equal length message (M_0, M_1), a revocation identity set $R^* = \{id_1', \cdots, id_l'\}$, where $l \leq k$ and $R^* \bigcap S^* = \emptyset$. Let $S = S^* + R^*$, $\mathcal{B}$ chooses a random bit $b \in \{0,1\}$, $r', t \in Z_p$, $\sigma \in G_T$ and a random identity $id_i \notin S$. $\mathcal{B}$ first issues (id_i) to H_1 and (σ) to H_2, and gets the corresponding λ_i and η. $\mathcal{B}$ then

computes ${C'_M}^* = M_b \cdot e(Q^{r'}, g^{-t\lambda_i})$, ${C'_1}^* = Q^{r'}$ and ${C'_2}^* = \eta \cdot g^{t\lambda_i}$. $\mathcal{B}$ further computes

$$ {C'_3}^* = \eta \cdot T^{\prod_{t+1+s^*}^{t+n} \lambda_i} \cdot e(g_0^{\psi(\alpha)}, \mu_0^{r\alpha \cdot f(\alpha)}), $$

where $\psi(\alpha) = \frac{1}{\alpha}\left(\prod_{i=t+1+s^*}^{t+n}(\alpha+\lambda_i) - \prod_{i=t+1+s^*}^{t+n}\lambda_i\right)$. $\mathcal{B}$ then computes

$$ {C'_4}^* = g_0^{r \cdot g(\alpha)}, $$

and

$$ {C'_5}^* = \mu_0^{r\cdot\alpha\cdot f(\alpha)}. $$

Case 1: $R^* = \emptyset$. If $T = e(g_0, \mu_0)^{r\cdot f(\alpha)}$, we have

$$ \begin{aligned} &{C'_3}^* \\ =&\eta \cdot T^{\prod_{t+1+s^*}^{t+n}\lambda_i} \cdot e(g_0^{\psi(\alpha)}, \mu_0^{r\alpha\cdot f(\alpha)}) \\ =&\eta \cdot e(g_0, \mu_0)^{r\cdot f(\alpha)\cdot \prod_{t+1+s^*}^{t+n}\lambda_i} \cdot e(g_0^{\psi(\alpha)}, \mu_0^{r\alpha\cdot f(\alpha)}) \\ =&\eta \cdot e(g_0^{\prod_{i=t+1+s^*}^{t+n}(\alpha+\lambda_i)}, \mu_0^{f(\alpha)})^r \\ =&\eta \cdot e(g, \mu)^r, \end{aligned} $$

$$ \begin{aligned} {C'_4}^* &= g_0^{r\cdot g(\alpha)} \\ &= g_0^{r\cdot\prod_{i=t+1}^{t+n}(\alpha+\lambda_i)} \\ &= g_0^{r\cdot\prod_{i=t+1}^{t+s^*}(\alpha+\lambda_i)\cdot\prod_{i=t+1+s^*}^{t+n}(\alpha+\lambda_i)} \\ &= g^{r\cdot\prod_{id^*\in S^*}(\alpha+H_1(id^*))}, \end{aligned} $$

and

$$ \begin{aligned} {C'_5}^* &= \mu_0^{r\cdot\alpha\cdot f(\alpha)} \\ &= \mu_1^r. \end{aligned} $$

is a valid challenge ciphertext. If T is a random value in G_T, the challenge ciphertext CT^* is independent of b in the adversary's view.

Case 2: $R^* \neq \emptyset$. $\mathcal{B}$ randomly chooses $r^* \in Z_p$ and computes

$$ {C'_3}^* = \eta \cdot \nu^{r^*\cdot\frac{\prod_{i=1}^{l}(\alpha+H_1(id'_i))}{\prod_{i=1}^{l}H_1(id'_i)}}, $$

$$ {C'_4}^* = g^{r^*\cdot\frac{\prod_{id\in S}(\alpha+H_1(id))}{\prod_{i=1}^{l}H_1(id'_i)}}, $$

$$ {C'_5}^* = \mu_1^{r^*\cdot\frac{\prod_{i=1}^{l}(\alpha+H_1(id'_i))}{\prod_{i=1}^{l}H_1(id'_i)}}. $$

Finally, $\mathcal{B}$ returns the challenge ciphertext

$$ CT^* = ({C'_M}^*, {C'_1}^*, {C'_2}^*, {C'_3}^*, {C'_4}^*, {C'_5}^*). $$

Note that, we denote $r = r^* \cdot \frac{\prod_{id\in S}(\alpha+H_1(id))}{\prod_{i=1}^{l}H_1(id'_i)}$, then the challenge ciphertext can be compute as

$$ \begin{aligned} {C'_3}^* &= \eta \cdot \nu^{r^*\cdot\frac{\prod_{i=1}^{l}(\alpha+H_1(id'_i))}{\prod_{i=1}^{l}H_1(id'_i)}} \\ &= \eta \cdot \nu^r, \end{aligned} $$

$$ \begin{aligned} {C'_4}^* &= g^{r^*\cdot\frac{\prod_{id\in S}(\alpha+H_1(id))}{\prod_{i=1}^{l}H_1(id'_i)}} \\ &= g^{r^*\cdot\frac{\prod_{i\in 1}^{l}(\alpha+H_1(id_i))}{\prod_{i=1}^{l}H_1(id'_i)}\cdot\prod_{id\in S^*}(\alpha+H_1(id))} \\ &= g^{r\cdot\prod_{id\in S^*}(\alpha+H_1(id))} \end{aligned} $$

and

$$ \begin{aligned} {C'_5}^* &= \mu_1^{r^*\cdot\frac{\prod_{i=1}^{l}(\alpha+H_1(id'_i))}{\prod_{i=1}^{l}H_1(id'_i)}} \\ &= \mu_1^r. \end{aligned} $$

We can see that $CT^* = ({C'_M}^*, {C'_1}^*, {C'_2}^*, {C'_3}^*, {C'_4}^*, {C'_5}^*)$ is a valid challenge ciphertext when $T = e(g_0, \mu_0)^{r\cdot f(\alpha)}$. When T is a random value in G_T, the challenge ciphertext CT^* is independent of b in the adversary's view.

5) Phase II. $\mathcal{A}$ continues making queries as in the query phase I except the restrictions described in the IND-CPA-Re game.
6) **Guess.** $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$. If $b' = b$, $\mathcal{B}$ outputs 1 to guess $T = e(g_0, \mu_0)^{r\cdot f(\alpha)}$; otherwise $\mathcal{B}$ outputs 0 to guess that T is a random element in G_T.

This completes the simulation. We here analyze the probability and time of the simulator $\mathcal{B}$ to solve the $(f,g,F)-GDDHE$ assumption. If $T \neq e(g_0,\mu_0)^{r\cdot f(\alpha)}$, the view of adversary $\mathcal{A}$ is independent of b, that means $Pr[\mathcal{B}(\vec{y},T) = 1 | T \neq e(g_0,\mu_0)^{r\cdot f(\alpha)}] = \frac{1}{2}$. If $T = e(g_0,\mu_0)^{r\cdot f(\alpha)}$, the simulator $\mathcal{B}$'s output is dependent on $\mathcal{A}$'s output. More specifically, $Pr[\mathcal{B}(\vec{y},T) = 1 | T = e(g_0,\mu_0)^{r\cdot f(\alpha)}] = \frac{1}{2} + \epsilon$. Thus, the simulator $\mathcal{B}$'s advantage of solve the $(f,g,F)-GDDHE$ assumption is $\epsilon' = |Pr[\mathcal{B}(\vec{y},T) = 1 | T = e(g_0,\mu_0)^{r\cdot f(\alpha)}] - Pr[\mathcal{B}(\vec{y},T) = 1 | T \neq e(g_0,\mu_0)^{r\cdot f(\alpha)}]| = |\frac{1}{2} + \epsilon - \frac{1}{2}| = \epsilon$. The running time of $\mathcal{B}$ is bound by $time' \leqslant time + \mathcal{O}(t(t_{H_1} + t_{H_2} + t_e + t_p))$ where t is the total number of queries that can be made by the adversary, t_{H_1} is the running time of H_1 hash function, t_{H_2} is the running time of H_2 hash function, t_e is the running time of exponentiation in group G and G_T, and t_p is the running time of a pairing.

This completes the proof of Lemma 2.

In summary, with definition 4, Lemma 1, and Lemma 2, we completes the proof of Theorem 1.

D. Collusion Resistant and Non-transferability

In this subsection, we discuss the collusion resistant and non-transferability properties of our scheme.

Collusion resistant. The collusion resistant property ensures the proxy cannot reveal the delegator's private key by colluding with a set of delegatees. In our proposed scheme, the delegator's private key sk_{id} is randomized by the random element Q^t as $rk_1 = sk_{id} \cdot Q^t$, where $Q \in G$ is a public parameter and $t \in Z_p$ is a randomly chosen element. The rk_4, rk_5 and $\{rk_{6,i}\}_{i\in\{1,\cdots,k+1\}}$ are the broadcast encryption ciphertext of σ under the set S. When the proxy colludes

TABLE II: Computation Comparison with [25], [8] and [31].

Schemes	Extract	Enc	RKeyGen	ReEnc	Dec(Or)	Dec(Re)
scheme [25]	$3e$	$p+3e$	$p+4e$	$2p$	$2p$	$3p$
scheme [8]	e	$\mathcal{O}(\|S\|)e$	$\mathcal{O}(\|S\|)e$	$\mathcal{O}(\|S\|)e+2p$	$\mathcal{O}(\|S\|)e+2p$	$\mathcal{O}(\|S\|)e+3p$
scheme [31]	e	$\mathcal{O}(\|S\|)e+p$	$\mathcal{O}(\|S\|)e+p$	$\mathcal{O}(\|S\|)e+8p$	$\mathcal{O}(\|S\|)e+8p$	$\mathcal{O}(\|S\|)e+7p$
Ours	e	$4e$	$\mathcal{O}(\|S\|)e$	$e+2p$	$e+2p$	$\mathcal{O}(\|S\|)e+3p$

TABLE III: Execute Time.

Algorithms	Extract (ms)	Enc (ms)	RKeyGen (ms)	Revoke (ms)	ReEnc (ms)	Dec(Or) (ms)	Dec(Re) (ms)
$\|S\|=20, k=12, l=10$	3.236	7.593	66.052	6.976	4.127	2.070	39.863
$\|S\|=30, k=18, l=15$	3.235	7.548	84.809	6.977	4.030	2.000	55.925
$\|S\|=40, k=24, l=20$	3.221	7.587	105.55	6.969	3.970	1.973	72.629
$\|S\|=50, k=30, l=25$	3.243	7.584	123.920	6.981	4.071	1.993	88.660
$\|S\|=60, k=36, l=30$	3.240	7.605	144.000	6.967	4.062	2.019	106.179

with a delegatee id', $id' \in S$, they can get the element σ and further compute g^t. However, they cannot reveal the element Q^t from elements (g, g^t, Q) because essentially it is solving a computational Diffie-Hellman assumption. Without the element Q^t that is used to randomize the delegator's private key sk_{id} in rk_1, nobody can reveal sk_{id}. Thus, our proposed scheme is collusion resistant.

Non-transferability. The non-transferability implies that the proxy colluding with a set of delegatees cannot re-delegate decryption rights [22]. That means with a re-encryption key $rk_{id \to id'}$, a delegatee's private key $sk_{id'}$ and a public key $sk_{id''}$, the proxy cannot produce a new re-encryption key $rk_{id \to id''}$ by colluding with a delegatee. We are not aware of any identity-based proxy re-encryption schemes that achieve this property. In our scheme, the proxy can collude with a delegatee from S to generate a new re-encryption key. However, as discussed in [22], the transferability is "mild-harmful", as the proxy colluding with a delegatee can always disclose the underlying plaintext and forward it to id''.

V. PERFORMANCE

A. Efficiency Theoretical Analysis

The comparison of computation cost between our scheme with [25], [8] and [31] is listed in Table II. In the table, $|S|$ denotes the total number of an identity set S and p denotes the computation cost of a bilinear pairing and e denotes an exponentiation in a group G or G_T. Let $Dec(Or)$ and $Dec(Re)$ denote the decryption operation of an original ciphertext and re-encrypted ciphertext. We omit the computation cost of hash functions as it is much less than the computation of a bilinear paring and exponentiation in group.

From Table II, we can see that our scheme is almost as efficient as the scheme [25] in $Extract$, Enc, $ReEnc$ and $Dec(Or)$ algorithms while less efficient in $RkeyGen$ and $Dec(Re)$ algorithms. However, this makes sense as our scheme supports the broadcast re-encryption functionality. In the $RkeyGen$ and $Dec(Re)$ algorithms, both of them should construct a re-encryption key/decrypt re-encrypted ciphertext for an identity set. When compared to [8], [31], our scheme is almost as efficient as [8], [31] in $RkeyGen$ and $Dec(Re)$ algorithms. Although our scheme does not significantly improve the efficiency compared to the previous IBPRE schemes [8], [31], we emphasize that only our scheme offers a unique *revocable functionality* feature. The revocable mechanism can highly reduce the cost of key maintenance because re-encryption key regeneration is not required when entities have been revoked.

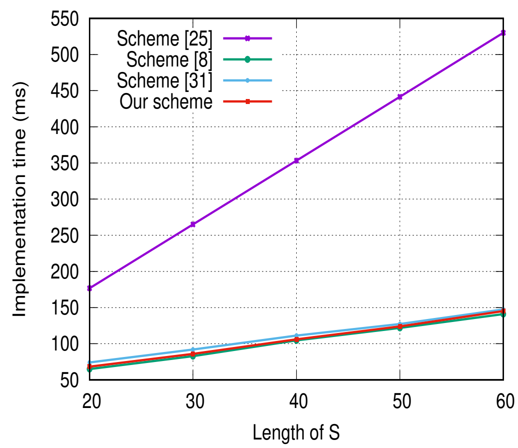


Fig. 2: $RKeyGen$ Execute Time Comparison

B. Implementation

For the experiments, the PBC package [34] written in Golang is used to implement our scheme. The PBC package not only provides a wrapper to a C language open source Pairing-Based Cryptography library (PBC) [35], but also offers structures for building pairing-based cryptosystems. Our Hardware is Intel(R) Core(TM) i5-8250U CPU @ 1.60GHZ 8GB RAM. The operation system is Linux Mint 18.1 Serena and programming language is GO 1.9.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2899300, IEEE Transactions on Dependable and Secure Computing

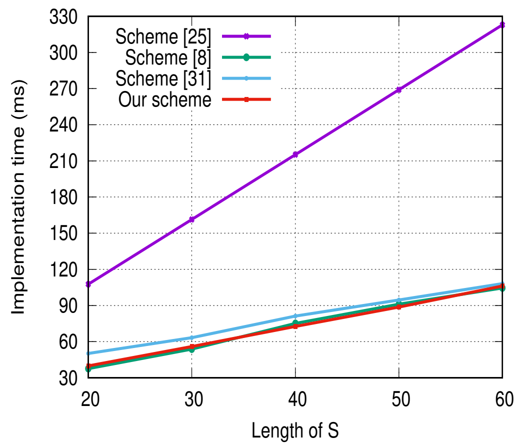


Fig. 3: $Dec(Re)$ Execute Time Comparison

We choose the elliptic curve $Y^2 = X^3 + X$ and the group order is 160 bit. We run each experiment for 20 times to obtain the average execution time.

1) Execute Time: In our experiment we set the maximum size of the set of delegatees in one encryption $N = 100$. We varied $|S|$ from 20 to 60 with step 10, and at the meanwhile varied k from 12 to 36 with step 6 and l from 10 to 30 with step 5. The execute time is summarized in Table III.

Table III plot the execution time of the algorithms run by the data user and the proxy. We observe that the execution time of $Extract$, Enc, $ReEnc$, $Revoke$ and $Dec(Or)$ algorithms are almost constant. While the execution time of $RKeyGen$ and $Dec(Re)$ algorithms are almost linear with the size of S. This coincide with the theoretical analysis in Table II.

2) Execute Time Comparison: In this experiment, we compare our scheme with [25] and [8] in $RKeyGen$ and $Dec(Re)$ algorithms as the execution time is linear with $|S|$. Further, we execute $RKeyGen$ and $Dec(Re)$ in [25] $|S|$ times to achieve the same broadcast effect. The execute time comparison is showed in Fig.1 for $RKeyGen$ algorithm and Fig.2 for $Dec(Re)$ algorithm.

Figure 2 and Figure 3 show that, our scheme is almost as efficient as [8] and [31] in $RkeyGen$ and $Dec(Re)$ algorithms. However, our proposed scheme achieves the *revocable functionality* that is not provided in [8] and [31]. When compared with [25], our scheme is much more efficient, especially when $|S|$ grows.

VI. CONCLUSION

In this paper, we defined revocable identity-based broadcast proxy re-encryption, proposed a concrete construction under the definition and proved our scheme is CPA secure in the random oracle model. More importantly, the property and performance comparison reveals that our proposed scheme is efficient and practical. Furthermore, our RIB-BPRE scheme can nicely support key revocation for a data sensitive system in a cloud environment, for example, a volunteer based genome research system. While this work has resolved the issue of key revocation for data sharing, it motivates some interesting open problems such designing RIB-BPRE scheme without random oracles and how to support more expressive on identities.

ACKNOWLEDGMENTS

Chunpeng Ge is supported by the National Natural Science Foundation of China (Grant No.61702236, 61672270) and Changzhou Sci&Tech Program, (Grant No.CJ20179027), Zhe Liu is supported by the National Natural Science Foundation of China (Grant No. 61802180), the Natural Science Foundation of Jiangsu Province (Grant No.SBK2018043466), the National Cryptography Development Fund (Grant No.MMJJ20180105) and the Fundamental Research Funds for the Central Universities (Grant No.NE2018106) and Liming Fang is supported by the National Natural Science Foundation of China (No.61872181).

We would like to thank the anonymous reviewers for very useful comments to an earlier version of this paper.

REFERENCES

[1] B. Dan and M. Franklin, "Identity-based encryption from the weil pairing," in *International Cryptology Conference*, 2001, pp. 213–229.
[2] C. Cocks, "An identity based encryption scheme based on quadratic residues," in *Cryptography and Coding, Ima International Conference, Cirencester, Uk, December*, 2015, pp. 360–363.
[3] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *International Conference on Theory and Applications of Cryptographic Techniques*, 2005, pp. 457–473.
[4] K. Liang and W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1981–1992, 2017.
[5] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *International Conference on the Theory and Applications of Cryptographic Techniques*, 1998, pp. 127–144.
[6] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *International Conference on Applied Cryptography and Network Security*, 2007, pp. 288–306.
[7] C. K. Chu, J. Weng, S. S. M. Chow, J. Zhou, and R. H. Deng, "Conditional proxy broadcast re-encryption," *Lecture Notes in Computer Science*, vol. 5594, pp. 327–342, 2009.
[8] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin, "Conditional identity-based broadcast proxy re-encryption and its application to cloud email," *IEEE Transactions on Computers*, vol. 65, no. 1, pp. 66–79, 2015.
[9] S. Berkovits, "How to broadcast a secret," in *International Conference on Theory and Application of Cryptographic Techniques*, 1991, pp. 535–541.
[10] A. Fiat and M. Naor, "Broadcast encryption," in *International Cryptology Conference*, 1993, pp. 480–491.
[11] J. Anzai, N. Matsuzaki, and T. Matsumoto, "A quick group key distribution scheme with efficient ntity revocation," *Proc Asiacrypt*, vol. 1716, pp. 333–347, 1999.
[12] D. Halevy and A. Shamir, "The lsd broadcast encryption scheme," in *International Cryptology Conference on Advances in Cryptology*, 2002, pp. 47–60.
[13] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," *Crypto*, vol. 2001, pp. 41–62, 2001.
[14] R. Sakai and J. Furukawa, "Identity-based broadcast encryption," *Journal of Electronics and Information Technology*, vol. 33, no. 4, pp. 1047–1050, 2007.
[15] C. Delerabl, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in *Advances in Crypotology International Conference on Theory and Application of Cryptology and Information Security*, 2007, pp. 200–215.
[16] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *ACM Conference on Computer and Communications Security*, 2008, pp. 417–426.
[17] W. Susilo, R. Chen, F. Guo, G. Yang, Y. Mu, and Y. W. Chow, "Recipient revocable identity-based broadcast encryption: How to revoke some recipients in ibbe without knowledge of the plaintext," in *ACM on Asia Conference on Computer and Communications Security*, 2016, pp. 201–210.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2899300, IEEE Transactions on Dependable and Secure Computing

[18] T. Ignatenko and M. Asim, "Attribute-based encryption," Feb. 20 2014, wO Patent 2,014,027,263.

[19] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *ACM Conference on Computer and Communications Security*, 2006, pp. 89–98.

[20] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy*, 2007, pp. 321–334.

[21] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," *Lecture Notes in Computer Science*, vol. 2008, pp. 321–334, 2011.

[22] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *Acm Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, 2006.

[23] D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy re-encryption," in *Practice and Theory in Public Key Cryptography, International Conference on Public Key Cryptography*, 2008, pp. 360–379.

[24] C. Ran and S. Hohenberger, "Chosen-ciphertext secure proxy re-encryption," in *ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, Usa, October*, 2007, pp. 185–194.

[25] C. K. Chu and W. G. Tzeng, "Identity-based proxy re-encryption without random oracles," in *International Conference on Information Security*, 2007, pp. 189–202.

[26] L. Wang, L. Wang, M. Mambo, and E. Okamoto, "New identity-based proxy re-encryption schemes to prevent collusion attacks," in *International Conference on Pairing-Based Cryptography*, 2010, pp. 327–346.

[27] C. Ge, W. Susilo, J. Wang, and L. Fang, "Identity-based conditional proxy re-encryption with fine grain policy," *Computer Standards and Interfaces*, vol. 52, pp. 1–9, 2017.

[28] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *International Symposium on Information, Computer, and Communications Security*, 2009, pp. 276–286.

[29] K. Liang, L. Fang, W. Susilo, and D. S. Wong, "A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security," in *International Conference on Intelligent NETWORKING and Collaborative Systems*, 2013, pp. 552–559.

[30] C. Ge, W. Susilo, J. Wang, Z. Huang, L. Fang, and Y. Ren, "A key-policy attribute-based proxy re-encryption without random oracles," *Computer Journal*, no. 7, 2016.

[31] M. Sun, C. Ge, L. Fang, and J. Wang, "A proxy broadcast re-encryption for cloud data sharing," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 10 455–10 469, 2018.

[32] B. Dan and X. Boyen, "Efficient selective-id secure identity-based encryption without random oracles," *Proc of Eurocrypt*, vol. 2004, no. 4, p. 172, 2004.

[33] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," *SIAM Journal on Computing*, vol. 33, no. 1, pp. 167–226, 2003.

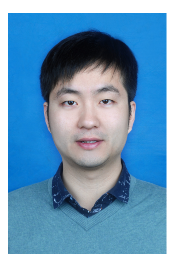
[34] Nik-U, "Pbc package," https://github.com/Nik-U/pbc, 2015.

[35] B. Lynn *et al.*, "Pbc library," *Online: http://crypto.stanford.edu/pbc*, 2006.

Chunpeng Ge received the Ph.D degree in Computer Science from Nanjing University of Aeronautics and Astronautics in 2016. He is now a research fellow of Singapore University of Technology and Design. His current research interests include cryptography, information security and privacy preserving for blockchain. His recent work has focused on the topics of public key encryption with keyword search, proxy re-encryption, identity-based encryption, and techniques for resistance to CCA attacks.

Zhe Liu received the BS and MS degrees in Shandong University in 2008 and 2011, respectively. He is a professor in College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics(NUAA), China. Before joining NUAA, he was a researcher in SnT, University of Luxembourg, Luxembourg. He received his Ph.D degree Laboratory of Algorithmics, Cryptology and Security (LACS), University of Luxembourg, Luxembourg in 2015. His Ph.D thesis has received the prestigious FNR Awards 2016 – Outstanding PhD Thesis Award for his contributions in cryptographic engineering on IoT devices. His research interests include computer arithmetic and information security. He has co-authored more than 70 research peer-reviewed journal and conference papers.

Jinyue Xia received the Ph.D degree in Computer Science from University of North Carolina at Charlotte, USA in 2017. His current research interests include data security, cryptography and information security. His recent work has focused on the topics of public key encryption with proxy re-encryption and identity-based encryption.



Liming Fang received the Ph.D degree in Computer Science from Nanjing University of Aeronautics and Astronautics in 2012, and has been a postdoctor in the information security from City University of Hong Kong. He is the associate professor at the School of Computer Science, Nanjing University of Aeronautics and Astronautics. Now, he is a visiting scholar of the Department of Electrical and Computer Engineering New Jersey Institute of Technology. His current research interests include cryptography and information security. His recent work has focused on the topics of public key encryption with keyword search, proxy re-encryption, identity-based encryption, and techniques for resistance to CCA attacks.