Fully Secure Unidirectional Identity-Based Proxy Re-encryption*

Song Luo[1,3,4], Qingni Shen[2,**], and Zhong Chen[2,3,4]

[1] College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China
[2] School of Software and Microelectronics & MoE Key Lab of Network and Software Assurance, Peking University, Beijing, China
[3] Institute of Software, School of Electronics Engineering and Computer Science, Peking University
[4] Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education
{luosong,shenqn,chen}@infosec.pku.edu.cn

Abstract. Proxy re-encryption (PRE) allows the proxy to translate a ciphertext encrypted under Alice's public key into another ciphertext that can be decrypted by Bob's secret key. Identity-based proxy re-encryption (IB-PRE) is the development of identity-based encryption and proxy re-encryption, where ciphertexts are transformed from one identity to another. In this paper, we propose two novel unidirectional identity-based proxy re-encryption schemes, which are both non-interactive and proved secure in the standard model. The first scheme is a single-hop IB-PRE scheme and has master secret security, allows the encryptor to decide whether the ciphertext can be re-encrypted. The second scheme is a multi-hop IB-PRE scheme which allows the ciphertext re-encrypted multiple times but without the size of ciphertext growing linearly as previous multi-hop IB-PRE schemes.

Keywords: Proxy Re-encryption, Identity-Based Encryption, Single-hop, Multi-hop.

1 Introduction

The primitive of proxy re-encryption (PRE) is first proposed by Blaze et al. [2] which involves three parties: Alice, Bob, and a proxy. PRE allows allows Alice to temporarily delegate the decryption rights to Bob via a proxy, i.e., the proxy with proper re-encryption key can translate a ciphertext encrypted under Alice's public key into another ciphertext that can be decrypted by Bob's secret key. Unlike the traditional proxy decryption scheme, PRE doesn't need users to store any additional decryption key, in other words, any decryption would be finished

* Supported by National Natural Science Foundation of China (No.60873238, 61073156, 60970135, 60821003, 61170263).
** Corresponding author.

H. Kim (Ed): ICISC 2011, LNCS 7259, pp. 109–126, 2012.

using only his own secret keys. PRE can be used in many scenarios, such as email forwarding, distributed file system, and the DRM of Apple's iTunes.

The concept of identity-based encryption (IBE) was first introduced by Shamir [16]. In an IBE system, arbitrary strings such as e-mail addresses or IP addresses can be used to form public keys for users. After Boneh and Franklin [5] proposed a practical identity-base encryption scheme, Green and Ateniese [11] proposed the first identity-based proxy re-encryption (IB-PRE). It allows the proxy to convert an encryption under Alice's identity into the encryption under Bob's identity. Due to the simplification of public-key infrastructure in identity-based framework, IB-PRE schemes are more desirable than non-identity-based ones.

According to the direction of transformation, IB-PRE schemes can be classified into two types, one is bidirectional, i.e., the proxy can transform from Alice to Bob and vice versa; the other is unidirectional, i.e., the proxy can only convert in one direction. Blaze et al. [2] also gave another method to classify IB-PRE schemes: single-hop, where the ciphertext can be transformed only once; and multi-hop, where the ciphertext can be transformed from Alice to Bob to Charlie and so on.

IB-PRE schemes are different from PRE schemes in which there exists a trusted private key generator (PKG) to generate all secret keys for identities. If Alice can compute re-encryption keys without the participation of Bob or PKG, the scheme is called non-interactive, or else called interactive. Obviously, it would be a hard work if all re-encryption keys are computed by the PKG. Therefore, it is more desirable to find non-interactive IB-PRE schemes. However, when generating secret keys, PKG insert the master key to users' secret keys. Obviously, re-encryption must involve some information of master key. But it is always hard to extract the part of master key from secret key to generate re-encryption, since elements of secret keys are always group elements and hard to get the discrete log based on a random generator.

Up to now, there are two ways to generate the re-encryption keys. One is proposed by Green and Ateniese [11]. In Green-Ateniese paradigm, to form a re-encryption key from Alice to Bob, a token is inserted in Alice's secret key and the token is encrypted to Bob, then these two parts form the re-encryption key. It is non-interactive in the generation of the re-encryption key and the re-encryption can be multi-hop where the ciphertext can be re-encrypted again and again. But the drawback of this method is that after one re-encryption, the encryption of the token would be attached to the ciphertext. So the ciphertext will grow linearly with the re-encryption times. The other is interactive proposed by Matsuo [15] in which the re-encryption key is generated by the private key generator or an extra re-encryption key generator which also owns the master key. This type of IB-PRE schemes are always single-hop where the re-encrypted ciphertext cannot be re-encrypted again.

1.1 Our Contribution

We present two novel unidirectional identity-based proxy re-encryption schemes. The first scheme is a single-hop scheme with master secret security. To make

this scheme be unidirectional, we present two kinds of ciphertexts, the original ciphertext is called the second level ciphertext which is nearly the same as Lewko-Waters IBE scheme's ciphertext, the transformed ciphertext is called the first level ciphertext and cannot be re-encrypted more. Our way of generating re-encryption keys are different from Green-Ateniese and Matsuo. To make the re-encryption key be generated by the user itself, we introduce non-group elements containing part information of master keys in user's secret keys and provide re-randomization to avoid collusion of proxy and users.

Based on our single-hop scheme, we present a multi-hop scheme in which the decryption cost and size of ciphertext do not grow linearly with the re-encryption times. To the best of our knowledge, this scheme is the first unidirectional IB-PRE scheme without growing linearly in the size of ciphertext as the re-encryption times increasing. Both schemes are non-interactive, which means the re-encryption key can be generated by Alice without the participation of Bob or the private key generator. We construct our schemes in composite order groups and use dual system encryption to prove the security of proposed schemes.

1.2 Related Works

Identity-Based Encryption. The first practical IBE scheme, proposed by Boneh and Franklin [5], was proven secure in the random oracle model. To remove random oracles, Canetti, Halevi, and Katz [7] suggested a weaker security notion for IBE, known as selective identity (selective-ID) security, relative to which they were able to build an inefficient but secure IBE scheme without using random oracles. Boneh and Boyen [3] proposed two new efficient selective-ID secure IBE schemes without random oracles. Later Boneh and Boyen [4], Waters [20] proposed new IBE schemes with full security. In Eurocrypt'06, Gentry [10] proposed an efficient identity based encryption with tight security reduction in the standard model but based on a stronger assumption.

By using dual system encryption, Waters [21] proposed the first fully secure IBE and HIBE schemes with short parameters under simple assumptions. But Waters's HIBE scheme does not have constant ciphertext size. Afterwards, another two fully secure HIBE schemes with constant size ciphertexts were proposed in composite order groups [8, 13].

Identity-Based Proxy Re-encryption. Ateniese et al. [1] presented the first unidirectional and single-use proxy re-encryption scheme. In 2007, Green and Ateniese [11] provided the first identity-based proxy re-encryption scheme but their scheme is secure in the random oracle model. Chu and Tzeng [9] proposed a new multi-hop unidirectional identity-based proxy re-encryption scheme in the standard model. However, their scheme is not chosen-ciphertext secure, Shao *et al.* [17] pointed out that its transformed ciphertext can be modified to another well-formed transformed ciphertext by anyone. Recently Lai et al. [12] gave new constructions on IB-PRE based on identity-based mediated encryption. Luo et al. [14] also gave a new generic IB-PRE construction based on an

existing IBE scheme. Wang et al. [18] proposed the first multi-use CCA-secure unidirectional IB-PRE scheme. All these schemes follow Green-Ateniese token paradigm, which makes the decryption cost and size of ciphertext grow linearly with the re-encryption times. In addition, Matsuo [15] proposed a new proxy re-encryption system for identity-based encryption, but his solution needs a re-encryption key generator (RKG) to generate re-encryption keys. Wang et al. [19] followed the route of Matsuo and proposed new secure IB-PRE schemes which let the PKG take part in generating the re-encryption keys.

1.3 Organization

The remaining paper is organized as follows. In Section 2, we review the definitions related to our proposals. In what follows, we present the single-hop scheme and its security analysis, and the multi-hop scheme and its security analysis, in Section 3 and Section 4, respectively. In Section 5 we discuss some extensions of the two schemes. Finally, we conclude the paper in Section 6.

2 Backgroud

2.1 Multi-hop Identity-Based Proxy Re-encryption

Definition 1. *A multi-hop unidirectional IB-PRE scheme consists of the following six algorithms:* **Setup**, **KeyGen**, **ReKeyGen**, **Enc**, **ReEnc**, *and* **Dec**.

Setup(1^λ)**.** *This algorithm takes the security parameter* λ *as input and generates a public key* PK*, a master secret key* MK.

KeyGen$(\mathrm{MK}, \mathcal{I})$**.** *This algorithm takes* MK *and an identity* $\mathcal{I}$ *as input and generates a secret key* $SK_{\mathcal{I}}$ *associated with* $\mathcal{I}$.

ReKeyGen$(SK_{\mathcal{I}}, \mathcal{I}')$**.** *This algorithm takes a secret key* $SK_{\mathcal{I}}$ *and an identity* $\mathcal{I}'$ *as input and generates a re-encryption key* $RK_{\mathcal{I}\to\mathcal{I}'}$.

Enc$(\mathrm{PK}, M, \mathcal{I})$**.** *This algorithm takes* PK*, a message* M*, and an identity* $\mathcal{I}$ *as input, and generates a ciphertext* $\mathrm{CT}_{\mathcal{I}}$.

ReEnc$(\mathrm{CT}_{\mathcal{I}}, RK_{\mathcal{I}\to\mathcal{I}'})$**.** *This algorithm takes a a ciphertext* $\mathrm{CT}_{\mathcal{I}}$ *encrypted to* $\mathcal{I}$ *and a re-encryption key* $RK_{\mathcal{I}\to\mathcal{I}'}$ *as input, generates a ciphertext* $\mathrm{CT}_{\mathcal{I}'}$ *encrypted to* $\mathcal{I}'$.

Dec$(\mathrm{CT}_{\mathcal{I}}, SK_{\mathcal{I}})$**.** *This algorithm takes a ciphertext* $\mathrm{CT}_{\mathcal{I}}$ *and* $SK_{\mathcal{I}}$ *associated with* $\mathcal{I}$ *as input and returns the message* M *or the error symbol* $\perp$ *if* $\mathrm{CT}_{\mathcal{I}}$ *is invalid.*

Correctness. A multi-hop unidirectional IB-PRE scheme should satisfy the following requirements:

1. $\mathbf{Dec}(\mathbf{Enc}(\mathrm{PK}, M, \mathcal{I}), SK_{\mathcal{I}}) = M$;
2. $\mathbf{Dec}(\mathbf{ReEnc}((\cdots\mathbf{ReEnc}(\mathbf{Enc}(\mathrm{PK}, M, \mathcal{I}), RK_{\mathcal{I}\to\mathcal{I}_1})\cdots), RK_{\mathcal{I}_{n-1}\to\mathcal{I}_n}), SK_{\mathcal{I}_n}) = M$;

We describe the game-based security definitions for multi-hop unidirectional IB-PRE systems as follows.

Definition 2. *The security of a multi-hop unidirectional IB-PRE scheme is defined according to the following* IND-PrID-ATK *game, where* ATK $\in \{\text{CPA}, \text{CCA}\}$.

Setup. *Run the* **Setup** *algorithm and give* PK *to the adversary* $\mathcal{A}$.

Phase 1. $\mathcal{A}$ *makes the following queries.*

- ***Extract***$(\mathcal{I})$: $\mathcal{A}$ *submits an identity* $\mathcal{I}$ *for a* **KeyGen** *query, return the corresponding secret key* $SK_{\mathcal{I}}$.
- ***RKExtract***$(\mathcal{I}, \mathcal{I}')$: $\mathcal{A}$ *submits an identity pair* $(\mathcal{I}, \mathcal{I}')$ *for a* **ReKeyGen** *query, return the re-encryption key* $RK_{\mathcal{I} \to \mathcal{I}'}$.

If ATK = CCA, $\mathcal{A}$ *can make the additional queries:*

- ***Reencrypt***$(\text{CT}_{\mathcal{I}}, \mathcal{I}, \mathcal{I}')$: $\mathcal{A}$ *submits a ciphertext* $\text{CT}_{\mathcal{I}}$ *encrypted for* $\mathcal{I}$ *and an identity* $\mathcal{I}'$ *for a* **ReEnc** *query, return the re-encrypted ciphertext* $\text{CT}_{\mathcal{I}'} = \textbf{ReEnc}(\text{CT}_{\mathcal{I}}, RK_{\mathcal{I} \to \mathcal{I}'})$ *where* $RK_{\mathcal{I} \to \mathcal{I}'} = \textbf{ReKeyGen}(SK_{\mathcal{I}}, \mathcal{I}')$ *and* $SK_{\mathcal{I}} = \textbf{KeyGen}(\text{MK}, \mathcal{I})$.
- ***Decrypt***$(\text{CT}_{\mathcal{I}}, \mathcal{I})$: $\mathcal{A}$ *submits a ciphertext* $\text{CT}_{\mathcal{I}}$ *encrypted for* $\mathcal{I}$ *for a* **Dec** *query, return the corresponding plaintext* $M = \textbf{Dec}(\text{CT}_{\mathcal{I}}, SK_{\mathcal{I}})$, *where* $SK_{\mathcal{I}} = \textbf{KeyGen}(\text{MK}, \mathcal{I})$.

Note that $\mathcal{A}$ *is not permitted to choose* $\mathcal{I}^*$ *which will be submitted in* **Challenge** *phase such that trivial decryption is possible using keys extracted during this phase (e.g., by using extracted re-encryption keys to translate from* $\mathcal{I}^*$ *to some identity for which* $\mathcal{A}$ *holds a decryption key).*

Challenge. $\mathcal{A}$ *submits a challenge identity* $\mathcal{I}^*$ *and two equal length messages* M_0, M_1 *to* $\mathcal{B}$. $\mathcal{B}$ *flips a random coin* b *and passes the ciphertext* $\text{CT}^* = \textbf{Enc}(PK, M_b, \mathcal{I}^*)$ *to* $\mathcal{A}$.

Phase 2. *Phase 1 is repeated with the following restrictions. Let* $\mathcal{C}$ *be a set of ciphertext/identity pairs, initially containing the single pair* $\langle \mathcal{I}^*, \text{CT}^* \rangle$. *For all* $\text{CT} \in \mathcal{C}$ *and for all RK given to* $\mathcal{A}$, *let* $\mathcal{C}'$ *be the set of all possible values derived via (one or more) consecutive calls to* ***Reencrypt****:*

- $\mathcal{A}$ *is not permitted to issue any query* ***Decrypt***$(\text{CT}, \mathcal{I})$ *where* $\langle \text{CT}, \mathcal{I} \rangle \in (\mathcal{C} \cap \mathcal{C}')$;
- $\mathcal{A}$ *is not permitted to issue any query* ***Extract***$(\mathcal{I})$ *or* ***RKExtract***$(\mathcal{I}, \mathcal{I}')$ *that would permit trivial decryption of any ciphertext in* $(\mathcal{C} \cap \mathcal{C}')$;
- $\mathcal{A}$ *is not permitted to issue any query* ***Reencrypt***$(\text{CT}, \mathcal{I}, \mathcal{I}')$ *where* $\mathcal{A}$ *possesses the keys to trivially decrypt ciphertexts under* $\mathcal{I}'$ *and* $\langle \text{CT}, \mathcal{I} \rangle \in (\mathcal{C} \cap \mathcal{C}')$. *On successful execution of any re-encrypt query, let* CT' *be the result and add the pair* $\langle \text{CT}', \mathcal{I}' \rangle$ *to the set* $\mathcal{C}$.

Guess. $\mathcal{A}$ *outputs its guess* b' *of* b.

The advantage of $\mathcal{A}$ *in this game is defined as* $Adv_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$ *where the probability is taken over the random bits used by the challenger and the adversary. We say that a multi-hop unidirectional IB-PRE scheme is* IND-PrID-ATK *secure, where* ATK $\in \{\text{CPA}, \text{CCA}\}$, *if no probabilistic polynomial time adversary* $\mathcal{A}$ *has a non-negligible advantage in winning the* IND-PrID-ATK *game.*

2.2 Single-hop Identity-Based Proxy Re-encryption

Single-hop IB-PRE can be viewed as a weaker concept than multi-hop IB-PRE, in which the ciphertext can be re-encrypted only once or not. According to the re-encryption time, its ciphertext is divided into two levels: second level ciphertext and first level ciphertext. A second ciphertext can be re-encrypted into a first level one (intended for a possibly different receiver) using the suitable re-encryption key and a first level ciphertext cannot be re-encrypted for another party. So the algorithms **Enc** and **Dec** are divided into two sub-algorithms $\mathbf{Enc}_2$ and $\mathbf{Enc}_1$, $\mathbf{Dec}_2$ and $\mathbf{Dec}_1$, respectively. The other algorithms are similar to multi-hop IB-PRE schemes. Furthermore, a single-hop unidirectional IB-PRE scheme should satisfy the following requirements:

1. $\mathbf{Dec}_2(\mathbf{Enc}_2(\mathrm{PK}, M, \mathcal{I}), SK_{\mathcal{I}}) = M$;
2. $\mathbf{Dec}_1(\mathbf{Enc}_1(\mathrm{PK}, M, \mathcal{I}), SK_{\mathcal{I}}) = M$;
3. $\mathbf{Dec}_1(\mathbf{ReEnc}(\mathbf{Enc}_2(\mathrm{PK}, M, \mathcal{I}), RK_{\mathcal{I}\to\mathcal{I}'}), SK_{\mathcal{I}'}) = M$.

The game-based security definitions for single-hop unidirectional IB-PRE systems are derived from previous multi-hop IB-PRE systems. Since single-hop unidirectional IB-PRE system has two level ciphertexts, there are two level securities called IND-2PrID-CPA(CCA) security and IND-1PrID-CPA(CCA) security.

Definition 3. *The security of a single-hop unidirectional IB-PRE scheme at the second level is defined according to the following* IND-2PrID-ATK *game, where* $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA}\}$.

Setup. *Run the* **Setup** *algorithm and give* PK *to the adversary* $\mathcal{A}$.

Phase 1. $\mathcal{A}$ *makes the following queries.*

- ***Extract***$(\mathcal{I})$: $\mathcal{A}$ *submits an identity* $\mathcal{I}$ *for a* **KeyGen** *query, return the corresponding secret key* $SK_{\mathcal{I}}$.
- ***RKExtract***$(\mathcal{I}, \mathcal{I}')$: $\mathcal{A}$ *submits an identity pair* $(\mathcal{I}, \mathcal{I}')$ *for a* **ReKeyGen** *query, return the re-encryption key* $RK_{\mathcal{I}\to\mathcal{I}'}$.

If $\mathrm{ATK} = \mathrm{CCA}$, $\mathcal{A}$ *can make the additional queries:*

- ***Reencrypt***$(\mathrm{CT}_{\mathcal{I}}, \mathcal{I}, \mathcal{I}')$: $\mathcal{A}$ *submits a second level ciphertext* $\mathrm{CT}_{\mathcal{I}}$ *encrypted for* $\mathcal{I}$ *and an identity* $\mathcal{I}'$ *for a* **ReEnc** *query, the challenger gives the adversary the re-encrypted ciphertext* $\mathrm{CT}_{\mathcal{I}'} = \mathbf{ReEnc}(\mathrm{CT}_{\mathcal{I}}, RK_{\mathcal{I}\to\mathcal{I}'})$ *where* $RK_{\mathcal{I}\to\mathcal{I}'} = \mathbf{ReKeyGen}(SK_{\mathcal{I}}, \mathcal{I}')$ *and* $SK_{\mathcal{I}} = \mathbf{KeyGen}(\mathrm{MK}, \mathcal{I})$.
- ***Decrypt***$(\mathrm{CT}_{\mathcal{I}}, \mathcal{I})$: $\mathcal{A}$ *submits a first level ciphertext* $\mathrm{CT}_{\mathcal{I}}$ *encrypted for* $\mathcal{I}$ *for a* $\mathbf{Dec}_1$ *query, return the corresponding plaintext* $M = \mathbf{Dec}_1(\mathrm{CT}_{\mathcal{I}}, SK_{\mathcal{I}})$, *where* $SK_{\mathcal{I}} = \mathbf{KeyGen}(\mathrm{MK}, \mathcal{I})$.

Challenge. $\mathcal{A}$ *submits a challenge identity* $\mathcal{I}^*$ *and two equal length messages* M_0, M_1 *to* $\mathcal{B}$. *If the queries*

- ***Extract***$(\mathcal{I}^*)$; *and*
- ***RKExtract***$(\mathcal{I}^*, \mathcal{I}')$ *and* ***Extract***$(\mathcal{I}')$ *for any identity* $\mathcal{I}'$

are never made, then flip a random coin b *and pass the ciphertext* $\mathrm{CT}^* = \mathbf{Enc}_2(PK, M_b, \mathcal{I}^*)$ *to* $\mathcal{A}$.

Phase 2. *Phase 1 is repeated with the restriction that* $\mathcal{A}$ *cannot make the following queries:*

- ***Extract***$(\mathcal{I}^*)$;
- ***RKExtract***$(\mathcal{I}^*, \mathcal{I}')$ *and* ***Extract***$(\mathcal{I}')$ *for any identity* $\mathcal{I}'$;
- ***Reencrypt***$(\mathrm{CT}^*, \mathcal{I}^*, \mathcal{I}')$ *and* ***Extract***$(\mathcal{I}')$ *for any identity* $\mathcal{I}'$;
- ***Decrypt***$(\mathrm{CT}_{\mathcal{I}'}, \mathcal{I}')$ *for any identity* $\mathcal{I}'$, *where* $\mathrm{CT}_{\mathcal{I}'} = \mathbf{ReEnc}(\mathrm{CT}^*, \mathcal{I}^*, \mathcal{I}')$.

Guess. $\mathcal{A}$ *outputs its guess* b' *of* b.

The advantage of $\mathcal{A}$ *in this game is defined as* $Adv_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$ *where the probability is taken over the random bits used by the challenger and the adversary. We say that a single-hop unidirectional IB-PRE scheme is* IND-2PrID-ATK *secure, where* $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA}\}$, *if no probabilistic polynomial time adversary* $\mathcal{A}$ *has a non-negligible advantage in winning the* IND-2PrID-ATK *game.*

Note that in the ***Decrypt*** query, we only provide the first level ciphertext decryption because any second level ciphertext can be re-encrypted to a first level ciphertext and then be queried for decryption.

Definition 4. *The security of a single-hop unidirectional IB-PRE scheme at the first level is defined according to the following* IND-1PrID-ATK *game, where* $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA}\}$.

Setup. *Run the* **Setup** *algorithm and give* PK *to the adversary* $\mathcal{A}$.

Phase 1. $\mathcal{A}$ *makes the following queries.*

- ***Extract***$(\mathcal{I})$: $\mathcal{A}$ *submits an identity* $\mathcal{I}$ *for a* **KeyGen** *query, return the corresponding secret key* $SK_{\mathcal{I}}$.
- ***RKExtract***$(\mathcal{I}, \mathcal{I}')$: $\mathcal{A}$ *submits an identity pair* $(\mathcal{I}, \mathcal{I}')$ *for a* **ReKeyGen** *query, return the re-encryption key* $RK_{\mathcal{I} \to \mathcal{I}'}$.

If $\mathrm{ATK} = \mathrm{CCA}$, $\mathcal{A}$ *can make the additional queries:*

- ***Decrypt***$(\mathrm{CT}_{\mathcal{I}}, \mathcal{I})$: $\mathcal{A}$ *submits a first level ciphertext* $\mathrm{CT}_{\mathcal{I}}$ *encrypted to* $\mathcal{I}$ *for a* $\mathbf{Dec}_1$ *query, return the corresponding plaintext* $M = \mathbf{Dec}_1(\mathrm{CT}_{\mathcal{I}}, SK_{\mathcal{I}})$, *where* $SK_{\mathcal{I}} = \mathbf{KeyGen}(\mathrm{MK}, \mathcal{I})$.

Challenge. $\mathcal{A}$ *submits a challenge identity* $\mathcal{I}^*$ *and two equal length messages* M_0, M_1 *to* $\mathcal{B}$. *If the query* ***Extract***$(\mathcal{I}^*)$ *is never made, then* $\mathcal{C}$ *flips a random coin* b *and passes the ciphertext* $\mathrm{CT}^* = \mathbf{Enc}_1(PK, M_b, \mathcal{I}^*)$ *to* $\mathcal{A}$.

Phase 2. *Phase 1 is repeated with the restriction that* $\mathcal{A}$ *cannot make the following queries:*

- ***Extract***$(\mathcal{I}^*)$;
- ***Decrypt***$(\mathrm{CT}^*, \mathcal{I}^*)$.

Guess. $\mathcal{A}$ *outputs its guess* b' *of* b.

The advantage of $\mathcal{A}$ *in this game is defined as* $Adv_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$ *where the probability is taken over the random bits used by the challenger and the adversary. We say that a single-hop unidirectional IB-PRE scheme is* IND-1PrID-ATK *secure, where* $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA}\}$, *if no probabilistic polynomial time adversary* $\mathcal{A}$ *has a non-negligible advantage in winning the* IND-1PrID-ATK *game.*

2.3 Master Secret Security

Master secret security is an important property for unidirectional PRE defined by Ateniese et al. [1]. Roughly speaking, even if the dishonest proxy colludes with the delegatee, it is still impossible for them to derive the delegator's secret key in full.

Definition 5. *The master secret security of a single-hop or multi-hop unidirectional IB-PRE scheme is defined according to the following master secret security game.*

Setup. *Run the* **Setup** *algorithm and give* PK *to the adversary* $\mathcal{A}$.
Phase 1. $\mathcal{A}$ *makes the following queries.*
- ***Extract***$(\mathcal{I})$: $\mathcal{A}$ *submits an identity* $\mathcal{I}$ *for a* **KeyGen** *query, return the corresponding secret key* $SK_{\mathcal{I}}$.
- ***RKExtract***$(\mathcal{I}, \mathcal{I}')$: $\mathcal{A}$ *submits an identity pair* $(\mathcal{I}, \mathcal{I}')$ *for a* **ReKeyGen** *query, return the re-encryption key* $RK_{\mathcal{I}\to\mathcal{I}'}$.

Challenge. $\mathcal{A}$ *submits a challenge identity* $\mathcal{I}^*$ *and query* ***Extract***$(\mathcal{I}^*)$ *is never made.*
Phase 2. *Phase 1 is repeated with the restriction that* $\mathcal{A}$ *cannot make query* ***Extract***$(\mathcal{I}^*)$.
Output. $\mathcal{A}$ *outputs the secret key* $SK_{\mathcal{I}^*}$ *for the challenge identity* $\mathcal{I}^*$.

The advantage of $\mathcal{A}$ *in this game is defined as* $Adv_{\mathcal{A}} = \Pr[\mathcal{A}\ succeeds]$. *A single-hop or multi-hop IB-PRE scheme has master secret security if no probabilistic polynomial time adversary* $\mathcal{A}$ *has a non-negligible advantage in winning the master secret security game.*

For single-hop unidirectional IB-PRE schemes, it is easy to see that the master secret security is implied by the first level plaintext security. We have the following result.

Lemma 1. *For a single-hop unidirectional IB-PRE scheme, the master secret security is implied by the first level plaintext security. That is, if there exists an adversary* $\mathcal{A}$ *who can break the master secret security of a single-hop unidirectional IB-PRE scheme* $\mathcal{E}$, *then there also exists an adversary* $\mathcal{B}$ *who can also break* $\mathcal{E}$*'s* IND-1PrID-CPA *security.*

Lemma 1 is obvious, so we omit its proof here.

2.4 Composite Order Bilinear Groups

Composite order bilinear groups were first introduced by Boneh, Goh and Nissim in [6].

Definition 6. *Let* $\mathcal{G}$ *be an algorithm called a* ***bilinear group generator*** *that takes as input a security parameter* λ *and outputs a tuple* $(N = p_1p_2p_3, G, G_T, e)$ *where* p_1, p_2 *and* p_3 *are three distinct primes,* G *and* G_T *are two multiplicative abelian groups of order* N, *and* $e : G \times G \to G_T$ *is an efficiently computable map (or "pairing") satisfying the following properties:*

– *(Bilinear)* $\forall g, h \in G, a, b \in \mathbb{Z}_N$, $e(g^a, h^b) = e(g, h)^{ab}$.
– *(Non-degenerate)* $\exists g \in G$ *such that* $e(g, g)$ *has order* N *in* G_T.

We assume that the group action in G *and* G_T *as well as the bilinear map* e *are all polynomial time computable in* λ. *Furthermore, we assume that the description of* G *and* G_T *includes a generator of* G *and* G_T *respectively.*

We say that G, G_T *are* bilinear groups *if the group operation in* G *and the bilinear map* $e : G \times G \to G_T$ *are both efficiently computable.*

We let G_{p_1}, G_{p_2} and G_{p_3} denote the subgroups of order p_1, p_2 and p_3 in G respectively. There is an important property called "orthogonality" between two different order subgroups under the bilinear map e, i.e., if $g \in G_{p_i}$ and $h \in G_{p_j}$ where $i \neq j$, then $e(g, h) = 1$. If g_1 generates G_{p_1}, g_2 generates G_{p_2} and g_3 generates G_{p_3}, then every element h of G can be expressed as $g_1^x g_2^y g_3^z$ for some values $x, y, z \in \mathbb{Z}_N$.

2.5 Complexity Assumptions

We use the notation $X \xleftarrow{\mathrm{R}} S$ to express that X is chosen uniformly randomly from the finite set S.

Assumption 1. Given a bilinear group generator $\mathcal{G}$, we define the following distribution:

$$\begin{gathered}\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, e) \xleftarrow{\mathrm{R}} \mathcal{G}(\lambda),\\ g \xleftarrow{\mathrm{R}} G_{p_1}, X_3 \xleftarrow{\mathrm{R}} G_{p_3},\\ D = (\mathbb{G}, g, X_3),\\ T_1 \xleftarrow{\mathrm{R}} G_{p_1}, T_2 \xleftarrow{\mathrm{R}} G_{p_1 p_2}.\end{gathered}$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 1 to be

$$Adv^{\mathrm{A1}}_{\mathcal{A},\mathcal{G}}(\lambda) := \left|\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]\right|.$$

Definition 7. *We say that* $\mathcal{G}$ *satisfies Assumption 1 if* $Adv^{\mathrm{A1}}_{\mathcal{A},\mathcal{G}}(\lambda)$ *is a negligible function of* λ *for any probabilistic polynomial-time algorithm* $\mathcal{A}$.

Assumption 2. Given a bilinear group generator $\mathcal{G}$, we define the following distribution:

$$\begin{gathered}\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, e) \xleftarrow{\mathrm{R}} \mathcal{G}(\lambda),\\ g, X_1 \xleftarrow{\mathrm{R}} G_{p_1}, X_2, Y_2 \xleftarrow{\mathrm{R}} G_{p_2}, X_3, Y_3 \xleftarrow{\mathrm{R}} G_{p_3},\\ D = (\mathbb{G}, g, X_1 X_2, X_3, Y_2 Y_3),\\ T_1 \xleftarrow{\mathrm{R}} G_{p_1 p_3}, T_2 \xleftarrow{\mathrm{R}} G.\end{gathered}$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 to be

$$Adv^{\mathrm{A2}}_{\mathcal{A},\mathcal{G}}(\lambda) := \left|\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]\right|.$$

Definition 8. *We say that $\mathcal{G}$ satisfies Assumption 2 if $Adv^{\mathrm{A2}}_{\mathcal{A},\mathcal{G}}(\lambda)$ is a negligible function of λ for any probabilistic polynomial-time algorithm $\mathcal{A}$.*

Assumption 3. Given a bilinear group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1p_2p_3, G, G_T, e) \xleftarrow{\mathrm{R}} \mathcal{G}(\lambda), \alpha, s \xleftarrow{\mathrm{R}} \mathbb{Z}_N,$$
$$g \xleftarrow{\mathrm{R}} G_{p_1}, X_2, Y_2, Z_2 \xleftarrow{\mathrm{R}} G_{p_2}, X_3 \xleftarrow{\mathrm{R}} G_{p_3}$$
$$D = (\mathbb{G}, g, g^{\alpha}X_2, X_3, g^{s}Y_2, Z_2),$$
$$T_1 = e(g,g)^{\alpha s}, T_2 \xleftarrow{\mathrm{R}} G_T.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 to be

$$Adv^{\mathrm{A3}}_{\mathcal{A},\mathcal{G}}(\lambda) := \left|\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]\right|.$$

Definition 9. *We say that $\mathcal{G}$ satisfies Assumption 3 if $Adv^{\mathrm{A3}}_{\mathcal{A},\mathcal{G}}(\lambda)$ is a negligible function of λ for any probabilistic polynomial-time algorithm $\mathcal{A}$.*

3 Single-hop IB-PRE Scheme

In this section, we present a single-hop IB-PRE scheme. Our construction is based on Lewko-Waters IBE scheme [13] with small modification. We use its ciphertext as the second level ciphertext and add an extra element to make the re-encryption feasible. The scheme is constructed as follows.

3.1 Construction

Setup(1^{λ}). Given the security parameter λ, this algorithm first gets a bilinear group G of order $N = p_1p_2p_3$ from $\mathcal{G}(\lambda)$ where p_1 and p_2 are distinct primes. Let G_{p_i} denote the subgroup of order p_i in G. It then chooses $a, b, c, d, \alpha, \beta, \gamma \in \mathbb{Z}_N$ and $g \in G_{p_1}$ randomly. Next it computes $u_1 = g^a$, $h_1 = g^b$, $u_2 = g^c$, $h_2 = g^d$, $w = g^{\beta}$, and $v = g^{\gamma}$. The public parameters are published as

$$\mathrm{PK} = \{N, g, u_1, h_1, u_2, h_2, w, v, e(g,g)^{\alpha}\}.$$

The master secret key MK is $\{\alpha, \beta, \gamma, a, b, c, d\}$ and a generator of G_{p_3}.

The identity space is $\mathbb{Z}_N$ and the message space is G_T.

KeyGen(MK, $\mathcal{I}$). Given an identity $\mathcal{I} \in \mathbb{Z}_N$, this algorithm chooses r, t, t', x, y, $z \in \mathbb{Z}_N$ and R_3, R'_3, $\hat{R}_3$, $\hat{R}'_3 \in G_{p_3}$ randomly, and computes $D_1 = g^{\alpha}(u_1^{\mathcal{I}}h_1)^r R_3$, $D_2 = g^r R'_3$, $E_1 = \frac{c+x}{a\mathcal{I}+b}$, $E_2 = g^{\beta x}$, $F_1 = \frac{d+y}{a\mathcal{I}+b}$, $F_2 = g^{\beta y}$, $Z_1 = \frac{z}{a\mathcal{I}+b}$, $Z_2 = g^{\beta z}$, $K_1 = \frac{t}{\beta(c\mathcal{I}+d)}$, $K_2 = g^{\alpha}g^{t+\gamma t'}\hat{R}_3$, $K_3 = g^{t'}\hat{R}'_3$. We require that the PKG always use the same random value t for $\mathcal{I}$. This can be accomplished by using a pseudo-random function (PRF) or an internal log to ensure consistency.

The secret key is $SK_{\mathcal{I}} = (D_1, D_2, E_1, E_2, F_1, F_2, Z_1, Z_2, K_1, K_2, K_3)$.

ReKeyGen$(SK_{\mathcal{I}}, \mathcal{I}')$**.** Given a secret key $SK_{\mathcal{I}} = (D_1, D_2, E_1, E_2, F_1, F_2, Z_1, Z_2, K_1, K_2)$ for $\mathcal{I}$ and an identity $\mathcal{I}' \neq \mathcal{I}$, this algorithm chooses $k_1, k_2 \in \mathbb{Z}_N$ randomly and computes $rk_1 = (E_1 + k_1 \cdot Z_1) \cdot \mathcal{I}' + (F_1 + k_2 \cdot Z_1)$, $rk_2 = (E_2 \cdot Z_2^{k_1})^{\mathcal{I}'} \cdot (F_2 \cdot Z_2^{k_2})$.

The re-encryption key is $RK_{\mathcal{I}\to\mathcal{I}'} = (rk_1, rk_2)$.

Enc$_2$**(**PK$, M, \mathcal{I})$**.** To encrypt a message $M \in G_T$ for an identity $\mathcal{I}$, this algorithm chooses $s \in \mathbb{Z}_N$ randomly and computes $C = M \cdot e(g,g)^{\alpha s}$, $C_1 = (u_1^{\mathcal{I}} h_1)^s$, $C_2 = g^s$, $C_3 = v^s$.

The second level ciphertext is $\mathrm{CT}_{\mathcal{I}} = (C, C_1, C_2, C_3)$.

Enc$_1$**(**PK$, M, \mathcal{I})$**.** To encrypt a message $M \in G_T$ for an identity $\mathcal{I}$, this algorithm chooses $s \in \mathbb{Z}_N$ randomly and computes $C = M \cdot e(g,g)^{\alpha s}, C_1' = e(u_2^{\mathcal{I}} h_2, w)^s$, $C_2 = g^s$, $C_3 = v^s$.

The first level ciphertext is $\mathrm{CT}_{\mathcal{I}} = (C, C_1', C_2, C_3)$.

ReEnc($\mathrm{CT}_{\mathcal{I}}, RK_{\mathcal{I}\to\mathcal{I}'}$**).** Given a second level ciphertext $\mathrm{CT}_{\mathcal{I}} = (C, C_1, C_2, C_3)$ and a re-encryption key $RK_{\mathcal{I}\to\mathcal{I}'} = (rk_1, rk_2)$, this algorithm computes $C_1' = e(C_1, w)^{rk_1} e(C_2, rk_2)^{-1}$.

The re-encrypted ciphertext is $\mathrm{CT}_{\mathcal{I}'} = (C, C_1', C_2, C_3)$.

Dec$_2$**(**$\mathrm{CT}_{\mathcal{I}}, SK_{\mathcal{I}}$**).** Let $\mathrm{CT}_{\mathcal{I}} = (C, C_1, C_2, C_3)$ be a second level ciphertext for identity $\mathcal{I}$, it can be decrypted as

$$M = C \cdot \frac{e(D_2, C_1)}{e(D_1, C_2)}.$$

Dec$_1$**(**$\mathrm{CT}_{\mathcal{I}}, SK_{\mathcal{I}}$**).** Let $\mathrm{CT}_{\mathcal{I}} = (C, C_1', C_2)$ be a first level ciphertext for identity $\mathcal{I}$, it can be decrypted as

$$M = C \cdot (C_1')^{K_1} \cdot \frac{e(K_3, C_3)}{e(K_2, C_2)}.$$

Correctness at Second Level

$$\frac{e(D_2, C_1)}{e(D_1, C_2)} = \frac{e(g^r R_3', (u_1^{\mathcal{I}} h_1)^s)}{e(g^\alpha (u_1^{\mathcal{I}} h_1)^r R_3, g^s)} = e(g,g)^{-\alpha s}.$$

Correctness at First Level

$$(C_1')^{K_1} \cdot \frac{e(K_3, C_3)}{e(K_2, C_2)} = e(u_2^{\mathcal{I}} h_2, w)^{s \cdot \frac{t}{\beta(c\mathcal{I}+d)}} \cdot \frac{e(g^{t'} \hat{R}_3', g^{\gamma s})}{e(g^\alpha g^{t+\gamma t'} \hat{R}_3, g^s)} = e(g,g)^{-\alpha s}.$$

3.2 Security

We have the following results for our proposed single-hop IB-PRE scheme.

Theorem 1. *If Assumptions 1, 2, 3 hold, then our single-hop IB-PRE scheme is* IND-2PrID-CPA *secure.*

Theorem 2. *If Assumptions 1, 2, 3 hold, then our single-hop IB-PRE scheme is* IND-1PrID-CPA *secure.*

It is easy to get the following result from Lemma 1 and Theorem 2.

Corollary 1. *Our single-hop IB-PRE scheme has master secret security.*

We use the dual system encryption technique to prove Theorem 1 and Theorem 2. First we define two additional structures: semi-functional keys and semi-functional ciphertexts. According to the encryption algorithms, there are two types of semi-functional ciphertext: second level semi-functional ciphertext and first level semi-functional ciphertext. These will not be used in the real system, but they will be used in our proof.

Second Level Semi-functional Ciphertext. Let g_2 denote a generator of the subgroup G_{p_2}. A second level semi-functional ciphertext is created as follows. The algorithm first runs the $\mathbf{Enc}_2$ algorithm to generate a normal second level ciphertext $\hat{C}, \hat{C}_1, \hat{C}_2, \hat{C}_3$, chooses $x, y \in \mathbb{Z}_N$ randomly and sets $C = \hat{C}, C_1 = \hat{C}_1 g_2^{xy}, C_2 = \hat{C}_2 g_2^{y}, C_3 = \hat{C}_3 g_2^{\gamma y}$.

First Level Semi-functional Ciphertext. A first level semi-functional ciphertext is created as follows. The algorithm first runs the $\mathbf{Enc}_1$ algorithm to generate a normal first level ciphertext $\hat{C}, \hat{C}'_1, \hat{C}_2, \hat{C}_3$, chooses $y \in \mathbb{Z}_N$ randomly and sets $C = \hat{C}, C'_1 = \hat{C}'_1, C_2 = \hat{C}_2 g_2^{y}, C_3 = \hat{C}_3 g_2^{\gamma y}$.

Semi-functional Key. A semi-functional key is created as follows. The algorithm first runs the **KeyGen** algorithm to generate a normal secret key $\hat{D}_1$, $\hat{D}_2$, $\hat{E}_1$, $\hat{E}_2$, $\hat{F}_1$, $\hat{F}_2$, $\hat{Z}_1$, $\hat{Z}_2$, $\hat{K}_1$, $\hat{K}_2$, $\hat{K}_3$, chooses $\eta, \delta, z_1, z_2 \in \mathbb{Z}_N$ randomly and sets $D_1 = \hat{D}_1 g_2^{\eta z_1}$, $D_2 = \hat{D}_2 g_2^{\eta}$, $E_1 = \hat{E}_1$, $E_2 = \hat{E}_2$, $F_1 = \hat{F}_1$, $F_2 = \hat{F}_2$, $Z_1 = \hat{Z}_1$, $Z_2 = \hat{Z}_2$, $K_1 = \hat{K}_1$, $K_2 = \hat{K}_2 g_2^{\delta z_2}$, $K_3 = \hat{K}_3 g_2^{\delta}$.

We will prove the security of our system from Assumptions 1, 2, 3 using a hybrid argument over a sequence of games. We let q denote the number of key queries made by the attacker. We define these games as follows:

$\mathbf{Game}_{2,Real}$: The IND-2PrID-CPA game defined previously in which the ciphertext and all the keys are normal.

$\mathbf{Game}_{2,Restricted}$: This is like the real IND-2PrID-CPA game except that the attacker cannot ask for keys for identities which are equal to the challenge identity modulo p_2.

$\mathbf{Game}_{2,i}$, $0 \leq i \leq q$: This is like $\mathbf{Game}_{2,Restricted}$ except that the challenge ciphertext is semi-functional and the first i private key is semi-functional. The rest of the keys are normal.

$\mathbf{Game}_{2,Final}$: This is like $\mathbf{Game}_{2,q}$ except that the ciphertext is a semi-functional encryption of a random message, independent of the two messages provided by the attacker.

$\mathbf{Game}_{1,Real}$: The IND-1PrID-CPA game defined previously in which the ciphertext and all the keys are normal.

$\mathbf{Game}_{1,Restricted}$: This is like the real IND-1PrID-CPA game except that the attacker cannot ask for keys for identities which are equal to the challenge identity modulo p_2.

$\mathbf{Game}_{1,i}$, $0 \leq i \leq q$: This is like $\mathbf{Game}_{1,Restricted}$ except that the challenge ciphertext is semi-functional and the first i private key is semi-functional. The rest of the keys are normal.

$\mathbf{Game}_{1,Final}$: This is like $\mathbf{Game}_{1,q}$ except that the ciphertext is a semi-functional encryption of a random message, independent of the two messages provided by the attacker.

$\mathbf{Game}_{2,Restricted}$ and $\mathbf{Game}_{1,Restricted}$ are introduced in our proofs due to the same reason explained in Lewko-Waters IBE scheme's proof [13]. We note that $\mathbf{Game}_{2,*}$ and $\mathbf{Game}_{1,*}$ are defined differently due to the two base games IND-2PrID-CPA game and IND-1PrID-CPA game are different. In $\mathbf{Game}_{*,0}$ the challenge ciphertext is semi-functional, but all keys are normal and in $\mathbf{Game}_{*,q}$ all private keys are semi-functional. We will prove $\mathbf{Game}_{2,*}$ type games and $\mathbf{Game}_{1,*}$ type games are indistinguishable respectively.

Lemma 2. *Suppose there exists a polynomial time algorithm $\mathcal{A}$ where $\mathbf{Game}_{2,Real}Adv_{\mathcal{A}} - \mathbf{Game}_{2,Restricted}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage $\geq \frac{\epsilon}{2}$ in breaking either Assumption 1 or Assumption 2.*

Proof. With probability ϵ, $\mathcal{A}$ produces identities $\mathcal{I}$ and $\mathcal{I}^*$ such that $\mathcal{I} \neq \mathcal{I}^*$ modulo N and p_2 divides $\mathcal{I} - \mathcal{I}^*$. Let $a = \gcd(\mathcal{I} - \mathcal{I}^*, N)$ and $b = \frac{N}{a}$. We have $p_2 \mid a$ and $a < N$. Note that $N = p_1 p_2 p_3$, so there are two cases:

1. $p_1 \mid b$ which means $a = p_2, b = p_1 p_3$ or $a = p_2 p_3, b = p_1$.
2. $p_1 \nmid b$ which means $a = p_1 p_2, b = p_3$.

At least one of these cases must occur with probability $\geq \frac{\epsilon}{2}$. In case 1, $\mathcal{B}$ will break Assumption 1. Given g, X_3, T, $\mathcal{B}$ can confirm that it is case 1 by checking whether $g^b = 1$. Then $\mathcal{B}$ can test whether $T^b = 1$. If yes, then $T \in G_{p_1}$. If not, then $T \in G_{p_1 p_2}$.

In case 2, $\mathcal{B}$ will break Assumption 2. Given $g, X_1 X_2, X_3, Y_2 Y_3$, $\mathcal{B}$ can confirm that it is case 2 by checking whether $g^a = 1$. Then $\mathcal{B}$ can test whether $e((Y_2 Y_3)^b, T) = 1$. If yes, then $T \in G_{p_1 p_3}$. If not, then $T \in G$. □

Lemma 3. *Suppose there exists a polynomial time algorithm $\mathcal{A}$ where $\mathbf{Game}_{2,Restricted}Adv_{\mathcal{A}} - \mathbf{Game}_{2,0}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ to Assumption 1.*

Proof. $\mathcal{B}$ receives g, X_3 and T to simulate $\mathbf{Game}_{2,Restricted}$ or $\mathbf{Game}_{2,0}$ with $\mathcal{A}$ depending on whether $T \in G_{p_1}$ or $T \in G_{p_1 p_2}$.

$\mathcal{B}$ sets the public parameters as follows. $\mathcal{B}$ chooses random exponents α, β, γ, a, b, c, d and computes $u_1 = g^a$, $h_1 = g^b$, $u_2 = g^c$, $h_2 = g^d$, $w = g^\beta$, and $v = g^\gamma$. It sends these public parameters N, g, u_1, h_1, u_2, h_2, w, v, $e(g,g)^\alpha$ to $\mathcal{A}$. And $\mathcal{B}$ uses X_3 as a generator of G_{p_3}. Note that $\mathcal{B}$ has the actual master secret key, it simply runs the key generation to generate the normal keys to $\mathcal{A}$ for any identity $\mathcal{I}$.

At the challenge phase, $\mathcal{A}$ submits two equal-length messages M_0, M_1 and the challenge identity $\mathcal{I}^*$ to $\mathcal{B}$. It then flips a coin μ and computes the challenge ciphertext as follows:

$$C = M_\mu e(g,T)^\alpha, C_1 = T^{a\mathcal{I}^* + b}, C_2 = T, C_3 = T^\gamma.$$

If $T \in G_{p_1}$, this is a normal ciphertext. If $T \in G_{p_1p_2}$, then it can be written as $g^{s_1}g_2^{s_2}$ and the ciphertext is a semi-functional ciphertext with randomness $s = s_1$, $x = a\mathcal{I}^* + b$, $y = s_2$.

We can thus conclude that, if $T \in G_{p_1}$, then $\mathcal{B}$ has properly simulated $\mathbf{Game}_{2,Restricted}$. If $T \in G_{p_1p_2}$, then $\mathcal{B}$ has properly simulated $\mathbf{Game}_{2,0}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

Lemma 4. *Suppose there exists a polynomial time algorithm $\mathcal{A}$ where $\mathbf{Game}_{2,k-1}Adv_{\mathcal{A}} - \mathbf{Game}_{2,k}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ to Assumption 2.*

Proof. $\mathcal{B}$ receives g, X_1X_2, X_3, Y_2Y_3, T to simulate $\mathbf{Game}_{2,k-1}$ or $\mathbf{Game}_{2,k}$ with $\mathcal{A}$ depending on whether $T \in G_{p_1p_3}$ or $T \in G$.

$\mathcal{B}$ sets the public parameters as follows. $\mathcal{B}$ chooses random exponents α, β, γ, a, b, c, d and computes $u_1 = g^a$, $h_1 = g^b$, $u_2 = g^c$, $h_2 = g^d$, $w = g^\beta$ and $v = g^\gamma$. And $\mathcal{B}$ uses X_3 as a generator of G_{p_3}. It sends these public parameters N, g, u_1, h_1, u_2, h_2, w, v, $e(g,g)^\alpha$ to $\mathcal{A}$.

When $\mathcal{A}$ requests the i-th key for $\mathcal{I}_i$ where $i < k$, $\mathcal{B}$ returns a semi-functional key as follows. It chooses r_i, $\hat{r}_i$, $\hat{r}'_i$, t_i, t'_i, $\hat{t}_i$, $\hat{t}'_i$, x_i, y_i, $z_i \in \mathbb{Z}_N$ randomly and computes $D_1 = g^\alpha(u_1^{\mathcal{I}_i}h_1)^{r_i}(Y_2Y_3)^{\hat{r}_i}$, $D_2 = g^{r_i}(Y_2Y_3)^{\hat{r}'_i}$, $E_1 = \frac{c+x_i}{a\mathcal{I}_i+b}$, $E_2 = g^{\beta x_i}$, $F_1 = \frac{d+y_i}{a\mathcal{I}_i+b}$, $F_2 = g^{\beta y_i}$, $Z_1 = \frac{z_i}{a\mathcal{I}_i+b}$, $Z_2 = g^{\beta z_i}$, $K_1 = \frac{t_i}{\beta(c\mathcal{I}+d)}$, $K_2 = g^\alpha g^{t_i}g^{\gamma t'_i}(Y_2Y_3)^{\hat{t}_i}$, $K_3 = g^{t'_i}(Y_2Y_3)^{\hat{t}'_i}$.

When $i = k$, to response the key query for identity $\mathcal{I}_k$, $\mathcal{B}$ chooses r_k, r'_k, t_k, t'_k, $\hat{t}_k$, x_k, y_k, $z_k \in \mathbb{Z}_N$ randomly and computes $D_1 = g^\alpha T^{r_k(a\mathcal{I}_k+b)}X_3^{r'_k}$, $D_2 = T^{r_k}$, $E_1 = \frac{c+x_k}{a\mathcal{I}_k+b}$, $E_2 = g^{\beta x_k}$, $F_1 = \frac{d+y_k}{a\mathcal{I}_k+b}$, $F_2 = g^{\beta y_k}$, $Z_1 = \frac{z_k}{a\mathcal{I}_k+b}$, $Z_2 = g^{\beta z_k}$, $K_1 = \frac{t_k}{\beta(c\mathcal{I}+d)}$, $K_2 = g^\alpha g^{t_k}T^{\gamma\hat{t}_k}X_3^{t'_k}$, $K_3 = T^{\hat{t}_k}$. If $T \in G_{p_1p_3}$, this is a normal key. If $T \in G$, then it is a semi-functional key.

For $i > k$, we note that $\mathcal{B}$ has the actual master secret key, so it only need to run the key generation algorithm to generate the normal keys to $\mathcal{A}$ for any identity $\mathcal{I}$.

At the challenge phase, $\mathcal{A}$ submits two equal-length messages M_0, M_1 and the challenge identity $\mathcal{I}^*$ to $\mathcal{B}$. It then flips a coin μ and computes the challenge semi-functional ciphertext as follows:

$$C = M_\mu e(g, X_1X_2)^\alpha, C_1 = (X_1X_2)^{a\mathcal{I}^*+b}, C_2 = X_1X_2, C_3 = (X_1X_2)^\gamma.$$

We can thus conclude that, if $T \in G_{p_1p_3}$, then $\mathcal{B}$ has properly simulated $\mathbf{Game}_{2,k-1}$. If $T \in G$, then $\mathcal{B}$ has properly simulated $\mathbf{Game}_{2,k}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

Lemma 5. *Suppose there exists a polynomial time algorithm $\mathcal{A}$ where $\mathbf{Game}_{2,q}Adv_{\mathcal{A}} - \mathbf{Game}_{2,Final}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ to Assumption 3.*

Proof. $\mathcal{B}$ receives $g, g^\alpha X_2, X_3, g^sY_2, Z_2, T$ to simulate $\mathbf{Game}_{2,q}$ or $\mathbf{Game}_{2,Final}$ with $\mathcal{A}$ depending on whether $T = e(g,g)^{\alpha s}$ or T is a random element of G_T.

$\mathcal{B}$ sets the public parameters as follows. $\mathcal{B}$ chooses random exponents β, γ, a, b, c, d and computes $u_1 = g^a$, $h_1 = g^b$, $u_2 = g^c$, $h_2 = g^d$, $w = g^\beta$ and $v = g^\gamma$. And $\mathcal{B}$ uses X_3 as a generator of G_{p_3}. It sends these public parameters N, g, u_1, h_1, u_2, h_2, w, v, $e(g, g^\alpha X_2) = e(g, g)^\alpha$ to $\mathcal{A}$. Note that α is unknown to $\mathcal{B}$.

When responding a key query from $\mathcal{A}$ for identity $\mathcal{I}_i$, $\mathcal{B}$ returns a semi-functional key as follows. It chooses r_i, t_i, $\hat{t}_i$, x_i, y_i, $\hat{x}_i$, w_i, w_i', $\hat{w}_i$, $\hat{w}_i'$, z_i, z_i', $\hat{z}_i$, $\hat{z}_i' \in \mathbb{Z}_N$ randomly and computes $D_1 = g^\alpha X_2 (u_1^{\mathcal{I}_i} h_1)^{r_i} Z_2^{w_i} X_3^{z_i}$, $D_2 = g^{r_i} Z_2^{w_i'} X_3^{z_i'}$, $E_1 = \frac{c+x_i}{a\mathcal{I}_i+b}$, $E_2 = g^{\beta x_i}$, $F_1 = \frac{d+y_i}{a\mathcal{I}_i+b}$, $F_2 = g^{\beta y_i}$, $Z_1 = \frac{\hat{x}_i}{a\mathcal{I}_i+b}$, $Z_2 = g^{\beta \hat{x}_i}$, $K_1 = \frac{t_i}{\beta(c\mathcal{I}+d)}$, $K_2 = g^\alpha X_2 g^{t_i} g^{\gamma \hat{t}_i} Z_2^{\hat{w}_i} X_3^{\hat{z}_i}$, $K_3 = g^{\hat{t}_i} Z_2^{\hat{w}_i'} X_3^{\hat{z}_i'}$.

At the challenge phase, $\mathcal{A}$ submits two equal-length messages M_0, M_1 and the challenge identity $\mathcal{I}^*$ to $\mathcal{B}$. It then flips a coin μ and computes the challenge semi-functional ciphertext as follows:

$$C = M_\mu T, C_1 = (g^s Y_2)^{a\mathcal{I}^* + b}, C_2 = g^s Y_2, C_3 = (g^s Y_2)^\gamma.$$

If $T = e(g, g)^{\alpha s}$, then this is a properly distributed semi-functional ciphertext with message M_μ. If T is a random element of G_T, then this is a semi-functional ciphertext with a random message. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

Proof of Theorem 1. If Assumptions 1, 2, 3 hold then we have proved by Lemma 2, 3, 4, 5 that the real security game is indistinguishable from $\mathbf{Game}_{2,Final}$, in which the value of μ is information-theoretically hidden from the attacker. So there is no attacker that can obtain non-negligible advantage in winning the IND-2PrID-CPA game. □

Proof of Theorem 2 is similar but uses the games $\mathbf{Game}_{1,*}$, so the concrete proof is omitted here and provided in the full version of our paper due to similarity and space limitation.

4 Multi-hop IB-PRE Scheme

Now we construct a multi-hop IB-PRE scheme based on our single-hop IB-PRE scheme proposed in previous section. We observe that if we set $a = c$ and $b = d$, then the first level ciphertext can be re-encrypted using the same re-encryption key and has the same form. This means from the first level ciphertext, we can get a new multi-hop IB-PRE scheme. The new scheme is constructed as follows.

4.1 Construction

Setup(1^λ)**.** Given the security parameter λ, this algorithm first gets a bilinear group G of order $N = p_1 p_2 p_3$ from $\mathcal{G}(\lambda)$ where p_1, p_2 and p_3 are distinct primes. Let G_{p_i} denote the subgroup of order p_i in G. It then chooses $a, b, \alpha, \beta \in \mathbb{Z}_N$ and $g \in G_{p_1}$ randomly. Next it computes $u = g^a$, $h = g^b$, $w = g^\beta$ and $v = g^\gamma$. The public parameters are published as

$$\mathrm{PK} = \{N, g, u, h, w, v, e(g, g)^\alpha\},$$

the master secret key is $\mathrm{MK} = \{\alpha, \beta, \gamma, a, b\}$ and a generator of G_{p_3}.
The identity space is $\mathbb{Z}_N$ and the message space is G_T.

KeyGen$(\mathrm{MK}, \mathcal{I})$**.** Given an identity $\mathcal{I} \in \mathbb{Z}_N$, this algorithm chooses $t, r, x, y, z \in \mathbb{Z}_N$ and $R_3, R_3' \in G_{p_3}$ randomly, and computes $D_1 = \frac{t}{\beta(a\mathcal{I}+b)}$, $D_2 = g^{\alpha} g^{t+\gamma r} R_3$, $D_3 = g^r R_3'$, $E_1 = \frac{a+x}{a\mathcal{I}+b}$, $E_2 = g^{\beta x}$, $F_1 = \frac{b+y}{a\mathcal{I}+b}$, $F_2 = g^{\beta y}$, $Z_1 = \frac{z}{a\mathcal{I}+b}$, $Z_2 = g^{\beta z}$. We also require that the PKG always use the same random value t for $\mathcal{I}$.
The secret key is $SK_{\mathcal{I}} = (D_1, D_2, D_3, E_1, E_2, F_1, F_2, Z_1, Z_2)$.

ReKeyGen$(SK_{\mathcal{I}}, \mathcal{I}')$**.** Given a secret key $SK_{\mathcal{I}} = (D_1, D_2, E_1, E_2, F_1, F_2, Z_1, Z_2)$ for $\mathcal{I}$ and an identity $\mathcal{I}' \neq \mathcal{I}$, this algorithm chooses $k_1, k_2 \in \mathbb{Z}_N$ randomly and computes $rk_1 = (E_1 + k_1 \cdot Z_1) \cdot \mathcal{I}' + (F_1 + k_2 \cdot Z_1)$, $rk_2 = (E_2 \cdot Z_2^{k_1})^{\mathcal{I}'} \cdot (F_2 \cdot Z_2^{k_2})$.
The re-encryption key is $RK_{\mathcal{I} \to \mathcal{I}'} = (rk_1, rk_2)$.

Enc$(\mathrm{PK}, M, \mathcal{I})$**.** To encrypt a message $M \in G_T$ for an identity $\mathcal{I}$, this algorithm $s \in \mathbb{Z}_N$ randomly and computes $C = M \cdot e(g,g)^{\alpha s}$, $C_1 = e(u^{\mathcal{I}} h, w)^s$, $C_2 = g^s$, $C_3 = v^s$.
The ciphertext is $\mathrm{CT}_{\mathcal{I}} = (C, C_1, C_2, C_3)$.

ReEnc$(\mathrm{CT}_{\mathcal{I}}, RK_{\mathcal{I} \to \mathcal{I}'})$**.** Given a second level ciphertext $\mathrm{CT}_{\mathcal{I}} = (C, C_1, C_2, C_3)$ and a re-encryption key $RK_{\mathcal{I} \to \mathcal{I}'} = (rk_1, rk_2)$, this algorithm computes $C_1' = (C_1)^{rk_1} \cdot e(C_2, rk_2)^{-1}$.
The re-encrypted ciphertext is $\mathrm{CT}_{\mathcal{I}'} = (C, C_1', C_2, C_3)$.

Dec$(\mathrm{CT}_{\mathcal{I}}, SK_{\mathcal{I}})$**.** Let $\mathrm{CT}_{\mathcal{I}} = (C, C_1, C_2, C_3)$ be a ciphertext for identity $\mathcal{I}$, it can be decrypted as

$$M = C \cdot (C_1)^{D_1} \cdot \frac{e(D_3, C_3)}{e(D_2, C_2)}.$$

The correctness of decryption process is easily observable.

4.2 Security

We have the following result for our proposed multi-hop IB-PRE scheme.

Theorem 3. *If Assumptions 1, 2, 3 hold, then our multi-hop IB-PRE scheme is* IND-PrID-CPA *secure.*

Proof of Theorem 3 is similar to proofs of Theorem 1 and Theorem 2, so we give the concrete proof in the full version due to similarity and space limitation.

5 Discussion

5.1 Re-encryption Control

In the single-hop proxy re-encryption scheme, we can see that the element $C_3 = v^s$ is of no use in the Dec_2 algorithm and it is only used in the Dec_1 algorithm. If the encryptor doesn't provide v^s in the second level ciphertext, the second level decryption is not affected but the decryption of re-encrypted ciphertext cannot go on. So the encryptor can decide whether the second level ciphertext can be re-encrypted (in fact he can decide whether the re-encrypted ciphertext can be decrypted).

5.2 Transitivity and Transferability

Transitivity means the proxy can redelegate decryption rights. For example, from $RK_{\mathcal{I}_1 \to \mathcal{I}_2}$ and $RK_{\mathcal{I}_2 \to \mathcal{I}_3}$, he can produce $RK_{\mathcal{I}_1 \to \mathcal{I}_3}$. Transferability means the proxy and a set of delegatees can redelegate decryption rights. For example, from $RK_{\mathcal{I}_1 \to \mathcal{I}_2}$ and $SK_{\mathcal{I}_2}$, they can produce $RK_{\mathcal{I}_1 \to \mathcal{I}_3}$. Note that the user $\mathcal{I}_2$ can produce the re-encryption key $RK_{\mathcal{I}_2 \to \mathcal{I}_3}$, so transferability is implied by transitivity. Our multi-hop scheme has such transitivity that the proxy can produce $RK_{\mathcal{I}_1 \to \mathcal{I}_3}$ by $RK_{\mathcal{I}_1 \to \mathcal{I}_2}$ and $RK_{\mathcal{I}_2 \to \mathcal{I}_3}$ as follows:

Let $RK_{\mathcal{I}_1 \to \mathcal{I}_2} = (rk_1, rk_2)$ and $RK_{\mathcal{I}_2 \to \mathcal{I}_3} = (rk'_1, rk'_2)$. It computes $rk''_1 = rk_1 \cdot rk'_1$ and $rk''_2 = (rk_2)^{rk'_1} \cdot rk'_2$. Then $RK_{\mathcal{I}_2 \to \mathcal{I}_3}$ is (rk''_1, rk''_2).

6 Conclusion

In this paper, we propose two novel unidirectional identity-based proxy re-encryption schemes, which are both non-interactive and proved secure in the standard model. The first scheme is a single-hop IB-PRE scheme and has master secret security, allows the encryptor to decide whether the ciphertext can be re-encrypted. The second scheme is a multi-hop IB-PRE scheme which allows the ciphertext re-encrypted many times but without the cost of ciphertext size growing linearly as previous multi-hop IB-PRE schemes.

References

1. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. In: Proceedings of the Network and Distributed System Security Symposium, NDSS 2005. The Internet Society (2005)
2. Blaze, M., Bleumer, G., Strauss, M.J.: Divertible Protocols and Atomic Proxy Cryptography. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 127–144. Springer, Heidelberg (1998)
3. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
4. Boneh, D., Boyen, X.: Secure Identity Based Encryption Without Random Oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
5. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
6. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF Formulas on Ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
7. Canetti, R., Halevi, S., Katz, J.: A Forward-secure Public-key Encryption Scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)

8. Caro, A.D., Iovino, V., Persiano, G.: Fully secure anonymous hibe and secret-key anonymous ibe with short ciphertexts. Cryptology ePrint Archive, Report 2010/197 (2010), http://eprint.iacr.org/
9. Chu, C.-K., Tzeng, W.-G.: Identity-Based Proxy Re-encryption Without Random Oracles. In: Garay, J.A., Lenstra, A.K., Mambo, M., Peralta, R. (eds.) ISC 2007. LNCS, vol. 4779, pp. 189–202. Springer, Heidelberg (2007)
10. Gentry, C.: Practical Identity-Based Encryption Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
11. Green, M., Ateniese, G.: Identity-Based Proxy Re-encryption. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 288–306. Springer, Heidelberg (2007)
12. Lai, J., Zhu, W., Deng, R., Liu, S., Kou, W.: New constructions for identity-based unidirectional proxy re-encryption. Journal of Computer Science and Technology, 793–806 (2010)
13. Lewko, A., Waters, B.: New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
14. Luo, S., Hu, J., Chen, Z.: New construction of identity-based proxy re-encryption. In: Proceedings of the Tenth Annual ACM Workshop on Digital Rights Management, DRM 2010, pp. 47–50. ACM, New York (2010)
15. Matsuo, T.: Proxy Re-encryption Systems for Identity-Based Encryption. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 247–267. Springer, Heidelberg (2007)
16. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
17. Shao, J., Cao, Z.: CCA-Secure Proxy Re-encryption without Pairings. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 357–376. Springer, Heidelberg (2009)
18. Wang, H., Cao, Z., Wang, L.: Multi-use and unidirectional identity-based proxy re-encryption schemes. Information Sciences (2010)
19. Wang, L., Wang, L., Mambo, M., Okamoto, E.: New Identity-Based Proxy Re-encryption Schemes to Prevent Collusion Attacks. In: Joye, M., Miyaji, A., Otsuka, A. (eds.) Pairing 2010. LNCS, vol. 6487, pp. 327–346. Springer, Heidelberg (2010)
20. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
21. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)