New Construction of Identity-based Proxy Re-encryption

Song Luo, Jianbin Hu and Zhong Chen

Peking University
Email: {luosong, hjbin, chen}@infosec.pku.edu.cn

Abstract. A proxy re-encryption (PRE) scheme involves three parties: Alice, Bob, and a proxy. PRE allows the proxy to translate a ciphertext encrypted under Alice's public key into one that can be decrypted by Bob's secret key. We present a general method to construct an identity-based proxy re-encryption scheme from an existing identity-based encryption scheme. The transformed scheme satisfies the properties of PRE, such as unidirectionality, non-interactivity and multi-use. Moreover, the proposed scheme has master key security, allows the encryptor to decide whether the ciphertext can be re-encrypted.

Keywords: Proxy Re-encryption, Identity-Based Encryption, Transformation

1 Introduction

A proxy re-encryption (PRE) scheme involves three parties: Alice, Bob, and a proxy. PRE allows the proxy to translate a ciphertext encrypted under Alice's public key into one that can be decrypted by Bob's secret key. PRE can be used in many scenarios, such as email forwarding, distributed file system, and the DRM of Apple's iTunes. 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 using only his own secret keys.

The concept of identity-based encryption (IBE) was first introduced by Shamir [9]. 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 [4] proposed a practical identity-base encryption scheme, Green and Ateniese [6] proposed the first identity-based PRE (IB-PRE). It allows the proxy to convert an encryption under Alice's identity into the encryption under Bob's identity.

Our Contribution. We present a general method to construct an identity-based proxy re-encryption scheme from an existing identity-based encryption scheme. The transformed scheme satisfies the following properties of PRE, which are mentioned in [1, 6]:

- *Unidirectionality.* Alice can delegate decryption rights to Bob without permitting her to decrypt Bob's ciphertext.
- *Non-Interactivity.* Alice can compute re-encryption keys without the participation of Bob or the private key generator (PKG).
- *Multi-Use.* The proxy can re-encrypt a ciphertext multiple times, e.g. re-encrypt from Alice to Bob, and then re-encrypt the result from Bob to Carol.

Moreover, our scheme has the other two properties:

- *Master Key Security.* A valid proxy designated by Alice, other users who are able to decrypt Alice's ciphertext with the help from the proxy can not collude to obtain Alice's secret key.
- *Re-encryption Control.* The encryptor can decide whether the ciphertext can be re-encrypted.

Related Work. Mambo and Okamoto [7] first introduced the notion of PRE. Blaze et al. [2] later proposed the first concrete scheme of proxy re-encryption which allows the keyholder to publish the proxy function and have it applied by untrusted parties without further involvement by the original keyholder. Their scheme is bidirectional and has multi-use property. Ateniese et al. [1] presented the first unidirectional and single-use proxy re-encryption scheme. In 2007, Green and Ateniese [6] provided the first identity-based proxy re-encryption scheme but their scheme is secure in the random oracle model. Chu et al. [5] proposed new identity-based proxy re-encryption scheme in the standard model. Matsuo [8] also proposed new proxy re-encryption system for identity-based encryption, but his solution needs a re-encryption key generator (RKG) to generate re-encryption keys.

2 Preliminaries

In this section, we first review the basic concept of the bilinear maps, then we describe the concepts of IB-PRE and its security model.

Definition 1. *Let $\mathbb{G}$, $\mathbb{G}_1$ be two cyclic multiplicative groups with prime order p. Let g be be a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ be a bilinear map with the following properties:*

1. Bilinearity: $\forall u, v \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{ab}$.

2. Non-degeneracy: The map does not send all pairs in $\mathbb{G} \times \mathbb{G}$ to the identity in $\mathbb{G}_1$. Observe that since $\mathbb{G}, \mathbb{G}_1$ are groups of prime order this implies that if g is a generator of $\mathbb{G}$ then $e(g, g)$ is a generator of $\mathbb{G}_1$.

We say that $\mathbb{G}$ is a bilinear group if the group operation in $\mathbb{G}$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ are both efficiently computable.

We assume that there is an efficient algorithm *Gen* for generating bilinear groups. The algorithm *Gen*, on input a security parameter κ, outputs a tuple $G = [p, \mathbb{G}, \mathbb{G}_1, g \in \mathbb{G}, e]$ where $\log(p) = \Theta(\kappa)$.

Definition 2. *An IB-PRE scheme consists of the following six algorithms: **Setup**, **KeyGen**, **Encrypt**, **RKGen**, **Reencrypt**, and **Decrypt**.*

Setup(1^κ)**.** *This algorithm takes the security parameter κ 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 $\mathcal{I}$.*

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

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

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

Decrypt$(\mathrm{CT}_\mathcal{I}, SK_\mathcal{I})$**.** *This algorithm takes $\mathrm{CT}_\mathcal{I}$ and $SK_\mathcal{I}$ associated with $\mathcal{I}$ as input and returns the message M.*

Definition 3. *The security of an IB-PRE scheme is defined according to the following* IND-PrID-ATK *game, where* $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA}\}$.

Setup. *The challenger runs the* **Setup** *algorithm and gives* 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, the challenger gives the adversary the secret key $SK_\mathcal{I}$.*
- ***RKExtract**($\mathcal{I}$,$\mathcal{I}'$): $\mathcal{A}$ submits an identity pair $(\mathcal{I}, \mathcal{I}')$ for a* **RKGen** *query, the challenger gives the adversary the re-encryption key $RK_{\mathcal{I}\rightarrow\mathcal{I}'}$.*

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

- ***Reencrypt**($\mathrm{CT}_\mathcal{I}$,$\mathcal{I}$,$\mathcal{I}'$): $\mathcal{A}$ submits a ciphertext $\mathrm{CT}_\mathcal{I}$ encrypted for $\mathcal{I}$ and an identity $\mathcal{I}'$ for a* **Reencrypt** *query, the challenger gives the adversary the re-encrypted ciphertext $\mathrm{CT}_{\mathcal{I}'} =$* **Reencrypt**$(\mathrm{CT}_\mathcal{I}, RK_{\mathcal{I}\rightarrow\mathcal{I}'})$ *where $RK_{\mathcal{I}\rightarrow\mathcal{I}'} =$* **RKGen**$(SK_\mathcal{I}, \mathcal{I}')$ *and $SK_\mathcal{I} =$* **KeyGen**$(\mathrm{MK}, \mathcal{I})$.
- ***Decrypt**($\mathrm{CT}_\mathcal{I}$,$\mathcal{I}$): $\mathcal{A}$ submits a ciphertext $\mathrm{CT}_\mathcal{I}$ encrypted for $\mathcal{I}$ for a* **Decrypt** *query, the challenger gives the corresponding plaintext $M =$* **Decrypt**$(\mathrm{CT}_\mathcal{I}, SK_\mathcal{I})$*, where $SK_\mathcal{I} =$* **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 $\mathcal{C}$ flips a random coin b and passes the ciphertext $\mathrm{CT}^* =$ **Encrypt**$(PK, M_b, \mathcal{I}^*)$ *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* $\mathcal{I}'$*;*
- ***RKExtract***$(\mathcal{I}^*, \mathcal{I}')$ *and* ***Decrypt***$(\mathrm{CT}_{\mathcal{I}'}, \mathcal{I}')$ *for any identity* $\mathcal{I}'$ *and any ciphertext* $\mathrm{CT}_{\mathcal{I}'}$*;*
- ***Reencrypt***$(\mathrm{CT}^*, \mathcal{I}^*, \mathcal{I}')$ *and* ***Extract***$(\mathcal{I}')$ *for any identity* $\mathcal{I}'$*;*
- ***Decrypt***$(\mathrm{CT}^*, \mathcal{I}^*)$*;*
- ***Decrypt***$(\mathrm{CT}_{\mathcal{I}'}, \mathcal{I}')$ *for any identity* $\mathcal{I}'$*, where* $\mathrm{CT}_{\mathcal{I}'} = \mathbf{Reencrypt}(\mathrm{CT}^*, \mathcal{I}^*, \mathcal{I}')$.

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 an IB-PRE scheme is* IND-PrID-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-PrID-ATK *game.*

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

Definition 4. *The master key security of an IB-PRE scheme is defined according to the following master key security game.*

Setup. *The challenger runs the* **Setup** *algorithm and gives* 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, the challenger gives the adversary the secret key $SK_{\mathcal{I}}$.*
- ***RKExtract***$(\mathcal{I}, \mathcal{I}')$*: $\mathcal{A}$ submits an identity pair $(\mathcal{I}, \mathcal{I}')$ for a* **RKGen** *query, the challenger gives the adversary the re-encryption key $SK_{\mathcal{I} \to \mathcal{I}'}$.*

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 IB-PRE scheme has master key security if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in winning the master key security game.*

3 Our Construction

Requirements. Let $\mathcal{E}$ be an IBE scheme and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is the bilinear map used in $\mathcal{E}$. The requirements for $\mathcal{E}$ are: 1) $\mathcal{E}$ doesn't use random oracles; 2) random elements of $\mathbb{G}$ are obtained by taking a generator of $\mathbb{G}_1$ and raising it to random exponents. We suppose a message $M \in \mathbb{G}_1$ is randomized as $M \cdot Y$ in the encryption process, here $Y \in \mathbb{G}_1$ is computed from public key by the encryptor. And we suppose the ciphertext length of $\mathcal{E}$ is $k + 1$. Then its four algorithms are as follows:

Setup. It outputs public key PK and master secret key MK.
KeyGen$(\mathrm{MK}, \mathcal{I})$**.** For an identity $\mathcal{I}$, it outputs the corresponding secret key $SK_{\mathcal{I}} = (d_1, d_2, \cdots, d_k)$.
Encrypt$(\mathrm{PK}, M, \mathcal{I})$**.** For a message M and an identity $\mathcal{I}$, it outputs ciphertext $\mathrm{CT} = (C, C_1, C_2, \cdots, C_k)$.
Decrypt$(\mathrm{CT}, SK_{\mathcal{I}})$**.** $M = C \cdot e(d_1, C_1) \cdot e(d_2, C_2) \cdots e(d_k, C_k)$.

Let A a non-empty set, $t \in \mathbb{Z}_p^*$, we define $A^t := \{x^t | x \in A\}$. Let g be a generator of $\mathbb{G}$. We transform the above scheme to an proxy re-encryption scheme as follows:

Setup. It chooses a random integer $t \in \mathbb{Z}_p^*$, computes $h = g^t$, outputs public key $\mathrm{PK} \cup \mathrm{PK}^t \cup \{h\}$ and master secret key $\mathrm{MK} \cup \{t\}$. Let $E : \mathbb{G} \to \mathbb{G}_1$ be an encoding between $\mathbb{G}$ and $\mathbb{G}_1$.
KeyGen$(\mathrm{MK}, \mathcal{I})$**.** For an identity $\mathcal{I}$, it outputs the corresponding secret key $SK_{\mathcal{I}} = (d_1, d_2, \cdots, d_k)$.
RKGen$(SK_{\mathcal{I}}, \mathcal{I}')$**.** Let the secret key for $\mathcal{I}$ be $SK_{\mathcal{I}} = (d_1, d_2, \cdots, d_k)$. There must be some component containing information of identity, for simplicity, we suppose it is d_k. To generate a re-encryption key for $\mathcal{I} \to \mathcal{I}'$, chooses $d \in_R \mathbb{Z}_p$ and computes $\mathbb{C}$ which is the ciphertext of $E(g^d)$ for an identity $\mathcal{I}'$, i.e., $\mathbb{C} = \mathbf{Encrypt}(\mathrm{PK}, E(g^d), \mathcal{I}')$. Then the re-encryption key is $RK_{\mathcal{I} \to \mathcal{I}'} = (d_1, d_2, \cdots, d_k h^d, \mathbb{C}) = (d'_1, d'_2, \cdots, d'_k, \mathbb{C})$.

Encrypt(PK, $M, \mathcal{I}$). For a message M and an identity $\mathcal{I}$, it outputs ciphertext $\mathrm{CT}_{\mathcal{I}} = (C, C_1, C_2, \cdots, C_k, C_k^t)$. Here C_k^t is computed like C_k, but using the corresponding parameters from PK^t. Note that under the requirements for $\mathbb{G}$ and recall that $\mathbb{G}$ is a cyclic multiplicative group, we can suppose $C_k = g_1^{s_1} \cdots g_n^{s_n}$, here $g_1, \cdots, g_n$ are from PK and $s_1, \cdots, s_n$ are computed by the encryptor. So $C_k^t = (g_1^{s_1} \cdots g_n^{s_n})^t = (g_1^t)^{s_1} \cdots (g_n^t)^{s_n}$ which can be generated from PK^t and $\mathcal{I}$.

Reencrypt($\mathrm{CT}_{\mathcal{I}}, RK_{\mathcal{I} \to \mathcal{I}'}$). Let $RK_{\mathcal{I} \to \mathcal{I}'} = (d_1', d_2', \cdots, d_k', \mathbb{C})$ be a re-encryption key for $\mathcal{I} \to \mathcal{I}'$, $\mathrm{CT}_{\mathcal{I}}$ be a well-formed ciphertext for identity $\mathcal{I}$, it computes $C' = e(d_1', C_1) \cdot e(d_2', C_2) \cdots e(d_k', C_k)$, sets $\bar{C} = C_k^t$ and outputs the re-encrypted ciphertext $\mathrm{CT}_{\mathcal{I}'} = (C, C', \bar{C}, \mathbb{C})$.

Decrypt($\mathrm{CT}_{\mathcal{I}}, SK_{\mathcal{I}}$). Let $\mathrm{CT}_{\mathcal{I}}$ be a ciphertext for identity $\mathcal{I}$, it can be decrypted as follows:

- If $\mathrm{CT}_{\mathcal{I}}$ is an original well-formed ciphertext, then $M = C \cdot e(d_1, C_1) \cdot e(d_2, C_2) \cdots e(d_k, C_k)$.
- Else if $\mathrm{CT}_{\mathcal{I}}$ is a re-encrypted well-formed ciphertext, then
  1. Decrypts $E(g^d)$ from $\mathbb{C}$ using the secret key $SK_{\mathcal{I}}$ and decodes it to g^d,
  2. $M = C \cdot C' / e(g^d, \bar{C})$.
- Else if $\mathrm{CT}_{\mathcal{I}}$ is a multi-time re-encrypted well-formed ciphertext, decryption is similar with the above phases.

Note that $\mathbb{C}$ can be re-encrypted again. Thus, we could obtain $\mathrm{CT}_{\mathcal{I}''} = (C, C', \bar{C}, \mathbb{C}')$, where $\mathbb{C}'$ is obtained from the **Reencrypt** algorithm with the input of another $RK_{\mathcal{I}' \to \mathcal{I}''}$ and $\mathbb{C}$. The decryption cost and size of ciphertext grows linearly with the re-encryption times. As stated in [6], it seems to be inevitable for a non-interactive scheme.

We have the following security results for the transformed scheme.

Theorem 1. *Let $\mathcal{E}$ be an IBE scheme and $\mathcal{PE}$ be the corresponding transformed proxy re-encryption scheme. If $\mathcal{E}$ is* IND-ID-CPA *secure , then $\mathcal{PE}$ is* IND-PrID-CPA *secure.*

Intuition. Note that we don't know the concrete construction of the original scheme and therefore we don't know what assumption the original scheme is based on, so we can not directly reduce the semantic security of the transformed scheme to some assumption. We prove this by an indirect way which is based on the semantic security of the original scheme, that is, we construct an IND-PrID-CPA adversary, if she can break our transformed IB-PRE scheme, we can take the adversary as an oracle to break the original scheme.

Proof. We show how to construct a simulator $\mathcal{B}$ which can take the adversary $\mathcal{A}$ as an oracle to play the IND-ID-CPA game with the challenger $\mathcal{C}$ to break $\mathcal{E}$.

Let Ω be the identity space, $\mathcal{B}$ maintains a list with tuples $(\beta, \mathcal{I}_1, \mathcal{I}_2) \in \{0, 1\} \times \Omega \times \Omega$. Let $*$ denote the wildcard. Without loss of generality, we assume an input is queried to an oracle only once.

Setup. The challenger $\mathcal{C}$ generates the master public parameters PK and gives them to $\mathcal{B}$. $\mathcal{B}$ chooses random integers $t \in \mathbb{Z}_p^*$, computes $h = g^t$ and outputs the new public key $\mathrm{PK}' = \mathrm{PK} \cup \mathrm{PK}^t \cup \{h\}$ and the additional master secret key $\mathrm{MK}' = \{t\}$. Then $\mathcal{B}$ gives PK' to the adversary $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can make the following queries.

- **Extract**($\mathcal{I}$): $\mathcal{A}$ submits an identity $\mathcal{I}$ for a **KeyGen** query. $\mathcal{B}$ flips a biased coin $\beta \in \{0, 1\}$ that yields 1 with probability δ and 0 otherwise. If $\beta = 0$ or $(0, \mathcal{I}, *)$ or $(0, *, \mathcal{I})$ already exists on the list, $\mathcal{B}$ outputs a random bit and aborts. Otherwise, $\mathcal{B}$ sends the query to the challenger $\mathcal{C}$ to get the secret key $SK_{\mathcal{I}}$ and returns to $\mathcal{A}$. $\mathcal{B}$ also adds $(1, \mathcal{I}, \mathcal{I})$ to the list.
- **RKExtract**($\mathcal{I}_1, \mathcal{I}_2$): $\mathcal{A}$ submits an identity pair $(\mathcal{I}_1, \mathcal{I}_2)$ for a **RKGen** query. $\mathcal{B}$ chooses a random coin β as in the **Extract**. If $\beta = 1$ or $(1, \mathcal{I}_1, \mathcal{I}_1)$ or $(1, \mathcal{I}_2, \mathcal{I}_2)$ already exists on the list, $\mathcal{B}$ queries the key extraction oracle of $\mathcal{E}$ for $\mathcal{I}_1$, and then computes the re-encryption key $SK_{\mathcal{I}_1 \to \mathcal{I}_2}$ as the original scheme and returns it to $\mathcal{A}$. Otherwise, $\mathcal{B}$ queries the key extraction oracle of $\mathcal{E}$ for a random identity $\mathcal{I}$ to get a valid but random key $(d_1, d_2, \cdots, d_k)$, returns the re-encryption key as $(d_1, d_2, \cdots, d_k \cdot y, \textbf{Encrypt}(\mathrm{PK}, \mathcal{I}_2, z))$, where y, z are random elements of $\mathbb{G}, \mathbb{G}_1$ respectively. Finally, $\mathcal{B}$ adds $(\beta, \mathcal{I}_1, \mathcal{I}_2)$ to the list.

Challenge. $\mathcal{A}$ submits $(\mathcal{I}^*, M_0, M_1)$ to $\mathcal{B}$. If $(1, \mathcal{I}^*, *)$ exists on the table, $\mathcal{B}$ simply outputs a random bit and aborts. Otherwise, $\mathcal{B}$ submits the same challenge $(\mathcal{I}^*, M_0, M_1)$ to $\mathcal{C}$. When the challenger returns ciphertext CT^*, $\mathcal{B}$ returns CT^* to $\mathcal{A}$.

Phase 2. Phase 1 is repeated with the restriction that $\mathcal{A}$ cannot make the queries described in Definition 3.

- **Extract**$(\mathcal{I})$: $\mathcal{B}$ answers queries as in the **Phase 1**.
- **RKExtract**$(\mathcal{I}_1, \mathcal{I}_2)$: If $\mathcal{I}_1 \neq \mathcal{I}^*$, $\mathcal{B}$ queries the key extraction oracle of $\mathcal{E}$ for $\mathcal{I}_1$, and then computes the re-encryption key $SK_{\mathcal{I}_1 \to \mathcal{I}_2}$ as the original scheme and returns it to $\mathcal{A}$. Otherwise, $\mathcal{B}$ queries the key extraction oracle of $\mathcal{E}$ for a random identity $\mathcal{I}$ to get a valid but random key $(d_1, d_2, \cdots, d_k)$, returns the re-encryption key as $(d_1, d_2, \cdots, d_k \cdot y, \mathbf{Encrypt}(\mathrm{PK}, \mathcal{I}_2, z))$, where y, z are random elements of $\mathbb{G}, \mathbb{G}_1$ respectively. Finally, $\mathcal{B}$ adds $(0, \mathcal{I}_1, \mathcal{I}_2)$ to the list.

Guess. When $\mathcal{A}$ outputs its guess b' of b, $\mathcal{B}$ outputs b'.

We can see that if $\mathcal{B}$ does not abort during the game, the view of $\mathcal{A}$ is identical to the real attack except for some incorrect re-encryption keys (when $\beta = 0$). We will address this case later (in Lemma 1) by showing that $\mathcal{A}$ cannot distinguish these random generated keys from the real keys. So now we only need to calculate the probability that B aborts during the game. Suppose $\mathcal{A}$ makes a total of q_E private key extraction queries. The probability that $\mathcal{B}$ does not abort in phases 1 or 2 is δ^{q_E}. The probability that it does not abort during the challenge step is $1 - \delta$. Therefore, the probability that $\mathcal{B}$ does not abort during the game is $\delta^{q_E}(1 - \delta)$. This value is maximized at $\delta_{max} = 1 - 1/(q_E + 1)$. Using δ_{max}, the probability that B does not abort is at least $1/e(q_E + 1)$. So $\mathcal{B}$'s advantage is at least $\epsilon/e(q_E + 1)$.

Lemma 1. *If $\mathcal{E}$ is* IND-ID-CPA *secure, then the simulation in the proof of Theorem 3 is computationally indistinguishable from the real game.*

Proof. The simulation in the proof of Theorem 3 almost acts the same as the real scheme, except for the incorrect form of re-encryption keys for $\beta = 0$. Therefore we only consider the indistinguishability of the re-encryption key $(d_1, d_2, \cdots, d_k \cdot y, \mathbf{Encrypt}(\mathrm{PK}, \mathcal{I}_2, z))$ for randomly chosen identity and the real re-encryption key. Note that d_k contains information of identity, there must be a valid secret key for $\mathcal{I}_1$ with form $(d_1, d_2, \cdots, d_k \cdot w)$ where $w \in \mathbb{G}$, the problem is equivalent to the distinguishability of the encryption of a random element $z \in \mathbb{G}_1$ and the encryption of some $K \in \mathbb{G}_1$, where $K = E(k), k \in \mathbb{G}, k^t = w^{-1} \cdot y$. Therefore the simulation works if $\mathcal{E}$ is IND-ID-CPA secure.

Theorem 2. *Let $\mathcal{E}$ be an IBE scheme and $\mathcal{PE}$ be the corresponding transformed proxy re-encryption scheme. If $\mathcal{E}$ is* IND-ID-CPA *secure under some assumption, then $\mathcal{PE}$ has master key security.*

Proof. Note that if $\mathcal{E}$ is IND-ID-CPA secure under some assumption, $\mathcal{E}$ actually can provide the secret key for any identity in Phase 1 or Phase 2 of IND-ID-CPA game. That is, if we try to break the same assumption in our master key game, we can also generate all secret keys. Since the re-encryption key generation is trivial (recall that t and d are chosen by us), the broken of master key game means the adversary can break the assumption used in $\mathcal{E}$'s IND-ID-CPA game.

4 Discussions

We discuss a number of extensions to our IB-PRE construction of the previous section.

Re-encryption Control. In the **Decrypt** algorithm we can see that the recipient only needs g^d and C_k^t to decrypt the re-encrypted ciphertext. If the encryptor doesn't provide C_k^t in ciphertext, the original decryption is not affected but the decryption of re-encrypted ciphertext cannot go on. So the encryptor can decide whether the ciphertext can be re-encrypted.

Transformation under Selective Security. It's easy to see that our transformation can be applied in those schemes which is IND-sID-CPA secure. But there should be some modifications on the security model to fit selective identity. We give the definitions of selective security model and example in Appendix A. Master key security is also achieved in our IND-sID-CPA example.

5 Conclusions

In this paper, we present a novel construction of identity-based proxy re-encryption which can transform a current IBE scheme to a IB-PRE scheme. The transformed scheme is IND-PrID-CPA secure if the original scheme is IND-ID-CPA secure, has master key security and allows the encryptor to decide whether the ciphertext can be re-encrypted.

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.: Divertible protocols and atomic proxy cryptography. In: K.Nyberg (ed.) Advances in Cryptology - EUROCRYPT'98. LNCS, vol. 1403, pp. 127–144. Springer-Verlag (1998)
3. Boneh, D., Boyen, X.: Efficient selective-id secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology - EUROCRYPTO 2004. LNCS, vol. 3027, pp. 223–238. Springer-Verlag (2004)
4. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: J.Kilian (ed.) Advances in Cryptology - CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer-Verlag (2001)
5. Chu, C.K., Tzeng, W.G.: Identity-based proxy re-encryption without random oracles. In: Garay, J. (ed.) ISC 2007. LNCS, vol. 4779, pp. 189–202. Springer-Verlag (2007)
6. Green, M., Ateniese, G.: Identity-based proxy re-encryption. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 288–306. Springer-Verlag (2007)
7. Mambo, M., Okamoto, E.: Proxy cryptosystems: Delegation of the power to decrypt ciphertexts. IEICE transactions on fundamentals of electronics, Communications and computer sciences 80(1), 54–63 (1997)
8. Matsuo, T.: Proxy re-encryption systems for identity-based encryption. In: Takagi, T. (ed.) Pairing 2007. LNCS, vol. 4575, pp. 247–267. Springer-Verlag (2007)
9. Shamir, A.: Identity-based cryptosystems and signatures schemes. In: Advances in Cryptology - Crypto 1984. LNCS, vol. 196, pp. 47–53. Springer-Verlag (1984)

A Selective Security

Definition 5. *The selective security of an IB-PRE scheme is defined according to the following* IND-sPrID-ATK *game, where* $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA}\}$.

Init. *The adversary* $\mathcal{A}$ *submits a challenge identity* $\mathcal{I}^*$ *to the challenger* $\mathcal{B}$.
Setup. *The challenger runs the* **Setup** *algorithm and gives* PK *to the adversary* $\mathcal{A}$.
Phase 1. $\mathcal{A}$ *makes the following queries.*

- ***Extract***$(\mathcal{I})$ *and* $\mathcal{I} \neq \mathcal{I}^*$: $\mathcal{A}$ *submits an identity* $\mathcal{I}$ *for a* **KeyGen** *query, the challenger gives the adversary the secret key* $SK_{\mathcal{I}}$.
- ***RKExtract***$(\mathcal{I}, \mathcal{I}')$ *and* $\mathcal{I} \neq \mathcal{I}^*$: $\mathcal{A}$ *submits an identity pair* $(\mathcal{I}, \mathcal{I}')$ *for a* **RKGen** *query, the challenger gives the adversary the re-encryption key* $SK_{\mathcal{I} \to \mathcal{I}'}$.

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

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

Challenge. $\mathcal{A}$ *submits two equal length messages* M_0, M_1 *to* $\mathcal{B}$. $\mathcal{C}$ *flips a random coin* b *and passes the ciphertext* $\mathrm{CT}^* = \mathbf{Encrypt}(PK, M_b, \mathcal{I}^*)$ *to* $\mathcal{A}$.
Phase 2. *Phase 1 is repeated.*
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 an IB-PRE scheme is* IND-sPrID-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-sPrID-ATK *game.*

Definition 6. *The selective master key security of an IB-PRE scheme is defined according to the following selective master key security game.*

Init. *The adversary* $\mathcal{A}$ *submits a challenge identity* $\mathcal{I}^*$ *to the challenger* $\mathcal{B}$.

Setup. *The challenger runs the* **Setup** *algorithm and gives* PK *to the adversary $\mathcal{A}$.*

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

- ***Extract**($\mathcal{I}$) and $\mathcal{I} \neq \mathcal{I}^*$: $\mathcal{A}$ submits an identity $\mathcal{I}$ for a* **KeyGen** *query, the challenger gives the adversary the secret key $SK_{\mathcal{I}}$.*
- ***RKExtract**($\mathcal{I}$,$\mathcal{I}'$): $\mathcal{A}$ submits an identity pair ($\mathcal{I}$,$\mathcal{I}'$) for a* **RKGen** *query, the challenger gives the adversary the re-encryption key $SK_{\mathcal{I}\to\mathcal{I}'}$.*

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 IB-PRE scheme has selective master key security if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in winning the master key security game.

Theorem 3. *Let $\mathcal{E}$ be an IBE scheme and $\mathcal{PE}$ be the corresponding transformed proxy re-encryption scheme. If $\mathcal{E}$ is* IND-sID-CPA *secure , then $\mathcal{PE}$ is* IND-sPrID-CPA *secure.*

Proof. We will show that if an adversary $\mathcal{A}$ can break scheme $\mathcal{PE}$ in the IND-sPrID-CPA game, we can construct a simulator $\mathcal{B}$ which can take the adversary $\mathcal{A}$ as an oracle to play the IND-sID-CPA game with the challenger $\mathcal{C}$ to break $\mathcal{E}$.

Init. The adversary $\mathcal{A}$ submits a challenge identity $\mathcal{I}^*$ to the challenger $\mathcal{B}$. $\mathcal{B}$ submits the same challenge identity to $\mathcal{C}$.

Setup. The challenger $\mathcal{C}$ generates the master public parameters PK and gives them to $\mathcal{B}$. $\mathcal{B}$ chooses random integers $t \in \mathbb{Z}_p^*$, computes $h = g^t$ and outputs the new public key $\mathrm{PK}' = \mathrm{PK} \cup \mathrm{PK}^t \cup \{h\}$ and the additional master secret key $\mathrm{MK}' = \{t\}$. Then $\mathcal{B}$ gives PK$'$ to the adversary $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can make the following queries.

- **Extract**($\mathcal{I}$) and $\mathcal{I} \neq \mathcal{I}^*$: $\mathcal{B}$ sends the query to the challenger $\mathcal{C}$ to get the secret key $SK_{\mathcal{I}}$ and returns to $\mathcal{A}$.
- **RKExtract**($\mathcal{I}$,$\mathcal{I}'$) and $\mathcal{I} \neq \mathcal{I}^*$: $\mathcal{B}$ queries the key extraction oracle of $\mathcal{E}$ for $\mathcal{I}$ to get a secret key $(d_1, d_2, \cdots, d_k)$, chooses a random $d \in \mathbb{Z}_p$, and returns the re-encryption key as $(d_1, d_2, \cdots, d_k \cdot h^d, \mathbf{Encrypt}(\mathrm{PK}, \mathcal{I}', E(g^d)))$.

Challenge. $\mathcal{A}$ submits two equal-length messages M_0, M_1 to $\mathcal{B}$. $\mathcal{B}$ submits the same challenge M_0, M_1 to $\mathcal{C}$. The challenger flips a coin b, encrypts M_b and returns ciphertext $(C_1, \cdots, C_k)$ to $\mathcal{B}$. $\mathcal{B}$ returns $\mathrm{CT}^* = (C_1, \cdots, C_k, C_k^t)$ to $\mathcal{A}$.

Phase 2. Phase 1 is repeated.

Guess. When $\mathcal{A}$ outputs its guess b' of b, $\mathcal{B}$ outputs b'.

Since the simulator can distinguish the ciphertext if and only if the adversary can distinguish the transformed ciphertext, the simulator's advantage in the IND-sID-CPA game is exactly ϵ. □

Proof of selective master key security is different from Theorem 2 because in the IND-sID-CPA game the simulator cannot provide the secret key for the challenge identity. So the proof mode in Theorem 2 is infeasible in the proof of selective master key security. But in the following we will see some transformed scheme can also achieve selective master key security.

We give an example which is transformed from BB_1 [3].

Setup(1^κ)**.** Given the security parameter κ, the setup algorithm chooses random generators $g, g_2, g_3 \in \mathbb{G}$ and two random integers $\alpha, t \in \mathbb{Z}_p^*$. Then the setup algorithm sets $g_1 = g^\alpha, Y = e(g_1, g_2), h = g^t$. Let $E : \mathbb{G} \to \mathbb{G}_1$ be an encoding between $\mathbb{G}$ and $\mathbb{G}_1$. The public key PK is published as

$$\mathrm{PK} = (Y, g, g_1, g_2, g_3, h, E),$$

and the master key MK is

$$\mathrm{MK} = (g_2^\alpha, t).$$

KeyGen$(\mathrm{MK}, \mathcal{I})$**.** To generate the secret key $SK_{\mathcal{I}}$ for an identity $\mathcal{I} \in \mathbb{Z}_p$, the key extract algorithm chooses random $r \in \mathbb{Z}_p$ and outputs $SK_{\mathcal{I}}$ as

$$SK_{\mathcal{I}} = (g^r, g_2^{-\alpha}(g_1^{\mathcal{I}} g_3)^{-r}).$$

RKGen$(RK_{\mathcal{I}\to\mathcal{I}'})$**.** Given a secret key (a_1, a_2) for identity $\mathcal{I}$, to generate the re-encryption key for $\mathcal{I} \to \mathcal{I}'$, the algorithm chooses random $d \in \mathbb{Z}_p$ and outputs $RK_{\mathcal{I}\to\mathcal{I}'}$ as

$$RK_{\mathcal{I}\to\mathcal{I}'} = (a_1, a_2h^d, \mathbf{Encrypt}(\mathrm{PK}, \mathcal{I}', E(g^d))).$$

Encrypt$(\mathrm{PK}, \mathcal{I}, M)$**.** To encrypt a message $M \in \mathbb{G}_1$ for an identity $\mathcal{I}$, the algorithm chooses random integer $s \in \mathbb{Z}_p$ and outputs the ciphertext CT as

$$\mathrm{CT} = (MY^s, (g_1^{\mathcal{I}} g_3)^s, g^s, h^s).$$

Reencrypt$(\mathrm{CT}_{\mathcal{I}}, RK_{\mathcal{I}\to\mathcal{I}'})$**.** Let $RK_{\mathcal{I}\to\mathcal{I}'} = (d_1, d_2, \mathbb{C})$ be a re-encryption key for $\mathcal{I} \to \mathcal{I}'$, $\mathrm{CT}_{\mathcal{I}} = (C, C_1, C_2, C_3)$ be a well-formed ciphertext for identity $\mathcal{I}$, it computes $C' = e(d_1, C_1) \cdot e(d_2, C_2)$, sets $\bar{C} = C_3$ and outputs the re-encrypted ciphertext $\mathrm{CT}_{\mathcal{I}'} = (C, C', \bar{C}, \mathbb{C})$.

Decrypt$(SK_{\mathcal{I}}, \mathrm{CT})$**.** If CT is an original well-formed ciphertext, supposing that $\mathrm{CT} = (C, C_1, C_2, C_3)$ and the corresponding secret key $SK_{\mathcal{I}} = (a_1, a_2)$, the algorithm outputs

$$M = C \cdot e(a_1, C_1) \cdot e(a_2, C_2).$$

Else if $\mathrm{CT} = (C, C', \bar{C}, \mathbb{C})$ is a re-encrypted well-formed ciphertext, then

1. Decrypts $E(g^d)$ from $\mathbb{C}$ using the secret key $SK_{\mathcal{I}}$ and decodes it to g^d,
2.
$$M = \tilde{C} \cdot C'/e(\bar{C}, g^d).$$

Else if CT is a multi-time re-encrypted well-formed ciphertext, decryption is similar with the above phases.

Lemma 2. *If the DBDH assumption holds, the above scheme is selective master key secure.*

Proof. We will show that a simulator $\mathcal{B}$ can break the DBDH assumption with advantage ϵ if it takes an adversary $\mathcal{A}$, who can break our scheme in the selective master key security game with advantage ϵ, as oracle.

Given a DBDH challenge $[g, A, B, C, Z] = [g, g^a, g^b, g^c, Z]$ by the challenger where Z is either $e(g, g)^{abc}$ or random with equal probability, the simulator $\mathcal{B}$ creates the following simulation.

Init. The simulator $\mathcal{B}$ runs $\mathcal{A}$. $\mathcal{A}$ gives $\mathcal{B}$ a challenge identity $\mathcal{I}^*$.

Setup. $\mathcal{B}$ chooses random $\gamma_1, \gamma_2 \in \mathbb{Z}_p$, sets $g_1 = A, g_2 = B, Y = e(A, B), g_3 = A^{-\mathcal{I}^*} g^{\gamma_1}, h = Ag^{\gamma_2}$. Note that it implies the master key g^{ab} is unknown to $\mathcal{B}$.

Query. $\mathcal{A}$ makes the following queries.

- **Extract**$(\mathcal{I})$ and $\mathcal{I} \neq \mathcal{I}^*$: $\mathcal{A}$ submits an identity $\mathcal{I}$ for a **KeyGen** query where $\mathcal{I} \neq \mathcal{I}^*$. $\mathcal{B}$ chooses random $r \in \mathbb{Z}_p$, computes $a_1 = B^{\frac{-1}{\mathcal{I}-\mathcal{I}^*}} g^r$, $a_2 = B^{\frac{\gamma_1}{\mathcal{I}-\mathcal{I}^*}} (A^{\mathcal{I}-\mathcal{I}^*} g^{\gamma_1})^{-r}$. We claim that (a_1, a_2) a valid random secret key for $\mathcal{I}$. To see this, let $r' = r - b/(\mathcal{I} - \mathcal{I}^*)$. That is, $r = r' + b/(\mathcal{I} - \mathcal{I}^*)$. Then we have that

$$a_1 = g^{r'}, a_2 = g^{\frac{b\gamma_1}{\mathcal{I}-\mathcal{I}^*}} (g^{a(\mathcal{I}-\mathcal{I}^*)} g^{\gamma_1})^{-r'-b/(\mathcal{I}-\mathcal{I}^*)} = g^{-ab}(g_1^{\mathcal{I}} g_3)^{-r'}.$$

- **RKExtract**$(\mathcal{I}, \mathcal{I}')$: $\mathcal{A}$ submits an identity pair $(\mathcal{I}, \mathcal{I}')$ for a **RKGen** query, the challenger gives the adversary the re-encryption key $RK_{\mathcal{I}\to\mathcal{I}'}$ as follows.
  If $\mathcal{I} \neq \mathcal{I}^*$, then $\mathcal{B}$ runs normal **KeyGen** algorithm and get a secret key $SK_{\mathcal{I}} = (a_1, a_2)$. Next $\mathcal{B}$ chooses random $d \in \mathbb{Z}_p$, computes $RK_{\mathcal{I}\to\mathcal{I}'}$ as

$$RK_{\mathcal{I}\to\mathcal{I}'} = (a_1, a_2h^d, \mathbf{Encrypt}(\mathrm{PK}, \mathcal{I}', E(g^d))).$$

  If $\mathcal{I} = \mathcal{I}^*$, then $\mathcal{B}$ chooses random $r, d \in \mathbb{Z}_p$, computes $RK_{\mathcal{I}^*\to\mathcal{I}'}$ as

$$RK_{\mathcal{I}^*\to\mathcal{I}'} = (g^r, g^{-r\gamma_1} B^{\gamma_2} (Ag^{\gamma_2})^d, \mathbf{Encrypt}(\mathrm{PK}, \mathcal{I}', E(Bg^d))).$$

  We claim that $RK_{\mathcal{I}^*\to\mathcal{I}'}$ is a valid random re-encryption key for $\mathcal{I}^* \to \mathcal{I}'$. To see this, let $d' = b + d$. Then we have that

$$g^{-r\gamma_1} B^{\gamma_2} (Ag^{\gamma_2})^d = g^{-ab}(g^{a\cdot\mathcal{I}^*} g^{-a\cdot\mathcal{I}^*} g^{\gamma_1})^{-r} (g^a g^{\gamma_2})^{b+d} = g^{-ab}(g_1^{\mathcal{I}^*} g_3)^{-r} h^{d'}$$

Output. $\mathcal{A}$ outputs the secret key $SK_{\mathcal{I}^*}$ for the challenge identity $\mathcal{I}^*$.

If it is a valid secret key, $SK_{\mathcal{I}^*}$ should satisfy the following equation

$$e((g_1^{\mathcal{I}^*} g_3)^s, a_1)e(g^s, a_2) = e((g^{\gamma_1})^s, a_1)e(g^s, a_2) = e(g_1, g_2)^s,$$

where s is chosen randomly from $\mathbb{Z}_p$. Then $\mathcal{B}$ checks the equation

$$e((g^c)^{\gamma_1}, a_1)e(g^c, a_2) = e(g, g)^{abc} \overset{?}{=} Z$$

and breaks the DBDH assumption.