On the Role of PKG for Proxy Re-encryption in Identity Based Setting

Xu an Wang[1], Xiaoyuan Yang[1] and Fagen Li[2,3]

[1]Key Laboratory of Information and Network Security
Engineering College of Chinese Armed Police Force, P.R. China
wangxahq@yahoo.com.cn

[2]School of Computer Science and Engineering
University of Electronic Science and Technology of China, P.R. China
[3]Future University, Japan

Abstract. In 1998, Blaze, Bleumer, and Strauss proposed a kind of cryptographic primitive called proxy re-encryption[3]. In proxy re-encryption, a proxy can transform a ciphertext computed under Alice's public key into one that can be opened under Bob's decryption key. In 2007, Matsuo proposed the concept of four types of proxy re-encryption schemes: CBE(Certificate Based Public Key Encryption) to IBE(Identity Based Encryption)(type 1), IBE to IBE(type 2), IBE to CBE (type 3), CBE to CBE (type 4)[29]. Now CBE to IBE and IBE to IBE proxy re-encryption schemes are being standardized by IEEEP1363.3 working group[31]. In this paper, based on [29] we pay attention to the role of PKG for proxy re-encryption in identity based setting. We find that if we allow the PKG to use its master-key in the process of generating re-encryption key for proxy re-encryption in identity based setting, many open problems can be solved. Our main results are as following: We construct the first proxy re-encryption scheme from CBE to IBE which can resist malicious PKG attack, the first proxy re-encryption scheme from IBE to CBE, the second proxy re-encryption scheme based on a variant of BB_1 IBE[1], the first proxy re-encryption scheme based on BB_2 IBE, the first proxy re-encryption scheme based on SK IBE, we also prove their security in their corresponding security models.

1 Introduction

The concept of proxy re-encryption(PRE) comes from the work of Blaze, Bleumer, and Strauss in 1998[3]. The goal of proxy re-encryption is to securely enable the re-encryption of ciphertexts from one key to another, without relying on trusted parties. In 2005, Ateniese et al proposed a few new PRE schemes and discussed its several potential applications such as e-mail forwarding, law enforcement, cryptographic operations on storage-limited devices, distributed secure file systems and outsourced filtering of encrypted spam [2]. Since then, many excellent schemes have been proposed[12,27,24,28,18,29,13,32]. In ACNS'07, Green et al. proposed the first identity based proxy re-encryption schemes(IDPRE) [18]. In ISC'07, Chu et al. proposed the first IND-ID-CCA2 IDPRE schemes in the standard model, they constructed their scheme based on Water's IBE. But unfortunately Shao et al. found a flaw in their scheme and they fixed this flaw by proposing an improved scheme [32]. In Pairing'07, Matsuo proposed another few more PRE schemes in identity based setting [29]. Interestingly, they proposed the concept of four types of PRE: CBE(Certificate Based Public Key Encryption) to IBE(Identity Based Encryption)(type 1), IBE to IBE(type 2), IBE to CBE (type 3), CBE to CBE (type 4)[29], which can help the ciphertext circulate smoothly in the network. They constructed two PRE schemes: one is the hybrid PRE from CBE to IBE, the other is the PRE from IBE to IBE. Both of the schemes are now being standardized by P1363.3 workgroup [31].

[1] The first proxy re-encryption based on BB_1 IBE is M07B proxy re-encryption in [29]l.

1.1 Our Motivation

We extend Matsuo's research on PRE in identity based setting [29]. We observe that: 1)One feature of proxy re-encryption from CBE to IBE scheme in [29] is that it inherits the key escrow problem from IBE. That is, PKG can decrypt every re-encrypted ciphertext for IBE users. We ask question like this: is it possible that the malicious PKG can not decrypt the re-encryption ciphertext? 2) Can we construct a PRE scheme from IBE to CBE? 3)In [30] there was a conclusion that it is hard to construct PRE scheme based on BF and SK IBE. But we know that in P1363.3/D1[31] there are three IBE schemes which have been standardized. They are BF, BB_1, SK IBE[31]. Naturally we ask question like this: 4) Can we construct another PRE schemes based on BB_1 IBE? 5)can we construct PRE schemes based on BB_2 IBE? 6)And can we construct PRE schemes based on SK IBE?

1.2 Our Contribution

Our contributions are mainly as following: 1) Like the idea in certificateless public encryption[1,20], if the IBE users can have their own secret key during the generating re-encryption key process, the delegatee can decrypt the re-encrypted ciphertext using this secret key while PKG no longer can. 2)If we follow the principal that all the work PKG can do is just generating private keys for IBE users, it is indeed difficult for constructing proxy re-encryption from IBE to CBE and PRE based on SK IBE. But if we allow PKG generating re-encryption keys for PRE by using its $\mathsf{master}-\mathsf{key}$, we can easily construct PRE from IBE to CBE, PRE based on a variant of BB_1 IBE, PRE based on SK IBE and PRE based on BB_2 IBE.

1.3 Roadmap

We organize our paper as following. In Section 2, we give some preliminaries which are necessary to understand our paper. In Section 3, we show how to solve the key escrow problem for proxy re-encryption scheme from CBE to IBE in [29]. In Section 4, we propose our new proxy re-encryption scheme from IBE to CBE and prove its security. We propose our new proxy re-encryption scheme based on a variant of BB_1 IBE and prove its security in Section5. In Section 6, we propose our new proxy re-encryption scheme based on BB_2 IBE and prove its security. In Section 7, we propose our new proxy re-encryption scheme based on SK IBE and prove its security. We give our conclusions in Section 10.

2 Preliminaries

In the following, we sometimes use notations described in this section without notice. We denote the concatenation of a and b by $a||b$, denote random choice from a set S by $\stackrel{R}{\leftarrow} S$.

2.1 Bilinear groups

Let G and G_1 be multiplicative cyclic groups of prime order p, and g be generator of G. We say that G_1 has an admissible bilinear map $e : G \times G \rightarrow G_1$. if the following conditions hold.

1. $e(g^a, g^b) = e(g, g)^{ab}$ for all a, b.
2. $e(g, g) \neq 1$.
3. There is an efficient algorithm to compute $e(g^a, g^b)$ for all a, b and g.

2.2 Assumptions

Definition 1. *For randomly chosen integers a, b, c* $\xleftarrow{R} Z_p^*$*, a random generator* $g \xleftarrow{R} G$*, and an element* $R \xleftarrow{R} G$*, we define the advantage of an algorithm* $\mathcal{A}$ *in solving the Decision Bilinear Diffie-Hellman(DBDH) problem as follows:*

$$Adv_G{}^{dbdh}(\mathcal{A}) =\mid Pr[\mathcal{A}(g, g^a, g^b, g^c, e(g,g)^{abc}) = 0] - Pr[\mathcal{A}(g, g^a, g^b, g^c, R) = 0] \mid$$

where the probability is over the random choice of generator $g \in G$*, the randomly chosen integers a, b, c, the random choice of* $R \in G$*, and the random bits used by A. We say that the* (k, t, ϵ)*-DBDH assumption holds in* $\mathbb{G}$ *if no t-time algorithm has advantage at least* ϵ *in solving the DBDH problem in G under a security parameter k.*

Definition 2. *For randomly chosen integers a, b, c* $\xleftarrow{R} Z_p^*$*, a random generator* $g \xleftarrow{R} G$*, and an element* $R \xleftarrow{R} G$*, we define the advantage of an algorithm* $\mathcal{A}$ *in solving the modified Decision Bilinear Diffie-Hellman(mDBDH) problem as follows:*

$$Adv_G^{dbdh}(\mathcal{A}) = |Pr[\mathcal{A}(g, g^a, g^{a^2}, g^b, g^c, e(g,g)^{abc}) = 0] - Pr[\mathcal{A}(g, g^a, g^{a^2}, g^b, g^c, R) = 0]|$$

where the probability is over the random choice of generator $g \in G$*, the randomly chosen integers a, b, c, the random choice of* $R \in G$*, and the random bits used by A. We say that the* (k, t, ϵ)*-mDBDH assumption holds in* $\mathbb{G}$ *if no t-time algorithm has advantage at least* ϵ *in solving the mDBDH problem in G under a security parameter k.*

Definition 3. *For randomly chosen integers* $x \xleftarrow{R} Z_p^*$*, a random generator* $g_1, g_2 \xleftarrow{R} G$*, we define the advantage of an algorithm* $\mathcal{A}$ *in solving the* q_1*-BDHI problem as follows:*

$$Adv_G^{q_1-BDHI}(\mathcal{A}) = |Pr[e(g_1, g_2)^{\frac{1}{x}} \leftarrow \mathcal{A}(g_1, xg_2, x^2g_2, x^3g_2, \cdots, x^{q_1}g_2)]|$$

where the probability is over the random choice of generator $g_1, g_2 \in G$*, the randomly chosen integers x, and the random bits used by A. We say that the* (k, t, ϵ)*-*q_1*-BDHI assumption holds in* $\mathbb{G}$ *if no t-time algorithm has advantage at least* ϵ *in solving the* q_1*-BDHI problem in G under a security parameter k.*

2.3 Certificate Based Public Key Encryption

A traditional certificate-based Public Key Encryption(CBE) system consists of the following algorithms.

1. **KeyGen**$_{\mathsf{CBE}}$**(k, aux)**. Given a security parameters k and auxiliary input aux, generate a secret key sk and the corresponding public key pk.
2. **Enc**$_{\mathsf{CBE}}$**(pk, aux, M)**. Given the public key pk with aux, compute the encryption of a message M, C_{pk}.
3. **Dec**$_{\mathsf{CBE}}$**(sk, aux,** C_{pk}**)**. Given the secret key sk with aux, decrypt the CBE ciphertext C_{pk}.

2.4 Identity Based Encryption

An Identity Based Encryption(IBE) system consists of the following algorithms.

1. $\mathbf{SetUp_{IBE}}(\mathsf{k})$. Given a security parameter k, PKG generate a pair (parms, mk), where parms denotes the public parameters and mk is the $\mathsf{master-key}$.
2. $\mathbf{KeyGen_{IBE}}(\mathsf{mk, parms, ID})$. Given the $\mathsf{master-key}$ mk and an identity ID with parms, generate a secret key $\mathsf{sk_{ID}}$ for ID.
3. $\mathbf{Enc_{IBE}}(\mathbf{ID}, \mathsf{parms, M})$. Given a message M and the identity ID with parms, compute the encryption of M, $\mathsf{C_{ID}}$ for ID.
4. $\mathbf{Dec_{IBE}}(\mathsf{sk, parms, C_{ID}})$. Given the secret key sk, decrypt the ciphertext $\mathsf{C_{ID}}$.

3 How to Solve Key Escrow Problem for PRE from CBE to IBE

3.1 Our Definition for PRE from CBE to IBE

Definition 4. PRE *from* CBE *to* IBE *consists of: 1)the three algorithms making up a* CBE *system* $\mathsf{KeyGen_{CBE}}$, $\mathsf{Enc_{CBE}}$ *and* $\mathsf{Dec_{CBE}}$ *2)the four algorithms making up an* IBE *system* $\mathsf{SetUp_{IBE}}$, $\mathsf{KeyGen_{IBE}}$, $\mathsf{Enc_{IBE}}$ *and* $\mathsf{Dec_{IBE}}$ *3)and three algorithms for re-encryption, which are*

1. $\mathbf{KeyGen}_{PRO}(\mathsf{sk}, \mathbf{ID}, \mathsf{mk}, \mathsf{parms})$. *Given a* CBE *secret key* sk, *an* IBE *secret key* sk_{ID} *for the* IBE *user ID,* PKG*'s* $\mathsf{master-key}$ mk *with* parms, *generate a re-encryption key* rk *which can re-encrypt* CBE *ciphertexts for* pk *into the* IBE *ciphertexts for ID.*
2. $\mathbf{ReEnc}(\mathsf{rk, parms, C_{pk}, ID})$. *Given the re-encryption key* rk, *a ciphertext* $\mathsf{C_{pk}}$ *encrypted under the traditional public key, and ID with* parms, *re-encrypt ciphertext* $\mathsf{C_{pk}}$ *into* C_{ID} *that can be decrypted by the* IBE *user ID.*
3. $\mathbf{Check}(\mathsf{parms, C_{pk}, pk})$. *Given* $\mathsf{C_{pk}}$ *and* pk *with* parms, *output 0 if* $\mathsf{C_{pk}}$ *is a malformed ciphertext. Otherwise, output 1.*

Remark 1. Our definition is different from Matsuo's definition [29] about PRE from CBE to IBE. That is, we allow PKG generating re-encryption key directly by using its $\mathsf{master-key}$ mk while Matsuo's scheme only allow PKG helping the delegator and the delegatee generating re-encryption key indirectly.

Remark 2. Just like the PRE definition in Section 2.1 in [27], sometimes we can further distinguish the $\mathsf{Enc_{CBE}}$ and $\mathsf{Dec_{CBE}}$, $\mathsf{Enc_{IBE}}$ and $\mathsf{Dec_{IBE}}$ algorithms as two level algorithms. For example, we can distinguish $\mathsf{Enc_{IBE}}$ as $\mathsf{Enc1_{IBE}}$ and $\mathsf{Enc2_{IBE}}$ algorithms. $\mathsf{Enc2_{IBE}}$ outputs a second level ciphertext which can be re-encrypted as a first level ciphertext. $\mathsf{Enc1_{IBE}}$ outputs a first level ciphertext which can not be re-encrypted. In our proposed PRE from CBE to IBE3.4, we distinguish $\mathsf{Dec_{IBE}}$ as a two level algorithm. $\mathsf{Dec2_{IBE}}$ can only decrypts the second level ciphertext-normal IBE ciphertext while $\mathsf{Dec1_{IBE}}$ can only decrypts the first level ciphertext- the re-encrypted ciphertext.

3.2 Our Security Models for PRE from CBE to IBE

In this section, we give our security models for PRE from CBE to IBE which based on [12,27]. *Internal and External Security.* Our security model protects users from two types of attacks: those launched from parties outside the system (*External Security*), and those launched from

parties inside the system, such as the proxy, another partner, PKG, or some collusion between them (*Internal Security*). Generally speaking, internal adversaries are more powerful than external adversaries. And our scheme can achieve reasonable internal security. We just provide formalization of *internal security* notions.

Delegatee Security.

Because in PRE from CBE to IBE, PKG knows every IBE's normal secret key. So for every level 2 normal ciphertext, PKG can decrypt them[2]. Thus we only consider the case that proxy and delegator are colluding for level 2 ciphertext.

Definition 5. (IBE-LV2-IND-ID-CPA) *A* PRE *scheme from* CBE *to* IBE *is IBE-LV2-IND-ID-CPA*[3] *secure if the probability*

$$\begin{aligned}
Pr[\{(ID^\star, sk_{ID^\star})\} \leftarrow KeyGen_{IBE}(\cdot),\\
\{(pk_x, sk_x) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_x, sk_{ID_x}) \leftarrow KeyGen_{IBE}(\cdot)\},\\
\{(pk_h, sk_h) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_h, sk_{ID_h}) \leftarrow KeyGen_{IBE}(\cdot)\},\\
\{R_{hx} \leftarrow KeyGen_{PRO}(sk_h, ID_x, mk, parms)\}, \{R_{xh} \leftarrow KeyGen_{PRO}(sk_x, ID_h, mk, parms)\},\\
\{R_{hh} \leftarrow KeyGen_{PRO}(sk_h, ID_h, mk, parms)\}, \{R_{xx} \leftarrow KeyGen_{PRO}(sk_x, ID_x, mk, parms)\},\\
\{R_{x\star} \leftarrow KeyGen_{PRO}(sk_x, ID^\star, mk, parms)\}, \{R_{h\star} \leftarrow KeyGen_{PRO}(sk_h, ID^\star, mk, parms)\},\\
(m_0, m_1, St) \leftarrow A^{\mathcal{O}_{renc}}(ID^\star, \{(pk_x, sk_x)\}, \{(ID_x, sk_{ID_x})\}, \{pk_h\}, \{ID_h\}, \{R_{xh}\},\\
\{R_{hx}\}, \{R_{hh}\}, \{R_{xx}\}, \{R_{x\star}\}, \{R_{h\star}\}),\\
d^\star \xleftarrow{R} \{0,1\}, C^\star = Enc_{IBE}(m_{d^\star}, ID^\star, parms), d' \leftarrow \mathcal{A}^{\mathcal{O}_{renc}}(C^\star, St) : d' = d^\star]
\end{aligned}$$

is negligibly close to $1/2$ *for any PPT adversary* $\mathcal{A}$*. In our notation, St is a state information maintained by* $\mathcal{A}$ *while* $(ID^\star, sk_{ID^\star})$ *is the target user's pubic and private key pair generated by the challenger which also chooses other keys for corrupt and honest parties. For other honest parties, keys are subscripted by h and we subscript corrupt keys by x. Oracles* $\mathcal{O}_{renc}$ *proceeds as follows:*

- **Re-encryption** $\mathcal{O}_{renc}$**:** *on input* (pk_i, ID_j, C_{pk_i})*, where* C_{pk_i} *is the ciphertext under the public key* pk_i *,* pk_i *were produced by* $\mathsf{Keygen_{CBE}}$*,* ID_j *were produced by* $\mathsf{Keygen_{IBE}}$*, this oracle responds with* 'invalid' *if* C_{pk_i} *is not properly shaped w.r.t.* pk_i*. Otherwise the re-encrypted first level ciphertext* $C_{ID} = ReEnc(KeyGen_{PRO}(sk_i, ID_j, mk, parms), ID_j, parms, C_{pk_i})$ *is returned to* $\mathcal{A}$*.*

Remark 3. The Re-encryption Oracle $\mathcal{O}_{renc}$ can not give the adversary more help, because we consider the case the proxy and the delegator corrupted. When the proxy is corrupted, the adversary can do re-encryption himself. The reason why we do not delete the Re-encryption Oracle $\mathcal{O}_{renc}$ oracle in the above definition is that this makes our definition more general and consistent with other definitions in the literature[12,27].

In PRE from CBE to IBE, the delegator certainly can decrypt the ciphertext which will be re-encrypted. Thus we consider only the case that proxy and PKG are colluding, We must point

[2] normal IBE means the usually standardized IBE. normal secret key and normal ciphertext mean the secret key generated by $\mathsf{KeyGen_{IBE}}$(mk, parms, ID). and ciphertext generated by $\mathsf{Enc_{IBE}}$(ID, parms, M) in 2.4

[3] LV2 denotes Level 2 ciphertext.

out this model is not considered in the current literature. The goal of solving the key escrow problem for PRE from CBE to IBE is just constructing a scheme which can resist the malicious PKG attack. But we consider a stronger model which can resist the the malicious PKG and proxy colluding attack.

Definition 6. (IBE-LV1-IND-ID-CPA) *A* PRE *scheme from* CBE *to* IBE *is IBE-LV1-IND-ID-CPA*[4] *secure if the probability*

$$\begin{aligned}
&Pr[(parms, master-key) \leftarrow Setup_{IBE}(\cdot),\\
&\{(ID^\star, sk_{ID^\star}) \leftarrow KeyGen_{IBE}(\cdot)\}, \{(pk^\star, sk^\star) \leftarrow KeyGen_{CBE}(\cdot)\},\\
&\{(pk_x, sk_x) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_x, sk_{ID_x}) \leftarrow KeyGen_{IBE}(\cdot)\},\\
&\{(pk_h, sk_h) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_h, sk_{ID_h}) \leftarrow KeyGen_{IBE}(\cdot)\},\\
&\{R_{hx} \leftarrow KeyGen_{PRO}(sk_h, ID_x, mk, parms)\}, \{R_{xh} \leftarrow KeyGen_{PRO}(sk_x, ID_h, mk, parms)\},\\
&\{R_{hh} \leftarrow KeyGen_{PRO}(sk_h, ID_h, mk, parms)\}, \{R_{xx} \leftarrow KeyGen_{PRO}(sk_x, ID_x, mk, parms)\},\\
&\{R_{x\star} \leftarrow KeyGen_{PRO}(sk_x, ID^\star, mk, parms)\}, \{R_{h\star} \leftarrow KeyGen_{PRO}(sk_h, ID^\star, mk, parms)\},\\
&\{R_{\star h} \leftarrow KeyGen_{PRO}(sk^\star, ID_h, mk, parms)\}, \{R_{\star x} \leftarrow KeyGen_{PRO}(sk^\star, ID_x, mk, parms)\},\\
&\{R_{\star\star} \leftarrow KeyGen_{PRO}(sk^\star, ID^\star, mk, parms)\}\\
&(m_0, m_1, St) \leftarrow A^{O_{renc}}(ID^\star, pk^\star\{(pk_x, sk_x)\}, \{(ID_x, sk_{ID_x})\}, \{pk_h\}, \{ID_h\}, \{R_{xh}\},\\
&\{R_{hx}\}, \{R_{hh}\}, \{R_{xx}\}, \{R_{\star h}\}, \{R_{\star x}\}, \{R_{h\star}\}, \{R_{x\star}\}, \{R_{\star\star}\}, \{master-key\}),\\
&d^\star \xleftarrow{R} \{0,1\}, C^\star = ReEnc(Enc_{CBE}(m_{d^\star}, pk^\star), ID^\star, R_{\star\star}, parms), d' \leftarrow \mathcal{A}^{O_{renc}}(C^\star, St) : d' = d^\star]
\end{aligned}$$

is negligibly close to $1/2$ *for any PPT adversary* $\mathcal{A}$*. The notations in this game are same as Definition 5 except the definition of* Re-encryption Oracle $\mathcal{O}_{renc}$*. In this game, any input makes* $\mathcal{O}_{renc}$ *outputting* $C^\star$ *will be returned with* $\perp$*.*

Remark 4. In this definition, we set two target users - $pk^\star$, $ID^\star$. The reason is that the target ciphertext can be seen as the ciphertext for $ID^\star$ and its second level ciphertext can be seen as the ciphertext for $pk^\star$. In our definition, we consider the proxy being corrupted. That means, the proxy can know which second level ciphertext can be re-encrypted as the target first level ciphertext. Of course, if the proxy is not corrupted, and the proxy re-encryption is untraceable, the security model can allow any delegator corrupting including $pk^\star$.

Delegator Security.

In PRE from CBE and IBE, the delegator is a CBE user. In this case, we consider the delegatee, proxy and PKG are all colluding.

Definition 7. (CBE-IND-CPA) *A* PRE *scheme from* CBE *to* IBE *is CBE-IND-CPA secure if the probability*

$$\begin{aligned}
&\Pr[(parms, master-key) \leftarrow Setup_{IBE}(\cdot),\\
&\{(pk^\star, sk^\star) \leftarrow KeyGen_{CBE}(\cdot)\},\\
&\{(pk_x, sk_x) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_x, sk_{ID_x}) \leftarrow KeyGen_{IBE}(\cdot)\},
\end{aligned}$$

[4] LV1 denotes Level 1 ciphertext.

$$\{(pk_h, sk_h) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_h, sk_{ID_h}) \leftarrow KeyGen_{IBE}(\cdot)\},$$
$$\{R_{hx} \leftarrow KeyGen_{PRO}(sk_h, ID_x, mk, parms)\}, \{R_{xh} \leftarrow KeyGen_{PRO}(sk_x, ID_h, mk, parms)\},$$
$$\{R_{hh} \leftarrow KeyGen_{PRO}(sk_h, ID_h, mk, parms)\}, \{R_{xx} \leftarrow KeyGen_{PRO}(sk_x, ID_x, mk, parms)\},$$
$$\{R_{\star h} \leftarrow KeyGen_{PRO}(sk^\star, ID_h, mk, parms)\}, \{R_{\star x} \leftarrow KeyGen_{PRO}(sk^\star, ID_x, mk, parms)\},$$
$$(m_0, m_1, St) \leftarrow A^{O_{renc}}(pk^\star\{(pk_x, sk_x)\}, \{(ID_x, sk_{ID_x})\}, \{pk_h\}, \{ID_h\}, \{R_{xh}\},$$
$$\{R_{hx}\}, \{R_{hh}\}, \{R_{xx}\}, \{R_{\star h}\}, \{R_{\star x}\}, \{master - key\}),$$
$$d^\star \xleftarrow{R} \{0,1\}, C^\star = Enc_{CBE}(m_{d^\star}, pk^\star), d' \leftarrow A^{O_{renc}}(C^\star, St) : d' = d^\star]$$

is negligibly close to $1/2$ *for any PPT adversary* $\mathcal{A}$*. The notations in this game are same as Definition 5.*

PKG Security.

In PRE from CBE and IBE, PKG's $\mathsf{master-key}$ can not leverage even if the delegator, the delegatee and proxy collude.

Definition 8. (PKG-OW) *A* PRE *scheme from* CBE *to* IBE *is one way secure for* PKG *if the output*

$$Pr[\{(pk_x, sk_x) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_x, sk_{ID_x}) \leftarrow KeyGen_{IBE}(\cdot)\},$$
$$\{(pk_h, sk_h) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_h, sk_{ID_h}) \leftarrow KeyGen_{IBE}(\cdot)\},$$
$$\{R_{hx} \leftarrow KeyGen_{PRO}(sk_h, ID_x, mk, parms)\}, \{R_{xh} \leftarrow KeyGen_{PRO}(sk_x, ID_h, mk, parms)\},$$
$$\{R_{hh} \leftarrow KeyGen_{PRO}(sk_h, ID_h, mk, parms)\}, \{R_{xx} \leftarrow KeyGen_{PRO}(sk_x, ID_x, mk, parms)\},$$
$$mk' \leftarrow \mathcal{A}^{O_{renc}}(\{(pk_x, sk_x)\}, \{(ID_x, sk_{ID_x})\}, \{(pk_h, sk_h)\}, \{(ID_h, sk_{ID_h})\}, \{R_{xh}\},$$
$$\{R_{hx}\}, \{R_{xx}\}, \{R_{hh}\}) : mk' = mk]$$

is negligibly close to 0 *for any PPT adversary* $\mathcal{A}$*. The notations in this game are same as Definition 5.*

3.3 Review of the PRE from CBE to IBE in Pairing'07

The PRE from CBE to IBE involves the ElGamal-type CBE scheme and the BB_1 IBE scheme.

- The underlying IBE scheme ($\mathsf{BB_1}$ IBE scheme):
  1. $\mathbf{SetUp}_{\mathsf{IBE}}(k)$. Given a security parameter k, select a random generator $g \in G$ and random elements $g_2, h \in G$. Pick a random $\alpha \in Z_p^*$. Set $g_1 = g^\alpha$,$\mathsf{mk} = g_2^\alpha$, and $\mathsf{parms} = (g, g_1, g_2, h)$. Let mk be the $\mathsf{master-key}$ and let parms be the public parameters.
  2. $\mathbf{KeyGen}_{\mathsf{IBE}}(\mathsf{mk}, \mathsf{parms}, \mathbf{ID})$. Given $\mathsf{mk} = g_2^\alpha$ and ID with parms, pick a random $u \in Z_p^*$. Set $sk_{\mathsf{ID}} = (d_0, d_1) = (g_2^\alpha (g_1^{ID} h)^u, g^u)$.
  3. $\mathbf{Enc}_{\mathsf{IBE}}(\mathbf{ID}, \mathsf{parms}, M)$. To encrypt a message $M \in G_1$ under the public key $\mathsf{ID} \in Z_p^*$, pick a random $r \in Z_p^*$ and compute $\widetilde{C_{ID}} = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3}) = (g^r, (g_1^{ID} h)^r, Me(g_1, g_2)^r)$.
  4. $\mathbf{Dec}_{\mathsf{IBE}}(sk_{\mathsf{ID}}, \mathsf{parms}, C_{ID})$. Given ciphertext $C_{ID} = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3})$ and the secret key $sk_{ID} = (d_0, d_1)$ with parms, compute $M = \widetilde{C_3} e(d_1, \widetilde{C_2}) / e(d_0, \widetilde{C_1})$.
- The underlying CBE scheme (ElGamal-type CBE scheme):

1. **KeyGen**$_{\mathsf{CBE}}(k, \mathsf{parms})$. Given a security parameter k , parms, pick a random $\theta, \beta, \delta \in Z_p$. Set $g_3 = g^{\theta}, g_4 = g_1^{\beta}, g_5 = h^{\delta}$. The public key is $pk = (g_3, g_4, g_5)$. The secret random key is $sk = (\theta, \beta, \delta)$.
2. **Enc**$_{\mathsf{CBE}}(pk, parms, M)$. Given $pk = (g_3, g_4, g_5)$ and a message M with parms, pick a random $r \in Z_p^*$ and compute $C_{pk} = (C_1, C_2, C_3, C_4) = (g_3^r, g_4^r, g_5^r, Me(g_1, g_2)^r)$.
3. **Dec**$_{\mathsf{CBE}}(sk, parms, C_{pk})$. Given $C_{pk} = (C_1, C_2, C_3, C_4)$ and the secret key $sk = (\theta, \beta, \delta)$ with parms, compute $M = C_4/e(C_2^{1/\beta}, g_2)$.

– The delegation scheme:

1. **EGen(**$sk_{\mathsf{ID}}, \mathsf{parms}$). Given $sk_{ID} = (d_0, d_1) = (g_2^{\alpha}(g_1^{ID}h)^u, g^u)$ for ID with parms, set $e_{ID} = d_1 = g^u$.
2. **KeyGen**$_{PRO}(sk, e_{ID}, \mathsf{parms})$. Given $sk = (\theta, \beta, \delta)$ and $e_{ID} = g^u$ for ID with parms, set $rk_{pk \to ID} = (\theta, g^{u/\beta}, \delta)$.
3. **ReEnc(**$rk_{ID}, \mathsf{parms}, C_{pk}, ID$). Given a CBE ciphertext $C_{pk} = (C_1, C_2, C_3, C_4)$, the re-encryption key $rk_{pk \to ID} = (\theta, g^{u/\beta}, \delta)$ and ID with parms, re-encrypt the ciphertext C_{pk} into C_{ID} as follows. $\widehat{C_{ID}} = (\widehat{C_1}, \widehat{C_2}, \widehat{C_3}) = (C_1^{1/\theta}, C_3^{1/\delta}, C_4 e(g^{u/\beta}, C_2^{ID}))$.
4. **Check(**$\mathsf{parms}, C_{pk}, pk$). Given $C_{pk} = (C_1, C_2, C_3, C_4)$ and $pk = (g_3, g_4, g_5)$ with parms, set $v_1 = e(C_1, g_4)$, $v_2 = e(C_2, g_3)$, $v_3 = e(C_2, g_5)$ and $v_4 = e(C_3, g_4)$. If $v_1 = v_2, v_3 = v_4$ then output 1, otherwise output 0.

Remark 5. In this scheme, the $\mathsf{EGen}(sk_{\mathsf{ID}}, \mathsf{parms})$, $\mathsf{KeyGen}_{PRO}(sk, e_{ID}, \mathsf{parms})$ algorithms can be replaced with one algorithm $\mathsf{KeyGen}_{PRO}(\mathsf{sk}, \mathsf{ID}, \mathsf{mk}, \mathsf{parms})$, which outputs $rk_{pk \to ID} = (\theta, g^{u/\beta}, \delta)$. Then the algorithms will be consistent with Definition 4.

Remark 6. In this scheme,PKG knows everything about the delegatee,the private key $sk_{ID} = (d_0, d_1) = (g_2^{\alpha}(g_1^{ID}h)^u, g^u)$, the ephemeral key e_{ID} for re-encryption key generation, he certainly can decrypt the re-encrypted ciphertext if the delegatee can.

3.4 Our Proposed PRE from CBE to IBE Which Can Resist Malicious PKG Attack

We construct our scheme based on the above PRE scheme. Our scheme shares the same underlying CBE scheme (ElGamal-type CBE scheme) as [29] scheme. The difference lies in the underlying IBE scheme and delegation scheme.

– The underlying IBE scheme (Variant of $\mathsf{BB_1}$ IBE scheme):

1. **SetUp**$_{\mathsf{IBE}}(k)$. Same as the above IBE scheme3.3 .
2. **KeyGen**$_{\mathsf{IBE}}(mk, \mathsf{parms}, ID)$. Same as the above IBE scheme3.3 except the following: the delegatee chooses a collision resistent hash function $H : \{0,1\}^{3|p|} \to Z_p^*$ and a random seed $r \in Z_p^*$, and computes $k = H(pk, ID, r)$. The delegatee's private key is $sk_{\mathsf{ID}} = (d_0, d_1, k) = (g_2^{\alpha}(g_1^{ID}h)^u, g^u, k)$.
3. **Enc**$_{\mathsf{IBE}}(\mathsf{ID}, \mathsf{parms}, M)$. Same as the above IBE scheme3.3.
4. **Dec1**$_{\mathsf{IBE}}(sk_{\mathsf{ID}}, \mathsf{parms}, C_{\mathsf{ID}})$. Given a re-encrypted ciphertext $\widehat{C_{ID}} = (\widehat{C_1}, \widehat{C_2}, \widehat{C_3}, \widehat{C_4})$, $sk_{ID} = (d_0, d_1, k)$, parms, compute $M = (\frac{\widehat{C_3}\widehat{C_4}^k e(d_1, \widehat{C_2}^k)}{e(d_0, \widehat{C_1}^k)})^{\frac{1}{k}}$.
5. **Dec2**$_{\mathsf{IBE}}(sk_{\mathsf{ID}}, \mathsf{parms}, C_{\mathsf{ID}})$. Same as the above IBE scheme3.3.

– The underlying CBE scheme (ElGamal-type CBE scheme): Same as the above CBE scheme 3.3.

– The delegation scheme:
  1. **KeyGen**$_{PRO}(sk, sk_{ID}, mk, \mathsf{parms})$. On input (θ, β, δ) from the delegator and input (g^u, k) from the delegatee, outputs the re-encryption key $rk_{pk\to ID} = (1/\theta, g^{ku/\beta}, 1/\delta)$.
  2. **ReEnc**$(rk_{ID}, \mathsf{parms}, C_{pk}, ID)$. Given a CBE ciphertext $C_{pk} = (C_1, C_2, C_3, C_4)$, the re-encryption key $rk_{pk\to ID} = (1/\theta, g^{ku/\beta}, 1/\delta)$ and ID with parms, re-encrypt the ciphertext C_{pk}into C_{ID} as following: $\widehat{C_{ID}} = (\widehat{C_1}, \widehat{C_2}, \widehat{C_3}, \widehat{C_4}) = (C_1^{1/\theta}, C_3^{1/\delta}, e(g^{ku/\beta}, C_2^{ID}), C_4)$.
  3. **Check**$(\mathsf{parms}, C_{pk}, pk)$. Given $C_{pk} = (C_1, C_2, C_3, C_4)$ and $pk = (g_3, g_4, g_5)$ with parms, set $v_1 = e(C_1, g_4)$, $v_2 = e(C_2, g_3)$, $v_3 = e(C_2, g_5)$ and $v_4 = e(C_3, g_4)$. If $v_1 = v_2$ and $v_3 = v_4$, output 1, otherwise output 0.

We verify correctness of our scheme. Following the $\mathsf{Dec2}_{\mathsf{IBE}}(sk_{\mathsf{ID}}, \mathsf{parms}, C_{\mathsf{ID}})$ algorithm, we get

$$(\frac{\widehat{C_3}\widehat{C_4}^k e(d_1, \widehat{C_2}^k)}{e(d_0, \widehat{C_1}^k)})^{\frac{1}{k}} = (\frac{e(g^{ku/\beta}, C_2'^{ID}) M^k e(g_1, g_2)^{rk} e(g^u, h^{kr})}{e(g_2^\alpha (g_1^{ID} h)^u, g^{rk})})^{\frac{1}{k}} = (\frac{M^k e(g_1, g_2)^{rk} e(g^{uk}, (g_1^{ID} h)^r)}{e(g_2^\alpha (g_1^{ID} h)^u, g^{rk})})^{\frac{1}{k}}$$
$$= (\frac{M^k e(g_1, g_2)^{rk}}{e(g_2^\alpha, g^{rk})})^{\frac{1}{k}} = (M^k)^{\frac{1}{k}} = M$$

Remark 7. In our scheme, every IBE user has a self generated private key k. It's this k that can make our scheme resist malicious PKG decrypting IBE user's re-encrypted ciphertext.

3.5 Security Analysis

Theorem 1. *Suppose the DBDH assumption holds, then our scheme is IBE-LV2-IND-sID-CPA secure for the proxy and delegator's colluding.*

Proof. See appendix A.

Theorem 2. *Our scheme is IBE-LV1-IND-ID-CPA secure for the proxy and* PKG*'s colluding.*

Proof. See appendix B.

Theorem 3. *Our scheme is CBE-IND-CPA secure for the proxy,* PKG *and delegatee's colluding except the case of the target CBE ciphertext has not been re-encrypted by the proxy.*

Proof. See appendix C.

Theorem 4. *Our scheme is not* CBE*-IND-CPA secure for the proxy,* PKG *and delegatee's colluding in the case of the target* CBE *ciphertext has been re-encrypted by the proxy.*

Proof. See appendix D.

Theorem 5. *Suppose the DBDH assumption holds, then our scheme is PKG-OW secure for all of the proxy, delegatee and delegator's colluding.*

Proof. See appendix E.

4 PRE from IBE to CBE

4.1 Our Definition for PRE from IBE to CBE

Definition 9. PRE *from* IBE *to* CBE *consists of: 1)the four algorithms making up an* IBE *system* $\mathsf{SetUp_{IBE}}$, $\mathsf{KeyGen_{IBE}}$, $\mathsf{Enc_{IBE}}$ *and* $\mathsf{Dec_{IBE}}$ *2)the three algorithms making up a* CBE *system* $\mathsf{KeyGen_{CBE}}$, $\mathsf{Enc_{CBE}}$ *and* $\mathsf{Dec_{CBE}}$ *3)and three algorithms for re-encryption, which are*

1. $\mathbf{KeyGen}_{PRO}(\mathsf{sk_{ID}}, \mathsf{sk}, \mathsf{pk}, \mathsf{mk}, \mathsf{parms})$. *Given an* IBE *secret key* sk_{ID} *for the* IBE *user* ID*, a* CBE *secret key* sk*,* PKG*'s* $\mathsf{master-key}$ mk *with* parms,pk*, generate a re-encryption key* rk *which can re-encrypt the* IBE *ciphertexts for* ID *into* CBE *ciphertexts for* pk*.*
2. $\mathbf{ReEnc}(\mathsf{rk}, \mathsf{parms}, \mathsf{C_{ID}}, \mathsf{pk})$. *Given the re-encryption key* rk*, a ciphertext* C_{ID} *encrypted under the identity* ID*, and* pk *with* parms*, re-encrypt ciphertext* C_{ID} *for* ID *into* $\mathsf{C_{pk}}$ *that can be decrypted by* sk*.*
3. $\mathbf{Check}(\mathsf{parms}, \mathsf{C_{ID}}, \mathbf{ID})$. *Given* C_{ID} *and* ID *with* parms*, output 0 if* C_{ID} *is a malformed ciphertext. Otherwise, output 1.*

4.2 Our Security Models for PRE from IBE to CBE

Delegator Security.

In PRE from IBE and CBE, the delegator is a IBE user. In this case, we consider the delegatee, proxy are colluding.

Definition 10. (IBE-IND-ID-CPA) *A* PRE *scheme from* IBE *to* CBE *is IBE-IND-ID-CPA secure if the probability*

$$\begin{aligned}
&Pr[\{(ID^\star, sk_{ID^\star})\} \leftarrow KeyGen_{IBE}(\cdot),\\
&\{(pk_x, sk_x) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_x, sk_{ID_x}) \leftarrow KeyGen_{IBE}(\cdot)\},\\
&\{(pk_h, sk_h) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_h, sk_{ID_h}) \leftarrow KeyGen_{IBE}(\cdot)\},\\
&\{R_{hx} \leftarrow KeyGen_{PRO}(sk_{ID_h}, sk_x, pk_x, mk, \cdot)\}, \{R_{xh} \leftarrow KeyGen_{PRO}(sk_{ID_x}, sk_h, pk_h, mk, \cdot)\},\\
&\{R_{hh} \leftarrow KeyGen_{PRO}(sk_{ID_h}, sk_h, pk_h, mk, \cdot)\}, \{R_{xx} \leftarrow KeyGen_{PRO}(sk_{ID_x}, sk_x, pk_x, mk, \cdot)\},\\
&\{R_{\star x} \leftarrow KeyGen_{PRO}(sk_{ID^\star}, sk_x, pk_x, mk, \cdot)\}, \{R_{\star h} \leftarrow KeyGen_{PRO}(sk_{ID^\star}, sk_h, pk_h, mk, \cdot)\},\\
&(m_0, m_1, St) \leftarrow A^{\mathcal{O}_{renc}}(ID^\star, \{(pk_x, sk_x)\}, \{(ID_x, sk_{ID_x})\}, \{pk_h\}, \{ID_h\}, \{R_{xh}\},\\
&\{R_{hx}\}, \{R_{hh}\}, \{R_{xx}\}, \{R_{\star x}\}, \{R_{\star h}\}),\\
&d^\star \xleftarrow{R} \{0,1\}, C^\star = Enc_{IBE}(m_{d^\star}, ID^\star, parms), d' \leftarrow \mathcal{A}^{\mathcal{O}_{renc}}(C^\star, St) : d' = d^\star]
\end{aligned}$$

is negligibly close to $1/2$ *for any PPT adversary* $\mathcal{A}$*. The notations in this game are same as Definition 5.*

Delegatee Security.

In PRE from IBE and CBE, the delegatee is a CBE user. We consider the second level CBE ciphertext [5]. In this case, we assume the delegator, proxy and PKG are colluding.

[5] Second level ciphertext means the normal CBE ciphertext

Definition 11. (CBE-LV2-IND-CPA) *A* PRE *scheme from* IBE *to* CBE *is CBE-LV2-IND-CPA secure for* CBE *if the probability*

$$\begin{aligned}
&\Pr[(parms, master-key) \leftarrow Setup_{IBE}(\cdot),\\
&\{(pk^\star, sk^\star) \leftarrow KeyGen_{CBE}(\cdot)\},\\
&\{(pk_x, sk_x) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_x, sk_{ID_x}) \leftarrow KeyGen_{IBE}(\cdot)\},\\
&\{(pk_h, sk_h) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_h, sk_{ID_h}) \leftarrow KeyGen_{IBE}(\cdot)\},\\
&\{R_{hx} \leftarrow KeyGen_{PRO}(sk_{ID_h}, sk_x, pk_x, mk, \cdot)\}, \{R_{xh} \leftarrow KeyGen_{PRO}(sk_{ID_x}, sk_h, pk_h, mk, \cdot)\},\\
&\{R_{hh} \leftarrow KeyGen_{PRO}(sk_{ID_h}, sk_h, pk_h, mk, \cdot)\}, \{R_{xx} \leftarrow KeyGen_{PRO}(sk_{ID_x}, sk_x, pk_x, mk, \cdot)\},\\
&\{R_{x\star} \leftarrow KeyGen_{PRO}(sk_{ID_x}, sk^\star, pk^\star, mk, \cdot)\}, \{R_{h\star} \leftarrow KeyGen_{PRO}(sk_{ID_h}, sk^\star, pk^\star, mk, \cdot)\},\\
&(m_0, m_1, St) \leftarrow A^{O_{renc}}(pk^\star, \{(pk_x, sk_x)\}, \{(ID_x, sk_{ID_x})\}, \{pk_h\}, \{ID_h\}, \{R_{xh}\},\\
&\{R_{hx}\}, \{R_{hh}\}, \{R_{xx}\}, \{R_{h\star}\}, \{R_{x\star}\}, \{master-key\}),\\
&d^\star \xleftarrow{R} \{0,1\}, C^\star = Enc_{CBE}(m_{d^\star}, pk^\star), d' \leftarrow A^{O_{renc}}(C^\star, St) : d' = d^\star]
\end{aligned}$$

is negligibly close to $1/2$ *for any PPT adversary* $\mathcal{A}$*. The notations in this game are same as Definition 5.*

In PRE from IBE and CBE, the delegatee is a CBE user. We consider the first level CBE ciphertext [6]. In this case, we assume the proxy and PKG are colluding.

Definition 12. (CBE-LV1-IND-CPA) *A* PRE *scheme from* IBE *to* CBE *is CBE-LV1-IND-CPA secure for* CBE *if the probability*

$$\begin{aligned}
&Pr[(parms, master-key) \leftarrow Setup_{IBE}(\cdot),\\
&\{(ID^\star, sk_{ID^\star}) \leftarrow KeyGen_{IBE}(\cdot)\}, \{(pk^\star, sk^\star) \leftarrow KeyGen_{CBE}(\cdot)\},\\
&\{(pk_x, sk_x) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_x, sk_{ID_x}) \leftarrow KeyGen_{IBE}(\cdot)\},\\
&\{(pk_h, sk_h) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_h, sk_{ID_h}) \leftarrow KeyGen_{IBE}(\cdot)\},\\
&\{R_{hx} \leftarrow KeyGen_{PRO}(sk_{ID_h}, sk_x, pk_x, mk, \cdot)\}, \{R_{xh} \leftarrow KeyGen_{PRO}(sk_{ID_x}, sk_h, pk_h, mk, \cdot)\},\\
&\{R_{hh} \leftarrow KeyGen_{PRO}(sk_{ID_h}, sk_h, pk_h, mk, \cdot)\}, \{R_{xx} \leftarrow KeyGen_{PRO}(sk_{ID_x}, sk_x, pk_x, mk, \cdot)\},\\
&\{R_{x\star} \leftarrow KeyGen_{PRO}(sk_{ID_x}, sk^\star, pk^\star, mk, \cdot)\}, \{R_{h\star} \leftarrow KeyGen_{PRO}(sk_{ID_h}, sk^\star, pk^\star, mk, \cdot)\},\\
&\{R_{\star x} \leftarrow KeyGen_{PRO}(sk_{ID^\star}, sk_x, pk_x, mk, \cdot)\}, \{R_{\star h} \leftarrow KeyGen_{PRO}(sk_{ID^\star}, sk_h, pk_h, mk, \cdot)\},\\
&\{R_{\star\star} \leftarrow KeyGen_{PRO}(sk_{ID^\star}, sk^\star, pk^\star, mk, \cdot)\}\\
&(m_0, m_1, St) \leftarrow A^{O_{renc}}(ID^\star, pk^\star\{(pk_x, sk_x)\}, \{(ID_x, sk_{ID_x})\}, \{pk_h\}, \{ID_h\}, \{R_{xh}\},\\
&\{R_{hx}\}, \{R_{hh}\}, \{R_{xx}\}, \{R_{\star h}\}, \{R_{\star x}\}, \{R_{h\star}\}, \{R_{x\star}\}, \{R_{\star\star}\}, \{master-key\}),\\
&d^\star \xleftarrow{R} \{0,1\}, C^\star = ReEnc(Enc_{IBE}(m_{d^\star}, ID^\star), pk^\star, sk^\star, R_{\star\star}, parms), d' \leftarrow \mathcal{A}^{O_{renc}}(C^\star, St) : d' = d^\star]
\end{aligned}$$

is negligibly close to $1/2$ *for any PPT adversary* $\mathcal{A}$*. The notations in this game are same as Definition 5.*

PKG Security.

In PRE from IBE to CBE, PKG's $\mathsf{master-key}$ can not leverage even if the delegator, the delegatee and proxy collude.

[6] first level ciphertext means the re-encrypted CBE ciphertext

Definition 13. (PKG-OW) *A* PRE *scheme from* IBE *to* CBE *is one way secure for PKG if the probability*

$$\begin{aligned}
Pr[\{(pk_x, sk_x) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_x, sk_{ID_x}) \leftarrow KeyGen_{IBE}(\cdot)\},\\
\{(pk_h, sk_h) \leftarrow KeyGen_{CBE}(\cdot)\}, \{(ID_h, sk_{ID_h}) \leftarrow KeyGen_{IBE}(\cdot)\},\\
\{R_{hx} \leftarrow KeyGen_{PRO}(sk_{ID_h}, sk_x, pk_x, mk, \cdot)\}, \{R_{xh} \leftarrow KeyGen_{PRO}(sk_{ID_x}, sk_h, pk_h, mk, \cdot)\},\\
\{R_{hh} \leftarrow KeyGen_{PRO}(sk_{ID_h}, sk_h, pk_h, mk, \cdot)\}, \{R_{xx} \leftarrow KeyGen_{PRO}(sk_{ID_x}, sk_x, pk_x, mk, \cdot)\},\\
mk' \leftarrow \mathcal{A}^{O_{renc}}(\{(pk_x, sk_x)\}, \{(ID_x, sk_{ID_x})\}, \{(pk_h, sk_h)\}, \{(ID_h, sk_{ID_h})\}, \{R_{xh}\},\\
\{R_{hx}\}, \{R_{xx}\}, \{R_{hh}\}) : mk' = mk]
\end{aligned}$$

is negligibly close to 0 *for any PPT adversary* $\mathcal{A}$*. The notations in this game are same as Definition 5.*

4.3 Our Proposed PRE Scheme from IBE to CBE

The PRE scheme from IBE to CBE involves the ElGamal-type CBE scheme and the BB_1 IBE scheme.

- The underlying IBE scheme (BB_1 IBE scheme):
  1. $\mathbf{SetUp}_{\mathsf{IBE}}(k)$. Same as the IBE scheme in Section 3.3 except this time PKG choose (g_2, h) as following: it first choose a generator $g \in G$, then randomly choose $t_1, t_2 \in Z_q^*$ and computes $g_2 = g^{t_1}, h = g^{t_2}$.
  2. $\mathbf{KeyGen}_{\mathsf{IBE}}(mk, parms, ID)$. Same as the IBE scheme in Section 3.3
  3. $\mathbf{Enc}_{\mathsf{IBE}}(ID, parms, M)$. Same as the IBE scheme in Section 3.3
  4. $\mathbf{Dec}_{\mathsf{IBE}}(sk_{ID}, parms, C_{ID})$. Same as the IBE scheme in Section 3.3
- The underlying CBE scheme (ElGamal-type CBE scheme):
  1. $\mathbf{KeyGen}_{\mathsf{CBE}}(k, \mathsf{parms})$. Given a security parameter k , parms, pick a random $\theta \in Z_p^*, k \in Z_p^*$. Set $g_3 = {g_1}^{\theta}$. The public key is $pk = g_3$. The secret key is $sk = (d_0, d_1) = (\theta, k)$.
  2. $\mathbf{Enc}_{\mathsf{CBE}}(\mathsf{pk}, \mathsf{parms}, M)$. Given $pk = g_3$ and a message M with parms, pick a random $r \in Z_p^*$ and compute $C_{pk} = (g_3^r, Me(g_1, g_2)^r)$.
  3. $\mathbf{Dec1}_{\mathsf{CBE}}(sk, \mathsf{parms}, C_{pk})$. Given $C_{pk} = (C_1, C_2)$ and the secret key $sk = (d_0, d_1) = (\theta, k)$ with parms, compute $M = C_2/e(C_1^{1/d_0}, g_2)$.
  4. $\mathbf{Dec2}_{\mathsf{CBE}}(sk, \mathsf{parms}, C_{pk})$. Given a re-encrypted ciphertext $\widehat{C_{pk}} = (\widehat{C_1}, \widehat{C_2})$ and the secret key $sk = (d_0, d_1) = (\theta, k)$ with parms, compute $M = \widehat{C_2}/e(\widehat{C_1}^{\frac{1}{d_0 d_1}}, g_2)$.
- The delegation scheme:
  1. $\mathbf{KeyGen}_{PRO}(\mathsf{sk_{ID}}, \mathsf{sk}, \mathsf{pk}, \mathsf{mk}, \mathsf{parms})$. The PKG first chooses a collision resistent hash function $H : \{0,1\}^{3|p|} \rightarrow Z_p^*$ and a random seed $s_1, s_2 \in Z_p^*$, and computes $k_1 = H(ID, pk, s_1), k_2 = H(ID, pk, s_2)$. The PKG computes $(\frac{\alpha+k_1}{ID\alpha+t_2} \mod p, g_2^{k_1})$ and sends it to the proxy. The delegatee sends $k\theta$ to the proxy. The proxy sets the re-encryption key $rk_{\mathsf{ID}\rightarrow\mathsf{pk}} = (rk_1, rk_2) = (\frac{(\alpha+k_1)k\theta}{ID\alpha+t_2}, g_2^{k_1})$.
  2. $\mathbf{ReEnc}(rk_{\mathsf{ID}\rightarrow\mathsf{pk}}, \mathsf{parms}, C_{ID}, pk)$. Given an IBE ciphertext $\widetilde{C_{ID}} = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3}) = (g^r, (g_1^{ID}h)^r, Me(g_1, g_2)^r)$ and re-encryption key $rk_{\mathsf{ID}\rightarrow\mathsf{pk}} = (rk_1, rk_2)$, the proxy re-encrypt the ciphertext $\widetilde{C_{ID}}$ into $\widehat{C_{pk}}$ as following. $\widehat{C_{pk}} = (\widehat{C_1}, \widehat{C_2}) = (\widetilde{C_2}^{rk_1}, \widetilde{C_3}e(\widetilde{C_1}, rk_2))$.
  3. $\mathbf{Check}(\mathsf{parms}, \widetilde{C_{ID}})$. Given $\widetilde{C_{ID}} = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3})$ with parms, set $v_1 = e(\widetilde{C_1}, g_1^{ID}h)$, $v_2 = e(\widetilde{C_2}, g)$. If $v_1 = v_2$ then output "Valid", otherwise output "Invalid".

We can verify its correctness as the following

$$\widehat{C_2}/e(\widehat{C_1}^{\frac{1}{d_0 d_1}}, g_2) = \frac{\widetilde{C_3} e(\widetilde{C_1}, rk_2)}{e(\widetilde{C_2}^{rk_1 \cdot \frac{1}{d_0 d_1}}, g_2)} = \frac{Me(g_1, g_2)^r e(g^r, rk_2)}{e(((g_1^{ID} h)^{r \cdot \frac{\alpha+k_1}{ID\alpha+t_2}} \cdot k\theta)^{\frac{1}{k\theta}}, g_2)} = \frac{Me(g_1, g_2)^r e(g^r, g_2^{k_1})}{e((g_1^{ID} h)^{r \cdot \frac{\alpha+k_1}{ID\alpha+t_2}}, g_2)}$$
$$= \frac{Me(g_1, g_2)^r e(g^r, g_2^{k_1})}{e(g^{(\alpha+k_1)r}, g_2)} = \frac{Me(g_1, g_2)^r e(g^r, g_2^{k_1})}{e(g^{\alpha r}, g_2) e(g^{k_1 r}, g_2)} = \frac{Me(g_1, g_2)^r e(g^r, g_2^{k_1})}{e(g_1, g_2)^r e(g^r, g_2^{k_1})} = M$$

Remark 8. In our scheme, we must note that the PKG needs to compute a different k_1 for every different user pair $(\mathsf{ID}, \mathsf{pk})$. Otherwise, if the adversary know $\frac{\alpha+k_1}{ID\alpha+t_2} \mod p$ for three different ID_1, ID_2, ID_3 but the same k_1 and pk, he can compute α, t_2, which is not secure at all.

Remark 9. In our scheme, $rk_1 = \frac{\alpha+k_1}{ID\alpha+t_2} \mod p$. One may wonder that every rk_1 for ID has a factor of form $\frac{1}{ID\alpha+t_2} \mod p$ which can help the adversary find $ID\alpha + t_2$. We comment that this attack can not succeed for this reason: when k_1 runs along $(1, 2, \cdots, p-1)$, $rk_1 = \frac{\alpha+k_1}{ID\alpha+t_2} \mod p$ distribute uniformly over Z_p^* and this means $rk_1 = \frac{\alpha+k_1}{ID\alpha+t_2} \mod p$ can not help adversary to find $ID\alpha + t_2$.

4.4 Security Analysis

Theorem 6. *Suppose the mDBDH assumption holds, then our scheme is IBE-IND-sID-CPA secure for the proxy and delegatee's colluding.*

Proof. See appendix F.

Theorem 7. *Our scheme is CBE-LV2-IND-CPA and CBE-LV1-IND-CPA secure for the proxy, delegator and* PKG*'s colluding.*

Proof. See appendix G.

Theorem 8. *Suppose the mDBDH assumption holds, then our scheme is PKG-OW secure for the proxy, delegatee and delegator's colluding.*

Proof. See appendix H.

5 IBPRE Based on a Variant of $\mathbf{BB_1}$ IBE

5.1 Our Definition for IBPRE

In this section, we give our definition and security model for identity based PRE scheme, which is based on [18,34].

Definition 14. *An identity based* PRE *scheme is tuple of algorithms* (Setup, KeyGen, Encrypt, Decrypt, RKGen, Reencrypt):

- Setup(1^k). *On input a security parameter, the algorithm outputs both the master public parameters which are distributed to users, and the master secret key (msk) which is kept private.*
- KeyGen(params, msk, ID). *On input an identity $ID \in \{0,1\}^*$ and the master secret key, outputs a decryption key sk_{ID} corresponding to that identity.*

- Encrypt(params, ID, m). *On input a set of public parameters, an identity* $ID \in \{0,1\}^*$ *and a plaintext* $m \in M$*, output* c_{ID}*, the encryption of* m *under the specified identity.*
- RKGen(params, msk, sk_{ID_1}, sk_{ID_2}, ID_1, ID_2). *On input secret keys* msk, sk_{ID_1}, sk_{ID_2}, *and identities* $ID \in \{0,1\}^*$, PKG, *the delegator and the delegatee interactively generat the re-encryption key* $rk_{ID_1 \to ID_2}$*, the algorithm output it.*
- Reencrypt(params, $rk_{ID_1 \to ID_2}$, c_{ID_1}). *On input a ciphertext* c_{ID_1} *under identity* ID_1*, and a re-encryption key* $rk_{ID_1 \to ID_2}$*, outputs a re-encrypted ciphertext* c_{ID_2}.
- Decrypt(params, sk_{ID}, c_{ID}). *Decrypts the ciphertext* c_{ID} *using the secret key* sk_{ID}*, and outputs* m *or* $\perp$.

Remark 10. This definition is different from the Definitions 49 which come from the work of [29]. We insist this is a more natural and general Definition for PRE from IBE to IBE. This definition is consistent with the work of [18,34].

5.2 Our Security Models for IBPRE

In PRE from IBE to IBE, there is no necessary to consider the malicious PKG attack, so we omit PKG in our security model when considering delegator security and delegatee security.

Delegator Security.

In PRE from IBE to IBE, we consider the case that proxy and delegatee are corrupted.

Definition 15. (DGA-IBE-IND-ID-CPA) *A* PRE *scheme from* IBE *to* IBE *is DGA*[7]*-IBE-IND-ID-CPA secure if the probability*

$$\begin{aligned} Pr[\{(ID^\star, sk_{ID^\star}) \leftarrow KeyGen(\cdot)\}\{(ID_x, sk_{ID_x}) \leftarrow KeyGen(\cdot)\}, \{(ID_h, sk_{ID_h}) \leftarrow KeyGen(\cdot)\},\\ \{R_{hx} \leftarrow RKGen(msk, sk_{ID_h}, sk_{ID_x}, \cdot)\}, \{R_{xh} \leftarrow RKGen(msk, sk_{ID_x}, sk_{ID_h}, \cdot)\},\\ \{R_{hh} \leftarrow RKGen(msk, sk_{ID_h}, sk_{ID_h}, \cdot)\}, \{R_{xx} \leftarrow RKGen(msk, sk_{ID_x}, sk_{ID_x}, \cdot)\},\\ \{R_{\star h} \leftarrow RKGen(msk, sk_{ID^\star}, sk_{ID_h}, \cdot)\}, \{R_{\star x} \leftarrow RKGen(msk, sk_{ID^\star}, sk_{ID_x}, \cdot)\},\\ (m_0, m_1, St) \leftarrow A^{O_{renc}}(ID^\star, \{sk_{ID_x}\}, \{R_{xh}\}, \{R_{hx}\}, \{R_{hh}\}, \{R_{xx}\}, \{R_{\star h}\}, \{R_{\star x}\}),\\ d^\star \xleftarrow{R} \{0,1\}, C^\star = Encrypt(m_{d^\star}, ID^\star), d' \leftarrow A^{\emptyset_{renc}}(C^\star, St) : d' = d^\star] \end{aligned}$$

is negligibly close to $1/2$ *for any PPT adversary A. The notations in this game are same as Definition 5.*

Delegatee Security.

In PRE from IBE to IBE, we consider the case that proxy and delegator are corrupted.

Definition 16. (DGE-IBE-IND-ID-CPA) *A* PRE *scheme from* IBE *to* IBE *is DGE*[8]*-IBE-IND-ID-CPA secure if the probability*

$$\begin{aligned} Pr[\{(ID^\star, sk_{ID^\star}) \leftarrow KeyGen(\cdot)\}\{(ID_x, sk_{ID_x}) \leftarrow KeyGen(\cdot)\}, \{(ID_h, sk_{ID_h}) \leftarrow KeyGen(\cdot)\},\\ \{R_{hx} \leftarrow RKGen(msk, sk_{ID_h}, sk_{ID_x}, \cdot)\}, \{R_{xh} \leftarrow RKGen(msk, sk_{ID_x}, sk_{ID_h}, \cdot)\}, \end{aligned}$$

[7] DGA means Delegator
[8] DGE means Delegatee.

$$\{R_{hh} \leftarrow RKGen(msk, sk_{ID_h}, sk_{ID_h}, \cdot)\}, \{R_{xx} \leftarrow RKGen(msk, sk_{ID_x}, sk_{ID_x}, \cdot)\},$$
$$\{R_{h\star} \leftarrow RKGen(msk, sk_{ID_h}, sk_{ID^\star}, \cdot)\}, \{R_{x\star} \leftarrow RKGen(msk, sk_{ID_x}, sk_{ID^\star}, \cdot)\},$$
$$(m_0, m_1, St) \leftarrow A^{O_{renc}}(ID^\star, \{sk_{ID_x}\}, \{R_{xh}\}, \{R_{hx}\}, \{R_{hh}\}, \{R_{xx}\}, \{R_{h\star}\}, \{R_{x\star}\}),$$
$$d^\star \xleftarrow{R} \{0,1\}, C^\star = Encrypt(m_{d^\star}, ID^\star), d' \leftarrow A^{\emptyset_{renc}}(C^\star, St) : d' = d^\star]$$

is negligibly close to $1/2$ *for any PPT adversary A. The notations in this game are same as Definition 5.*

PKG Security.

In PRE from IBE and IBE, PKG's master key can not leverage even if the delegator, the delegatee and proxy collude.

Definition 17. (PKG-OW) *A* PRE *scheme from* IBE *to* IBE *is one way secure for* PKG *if the probability*

$$Pr[\{(ID_x, sk_{ID_x}) \leftarrow KeyGen(\cdot)\}, \{(ID_h, sk_{ID_h}) \leftarrow KeyGen(\cdot)\},$$
$$\{R_{hx} \leftarrow RKGen(msk, sk_{ID_h}, sk_{ID_x}, \cdot)\}, \{R_{xh} \leftarrow RKGen(msk, sk_{ID_x}, sk_{ID_h}, \cdot)\},$$
$$\{R_{hh} \leftarrow RKGen(msk, sk_{ID_h}, sk_{ID_h}, \cdot)\}, \{R_{xx} \leftarrow RKGen(msk, sk_{ID_x}, sk_{ID_x}, \cdot)\},$$
$$mk' \leftarrow A^{O_{renc}}(\{sk_{ID_x}\}, \{sk_{ID_h}\}, \{R_{xh}\}, \{R_{hx}\}, \{R_{hh}\}, \{R_{xx}\}, \{parms\}) : mk = mk']$$

is negligibly close to 0 *for any PPT adversary A. The notations in this game are same as Definition 5.*

5.3 Our Proposed IND-Pr-sID-CPA Secure IBPRE Scheme Based on a Variant of BB_1 IBE

- The underlying IBE scheme: We give a variant of BB_1-IBE scheme as follows:
  Let G be a bilinear group of prime order p (the security parameter determines the size of G). Let $e : G \times G \rightarrow G_1$ be the bilinear map. For now, we assume public keys (ID) is element in Z_p^*. We later extend the construction to public keys over $\{0,1\}^*$ by first hashing ID using a collision resistant hash $H : \{0,1\}^* \rightarrow Z_p$. We also assume messages to be encrypted are elements in G. The IBE system works as follows:
  1. $\mathbf{SetUp_{IBE}(k)}$. Given a security parameter k, select a random generator $g \in G$ and random elements $g_2 = g^{t_1}, h = g^{t_2} \in G$. Pick a random $\alpha \in Z_p^*$. Set $g_1 = g^\alpha$,$mk = g_2^\alpha$, and $params = (g, g_1, g_2, h)$. Let mk be the master-secret key and let $params$ be the public parameters.
  2. $\mathbf{KeyGen_{IBE}(mk, params, ID)}$. Given $mk = g_2^\alpha$ and ID with $params$, the PKG picks random $s_0, s_1 \in Z_p^*$, choose a hash function $\widetilde{H} : \mathbb{Z}_p^* \times \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ and computes $u_0 = \widetilde{H}(s_0, ID)$, $u_1 = \widetilde{H}(s_1, ID)$. Set $sk_{ID} = (d_0, d_1, d_0') = (g_2^\alpha (g_1^{ID} h)^{u_0}, g^{u_0}, (g_2^\alpha (g_1^{ID} h)^{u_1}))$. The PKG preserves (s_0, s_1).
  3. $\mathbf{Enc_{IBE}(ID, params, M)}$. To encrypt a message $M \in G_1$ under the public key $ID \in Z_p^*$, pick a random $r \in Z_p^*$ and compute $C_{ID} = (g^r, (g_1^{ID} h)^r, Me(g_1, g_2)^r)$.
  4. $\mathbf{Dec_{IBE}(sk_{ID}, params, C_{ID})}$. Given ciphertext $C_{ID} = (C_1, C_2, C_3)$ and the secret key $sk_{ID} = (d_0, d_1)$ with $prams$, compute $M = \frac{C_3 e(d_1, C_2)}{e(d_0, C_1)}$.
- The delegation scheme:

1. $\mathbf{KeyGen_{PRO}}(\mathbf{sk_R}, \mathbf{params}, \mathbf{ID}, \mathbf{ID'})$. The PKG computes $u_1' = \widetilde{H}(s_1, ID')$ and randomly selects $k_1, k_2, k_3 \in Z_p^*$ and sets $rk_{ID\to ID'} = (rk_1, rk_2, rk_3, rk_4) = (\frac{\alpha ID'+t_2+k_1}{k_3(\alpha ID+t_2)} + k_2, g^{u_1' k_3}, g^{u_1' k_2 k_3}, g^{u_1' k_1})$ and sends them to the proxy via secure channel. We must note that the PKG computes a different (k_1, k_2, k_3) for every different user pair (ID, ID').
2. $\mathbf{Check}(\mathbf{params}, \mathbf{C_{ID}}, \mathbf{ID})$. Given the delegator's identity ID and $C_{ID} = (C_1, C_2, C_3)$ with $params$, compute $v_0 = e(C_1, g_1^{ID} h)$ and $v_1 = e(C_2, g)$. If $v_0 = v_1$ then output 1. Otherwise output 0.
3. $\mathbf{ReEnc}(\mathbf{rk_{ID\to ID'}}, \mathbf{params}, \mathbf{C_{ID}}, \mathbf{ID'})$. Given the identities ID, ID', $rk_{ID\to ID'} = (rk_1, rk_2, rk_3, rk_4) = (\frac{\alpha ID'+t_2+k_1}{k_3(\alpha ID+t_2)} + k_2, g^{u_1' k_3}, g^{u_1' k_2 k_3}, g^{u_1' k_1})$ with $params$, the proxy re-encrypt the ciphertext C_{ID} into $C_{ID'}$ as follows. First it runs "Check", if output 0, then return "Reject". Else computes $C_{2ID'} = (C_1', C_2', C_3', C_4', C_5', C_6', C_7') = (C_1, C_2, C_3, C_2^{\frac{\alpha ID'+t_2+k_1}{k'(\alpha ID+t_2)}+k_2}, rk_2, rk_3, rk_4)$.
4. $\mathbf{Dec1_{IBE}}(\mathbf{sk_{ID'}}, \mathbf{params}, \mathbf{C_{2ID'}})$. Given a re-encrypted ciphertext $C_{2ID'} = (C_1', C_2', C_3', C_4', C_5', C_6', C_7')$ and the secret key $sk_{ID} = (d_0, d_1, d_0')$ with $params$, computes

$$M = \frac{C_3' e(C_5', C_4')}{e(C_2', C_6') e(C_1', C_7') e(d_0', C_1')} = \frac{C_3' e(rk_2, C_4')}{e(C_2', rk_3) e(C_1', rk_4) e(d_0', C_1')}$$

5. $\mathbf{Dec2_{IBE}}(\mathbf{sk_{ID'}}, \mathbf{params}, \mathbf{C_{1ID'}})$. Given a normal ciphertext $C_{ID'} = (C_1, C_2, C_3)$ and the secret key $sk_{ID'} = (d_0, d_1, d_0')$ with $prams$, compute $M = \frac{C_3 e(d_1, C_2)}{e(d_0, C_1)}$.

We can verify its correctness as following

$$\frac{C_3' e(rk_2, C_4')}{e(C_2', rk_3) e(C_1', rk_4) e(d_0', C_1')} = \frac{M e(g_1, g_2)^r e(g^{k_3 u_1'}, (g_1^{ID} h)^{r(\frac{\alpha ID'+t_2+k_1}{k_3(\alpha ID+t_2)}+k_2)})}{e((g_1^{ID} h)^r, g^{u_1' k_2 k_3}) e(g^r, g^{k_1 u_1'}) e(g_2^{\alpha} (g_1^{ID'} h)^{u_1'}, g^r)}$$

$$= \frac{M e(g_1, g_2)^r e(g^{k_3 u_1'}, (g_1^{ID} h)^{k_2 r}) e(g^{k_3 u_1'}, (g_1^{ID'} h)^{\frac{r}{k_3}}) e(g^{k_3 u_1'}, g^{\frac{k_1 r}{k_3}})}{e((g_1^{ID} h)^r, g^{u_1' k_2 k_3}) e(g^r, g^{k_1 u_1'}) e(g_2^{\alpha} (g_1^{ID'} h)^{u_1'}, g^r)} = \frac{M e(g_1, g_2)^r}{e(g_2^{\alpha}, g^r)} = M$$

Remark 11. In our scheme, we must note that the PKG computes a different (k_1, k_2, k_3) for every different pair (ID, ID'). Otherwise, if the adversary knows $\frac{\alpha ID'+t_2+k_1}{k_3(\alpha ID+t_2)} + k_2$ for five different pairs (ID, ID') but the same $k_1, k_2, k_3, \alpha, t_2$, he can compute (α, t_2), which is not secure at all.

5.4 Security Analysis

Theorem 9. *Suppose the DBDH assumption holds, then our scheme proposed in Section 5.3 is* DGA-IBE-IND-sID-CPA *secure for the proxy and the delegatee's colluding.*

Proof. See appendix I.

Theorem 10. *Suppose the DBDH assumption holds, then our scheme proposed in Section 5.3 is* DGE-IBE-IND-sID-CPA *secure for the delegator and proxy's colluding.*

Proof. See appendix J.

Theorem 11. *Suppose the DBDH assumption holds, then our scheme proposed in Section 5.3 is* PKG-OW *secure for the delegator, delegatee and proxy's colluding.*

Proof. See appendix K.

5.5 Toward Chosen Ciphertext Security

As we all know, just considering IND-sID-CPA security is not enough for many applications. We consider construct IND-Pr-ID-CCA secure IBPRE based on a variant of BB_1 IBE. There are two ways to construct IND-Pr-ID-CCA secure IBPRE. One way is considering CHK transformation to hierarchal variant of BB_1 IBE to get IND-Pr-sID-CCA secure IBPRE or get IND-Pr-IDKEM-CCA secure IBPRE. The other way is considering variant of BB_1 IBE in the random oracle model. From a practical viewpoint, we construct an IND-Pr-ID-CCA secure IBPRE based on a variant of BB_1 IBE in the random oracle model.

5.6 Our Proposed IND-Pr-ID-CCA Secure IBPRE Scheme Based on a Variant of BB_1 IBE

Let G be a bilinear group of prime order p(the security parameter determines the size of G). Let $e : G \times G \to G_1$ be the bilinear map. Identities are represented using distinct arbitrary bit strings in $\{0,1\}^l$. The messages (or session keys) are bit strings in $\{0,1\}^l$ of some fixed length l. We require the availability of five hash functions viewed as random oracles:

- A hash function $H_1 : \{0,1\}^* \to Z_q^*$;
- A hash function $H_2 : G_1 \times \{0,1\}^l \to G$;
- A hash function $H_3 : G_1 \to \{0,1\}^l$;
- A hash function $H_4 : \{0,1\}^* \times G \times G \times G \times \{0,1\}^l \to G$;

1. **SetUp.** To generate IBE system parameters, first select three integers $\alpha, \beta, \gamma \in Z_p$ at random. Set $g_1 = g^\alpha$, $g_2 = g^{t_1}$ and $h = g^{t_2}$ in G, and compute $v_0 = e(g,g)^{\alpha\beta}$. The public system parameters *params* and the *masterkey* are given by: $params = (g, g_1, g_3, v_0)$, $masterkey = (\alpha, \beta, \gamma)$. Strictly speaking, the generator need not be kept secret, but since it will be used exclusively by the authority, it can be retained in *masterkey* rather than published in *params*.
2. **Extract.** To generate a private key d_{ID} for an identity $ID \in \{0,1\}^*$, using the *masterkey*, the PKG picks random $s_0, s_1 \in Z_p^*$, choose a hash function $\widetilde{H} : \mathbb{Z}_p^* \times \{0,1\}^* \to \mathbb{Z}_p^*$ and computes $u_0 = \widetilde{H}(s_0, ID)$, $u_1 = \widetilde{H}(s_1, ID)$. It outputs: $d_{ID} = (d_0, d_1) = (g_2^\alpha (g_1^{H_2(ID)} h)^{u_0}, g^{u_0}, g_2^\alpha (g_1^{H_2(ID)} h)^{u_1})$. The PKG preserves (s_0, s_1).
3. **Encrypt.** To encrypt a message $M \in \{0,1\}^l$ for a recipient $\{0,1\}^*$, the sender chooses a randomly $\delta \in G$ and computes $s = H_2(\delta, M)$, $k = v_0^s$, $C_1 = g^s$, $C_2 = h^s g_1^{H_1(ID)s}$, $C_3 = \delta \cdot k$, $C_4 = M \oplus H_3(\delta)$, $C_5 = H_4(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4)^s$, and then outputs $C = (C_1, C_2, C_3, C_4, C_5)$.
4. **ReKeyGen.** The PKG computes $u_1' = \widetilde{H}(s_1, ID')$ and randomly selects $k_1, k_2, k_3 \in Z_p^*$, sets $rk_{ID \to ID'} = (\frac{\alpha H_1(ID') + t_2 + k_1}{k_3(\alpha H_1(ID) + t_2)} + k_2, g^{u_1' k_3}, g^{u_1' k_2 k_3}, g^{u_1' k_1})$ and sends it to the proxy via secure channel. We must note that the PKG computes a different (k_1, k_2, k_3) for every different user pair (ID, ID').
5. **ReEnc.** Given the identities (ID, ID'), $rk_{ID \to ID'} = (rk_1, rk_2, rk_3, rk_4) = (\frac{\alpha H_1(ID') + t_2 + k_1}{k_3(\alpha H_1(ID) + t_2)} + k_2, g^{u_1' k_3}, g^{u_1' k_2 k_3}, g^{u_1' k_1})$, $C_{ID} = (C_1, C_2, C_3, C_4, C_5)$ with *params*, the proxy re-encrypts the ciphertext C_{ID} into $C_{ID'}$ as follows.
   (a) First it computes $v_0 = e(C_5, g)$ and $v_1 = e(H_4(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4), C_1)$. If $v_0 \neq v_1$, the ciphertext is rejected.

(b) Else computes $C_{ID'} = (C'_1, C'_2, C'_3, C'_4, C'_5, C'_6, C'_7, C'_8) = (C_1, C_2, C_3, C_2^{rk_1}, rk_2, rk_3, rk_4, C_4)$.

6. **Decrypt.**
   (a) To decrypt a normal ciphertext $C = (C_1, C_2, C_3, C_4, C_5)$ using the private key $d_{ID} = (d_0, d_1, d'_0)$, it computes $v_0 = e(C_5, g)$ and $v_1 = e(H_4(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4), C_1)$. If $v_0 \neq v_1$, the ciphertext is rejected. The recipient computes $k = \frac{e(C_1,d_0)}{e(C_2,d_1)}$. It then computes $\delta = \frac{C_3}{k}$, $M = H_4(\delta) \oplus C_4$. It computes $s' = H_2(\delta, M)$ and verifies that $C_1 = g^{s'}$, $C_2 = h^{s'} g_1^{H_1(ID)s'}$, if either checks fails, returns $\perp$, otherwise returns M.
   (b) To decrypt a re-encrypted ciphertext $C_{ID'} = (C'_1, C'_2, C'_3, C'_4, C'_5, C'_6, C'_7, C'_8)$ using the private key $d_{ID} = (d_0, d_1, d'_0)$, the recipient computes $k = \frac{C'_3 e(C'_5, C'_4)}{e(C'_2, C'_6) e(C'_1, C'_7) e(d'_0, C'_1)} = \frac{C'_3 e(rk_2, C'_4)}{e(C'_2, rk_3) e(C'_1, rk_4) e(d'_0, C'_1)}$. It then computes $\delta = \frac{C_3}{k}$, $M = H_3(\delta) \oplus C'_8$. It computes $s' = H(\delta, M)$ and verifies that $C_1 = g^{s'}$, $C_2 = h^{s'} g_1^{H_1(ID)s'}$, if either check fails, returns $\perp$, otherwise returns M.

5.7 Security Analysis

Theorem 12. *Suppose the DBDH assumption holds, then our scheme proposed in Section 5.6 is* DGA-IBE-IND-ID-CCA *secure for the proxy and delegatee's colluding.*

Proof. See appendix L.

Theorem 13. *Suppose the DBDH assumption holds, then our scheme proposed in Section 5.6 is* DGE-IBE-IND-ID-CCA *secure for the delegator and proxy's colluding.*

Proof. See appendix M.

Theorem 14. *Suppose the DBDH assumption holds, then our scheme proposed in Section 5.6 is* PKG-OW *secure for the delegator, proxy and delegatee's colluding.*

Proof. See appendix N.

6 IBPRE Based on BB_2 IBE

6.1 Review of the BB_2 Identity Based Encryption

Let $\mathbb{G}$ be a bilinear group of prime order p and g be a generator of $\mathbb{G}$. For now, we assume that the public keys (ID) are elements in Z_p^*. We show later that arbitrary identities in $\{0,1\}^*$ can be used by first hashing ID using a collision resistant hash $H : \{0,1\}^* \rightarrow Z_p^*$. We also assume that the messages to be encrypted are elements in $\mathbb{G}_1$. The IBE system works as follows:

1. **Setup:** To generate IBE parameters, select random elements $(x, y) \in Z_p^*$ and define $X = g^x$ and $Y = g^y$. The public parameters parms and the secret master − key are given by parms $= (g, g^x, g^y)$, master − key $= (x, y)$
2. **KeyGen(**master − key, **ID):** To create a private key for the public key ID $\in Z_p^*$:
   (a) pick a random $r \in Z_p$ and compute $K = g^{\frac{1}{(ID+x+ry)}} \in \mathbb{G}$,
   (b) output the private key $d_{ID} = (r, K)$. In the unlikely event that $x + ry + ID = 0 \mod p$, try again with a new random value for r.

3. **Encrypt(parms, ID, M):** To encrypt a message $M \in G_1$ under public key $ID \in Z_p^*$, pick a random $s \in Z_p^*$ and output the ciphertext $C = (g^{s \cdot ID} X^s, Y^s, e(g,g)^s \cdot M)$. Note that $e(g,g)$ can be precomputed once and for all so that encryption does not require any pairing computations.
4. **Decrypt(d_{ID}, C):** To decrypt a ciphertext $C = (A, B, C)$ using the private key $d_{ID} = (r, K)$, output $C/e(AB^r, K)$. Indeed, for a valid ciphertext we have

$$\frac{C}{e(AB^r, K)} = \frac{C}{e(g^{s(ID+x+ry)}, g^{1/(ID+x+ry)})} = \frac{C}{e(g,g)^s} = M$$

Remark 12. This scheme is an efficient identity based encryption and proved to be IND-sID-CPA secure in the standard model. In Eurocrypt'06, Gentry proposed a practical identity based encryption based on this scheme which can achieve IND-ID-CCA2 with tight security proof[21]. Thus this scheme plays an important role in the field of identity based encryption.

6.2 Our PRE Scheme Based on BB_2 Identity Based Encryption

1. **ReKeyGen$_{ID \to ID'}$:** PKG chooses a collision resistent hash function $H : \{0,1\}^{3|p|} \to Z_p^*$ and a random seed $t \in Z_p^*$, and computes $k = H(ID, ID', t)$. He computes $rk_{ID \to ID'} = (rk_1, rk_2, rk_3) = (r, \frac{ID'+x+r'y}{ID+x+ry} + k \mod p, g^{\frac{k}{(ID'+x+r'y)}})$ and sends them to the proxy as the re-encryption key. We note that PKG chooses a different k for every different user pair (ID, ID').
2. **Encrypt(parms, ID, M):** Same as the Encrypt algorithm in 6.1.
3. **ReEnc ($rk_{ID \to ID'}$, parms, C_{ID}, ID'):**. On input the ciphertext $\widetilde{C_{ID}} = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3}) = (g^{s \cdot ID} X^s, Y^s, e(g,g)^s \cdot M)$, the proxy computes $\widehat{C_{ID'}} = (\widehat{C_1}, \widehat{C_2}) = (\widetilde{C_1}\widetilde{C_2}^{rk_1}, \widetilde{C_3} e((\widetilde{C_1}\widetilde{C_2}^{rk_1})^{rk_2}, rk_3))$, and sends it to the delegatee.
4. **Decrypt$_1$($\widehat{C_{ID'}}, d_{ID'}$):** On input a re-encrypted ciphertext $\widehat{C_{ID'}} = (\widehat{C_1}, \widehat{C_2})$, the delegatee decrypts like this: $M = \frac{\widehat{C_2}}{e(\widehat{C_1}, {d_{ID'}}^2)} = \frac{\widehat{C_2}}{e(\widehat{C_1}, K')}$ and returns M.
5. **Decrypt$_2$(d_{ID}, C):** On input a normal ciphertext, the delegatee do the same as the Decrypt algorithm in 6.1.
6. **Check:**. On input a ciphertext $\widetilde{C_{ID}} = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3})$, the proxy computes $v_1 = e(\widetilde{C_1}, Y)$ and $v_2 = e(\widetilde{C_2}, g^{ID} X)$, if $v_1 = v_2$, then return "Valid", else return "Invalid".

First we verify our scheme's correctness as following.

$$\frac{\widehat{C_2}}{e(\widehat{C_1}, K')} = \frac{\widetilde{C_3} e(\widetilde{C_1}\widetilde{C_2}^{rk_1}, rk_3)}{e((\widetilde{C_1}\widetilde{C_2}^{rk_1})^{rk_2}, g^{\frac{1}{ID'+x+r'y'}})} = \frac{e(g,g)^s \cdot M \cdot e(g^{s \cdot ID} X^s Y^{sr}, g^{\frac{k}{(ID'+x+r'y)}})}{e((g^{s \cdot ID} X^s Y^{sr})^{\frac{ID'+x+r'y}{ID+x+ry}+k}, g^{\frac{1}{ID'+x+r'y'}})}$$

$$= \frac{e(g,g)^s \cdot M \cdot e(g^{s(ID+x+ry)}, g^{\frac{k}{(ID'+x+r'y)}})}{e(g^{s(ID'+x+r'y)}, g^{\frac{1}{ID'+x+r'y'}}) e(g^{sk(ID+x+ry)}, g^{\frac{1}{ID'+x+r'y'}})} = M$$

Remark 13. In our scheme, we let $rk_1 = r$ which is a part of delegator's secret key. We remark that let r be public should still preserve BB_2 IBE scheme's IND-sID-CPA security.

6.3 Security Analysis

Theorem 15. *Suppose Decision q-BDHI assumption holds in $\mathbb{G}$, then our scheme is DGA-IBE-IND-sID-CPA secure for the proxy and delegatee's colluding.*

Proof. See appendix O.

Theorem 16. *Suppose the q-BDHI assumption holds, then our scheme is DGE-IBE-IND-sID-CPA secure for the proxy and delegator's colluding.*

Proof. See appendix P.

Theorem 17. *Suppose the q-BDHI assumption holds, then our scheme is KGC-OW secure for the proxy, delegatee and delegator's colluding.*

Proof. See appendix Q.

7 IBPRE Based on SK IBE

7.1 Review of the SK Identity Based Encryption

SK-IBE is specified by four polynomial time algorithms:

1. **Setup.** Given a security parameter k, the parameter generator follows the steps.
   (a) Generate three cyclic groups G_1, G_2 and G_T of prime order q, an isomorphism φ from G_2 to G_1, and a bilinear pairing map $e : G_1 \times G_2 \to G_T$. Pick a random generator $P_2 \in G^*$ and set $P_1 = \varphi(P_2)$.
   (b) Pick a random $s \in Z_q^*$ and compute $P_{pub} = sP_1$.
   (c) Pick four cryptographic hash functions $H_1 : \{0,1\}^* \to Z_q^*, H_2 : G_T \to \{0,1\}^n, H_3 : \{0,1\}^n \times \{0,1\}^n \to Z_q^*$ and $H_4 : \{0,1\}^n \to \{0,1\}^n$ for some integer $n > 0$.

   The message space is $M = \{0,1\}^n$. The ciphertext space is $C = G_1^* \times \{0,1\}^n \times \{0,1\}^n$. The master public key is $M_{pk} = (q, G_1, G_2, G_T, \varphi, e, n, P_1, P_2, P_{pub}, H_1, H_2, H_3, H_4)$, and the master secret key is $M_{sk} = s$.
2. **Extract.** Given an identifier string $ID_A \in \{0,1\}^*$ of identity A, M_{pk} and M_{sk}, the algorithm returns $d_A = \frac{1}{s+H_1(ID_A)} P_2$.
3. **Encrypt.** Given a plaintext $m \in M$, ID_A and M_{pk}, the following steps are performed.
   (a) Pick a random $\sigma \in \{0,1\}^n$ and compute $r = H_3(\sigma, m)$.
   (b) Compute $Q_A = H_1(ID_A)P_1 + P_{pub}$, $g^r = e(P_1, P_2)^r$.
   (c) Set the ciphertext to $C = (rQ_A, \sigma \oplus H_2(g^r), m \oplus H_4(\sigma))$.
4. **Decrypt.** Given a ciphertext $C = (U, V, W) \in \mathcal{C}, ID_A, d_A$ and M_{pk}, follows the steps:
   (a) Compute $g' = e(U, d_A)$ and $\sigma' = V \oplus H_2(g')$.
   (b) Compute $m' = W \oplus H_4(\sigma)$ and $r' = H_3(\sigma', m')$.
   (c) if $U \neq r'(H_1(ID_A)P_1 + P_{pub})$, output $\perp$, else return m' as the plaintext.

7.2 Our Proposed PRE Based On SK Identity Based Encryption

Our proposed PRE scheme based on SK identity based encryption are as following:

1. **Setup.** Same as the above scheme 7.1.
2. **Extract.** Same as the above scheme 7.1.
3. **RKGen:** The PKG chooses a collision resistent hash function $H_5 : \{0,1\}^{3|p|} \rightarrow Z_p^*$ and random seeds $s_2, s_1 \in Z_p^*$, it computes $k_2 = H_5(ID, ID', s_2)$, $k_1 = H_5(ID, ID', s_1)k_2$. He computes $rk_{ID \rightarrow ID'} = (rk_1, rk_2, rk_3) = (\frac{s+H_1(ID')+k_1}{(s+H_1(ID))} \mod p, \frac{k_2}{(H_1(ID)+s)} \mod p, \frac{k_1}{k_2(s+H_1(ID'))}P_2)$.
4. **Encrypt.** Same as the above scheme 7.1.
5. **Reencrypt:**. On input the ciphertext $\widetilde{C_{ID}} = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3}) = (rQ_{ID}, \sigma \oplus H_2(g^r), m \oplus H_4(\sigma))$, the proxy computes $\widehat{C_{ID'}} = (\widehat{C_1'}, \widehat{C_2'}, \widehat{C_3'}, \widehat{C_4'}, \widehat{C_5'}) = (rk_1\widetilde{C_1}, e(rk_2\widetilde{C_1}, rk_3), \widetilde{C_2}, \widetilde{C_3}, \widetilde{C_1})$, and sends it to the delegatee.
6. **Decrypt$_1$.** Given a first level ciphertext - re-encrypted ciphertext $\widehat{C_{ID'}} = (\widehat{C_1'}, \widehat{C_2'}, \widehat{C_3'}, \widehat{C_4'}, \widehat{C_5'})$, follows the steps:
   (a) Compute $g' = \frac{e(\widehat{C_1'}, d_{ID'})}{\widehat{C_2'}}$ and $\sigma' = \widehat{C_3'} \oplus H_2(g')$.
   (b) Compute $m' = \widehat{C_4'} \oplus H_4(\sigma')$ and $r' = H_3(\sigma', m')$.
7. **Decrypt$_2$.** Given a second level ciphertext - normal ciphertext, do the same as the algorithm Decrypt in the above scheme 7.1.
8. **Verify.** If $\widehat{C_5'} \neq r'(H_1(ID)P_1 + P_{pub})$, output $\perp$, else return m' as the plaintext.

First we verify our scheme's correctness as following.

$$
\begin{aligned}
g' &= \frac{e(\widehat{C_1'}, d_{ID'})}{\widehat{C_2'}} = \frac{e(rk_1\widetilde{C_1}, d_{ID'})}{e(rk_2\widetilde{C_1}, rk_3)} = \frac{e(\frac{s+H_1(ID')+k_1}{s+H_1(ID)} \cdot rQ_{ID}, \frac{1}{s+H_1(ID')}P_2)}{e(\frac{k_2}{(H_1(ID)+s)} \cdot rQ_{ID}, \frac{k_1}{k_2(s+H_1(ID'))}P_2)} \\
&= \frac{e(rP_1, P_2)e(rk_1P_1, \frac{1}{s+H_1(ID')}P_2)}{e(rP_1, \frac{k_1}{s+H_1(ID')}P_2)} = e(P_1, P_2)^r = g^r \\
\sigma' &= C_3' \oplus H_2(g') = \sigma \oplus H_2(g^r) \oplus H_2(g^r) = \sigma \\
m' &= \widehat{C_4'} \oplus H_4(\sigma') = m \oplus H_4(\sigma) \oplus H_4(\sigma') = m \oplus H_4(\sigma) \oplus H_4(\sigma) = m, \\
r' &= H_3(\sigma', m') = H_3(\sigma, m) = r \\
\widehat{C_5'} &= \widetilde{C_1} = rQ_{ID} = r(H_1(ID)P_1 + P_{pub}) = r'(H_1(ID)P_1 + P_{pub})
\end{aligned}
$$

Remark 14. In our scheme, we must note that the PKG needs to compute different (k_1, k_2) for every different user pair (ID, ID'). Otherwise, if the adversary know $(\frac{s+H_1(ID')+k_1}{(s+H_1(ID))} \mod p, \frac{k_2}{(H_1(ID)+s)} \mod p)$ for two different pair (ID, ID') but the same (k_1, k_2), he can compute s, which is not secure at all.

Remark 15. In our scheme, we require $k_1 = H_5(ID, ID', s_1)k_2$. The reason of k_2 is a factor of k_1 is just for security proof which can be seen in I.

7.3 Security Analysis

Interestingly, our PRE based on SK IBE scheme even can achieve IND-Pr-ID-CCA2 secure while all the above PRE scheme can only achieve IND-Pr-sID-CPA secure.

Theorem 18. *Suppose q-BDHI assumption holds in $\mathbb{G}$, then our scheme is DGA-IBE-IND-ID-CCA2 secure for the proxy and delegatee's colluding.*

Proof. See appendix R.

Theorem 19. *Suppose q-BDHI assumption holds in $\mathbb{G}$, then our scheme is DGE-IBE-IND-ID-CCA2 secure for the proxy and delegator's colluding.*

Proof. See appendix S.

Theorem 20. *Suppose the q-BDHI assumption holds, then our scheme is PKG-OW secure for the proxy, delegatee and delegator's colluding.*

Proof. See appendix T.

Scheme	Security		W/O RO	Assum	SecMod	Colluding	UnderlyIBE	Remark
GA07A[18]	IND-Pr-ID-CPA ␣		RO	DBDH	Sec.3.1[18]	P and DGA or P and DGE	BF IBE	Weak
GA07B[18]	IND-Pr-ID-CCA ␣		RO	DBDH	Sec.3.1[18]	P and DGA or P and DGE	BF IBE	Strong
M07B [29]	IND-Pr-sID-CPA ␣		Std	DBDH	Sec.4.2[29]	P or DGA or DGE	BB_1 IBE	Weak
CT07[13]	IND-Pr-ID-CPA ␣		Std	DBDH	Sec.4.2[13]	P and DGA or P and DGE	Waters' IBE	Weak
SXC08[32]	IND-Pr-ID-CCA ␣		Std	DBDH	Sec.2.6[32]	P and DGA or P and DGE	Waters' IBE	Strong
OursC5.3	IND-Pr-sID-CPA ␣		Std	DBDH	5.2	P and DGA or P and DGE	Variant of BB_1 IBE	Weak
OursD5.6	IND-Pr-ID-CCA ␣		RO	DBDH	5.2	P and DGA or P and DGE	Variant of BB_1 IBE	Strong
OursE6.2	DGA	DGA-IBE-IND-sID-CPA	Std	q-BDHI	5.2	P and DGE	BB_2 IBE	Weak
OursE6.2	DGE	DGE-IBE-IND-ID-CCA	Std	q-BDHI	5.2	P and DGA	BB_2 IBE	Weak
OursE6.2	PKG	PKG-OW	Std	q-BDHI	5.2	P and DGA and DGE	BB_2 IBE	Strong
OursF7.2	DGA	DGA-IBE-IND-ID-CCA	RO	q-BDHI	5.2	P and DGE	SK IBE	Strong
OursF7.2	DGE	DGA-IBE-IND-ID-CCA	RO	q-BDHI	5.2	P and DGA	SK IBE	Strong
OursF7.2	PKG	PKG-OW	RO	q-BDHI	5.2	P and DGA and DGE	SK IBE	Strong

Table 1. IBPRE Security Comparison

Scheme	Enc	Check	Reenc	Dec		Ciph-Len	
				1stCiph[9]	2-ndCiph	1stCiph	2-ndCiph
GA07A[18][10]	$1t_e + 1t_p$	0	$1t_p$	$2t_p$	$1t_p$	$2\|G\| + 2\|G_e\|$[11]	$1\|G\| + 1\|G_e\|$
GA07B[18]	$1t_p + 1t_e$	$2t_p$	$2t_e + 2t_p$	$1t_e + 2t_p$	$2t_e + 2t_p$	$1\|G\| + 1\|G_e\|$ $+2\|m\| + \|id\|$	$1\|G\| + 1\|G_T\|$ $+1\|G_e\| + \|m\|$
M07B [29]	$1t_p + 2t_e$	$2t_p$	$1t_p$	$2t_p$	$2t_p$	$2\|G_e\| + 1\|G_T\|$	$2\|G_e\| + 1\|G_T\|$
CT07[13]	$3t_e + 1t_p + 1t_s$	$1t_v$	$2t_e$	$2t_e + 10t_p + 1t_v$	$2t_e + 3t_p$	$9\|G\| + 2\|G_T\|$ $+\|vk\| + \|s\|$	$3\|G\| + \|G_T\|$ $+\|vk\| + \|s\|$
SXC08[32]	$3t_e + 1t_p + 1t_s$	$1t_v$	$2t_e + 1t_s$	$2t_e + 10t_p + 2t_v$	$2t_e + 3t_p + 1t_v$	$9\|G\| + 2\|G_T\|$ $+2\|vk\| + 2\|s\|$	$3\|G\| + \|G_T\|$ $+1\|vk\| + 1\|s\|$
OursC5.3	$2t_e + 1t_p$	$2t_p$	$1t_e$	$4t_p$	$2t_p$	$6\|G\| + \|G_T\|$	$2\|G\| + \|G_T\|$
OursD5.6	$3t_e + 1t_{me}$	$2t_p$	$1t_e$	$4t_p + 1t_e + 1t_{me}$	$2t_p + 1t_e + 1t_{me}$	$7\|G\| + m$	$4\|G\| + m$
OursE6.2	$1t_{me} + 1t_e + 1t_p$	$2t_p$	$2t_e$	$1t_p$	$1t_e + 1t_p$	$2\|G_e\| + 1\|G_t\|$	$2\|G_e\| + 1\|G_t\|$
OursF7.2	$3t_e$[12]	$1t_e$	$2t_e + 1t_p$	$1t_p$	$1t_p$	$2\|G_e\| + 1\|G_t\|$ $+2\|n\|$	$2\|G_e\| + 2\|n\|$

Table 2. IBPRE Efficiency Comparison

8 Comparison

In this section, we give our comparison results with other identity based proxy re-encryption schemes [18,13,29,32] or hybrid proxy re-encryption schemes [29]. We compare our schemes with other schemes from two ways. First we concern about schemes' security, then we concern about schemes' efficiency.

Notations: In Table 1,3 we denote with/without random oracle as W/O RO, assumption as Assum, security model as SecMod, colluding attackers as Colluding, underlying IBE as UnderIBE, stand model as Std, , proxy as P, DGA as delegator, DGE as delegatee. P and DGA means that proxy colludes with delegator, P or DGA means that proxy or delegator is malicious adversary but they never collude. SymEnc-Sec means the security of symmetric encryption, CBE-ciph-no-re-encrypted means CBE ciphertext having not been re-encrypted, CBE-ciph-re-encrypted means the CBE ciphertext having been re-encrypted

From Table 1, we can know that our IBPRE scheme based on a variant of BB_1 IBE, IBPRE scheme based on SK IBE and SXC08 scheme are the most secure IBPRE. M07B scheme is the weakest IBPRE for it can only achieve IND-Pr-sID-CPA under separated proxy or delegator or delegatee attack.

In Table 2,4, we denote encryption as Enc, re-encryption as Reenc, decryption as Dec, ciphertext as Ciph and ciphertext length as Ciph-Len, resisting malicious PKG attack as ReMal.

[9] Our first level ciphertext maps second level ciphertext and second level ciphertext maps first level ciphertext in [18,13,32].

[10] GA07 and SXC08 are multi-hop IBPRE but we just consider their single-hop variant.

[11] Sometimes in our schemes we use $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ or $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, in the former cases, $\mathbb{G}$ maps to $\mathbb{G}_e$, $\mathbb{G}_1$ maps $\mathbb{G}_T$, in the latter case, $\mathbb{G}_1$ maps to $\mathbb{G}_e$, $\mathbb{G}_T$ maps $\mathbb{G}_T$.

[12] In SK IBE we can precomputation $e(P_1, P_2)$, so there is no paring computation in Encryption.

t_p, t_e and t_{me} represent the computational cost of a bilinear pairing, an exponentiation and a multi-exponentiation respectively, while t_{s} and t_{v} represent the computational cost of a one-time signature signing and verification respectively. $|\mathbb{G}|$, $|\mathbb{Z}_q|$, $|\mathbb{G}_e|$ and $|\mathbb{G}_T|$ denote the bit -length of an element in groups $\mathbb{G}$, $\mathbb{Z}_q$, $\mathbb{G}_e$ and $\mathbb{G}_T$ respectively. Here $\mathbb{G}$ and $\mathbb{Z}_q$ denote the groups used in our scheme, while $\mathbb{G}_e$ and $\mathbb{G}_T$ are the bilinear groups used in GA07, CT07, SXC08 schemes, i.e., the bilinear pairing is $e : \mathbb{G}_e \times \mathbb{G}_e \rightarrow \mathbb{G}_T$. Finally, $|vk|$ and $|s|$ denote the bit length of the one-time signature's public key and a one-time signature respectively.

Scheme	Security		W/O RO	Assum	SecMod	Colluding	UnderlyIBE	Remark
M07A [29]	IND-Pr-ID-CPA		Std	DBDH	Sec.3.4[29]	P or DGA or DGE	BB_1 IBE	Weak
OursA3.4	DGA	CBE-IND-CPA CBE-ciph-no-re-encrypted	Std	DBDH	3.2	P and DGE and PKG	BB_1 IBE	Weak
OursA3.4	DGA	No CBE-IND-CPA CBE-ciph-re-encrypted	␣	␣	3.2	P and DGE and PKG	BB_1 IBE	Weak
OursA3.4	DGE	IBE-LV2-IND-sID-CPA	Std	DBDH	3.2	P and DGA	BB_1 IBE	Weak
OursA3.4	DGE	IBE-LV1-IND-sID-CPA	Std	SymEnc-Sec	3.2	P and PKG and DGA	BB_1 IBE	Weak
OursA3.4	PKG	PKG-OW	Std	DBDH	3.2	P and DGA and DGE	BB_1 IBE	Strong
OursB4.3	DGA	IBE-IND-sID-CPA	Std	mDBDH	4.2	P and DGE	BB_1 IBE	Weak
OursB4.3	DGE	CBE-LV1-IND-CPA	Std	SymEnc-Sec	4.2	P and DGA and PKG	BB_1 IBE	Weak
OursB4.3	DGE	CBE-LV2-IND-CPA	Std	SymEnc-Sec	4.2	P and DGA and PKG	BB_1 IBE	Weak
OursB4.3	PKG	PKG-OW	Std	mDBDH	4.2	P and DGA and DGE	BB_1 IBE	Strong

Table 3. Hybrid PRE Security Comparison

Scheme	Type	$\mathsf{Enc}_{\mathsf{CBE}}$	$\mathsf{Enc}_{\mathsf{IBE}}$	Check	Reenc	Dec		Ciph-Len		ReMal
						1stCiph	2-ndCiph	1stCiph	2-ndCiph	
M07A [29]	$\mathsf{CBE} \rightarrow \mathsf{IBE}$	$3t_e + 1t_p$	$1t_p + 2t_e$	$4t_p$	$2t_e + 1t_p$	$2t_p$	$2t_p$	$2\|G_e\| + 1\|G_T\|$	$2\|G_e\| + 1\|G_T\|$	NO
OursA 3.4	$\mathsf{CBE} \rightarrow \mathsf{IBE}$	$3t_e + 1t_p$	$1t_p + 2t_e$	$4t_p$	$2t_e + 1t_p$	$4t_e + 1t_p$	$2t_p$	$3\|G_e\| + 1\|G_T\|$	$2\|G_e\| + 1\|G_T\|$	YES
OursB 4.3	$\mathsf{IBE} \rightarrow \mathsf{CBE}$	$2t_e + 1t_p$	$1t_p + 1t_e$	$2t_p$	$1t_e + 1t_p$	$1t_e + 1t_p$	$1t_e + 1t_p$	$1\|G_e\| + 1\|G_T\|$	$1\|G_e\| + 1\|G_T\|$	␣

Table 4. Hybrid PRE Efficiency Comparison

From Table 2, we can know that our IBPRE scheme based on SK IBE is the most efficient IBPRE for its only totally 3 pairing computation and relative shorter first level and second level ciphertext. Our schemes, GA07 and M07B schemes are much more efficient than CT07 and SXC08

scheme due to their underlying IBE is Waters' IBE. And for the proxy, CT07 and SXC08 scheme are much more efficient than others for their special paradigm, our IBPRE scheme based on SK IBE is more efficient than GA07B scheme and our other schemes, we think this is important for resisting DDos attack against the proxy.

From the above discussion, we conclude that our IBPRE scheme based on SK IBE is the most secure and efficiency IBPRE among existing IBPRE schemes.

From Table 3 and Table 4, we can know that the security models of our PRE from CBE to IBE and PRE from IBE to CBE schemes are stronger than the security model of M07A scheme. Thus our schemes are more secure than M07A scheme. We construct the first PRE from CBE to IBE which can resist malicious PKG attack. But we note that our scheme needs to add one more secret key k to the delegatee, and that neither of our schemes and M07A scheme can achieve IND-Pr-ID-CCA security, which is our further work.

9 Issues about PKG's Workload in Our Proposed Schemes

One core idea in our proposed schemes(except the first scheme) is that, PKG itself generates every delegation key -the re-encryption key. This idea looks first contradict with our intuition about PKG(That is, what PKG can only do is generating IBE user's secret key) and increases PKG's workload. But we think our idea is reasonable.

1. From a theoretical point, the idea about PKG generating re-encryption key comes from Matsuo's M2 proxy re-encryption[29]. In their scheme, $rk_{\mathsf{ID}\rightarrow\mathsf{ID}'} = g^{u'\alpha}$ is generated by exponentiating delegatee's secret key $g^{u'}$ with $\mathsf{master-key}$ α. Later in Inscrypt'08, Tang et al. proposed an inter-domain identity based proxy re-encryption [34]. In their scheme, generating the re-encryption key needs PKG. We quote it as follows:

   > $\mathsf{Pextract}(id, id', sk_{id}(sk_{id'}, mk_1, mk_2))$*: This algorithm takes the delegator's identifier* id*, the delegatee's identifier* id'*, the delegator's private key* sk_{id}*, and possibly also* $\{sk_{id'}, mk_1, mk_2\}$ *as input and outputs the proxy key* $rk_{id\rightarrow id'}$ *to the proxy. This algorithm will be run by the delegator and possibly with other parties, such as the delegatee and KGCs.*

   Furthermore, it seems difficult for constructing PRE in identity based setting which just needs the delegator and the delegatee to generate re-encryption key.
2. From a practical point, generating re-encryption key by PKG can make PRE in identity based setting much efficient for the proxy, which is important for practical IBE systems. Furthermore, many practical IBE systems let their PKG be online 24/7/365[16], which results PKG generating re-encryption key is tolerable for these systems.

10 Conclusions and Open Problems

In 2007, Matsuo proposed the concept of four types of PRE schemes: CBE to CBE, IBE to CBE, CBE to IBE and IBE to IBE [29]. Now CBE to IBE and IBE to IBE proxy re-encryption schemes are being standardized by IEEEP1363.3 working group[31]. We extend their research, we solved the key escrow problem of their PRE scheme from CBE to IBE. In Matsuo's scheme, they allow the PKG to help the delegator and the delegatee to generate re-encryption key. We explore this

feature further, if we allow PKG to generate re-encryption keys by directly using master − key, many open problems can be solved, such as constructing PRE from IBE to CBE, PRE based on a variant of BB_1 IBE, PRE based on BB_2 IBE and PRE based on SK IBE.

Although some excellent work[12,13,18,24,27,28,29,32,34] has been done in PRE in identity based setting, there are still many open problems need to be solved such as:

1. More reasonable security models for IBPRE and PRE. We note that our security model is stronger than security model in [29] for we considering colluding between proxy and delegator or delegatee. But we must point out that our security model just consider single-hop PRE, security models for multi-hop PRE maybe be different.
2. More stronger security results for our PRE scheme. We note most of our schemes can only achieve IND-Pr-ID-CPA secure, which is not enough for most applications.
3. More interesting applications for IBPRE and PRE. From a theoretical point, Obfuscating PRE is the only positive results for obfuscation of natural cryptographic tasks, maybe this primitive can find other applications in oblivious transfer, multi-party computation in common reference string model etc. From a practical point, PRE can bridge between different types of encryption just as Matsuo's PRE from CBE to IBE[29]. This feature is important to our life, which can help ciphertexts circulate smoothly in the network. PRE can have other applications in e-mail forwarding, law enforcement, mobile equipment with limited computation ability, access control in secure distributed file storage. But IBPRE maybe have other interesting applications such as anonymous encryption, group encryption, one to many, many to one IBPRE [13] and even identity based broadcast encryption.

Acknowledgement

The authors would like to express their gratitude thanks to Dr. Jian Weng and Dr. Wei Wu's many helpful discussions, Dr. Qiang Tang and Dr. Jun Shao's many helpful comments.

References

1. S. S. Al-Riyami and K. Paterson. Certificateless public key cryptography. In *Advances in Cryptology, Proc. ASIACRYPT 2003,* LNCS 2894, pages 452–473. Springer–Verlag, 2003.
2. G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage.In *ACM Trans. Inf. Syst. Secur. 9 (2006), no. 1, pages 1–30.*
3. M. Blaze, G. Bleumer, M. Strauss, Divertible Protocols and Atomic Proxy Cryptography.In *Advances in Cryptology - Eurocrypt'98,* LNCS 1403, pp. 127–144.Springer–Verlag,1998.
4. D. Boneh, E. Goh, T. Matsuo.Proposal for P1363.3 Proxy Re-encryption.http://grouper.ieee.org/groups/1363/IBC /submissions/NTTDataProposal-for-P1363.3-2006-09-01.pdf.
5. D. Boneh and M. Franklin. Identity-based Encryption from the Weil Pairing. In *Advances in Cryptology - CRYPTO 2001,* LNCS 2139, pp. 213–229.Springer–Verlag,2001.
6. D. Boneh and X.Boyen. Efficient Selective-id Secure Identity Based Encryption without Random Oracles. In *Advances in Cryptology - EUROCRYPT 2004,* LNCS 3027, pp. 223–238. Springer–Verlag, 2004.
7. D. Boneh and X.Boyen. Secure Identity Based Encryption without Rando Oracles. In *Advances in Cryptology - CRYPTO 2004,* LNCS 3152, pp. 443–459. Springer–Verlag, 2004.
8. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil Pairing. In *Advances in Cryptology-ASIACRYPT2001*, pp.514–532. Springer–Verlag, 2001
9. X.Boyen. An introduction to Identity Based Encryption. In *International Lecture Series on Pairings*, Brisbane, 2007.

[13] Actually, Matsuo's M07B PRE is a many to one IBPRE.

10. M.Barbosa, L.Chen,Z.Cheng et al.SK−KEM:An Identity−based Kem.http://grouper.ieee.org/groups/1363/IBC/submissions/Barbosa-SK-KEM-2006-06.pdf.
11. R.Canetti, S.Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology - Eurocrypt'03,* LNCS 2656, pp. 255-271. Springer–Verlag,2003.
12. R. Canetti and S. Hohenberger, Chosen Ciphertext Secure Proxy Re-encryption.In *In Proceedings of the 14th ACM conference on Computer and Communications Security(CCS 2007)*,pp. 185–194.2007. Also available at Cryptology ePrint Archive: http://eprint.iacr.org/2007/171.pdf.
13. C.Chu and W.Tzeng. Identity-based Proxy Re-encryption without random oracles. In *ISC 2007*,LNCS 4779, pp. 189–202.Springer–Verlag,2007.
14. L.Chen and Z.Cheng. Security Proof of Sakai-Kasahara's Identity-Based Encryption Scheme. http://eprint.iacr.org/2005 /226.pdf, 2005.
15. Y. Dodis and A. Ivan. Proxy cryptography revisited.In *NDSS '03*,2003.
16. A. Dancer (CTO, Encryption Group, Trend Micro). Personal communication.
17. E.Fujisaki and T.Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology - CRYPTO 1999,* LNCS 1666, pp. 535–554. Springer–Verlag, 1999.
18. M.Green and G. Ateniese, Identity-Based Proxy Re-encryption. In *Applied Cryptography and Network Security'07*,LNCS 4521, pp. 288–306.Springer–Verlag,2007.
19. V.Goyal. Reducing Trust in Identity Based Cryptosystems.In *Advances in Cryptology - CRYPTO 2007,* LNCS 4622, pp. 430–447.Springer–Verlag,2007.
20. C.Gentry.Certificate-based encryption and the certificate revocation problem.In *Advances in Cryptology - EUCRYPT 2003,* LNCS 2656, pp. 272–293. Springer–Verlag, 2003.
21. C.Gentry.Practical Identity-Based Encryption without Random Oracles. In In *Advances in Cryptology - EUCRYPT 2006,* LNCS 4004, pp. 445–464. Springer–Verlag, 2006.
22. E. Goh and T. Matsuo.Proposal for P1363.3 Proxy Re-encryption.http://grouper.ieee.org/groups/1363/IBC/submissions/NTTDataProposal-for-P1363.3-2006-08-14.pdf.
23. S. Hohenberger. Advances in Signatures, Encryption, and E-Cash from Bilinear Groups. Ph.D. Thesis, MIT, May 2006.
24. S. Hohenberger, G. N. Rothblum, a. shelat, V. Vaikuntanathan. Securely Obfuscating Re-encryption.In *TCC'07*,LNCS 4392, pp. 233–252.Springer–Verlag,2007.
25. M. Jakobsson. On quorum controlled asymmetric Proxy Re-encryption.In *PKC '99*,pages 112–121,Springer–Verlag,1999.
26. L Ibraimi, Q Tang, P Hartel, W Jonker. A Type-and-Identity-based Proxy Re-encryption Scheme and its Application in Healthcare. http://eprints.eemcs.utwente.nl/12258/01/.
27. B. Libert and D. Vergnaud, Unidirectional Chosen-Ciphertext Secure Proxy Re-encryption.In *11th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2008*,LNCS 4939, pp. 360–379.Springer–Verlag,2008.
28. B. Libert and D. Vergnaud, Tracing Malicious Proxies in Proxy Re-Encryption.In *2th International Conference on Pairing-Based Cryptography - Pairing 2008*,Springer–Verlag,2008.
29. T. Matsuo, Proxy Re-encryption Systems for Identity-Based Encryption.In *First International Conference on Pairing-Based Cryptography - Pairing 2007*,LNCS 4575, pp. 247–267.Springer–Verlag,2007.
30. T.Matsuo, W.William. Exclusive use of BB2 in proxy re-encryptionscheme.http://grouper.ieee.org/groups/1363/email/discuss/msg00259.html, msg00276.html, msg00277.html, msg00278.html.
31. L.Martin(editor). P1363.3(TM)/D1, Draft Standard for Identity-based Public Cryptography Using Pairings, May 2008.
32. J.Shao,D.Xing and Z.Cao, Identity-Based Proxy Rencryption Schemes with Multiuse, Unidirection, and CCA Security.Cryptology ePrint Archive: http://eprint.iacr.org/2008/103.pdf,2008.
33. R.Sakai and M.Kasahara. ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report2003/054. 2003.
34. Q.Tang, P. Hartel and W Jonker. Inter-domain Identity-based Proxy Re-encryption. Accepted by Inscrypt'08. http://eprints.eemcs.utwente.nl/ 12259/01/.
35. Q.Tang. Type-based Proxy Re-encryption and its Construction. http://eprints.eemcs.utwente.nl/13024/01/.
36. L. d. Zhou, M. A. Marsh, F. B. Schneider, and A. Redz. Distributed blinding for ElGamal re-encryption.TR 1924, Cornell CS Dept.,2004.

A Proof for Theorem 1

Proof. Suppose $\mathcal{A}$ can attack our scheme, we construct an algorithm $\mathcal{B}$ solves the DBDH problem in $\mathbb{G}$. On input (g, g^a, g^b, g^c, T), algorithm $\mathcal{B}$'s goal is to output 1 if $T = e(g, g)^{abc}$ and 0 otherwise.

Let $g_1 = g^a, g_2 = g^b, g_3 = g^c$. Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective identity game as follows:

1. **Initialization**. The selective identity game begins with $\mathcal{A}$ first outputting an identity $ID^\star$ that it intends to attack.
2. **Setup**. To generate the system's parameters, algorithm $\mathcal{B}$ picks $\alpha' \in Z_p$ at random and defines $h = g_1^{-ID^\star} g^{\alpha'}$. It gives $\mathcal{A}$ the parameters $\mathsf{parms} = (g, g_1, g_2, h)$. Note that the corresponding $\mathsf{master-key}$, which is unknown to $\mathcal{B}$, is $g_2^a = g^{ab}$. B picks random $x_i, y_i, z_i \in Z_p$, computes $g_{i_1} = g^{x_i}, g_{i_2} = g^{y_i}, g_{i_3} = h^{z_i}$. It gives $\mathcal{A}$ the public key $pk_i = (g_{i_1}, g_{i_2}, g_{i_3})$.
3. **Phase 1**
   - *"$\mathcal{A}$ issues up to private key queries on ID_i."* $\mathcal{B}$ selects randomly $r_i \in {Z_p}^\star$ and $k' \in Z_p$, sets $sk_{ID_i} = (d_0, d_1, d_2) = (g_2^{\frac{-\alpha'}{ID_i - ID^\star}} (g_1^{(ID_i - ID^\star)} g^{\alpha'})^{r_i}, g_2^{\frac{-1}{ID_i - ID^\star}} g^{r_i}, k')$. We claim sk_{ID_i} is a valid random private key for ID_i. To see this, let $\widetilde{r_i} = r_i - \frac{b}{ID - ID^\star}$. Then we have that
     $d_0 = g_2^{\frac{-\alpha'}{ID_i - ID^\star}} (g_1^{(ID_i - ID^\star)} g^{\alpha})^{r_i} = g_2^{\alpha} (g_1^{(ID_i - ID^\star)} g^{\alpha})^{r_i - \frac{b}{ID - ID^\star}} = g_2^a (g_1^{ID_i} h)^{\widetilde{r_i}}$.
     $d_1 = g_2^{\frac{-1}{ID_i - ID^\star}} g^{r_i} = g^{\widetilde{r_i}}$.
   - *"$\mathcal{A}$ issues up to private key queries on pk_i".* $\mathcal{B}$ returns (x_i, y_i, z_i).
   - *"$\mathcal{A}$ issues up to re-encryption key queries on (pk_j, ID_i)".* The challenge $\mathcal{B}$ computes $rk_{pk_j \to ID_i} = (k'/x_j, (g_2^{\frac{-1}{ID_i - ID^\star}} g^{r_i})^{\frac{k'}{y_j}}, k'/z_j)$ and returns it to $\mathcal{A}$.
   - *"$\mathcal{A}$ issues up to re-encryption key queries on $(pk_j, ID^\star)$".* The challenge $\mathcal{B}$ randomly choose a $k' \in Z_p$, and computes $rk_{pk_j \to ID^\star} = (k'/x_j, (g^{u'})^{k'/y_j}, k'/z_j)$ where u' is a randomly choose from Z_p^* and returns it to A.
   - *"$\mathcal{A}$ issues up to re-encryption queries on (C, pk_j, ID_i) or $(C, pk_j, ID^\star)$"* The challenge $\mathcal{B}$ runs $\mathsf{ReEnc}(rk_{pk_j \to ID_i}, C, pk_j, ID_i)$ or $\mathsf{ReEnc}(rk_{pk_j \to ID^\star}, C, pk_j, ID^\star)$ and returns the results.
4. **Challenge** When $\mathcal{A}$ decides that Phase1 is over, it outputs two messages $M_0, M_1 \in G$. Algorithm $\mathcal{B}$ picks a random bit b and responds with the ciphertext $C^\star = (g^c, (g^\alpha)^c, M_b \cdot T)$. Hence if $T = e(g,g)^{abc} = e(g_1, g_2)^c$, then $C^\star$ is a valid encryption of M_b under $ID^\star$. Otherwise, $C^\star$ is independent of b in the adversary's view.
5. **Phase2** $\mathcal{A}$ issues queries as he does in Phase 1 excepts natural constraints.
6. **Guess** Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. Algorithm $\mathcal{B}$ concludes its own game by outputting a guess as follows. If $b = b'$, then $\mathcal{B}$ outputs 1 meaning $T = e(g,g)^{abc}$. Otherwise it outputs 0 meaning $T \neq e(g,g)^{abc}$.

When $T = e(g,g)^{abc}$ then $\mathcal{A}$'s advantage for breaking the scheme is same as $\mathcal{B}$'s advantage for solving DBDH problem.

B Proof for Theorem 2

Proof. The security proof follows the security of symmetrical encryption.

1. **Setup**. To generate the system's parameters, the challenger $\mathcal{B}$ picks $\alpha \in Z_p$, it randomly choose $x \in Z_q^*$,$y \in Z_q^*$ and computes $h = g^x$, $g_1 = g^\alpha$, $g_2 = g^y$, $\mathsf{master-key} = g_2^\alpha$. It gives $\mathsf{parms} = (g, g_1, g_2, h)$ to $\mathcal{A}$.
2. **Phase 1**

- *"$\mathcal{A}$ issues up to master-key query ".* The challenger $\mathcal{B}$ returns (α, g_2^{α}).
- *"$\mathcal{A}$ issues up to private key queries on ID".* Given $mk = g_2^{\alpha}$ and ID with parms, pick a random $u, k' \in Z_p^*$. Set $sk_{ID} = (d_0, d_1, d_2) = (g_2^{\alpha}(g_1^{ID}h)^u, g^u, k')$.
- *"$\mathcal{A}$ issues up to private key queries on pk".* B returns (θ, β, δ).
- *"$\mathcal{A}$ issues up to rekey generation queries on (pk, ID)".* The challenge $\mathcal{B}$ chooses randomly $k' \in Z_p^*$ and computes $rk_{pk \to ID} = (k'/\theta, g^{k'u/\beta}, k'/\delta)$ and returns it to $\mathcal{A}$.
- *"$\mathcal{A}$ issues up to re-encryption queries on (C, pk, ID)".* The challenge $\mathcal{B}$ runs $ReEnc(rk_{pk \to ID}, C, pk, ID)$ and return the results.

3. **Challenge** When $\mathcal{A}$ decides that Phase1 is over, it outputs two messages $M_0, M_1 \in G$ and the attack identity $ID^{\star}$, Algorithm $\mathcal{B}$ picks g^u as the $ID^{\star}$'s second item of its private key, he picks a random bit b and $r, k^{\star} \in Z_p^*$ responds with the ciphertext $C^{\star} = (g^r, h^r, e(g^{k^{\star}u}, g_1^{IDr}), M_b \cdot e(g_2, (g^{r\alpha}))$. Hence if $k^{\star}$ is the real secret key of $ID^{\star}$, then $C^{\star}$ is a valid encryption of M_b under $ID^{\star}$. Otherwise, $C^{\star}$ is independent of b in the adversary's view.
4. **Phase 2** $\mathcal{A}$ issues queries as he does in Phase 1 except natural constraints.
5. **Guess** Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. Algorithm $\mathcal{B}$ concludes its own game by outputting a guess as follows. If $b = b'$, then $\mathcal{B}$ outputs 1. Otherwise it outputs 0.

Thus the maximal probability of $\mathcal{A}$ successes is $1/p$, which is negligible.

C Proof for Theorem 3

Proof. In this case, the PKG and delegatee's colluding just likes [29]'s PRE scheme from CBE to IBE, the proof is the same as [29].

D Proof for Theorem 4

Proof. Suppose the target CBE ciphertext is $C_{pk} = (C_1, C_2, C_3, C_4)$ which has been re-encrypted by proxy to be $\widehat{C_{ID}} = (\widehat{C_1}, \widehat{C_2}, \widehat{C_3}, \widehat{C_4}) = (C_1, C_3^{1/\delta}, e(g^{ku/\beta}, C_2^{ID}), C_4)$, PKG can decrypt the ciphertext as following. Because $\widehat{C_1} = g^r$, he can compute $w = g^{r\alpha}$ and gets the plaintext by computing

$$\frac{\widehat{C_4}}{e(w, g_2)} = \frac{Me(g_1, g_2)^r}{e(g^{r\alpha}, g_2)} = \frac{Me(g_1, g_2)^r}{e(g_1, g_2)^r} = M$$

E Proof for Theorem 5

Proof. We just give the intuition for this theorem. When considering the proxy, delegatee and delegator's colluding, PKG only interacts with the delegatee-its IBE user. And we know the $\mathsf{BB_1}$ identity based encryption is $\mathsf{IND\text{-}sID\text{-}CPA}$ secure under DBDH assumption. That's imply the attacker can not recover the PKG's $\mathsf{master-key}$.

F Proof for Theorem 6

Proof. Suppose $\mathcal{A}$ can attack our scheme, we construct an algorithm $\mathcal{B}$ solves the mDBDH problem in $\mathbb{G}$. On input $(g, g^a, g^{a^2}, g^b, g^c, T)$, algorithm $\mathcal{B}$'s goal is to output 1 if $T = e(g, g)^{abc}$ and 0 otherwise. Let $g_1 = g^a, g_2 = g^b, g_3 = g^c$. Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective identity game as follows:

1. **Initialization**. The selective identity game begins with $\mathcal{A}$ first outputting an identity $ID^\star$ that it intends to attack.
2. **Setup**.To generate the system's parameters, algorithm B picks $\alpha' \in Z_p^*$ at random and defines $h = g_1^{-ID^\star} g^{\alpha'} \in \mathbb{G}$. It gives $\mathcal{A}$ the parameters $\mathsf{parms} = (g, g_1, g_2, h)$. Note that the corresponding *master*, which is unknown to $\mathcal{B}$, is $g_2^a = g^{ab}$. $\mathcal{B}$ picks random $(x_i, y_i, z_i) \in Z_p^*$, computes $g_{i_1} = g^{x_i}$. it gives $\mathcal{A}$ the public key $pk_i = g_{i_1}$.
3. **Phase 1**
   - *"$\mathcal{A}$ issues up to private key queries on ID_i"*. $\mathcal{B}$ selects randomly $r_i \in {Z_p}^*$ and $k' \in Z_p^*$, sets $sk_{ID_i} = (d_0, d_1) = (g_2^{\frac{-\alpha'}{ID_i - ID^\star}} (g_1^{(ID_i - ID^\star)} g^\alpha)^{r_i}, g_2^{\frac{-1}{ID_i - ID^\star}} g^{r_i})$. We claim sk_{ID_i} is a valid random private key for ID_i. To see this, let $\widetilde{r_i} = r_i - \frac{b}{ID - ID^\star}$. Then we have that
     $d_0 = g_2^{\frac{-\alpha'}{ID_i - ID^\star}} (g_1^{(ID_i - ID^\star)} g^\alpha)^{r_i} = g_2^\alpha (g_1^{(ID_i - ID^\star)} g^\alpha)^{r_i - \frac{b}{ID - ID^\star}} = g_2^a (g_1^{ID_i} h)^{\widetilde{r_i}}$.
     $d_1 = g_2^{\frac{-1}{ID_i - ID^\star}} g^{r_i} = g^{\widetilde{r_i}}$.
   - *"$\mathcal{A}$ issues up to private key queries on pk_i"*. $\mathcal{B}$ returns x_i.
   - *"$\mathcal{A}$ issues up to rekey generation queries on (ID, pk_i)"*. The challenge $\mathcal{B}$ chooses a randomly $x \in Z_p^*$, sets $rk_{ID,pk_1} = x$ and returns it to $\mathcal{A}$. he computes $rk_{ID,pk_3} = w = \frac{g_4^{(ID - ID^\star)x} g_1^{\alpha' x}}{g_4}$ and $rk_{ID,pk_2} = k' x_i$ where k' chosen randomly from Z_p^*, sends them to the proxy. We have
     $$(g_1^{ID} h)^x = g_1 g^{k_1}, g_1^{k_1} = (\frac{(g_1^{ID} h)^x}{g_1})^\alpha = \frac{(g_1^{ID - ID^\star} g^{\alpha'})^{\alpha x}}{g_1^\alpha} = w$$
     For the delegatee and the proxy, they can verify $e(g^{k_1}, g_1) = e(w, g)$ is always satisfied. Thus our simulation is a perfect simulation. But the delegator and delegatee cannot get any useful information from x.
   - *"$\mathcal{A}$ issues up to re-encryption queries on $(\widetilde{C_{ID}}, ID, pk_i)$"*. Challenge $\mathcal{B}$ runs $ReEnc(rk_{ID \to pk_i}, \widetilde{C_{ID}}, ID, pk_i)$ and returns the results.
4. **Challenge** When $\mathcal{A}$ decides that Phase1 is over, it outputs two messages $M_0, M_1 \in G$. Algorithm $\mathcal{B}$ picks a random bit b and responds with the ciphertext $\widetilde{C^\star} = (g^c, (g^{\alpha'})^c, M_b \cdot T)$. Hence if $T = e(g, g)^{abc} = e(g_1, g_2)^c$, then C is a valid encryption of M_b under $ID^\star$. Otherwise, C is independent of b in the adversary's view.
5. **Phase2** $\mathcal{A}$ issues queries as he does in Phase 1 except natural constraints.
6. **Guess** Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. Algorithm $\mathcal{B}$ concludes its own game by outputting a guess as follows. If $b = b'$, then $\mathcal{B}$ outputs 1 meaning $T = e(g, g)^{abc}$. Otherwise it outputs 0 meaning $T \neq e(g, g)^{abc}$.

When $T = e(g, g)^{abc}$ then $\mathcal{A}$'s advantage for breaking the scheme is same as $\mathcal{B}$'s advantage for solving mDBDH problem.

G Proof for Theorem 7

Proof. We just give the intuition for this theorem. The security proof follows the principle of symmetrical encryption. The only information about CBE user's private key just relying on $k\theta$. But even if the proxy, delegator and PKG are colluding, they can only get $k\theta$ where k blinding the private key θ perfectly. Thus they can only guess θ. The adversaries' success probability is at most $1/p$ which is negligible, whether for CBE first level ciphertext or for CBE second level ciphertext.

H Proof for Theorem 8

Proof. We just give the intuition for this theorem. When considering the proxy, delegatee and delegator colluding, the PKG only interact with delegator and proxy. The re-encryption key $rk = (\frac{(\alpha+k_1)k\theta}{ID\alpha+t_2}, g_2^{k_1})$ is distributed same as $(x, \frac{g_4^{(ID-ID^\star)x} g_1^{\alpha' x}}{g_4})$ where x is randomly choose from Z_p^*. That is to say, the adversaries can not get any information about α except randomly guessing. And we know the BB_1 identity based encryption is secure under DBDH assumption. That's imply the attacker can not recover the PKG's $\mathsf{master-key}$. Thus our scheme is PKG-OW secure for the proxy, delegatee and delegator's colluding.

I Proof for Theorem 9

Proof. Suppose $\mathcal{A}$ can attack our scheme, we construct an algorithm $\mathcal{B}$ solves the DBDH problem in G. On input $(g, g^a, g^{a^2}, g^b, g^c, T)$, algorithm $\mathcal{B}$'s goal is to output 1 if $T = e(g,g)^{abc}$ and 0 otherwise. Let $g_1 = g^a, g_2 = g^b, g_3 = g^c$. Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective identity game as follows:

1. **Initialization**. The selective identity game begins with $\mathcal{A}$ first outputting an identity ID^* that it intends to attack.
2. **Setup**.To generate the system's parameters, algorithm $\mathcal{B}$ picks $\alpha' \in Z_p$ at random and defines $h = g_1^{-ID^*} g^{\alpha'} \in G$. It gives $\mathcal{A}$ the parameters $params = (g, g_1, g_2, h)$. Note that the corresponding $master-key$, which is unknown to $\mathcal{B}$, is $g_2^a = g^{ab} \in G^*$.
3. **Phase 1**
   - *"$\mathcal{A}$ issues up to private key queries on ID_i".* $\mathcal{B}$ selects randomly $r_i, r_i' \in Z_p{}^*$ and $k' \in Z_p$, sets $sk_{ID_i} = (d_0, d_1, d_0') = (g_2^{\frac{-\alpha'}{ID_i - ID^*}} (g_1^{(ID_i - ID^*)} g^a)^{r_i}, g_2^{\frac{-1}{ID_i - ID^*}} g^{r_i}, g_2^{\frac{-\alpha'}{ID_i - ID^*}} (g_1^{(ID_i - ID^*)} g^a)^{r_i'})$. We claim sk_{ID_i} is a valid random private key for ID_i. To see this, let $\widetilde{r_i} = r_i - \frac{b}{ID - ID^*}$ and $\widetilde{r_i'} = r_i' - \frac{b}{ID - ID^*}$. Then we have that
     $d_0 = g_2^{\frac{-\alpha'}{ID_i - ID^*}} (g_1^{(ID_i - ID^*)} g^{\alpha'})^{r_i} = g_2^a (g_1^{(ID_i - ID^*)} g^{\alpha'})^{r_i - \frac{b}{ID - ID*}} = g_2^a (g_1^{ID_i} h)^{\widetilde{r_i}}$.
     $d_1 = g_2^{\frac{-1}{ID_i - ID^*}} g^{r_i} = g^{\widetilde{r_i}}$.
     $d_0' = g_2^{\frac{-\alpha'}{ID_i - ID^*}} (g_1^{(ID_i - ID^*)} g^{\alpha'})^{r_i'} = g_2^a (g_1^{(ID_i - ID^*)} g^{\alpha'})^{r_i' - \frac{b}{ID - ID*}} = g_2^a (g_1^{ID_i} h)^{\widetilde{r_i'}}$.
   - *"$\mathcal{A}$ issues up to rekey generation queries on (ID, ID')".*
     The challenge $\mathcal{B}$ chooses a randomly $x \in Z_p^*$, sets $rk_{ID \to ID'} = x$ and returns it to $\mathcal{A}$. He computes $w = \frac{(g^{H_1(ID)} h)^x}{(g^{H_1(ID)} h)}$ and sends it to the proxy. We observe that

$$rk_1 = \frac{\alpha ID' + t_2 + k_1}{k_3(\alpha ID + t_2)} + k_2$$

     but from the simulation, $\alpha = a$ and $t_2 = \alpha' - aID^*$, so we can get

$$rk_1 = \frac{aID' + \alpha' - aID^* + k_1}{k_3(aID + \alpha' - aID^*)} + k_2$$

Let $rk_1 = x$, we can get

$$\begin{aligned} k_1 &= k_3(aID + \alpha' - aID^*)(x - k_2) - (aID' + \alpha' - aID^*) \\ &= [k_3(x - k_2)a(ID - ID^*) - a(ID' - ID^*)] + k_3\alpha'(x - k_2) - \alpha' \end{aligned}$$

So the challenge $\mathcal{B}$ simulates as follows. He chooses a randomly $k_2, k_3 \in Z_p^*$, sets

$$x = \frac{ID' - ID^*}{k_3(ID - ID^*)} + k_2, k_1 = \alpha'(\frac{ID' - ID^*}{ID - ID^*}) - \alpha'$$

searches in `User-key-list` for item (ID', α', r, r')(we assume $sk_{ID'} = (d_0, d_1, d_0') = (g_2^{\frac{-\alpha'}{ID'-ID^*}}(g_1^{(ID'-ID^*)}g^a)^r, g_2^{\frac{-1}{ID'-ID^*}}g^r, g_2^{\frac{-\alpha'}{ID'-ID^*}}(g_1^{(ID'-ID^*)}g^a)^{r'})$ and computes

$$rk_1 = \frac{ID' - ID^*}{k_3(ID - ID^*)} + k_2, rk_2 = g_2^{\frac{-k_3}{ID'-ID^*}}g^{k_3r'}$$

$$rk_3 = g_2^{\frac{-k_2k_3}{ID'-ID^*}}g^{k_2k_3r'}, rk_4 = g_2^{\frac{\alpha'(\frac{ID'-ID^*}{ID-ID^*})-\alpha'}{ID'-ID^*}}g^{(\alpha'(\frac{ID'-ID^*}{ID-ID^*})-\alpha')r'}$$

returns them to $\mathcal{A}$. We can see

$$\frac{C_3' e(rk_2, C_4')}{e(C_2', rk_3)e(C_1', rk_4)e(d_0', C_1')}$$

can be reduced to

$$\frac{Me(g_1, g_2)^r e(g_2^{\frac{-k_3}{ID'-ID^*}}g^{k_3r'}, (g_1^{ID}h)^{r(\frac{ID'-ID^*}{k_3(ID-ID^*)}+k_2)})}{e((g_1^{ID}h)^r, g_2^{\frac{-k_2k_3}{ID'-ID^*}}g^{k_2k_3r'})e(g^r, g_2^{\frac{\alpha'(\frac{ID'-ID^*}{ID-ID^*})-\alpha'}{ID'-ID^*}}g^{(\alpha'(\frac{ID'-ID^*}{ID-ID^*})-\alpha')r'})e(g_2^{\frac{-\alpha'}{ID'-ID^*}}(g_1^{(ID'-ID^*)}g^a)^{r'}, g^r)}$$

which can then be reduced to

$$\frac{Me(g_1, g_2)^r}{e(g_2^\alpha, g^r)} = M$$

Thus our simulation is indistinguishable from the real algorithm running. Thus our simulation is indistinguishable from the real algorithm running.

- *"$\mathcal{A}$ issues up to re-encryption queries on (C_{ID}, ID, ID')".* The challenge $\mathcal{B}$ runs $ReEnc(rk_{ID\to ID'}, C_{ID}, ID, ID')$ and returns the results.

4. **Challenge** When $\mathcal{A}$ decides that Phase1 is over, it outputs two messages $M_0, M_1 \in G$. Algorithm $\mathcal{B}$ picks a random bit b and responds with the ciphertext $C = (g^c, (g^{\alpha'})^c, M_b \cdot T)$. Hence if $T = e(g, g)^{abc} = e(g_1, g_2)^c$, then C is a valid encryption of M_b under ID^*. Otherwise, C is independent of b in the adversary's view.
5. **Phase2** $\mathcal{A}$ issues queries as he does in Phase 1 except natural constraints.
6. **Guess** Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. Algorithm $\mathcal{B}$ concludes its own game by outputting a guess as follows. If $b = b'$, then $\mathcal{B}$ outputs 1 meaning $T = e(g, g)^{abc}$. Otherwise it outputs 0 meaning $T \neq e(g, g)^{abc}$.

When $T = e(g, g)^{abc}$ then $\mathcal{A}$'s advantage for breaking the scheme is same as $\mathcal{B}$'s advantage for solving DBDH problem.

J Proof for Theorem 10

Proof. The security proof is same as the above theorem except that it does not allow *"$\mathcal{A}$ issues up to rekey generation queries on (ID, ID^*)"*, for $\mathcal{B}$ does not know the private key corresponding to ID^*.

K Proof for Theorem 11

Proof. We just give the intuition for this theorem. The master-key is g_2^α, and delegator's private key is $sk_{ID} = (g_2^\alpha(g_1^{ID}h)^{u_0}, g^{u_0}, (g_2^\alpha(g_1^{ID}h)^{u_1}))$, the delegatee's private key is $sk_{ID'} = (g_2^\alpha(g_1^{ID'}h)^{u_0}, g^{u_0}, (g_2^\alpha(g_1^{ID'}h)^{u_1}))$, the proxy re-encryption key is $rk_{ID\to ID'} = (\frac{\alpha ID' + t_2 + k_1}{k_3(\alpha ID + t_2)} + k_2, g^{u_1' k_3}, g^{u_1' k_2 k_3}, g^{u_1' k_1})$. Because the re-encryption key $rk_{ID\to ID'}$ is uniformly distributed in $(Z_p^*, \mathbb{G}, \mathbb{G}, \mathbb{G})$, and the original BB_1 IBE is secure, we can conclude that g_2^α can not be disclosed by the proxy, delegatee and delegator's colluding.

L Proof for Theorem 12

Proof. Let $\mathcal{A}$ be a p.p.t. algorithm that has non-negligible advantage in attacking the scheme proposed in Section 5.6. We use $\mathcal{A}$ in order to construct a second algorithm $\mathcal{B}$ which has non-negligible advantage at solving the DBDH problem in G. Algorithm $\mathcal{B}$ accepts as input a properly-distributed tuple (g, g^a, g^b, g^c, R) and outputs 1 if $R = e(g,g)^{abc}$. We now describe the algorithm $\mathcal{B}$, which interacts with algorithm $\mathcal{A}$ as following.

$\mathcal{B}$ simulates the random oracles H_1, H_2, H_3, H_4 as follows.

1. $H_1 : \{0,1\}^* \to Z_q^*$. On receipt of a new query for $ID \neq ID^*$, return $t \leftarrow_R Z_q^*$ and record (ID, t); On receipt of a new query for ID^*, select randomly $T \in Z_q^*$, return T and record (ID^*, T).
2. $H_2 : G_1 \times \{0,1\}^l :\to Z_q^*$. On a new query (δ, M), returns $s \leftarrow_R G$ and record (δ, M, s).
3. $H_3 : G_1 :\to \{0,1\}^l$. On receipt of a new query δ, select $p \leftarrow \{0,1\}^l$ and return p. Record the tuple (δ, p).
4. $H_4 : \{0,1\}^* \times G \times G \times G \times \{0,1\}^l :\to G$. On receipt of a new query $(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4)$, select $z \in Z_q^*$ and return $g^z \in G$, record $(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4, z, g^z)$.

Our simulation proceeds as follows:

1. **Setup.** $\mathcal{B}$ generates the scheme's master parameter as following. First it lets $g_1 = g^a$, $g_2 = g^b$, $g_3 = g^c$, algorithm $\mathcal{B}$ picks $\alpha \in Z_p$ at random and defines $h = g_1^{-T} g^{\alpha'} \in G$ $\mathcal{B}$ lets $params = (G_1, H_1, H_2, H_3, H_4, g, g_1, g_2, g_3, h)$ and gives $params$ to $\mathcal{A}$.
2. **Find/Guess.** During the Find stage, there are no restrictions on which queries $\mathcal{A}$ may issue. The scheme permits only a single consecutive re-encryption, therefore, during the GUESS stage, $\mathcal{A}$ is restricted from issuing the following queries:
   (a) $(extract, ID^*)$ where ID^* is the challenge identity.
   (b) $(decrypt, ID^*, c^*)$ where c^* is the challenge ciphertext.
   (c) Any pair of queries $(rkextract, ID^*, ID_i)$, $(decrypt, ID_i, c_i)$ where c_i=Reencrypt($rk_{ID^*\to ID_i}$, c^*).

   In the Guess stage, let ID^* be the target identity, and parse the challenge ciphertext c^* as $(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$. In both phases, $\mathcal{B}$ responds to $\mathcal{A}$'s queries as follows.

– On $(extract, ID)$, where(in the Guess)stage $ID \neq ID^*$, $\mathcal{B}$ selects randomly $r_i \in Z_p^*$, sets $sk_{ID_i} = (d_0, d_1) = (g_2^{\frac{-\alpha'}{H_1(ID_i)-T}}(g_1^{(H_1(ID_i)-T)}g^{\alpha'})^{r_i}, g_2^{\frac{-1}{H_1(ID_i)-T}}g^{r_i})$. We claim sk_{ID_i} is a valid random private key for ID_i. To see this, let $\widetilde{r_i} = r_i - \frac{b}{H_1(ID_i)-T}$. Then we have that

$d_0 = g_2^{\frac{-\alpha'}{H_1(ID_i)-T}}(g_1^{(H_1(ID_i)-T)}g^{\alpha'})^{r_i} = g_2^a(g_1^{(H_1(ID_i)-T)}g^{\alpha'})^{r_i - \frac{b}{H_1(ID_i)-T}} = g_2^a(g_1^{H_1(ID_i)}h)^{\widetilde{r_i}}$.

$d_1 = g_2^{\frac{-1}{H(ID_i)-T}}g^{r_i} = g^{\widetilde{r_i}}$.

$d_0' = g_2^{\frac{-1}{H(ID_i)-T}}g^{r_i} = g^{\widetilde{r_i}}$.

– On $(rkextract, ID, ID')$, do the same as $\mathcal{A}$ handling re-encryption key query in Phase 13 in the above theorem.

– On $(decrypt, ID, c)$ where (in the Guess stage) $(ID, c) \neq (ID^*, c^*)$, check whether c is a level-1 (non re-encrypted) or level-2 (re-encrypted) ciphertext. In the Guess stage, parse c^* as $(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$.

For a level-1 ciphertext, $\mathcal{B}$ parses c as $(C_1, C_2, C_3, C_4, C_5)$ and:

(a) Looks up the value $(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4)$ in the H_4 table, to obtain the tuple $(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4, z, g^z)$. If $(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4)$ is not in the table, or if (in the Guess stage) $C_5 = C_5^*$, then $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$.

(b) Looks up the value (δ, M, s) in the H_2 table. Checks whether there exist an item of (δ, M, s) such that $S = g^{zs}$. If not, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$.

(c) Computes $k = \frac{e(C_1, d_0)}{e(C_2, d_1)}$, checks that $\delta = \frac{C}{k}$. If not, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$.

(d) Checks that $C_4 = H_3(\delta) \oplus M$. If not, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$.

(e) Otherwise, $\mathcal{B}$ returns M to $\mathcal{A}$.

For a level-2 ciphertext, $\mathcal{B}$ parses c as $(C_1', C_2', C_3', C_4', C_5', C_6', C_7', C_8')$ and:

(a) Computes

$$k = \frac{C_3' e(C_5', C_4')}{e(C_2', C_6')e(C_1', C_7')e(d_0', C_1')} = \frac{C_3' e(rk_2, C_4')}{e(C_2', rk_3)e(C_1', rk_4)e(d_0', C_1')}$$

(b) Checks that $\delta = \frac{C}{k}$. If not, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$.

(c) Checks that $C_2 = h^s g_1^{H_1(ID)s}$. If so, return M. Otherwise, return $\perp$.

– On $(reencrypt, C_{ID}, ID, ID')$. $\mathcal{B}$ runs $ReEnc(rk_{ID \to ID'}, C_{ID}, ID, ID')$ and returns the results.

At the end of the Find phase, $\mathcal{A}$ outputs (ID^*, M_0, M_1), with the condition that $\mathcal{A}$ has not previously issued $(extract, ID^*)$. At the end of the Guess stage, $\mathcal{A}$ outputs its guess bit i'.

3. **Choice and Challenge.** At the end of the Find phase, $\mathcal{A}$ outputs (ID^*, M_0, M_1). $\mathcal{B}$ forms the challenge ciphertext as follows:

(a) Choose $\delta \in G_1$ and $p \in \{0,1\}^n$ randomly, and insert (δ, p) in H_3 table.

(b) Insert $(\delta, M_b, ?, g_3, \delta \cdot R, M_b \oplus p)$ to H_2 table.

(c) Choose $z \in Z_p$ randomly, and insert $((g_3, g_3^{\alpha'}, \delta \cdot R, M_b \oplus p), z, g^z)$ in the H_4 table.

$\mathcal{B}$ outputs the challenge ciphertext $(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*) = (g_3, g_3^{\alpha'}, \delta \cdot R, M_b \oplus p, g_3^z)$ to $\mathcal{A}$ and begins the GUESS stage.

4. **Forgeries and Abort conditions** The adversary may forge C_5 on (C_1, C_2, C_3, C_4), but from the security of BLS short signature [8], this probability is negligible.

M Proof for Theorem 13

Proof. The security proof is same as the above theorem except that it does not allow *"$\mathcal{A}$ issues up to rekey generation queries on (ID, ID^*)"*, for $\mathcal{B}$ does not know the private key corresponding to ID^*.

N Proof for Theorem 14

Proof. The security proof is same as the proof for Theorem K.

O Proof for Theorem 15

Proof. Suppose $\mathcal{A}$ has advantage in attacking our PRE system. We build an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the Decision $q-BDHI$ problem in $\mathbb{G}$. Algorithm $\mathcal{B}$ is given as input a random $(q+2)$-tuple $(g, g^{\alpha}, g^{(\alpha^2)}, ..., g^{(\alpha^q)}, T) \in (\mathbb{G}^*)^{q+1} \times \mathbb{G}_1$ that is either sampled from P_{BDHI} (where $T = e(g,g)^{\frac{1}{\alpha}}$) or from R (where T is uniform and independent in $\mathbb{G}_1$). Algorithm $\mathcal{B}$'s goal is to output 1 if $T = e(g,g)^{1/\alpha}$ and 0 otherwise. Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective identity game as follows:
Preparation. Algorithm $\mathcal{B}$ builds a generator $h \in \mathbb{G}^*$ for which it knows $q-1$ pairs of the form $(w_i, h^{1/(\alpha+w_i)})$ for random $w_1, ..., w_{q-1} \in Z_p^*$. This is done as follows:

1. Pick random $w_1, ..., w_{q-1} \in Z_p^*$ and let $f(z)$ be the polynomial $f(z) = \prod_{i=1}^{q-1}(z+w_i)$. Expand the terms of f to get $f(z) = \sum_{i=0}^{q-1} c_i x_i$. The constant term c_0 is non-zero.
2. Compute $h = \prod_{i=0}^{q-1}(g^{(\alpha^i)})^{c_i} = g^{f(\alpha)}$ and $u = \prod_{i=1}^{q}(g^{(\alpha^i)})^{c_{i-1}} = g^{\alpha f(\alpha)}$. Note that $u = h^{\alpha}$.
3. Check that $h \in G^*$. Indeed if we had $h = 1$ in $\mathbb{G}$ this would mean that $w_j = -\alpha$ for some j easily identifiable w_j, at which point $\mathcal{B}$ would be able to solve the challenge directly. We thus assume that all $w_j \neq -\alpha$.
4. Observe that for any $i = 1, ..., q-1$, it is easy for $\mathcal{B}$ to construct the pair $(w_i, h^{1/(\alpha+w_i)})$. To see this, write $f_i(z) = f(z)/(z+w_i) = \sum_{i=0}^{q-2} d_i Z_i$. Then $h^{1/(\alpha+w_i)} = g^{f_i(\alpha)} = \prod_{i=0}^{q-2}(g^{(\alpha^i)})^{d_i}$.
5. Next $\mathcal{B}$ computes

$$T_h = T^{c_0 f(\alpha)} \cdot T_0$$
$$T_0 = \prod_{i=0}^{q-1}\prod_{j=0}^{q-2} e(g^{(\alpha^i)}, g^{(\alpha^j)})^{c_i c_{j+1}}$$

Observe that if $T = e(g,g)^{1/\alpha}$ then $T_h = e(g^{f(\alpha)/\alpha}, g^{f(\alpha)}) = e(h,h)^{1/\alpha}$. On the contrary, if T is uniform in G_1, then so is T_h.

We will be using the values h, u, T_h and the pairs $(w_i, h^{1/(\alpha+w_i)})$ for $i = 1, ..., q-1$ throughout the simulation.

1. **Initialization.** The selective identity game begins with $\mathcal{A}$ first outputting an identity $ID^{\star} \in Z_p^*$ that it intends to attack.
2. **Setup.** To generate the system parameters,algorithm $\mathcal{B}$ does the following:
   (a) Pick random $a, b \in Z_p^*$ under the constraint that $ab = ID^{\star}$.
   (b) Compute $X = u^{-a}h^{-ab} = h^{-a(\alpha+b)}$ and $Y = u = h^{\alpha}$.

(c) Publish $\mathsf{parms} = (h, X, Y)$ as the public parameters. Note that X, Y are independent of $ID^\star$ in the adversary's view.
(d) We implicitly define $x = -a(\alpha + b)$ and $y = \alpha$ so that $X = h^x$ and $Y = h^y$. Algorithm $\mathcal{B}$ does not know the value of x or y, but does know the value of $x + ay = -ab = -ID^\star$.

3. **Phase 1**.
– *"$\mathcal{A}$ issues up to $q_s < q$ private key queries".*
Consider the i-th query for the private key corresponding to public key $ID_i \neq ID^\star$. We need to respond with a private key $(r, h^{\frac{1}{(ID_i+x+ry)}})$ for a uniformly distributed $r \in Z_p$. Algorithm $\mathcal{B}$ responds to the query as follows:
(a) Let $(w_i, h^{1/(\alpha+w_i)})$ be the $i-$th pair constructed during the preparation step. Define $h_i = h^{1/(\alpha+w_i)}$.
(b) $\mathcal{B}$ first constructs an $r \in Z_p$ satisfying $(r-a)(\alpha + w_i) = ID_i + x + ry$. Plugging in the values of x and y the equation becomes

$$(r-a)(\alpha + w_i) = ID_i - a(\alpha + b) + r\alpha$$

We see that the unknown α cancels from the equation and we get $r = a + \frac{ID_i - ab}{w_i} \in Z_p$ which B can evaluate.
(c) Now $(r, h_i^{1/(r-a)})$ is a valid private key for ID for two reasons. First,

$$h_i^{1/(r-a)} = (h^{1/(\alpha+w)})^{1/(r-a)} = h^{1/(r-a)(\alpha+w_i)} = h^{1/(ID_i+x+ry)}$$

as required. Second, r is uniformly distributed among all elements in Z_p for which $ID_i + x + ry \neq 0$ and $r \neq a$. This is true since w is uniform in $Z_p/\{0, -\alpha\}$ and is currently independent of $\mathcal{A}$'s view. Algorithm $\mathcal{B}$ gives $\mathcal{A}$ the private key $(r, h_i^{1/(r-\alpha)})$. For completeness, we note that $\mathcal{B}$ can construct the private key for ID_i with $r = a$ as $(r, h^{1/ID_i - ID^\star})$. Hence,the r in the private key given to $\mathcal{A}$ can be made uniform among all $r \in Z_p$ for which $ID + x + ry \neq 0$ as required.
We point out that this procedure will fail to produce the private key for $ID_i = ID^\star$ since in that case we get $r = a$ and $ID + x + ry = 0$. Hence, B can generate private keys for all public keys except for $ID^\star$.
– *"$\mathcal{A}$ issues up to re-encryption key queries on (ID_i, ID_j)".*
The challenger $\mathcal{B}$ chooses a randomly $x \in Z_p^*$ and sets $rk_2 = \frac{ID_j+x+r_jy}{ID_i+x+r_iy} + k = x$, he computes re-encryption key as follows:

$$rk_1 = r_i, rk_2 = x$$
$$rk_3 = g^{\frac{k}{ID_j+x+r_jy}} = g^{\frac{x - \frac{ID_j+x+r_jy}{ID_i+x+r_iy}}{ID_j+x+r_jy}} = g^{\frac{x}{ID_j+x+r_jy}} \cdot g^{-\frac{1}{ID_i+x+r_iy}} = h_j^{\frac{x}{r_j-a}} \cdot h_i^{\frac{1}{r_i-a}}$$

thus our simulation is a perfect simulation. Because x is uniformly in Z_p^*, the adversary (including delegator and proxy colluding or delegatee and proxy colluding) can not get any useful information from it.
– *"$\mathcal{A}$ issues up to rekey generation queries on $(ID^\star, ID)$".*
Do the same as the above.
– *"$\mathcal{A}$ issues up to re-encryption queries on (C_{ID_i}, ID_i, ID_j)".*
The challenge $\mathcal{B}$ runs $\mathsf{ReEnc}(rk_{ID_i \to ID_j}, C_{ID_i}, ID_j)$ and returns the results.

4. **Challenge.** $\mathcal{A}$ outputs two messages $M_0, M_1 \in G$. Algorithm $\mathcal{B}$ picks a random bit $b \in \{0,1\}$ and a random $l \in Z_p^*$. It responds with the ciphertext $C^\star = (h^{-al}, h^l, T_h^l \cdot M_b)$. Define $s = l/\alpha$. On the one hand, if $T = e(h,h)^{1/\alpha}$ we have

$$\begin{aligned} h^{-al} &= h^{a\alpha(l/\alpha)} = h^{(x+ab)(l/\alpha)=h^{sID^\star} \cdot X^s} \\ h^l &= Y^{l/\alpha} = Y^s \\ T_h^l &= e(h,h)^{l/\alpha} = e(h,h)^s \end{aligned}$$

   It follows that $C^\star$ is a valid encryption of M_b under $ID^\star$, with the uniformly distributed randomization value $s = l/\alpha$. On the other hand, when T is uniform in G_1 ,then, in the adversary's view $C^\star$ is independent of the bit b.
5. **Phase 2.** $\mathcal{A}$ issues more private key queries, for a total of at most $q_s < q$. Algorithm $\mathcal{B}$ responds as before. $\mathcal{A}$ issues more other queries like in Phase 1 except natural constraints and Algorithm $\mathcal{B}$ responds as before.
6. **Guess.** Finally, $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$. If $b = b'$ then $\mathcal{B}$ outputs 1 meaning $T = e(g,g)^{1/\alpha}$. Otherwise,it outputs 0 meaning $T \neq e(g,g)^{1/\alpha}$.

When $T = e(g,g)^{1/\alpha}$ then $\mathcal{A}$'s advantage for breaking the scheme is same as $\mathcal{B}$'s advantage for solving q-BDHI problem.

P Proof for Theorem 16

Proof. The security proof is same as the above theorem except that it does not allow *"$\mathcal{A}$ issues up to rekey generation queries on $(ID, ID^\star)$"*, for $\mathcal{B}$ does not know the private key corresponding to $ID^\star$.

Q Proof for Theorem 17

Proof. We just give the intuition for this theorem. The master-key is (x, y), and delegator's private key is $(r_i, g^{\frac{1}{ID_i+x+r_iy}})$, the delegatee's private key is $(r_j, g^{\frac{1}{ID_j+x+r_jy}})$, the proxy re-encryption key is $(r_i, \frac{ID_j+x+r_jy}{ID_i+x+r_iy} + k \mod p, g^{\frac{k}{ID'+x+r'y}})$. Although $rk_1 = r_i$, this does not give adversary any more help for $g^{\frac{1}{ID_i+x+r_iy}}$ or x, y. Because the re-encryption key is uniformly distributed in Z_p^*, and the original BB_2 IBE is secure, we can conclude that (x, y) can not be disclosed by the proxy, delegatee and delegator's colluding.

R Proof for Theorem 18

Proof. The proof combines the following three lemmas.

Lemma 1. *Suppose that H is a random oracle and that there exists an IND-ID-CCA adversary $\mathcal{A}$ against* PRE-SK-IBE *with advantage $\varepsilon(k)$ which makes at most q_1 distinct queries to H (note that H can be queried directly by $\mathcal{A}$ or indirectly by an extraction query,a decryption query or the challenge operation).Then there exists an IND-CCA adversary $\mathcal{B}$ which runs in time $O(time(\mathcal{A}) + q_D \cdot (T + \Gamma_1))$ against the following* PRE-BasicPub^{hy} *scheme with advantage at least $\varepsilon(k)/q_1$ where T is the time of computing pairing and Γ_1 is the time of a multiplication operation*

1 in $\mathbb{G}_1$.

PRE-BasicPub^{hy} *is specified by six algorithms:* KeyGen, RKGen, Encrypt, Reencrypt, Decrypt$_1$, Decrypt$_2$.

1. **KeyGen:** *Given a security parameter* k*, the parameter generator follows the steps.*
   (a) *Identical with step 1 in Setup algorithm of* PRE-SK-IBE.
   (b) *The* PKG *pick a random* $s \in Z_q^*$ *and compute* $P_{pub} = sP$*. Randomly choose different elements* $h_i \in Z_q^*$ *and compute* $\frac{1}{h_i+s}P$ *for* $0 \leq i \leq q_1$*. Randomly choose different elements* $h_0' \in Z_q^*$ *and compute* $\frac{1}{h_0'+s}P$.
   (c) *Pick three cryptographic hash functions:* $H_2 : G_T \to \{0,1\}^n$, $H_3 : \{0,1\}^n \times \{0,1\}^n \to Z_q^*$ *and* $H_4 : \{0,1\}^n \to \{0,1\}^n$ *for some integer* $n > 0$.

   The message space is $M = \{0,1\}^n$*. The ciphertext space is* $C = G_1^* \times \{0,1\}^n \times \{0,1\}^n$*. The public key for delegator is* $K_{pubA} = (q, G_1, G_2, G_T, \varphi, e, n, P_1, P_2, P_{pub}, h_0, (h_1, \frac{1}{h_1+s}P_2), ..., (h_i, \frac{1}{h_i+s}P_2), ..., (h_{q_1-1}, \frac{1}{h_{q_1-1}+s}P_2), H_2, H_3, H_4)$ *and the private key is* $d_A = \frac{1}{h_0+s}P$*. Note that* $e(h_0P_1 + P_{pub}, d_A) = e(P_1, P_2)$*. The public key for delegatee is* $K_{pubB} = (q, G_1, G_2, G_T, \varphi, e, n, P_1, P_2, P_{pub}, h_0', (h_1, \frac{1}{h_1+s}P_2), ..., (h_i, \frac{1}{h_i+s}P_2), ..., (h_{q_1-1}, \frac{1}{h_{q_1-1}+s}P_2), H_2, H_3, H_4)$ *and the private key is* $d_B = \frac{1}{h_0'+s}P$*. Note that* $e(h_0'P_1 + P_{pub}, d_B) = e(P_1, P_2)$.
2. **RKGen:** *The* PKG *chooses a collision resistent hash function* $H_5 : \{0,1\}^{3|p|} \to Z_p^*$ *and random seeds* $t_1, t_2 \in Z_p^*$*, and computes* $k = H_5(h_0, h_0', t_1)$*. He computes* $rk_{A\to B} = (rk_1, rk_2, rk_3) = (\frac{s+h_0'+k_1}{s+h_0} \mod p, \frac{k_2}{s+h_0} \mod p, \frac{k_1}{k_2(s+h_0')}P_2)$*. He sends* $rk_{A\to B}$ *to the proxy as the re-encryption key via authenticated channel.*
3. **Encrypt:** *Given a plaintext* $m \in M$ *and the public key* K_{pubA} *and* K_{pubB}*,*
   (a) *Pick a random* $\sigma \in \{0,1\}^n$ *and compute* $r = H(\sigma, m)$*,and* $g^r = e(P_1, P_2)^r$.
   (b) *For the delegator, set the ciphertext to be* $C = (r(h_0P_1 + P_{pub}), \sigma \oplus H_2(g^r), m \oplus H(\sigma))$.
   (c) *For the delegatee, set the ciphertext to be* $C = (r(h_0'P_1 + P_{pub}), \sigma \oplus H_2(g^r), m \oplus H(\sigma))$.
4. **Reencrypt:**. *On input the ciphertext* $C_A = (C_1, C_2, C_3) = (rQ_{ID}, \sigma \oplus H_2(g^r), m \oplus H_4(\sigma))$*, the proxy computes* $C_B = (C_1', C_2', C_3', C_4', C_5') = (rk_1C_1, e(rk_2C_1, rk_3), C_2, C_3, C_1)$*, and sends it to the delegatee.*
5. **Decrypt$_1$:** *For the delegator, given a ciphertext* $C_A = (U, V, W)$*,* K_{pubA}*, and the private key* d_A
   (a) *Compute* $g' = e(U, d_A)$ *and* $\sigma' = V \oplus H(g')$*,*
   (b) *Compute* $m' = W \oplus H_4(\sigma')$ *and* $r' = H_3(\sigma', m')$*,*
   (c) *If* $U \neq r'(h_0P_1 + P_{pub})$*,reject the ciphertext, else return* m' *as the plaintext.*
6. **Decrypt$_2$:** *For the delegatee, given a ciphertext* $C_B = (C_1', C_2', C_3', C_4', C_5')$*:*
   (a) *Compute* $g' = \frac{e(C_1', d_B)}{C_2'}$ *and* $\sigma' = C_3' \oplus H_2(g')$.
   (b) *Compute* $m' = C_4' \oplus H_4(\sigma)$ *and* $r' = H_3(\sigma', m')$.
   (c) *If* $C_5' \neq r'(h_0P_1 + P_{pub})$*, output* $\perp$*, else return* m' *as the plaintext.*

Proof. The proof for this lemma is similar as Lemma 1 in Section 3.2 in [14].

Lemma 2. *Let* H_3, H_4 *be random oracles. Let* $\mathcal{A}$ *be an IND-CCA adversary against* PRE-BasicPub^{hy} *defined in Lemma 1 with advantage* $\epsilon(k)$*. Suppose* $\mathcal{A}$ *has running time* $t(k)$*, makes at most* q_D *decryption queries, and makes* q_3 *and* q_4 *queries to* H_3 *and* H_4 *respectively. Then there*

exists an IND-CPA adversary $\mathcal{B}$ against the following PRE-BasicPub *scheme with advantage $\epsilon_1(k)$ and running time $t_1(k)$ where*

$$\epsilon_1(k) \geq \frac{1}{2(q_3+q_4)}[(\epsilon(k)+1)(1-\frac{2}{q})^{q_D}-1]$$
$$t_1(k) \leq t(k) + O((q_3+q_4)\cdot(n+logq)).$$

PRE-BasicPub *is specified by six algorithms:* KeyGen, RKGen, Encrypt, Reencrypt, Decrypt_1, Decrypt_2.

1. **KeyGen:** *Given a security parameter k, the parameter generator follows the steps.*
   (a) Identical with step 1 in algorithm KeyGen *of* PRE-BasicPub^{hy}.
   (b) Identical with step 2 in algorithm KeyGen *of* PRE-BasicPub^{hy}.
   (c) Pick a cryptographic hash function $H_2 : G_T \to \{0,1\}^n$ for some integer $n > 0$.
   The message space is $M = \{0,1\}^n$. The ciphertext space is $C = G_1^* \times \{0,1\}^n \times \{0,1\}^n$. The public key for delegator is $K_{pubA} = (q, G_1, G_2, G_T, \varphi, e, n, P_1, P_2, P_{pub}, h_0, (h_1, \frac{1}{h_1+s}P_2), ..., (h_i, \frac{1}{h_i+s}P_2), ..., (h_{q_1-1}, \frac{1}{h_{q_1-1}+s}P_2), H_2, H_3, H_4)$ and the private key is $d_A = \frac{1}{h_0+s}P$. Note that $e(h_0P_1+P_{pub}, d_A) = e(P_1, P_2)$. The public key for delegatee is $K_{pubB} = (q, G_1, G_2, G_T, \varphi, e, n, P_1, P_2, P_{pub}, h_0', (h_1, \frac{1}{h_1+s}P_2), ..., (h_i, \frac{1}{h_i+s}P_2), ..., (h_{q_1-1}, \frac{1}{h_{q_1-1}+s}P_2), H_2, H_3, H_4)$ and the private key is $d_B = \frac{1}{h_0'+s}P$. Note that $e(h_0'P_1 + P_{pub}, d_B) = e(P_1, P_2)$.
2. **ReKeyGen:** *Identical with* RKGen *of* PRE-BasicPub^{hy} *except no s generation.*
3. **Encrypt:** *Given a plaintext $m \in M$ and the public key K_{pub}, choose a random $r \in Z_q^*$ and compute ciphertext $C = (rP_1, r(h_0P_1 + P_{pub}), m \oplus H_2(g^r))$ where $g^r = e(P_1, P_2)^r$.*
4. **Reencrypt:** *Identical with* Reencrypt *of* PRE-BasicPub^{hy}.
5. **Decrypt_1:** *Given a ciphertext $C = (U_1, U_2, V)$, K_{pub}, and the private key d_A, compute $g' = e(U_2, d_A)$ and plaintext $m = V \oplus H_2(g')$.*
6. **Decrypt_2:***Identical with* Decrypt_2 *of* PRE-BasicPub^{hy} *except no step 3(no checking step).*

Proof. The proof for this lemma is similar as lemma 2 in Section 3.2 in [14], actually this is the Fujisaki-Okamoto transformation[17].

Lemma 3. *Let H_2 be a random oracle. Suppose there exists an IND-CPA adversary $\mathcal{A}$ against the* PRE-BasicPub *defined in Lemma 2 which has advantage $\epsilon(k)$ and queries H at most q_2 times. Then there exists an algorithm $\mathcal{B}$ to solve the $q_1 - BDHI$ problem with advantage at least $2\epsilon(k)/q_2$ and running time $O(time(\mathcal{A}) + q_1^2 \cdot T_2)$ where T_2 is the time of a multiplication operation in G_2.*

Proof. Algorithm $\mathcal{B}$ is given as input a random $q_1 - BDHI$ instance (q,G_1, G_2,G_T, φ, P_1, P_2, xP_2, x^2P_2, ...,$x^{q_1}P_2$) where x is a random element from Z_q^*. Algorithm $\mathcal{B}$ finds $e(P_1, P_2)^{\frac{1}{x}}$ by interacting with $\mathcal{A}$ as follows: Algorithm $\mathcal{B}$ first simulates algorithm keygen of BasicPub, which was defined in Lemma 2, to create the public key as below.

1. Randomly choose different $h_0, ..., h_{q_1-1} \in Z$ and let $f(z)$ be the polynomial $f(z) = \prod_{i=1}^{q_1-1}(z+h_i)$. Reformulate f to get $f(z) = \prod_{i=0}^{q_1-1} c_i z_i$. The constant term c_0 is non-zero because $h_i \neq 0$ and c_i are computable from h_i.
2. Compute $Q_2 = \sum_{i=0}^{q_1-1} c_i x^i P_2 = f(x)P_2$ and $xQ_2 == \sum_{i=0}^{q_1-1} c_i x^{i+1} P_2 = xf(x)P_2$.

3. Check that $Q_2 \in G_2^*$. If $Q_2 = 1_{G_2}$, then there must exist an $h_i = -x$ which can be easily identified, and so, $\mathcal{B}$ solves the $q_1 - BDHI$ problem directly. Otherwise $\mathcal{B}$ computes $Q_1 = \varphi(Q_2)$ and continues.
4. Compute $f_i(z) = f(z)/(z + h_i) = \sum_{j=0}^{q_1-2} d_j z^j$ and $\frac{1}{x+h_i}Q_2 = f_i(x)P_2 = \sum_{j=0}^{q_1-2} d_j x^j P_2$ for $1 \le i < q_1$.
5. Set $T' = \sum_{i=0}^{q_1-1} c_i x^{i-1} P_2$ and compute $T_0 = e(\varphi(T'), Q_2 + c_0 P_2)$.
6. Now $\mathcal{B}$ passes $\mathcal{A}$ the public key $K_{pub} = (q, G_1, G_2, G_T, \varphi, e, n, Q_1, Q_2, xQ_1 - h_0Q_1, h_0, (h_2 + h_0, \frac{1}{h_2+x}Q_2), ..., (h_i + h_0, \frac{1}{h_i+x}Q_2), ..., (h_{q_1-1} + h_0, \frac{1}{h_{q_1-1}+x}Q_2), H_2)$(ie. setting $P_{pub} = xQ_1 - h_0Q_1$, $H_1(ID_A) = h_0$, $H_1(ID_B) = h_1 + h_0$), and the private key for A is $d_A = \frac{1}{x}Q_2$, which $\mathcal{B}$ does not know. The private key for B is $d_B = \frac{1}{h_1+x}Q_2$, which $\mathcal{B}$ knows. H_2 is a random oracle controlled by $\mathcal{B}$. Note that $e((h_i + h_0)Q_1 + P_{pub}, \frac{1}{h_i+x}Q_2) = e(Q_1, Q_2)$ for $i = 2, \cdots, q_1 - 1$, $e(h_0Q_1 + P_{pub}, d_A) = e(Q_1, Q_2)$, $e((h_1 + h_0)Q_1 + P_{pub}, d_B) = e(Q_1, Q_2)$. Hence K_{pub} is a valid public key of A in BasicPub .

Now $\mathcal{B}$ starts to respond to queries as follows.

1. **Phase 1**
   (a) **$\mathbf{H}_2$ Query(X_i).** At any time algorithm $\mathcal{A}$ can issue queries to the random oracle H_2. To respond to these queries C maintains a list of tuples called H_2^{list}. Each entry in the list is a tuple of the form (X_i, ζ_i) indexed by X_i. To respond to a query on X_i, $\mathcal{B}$ does the following operations:
      i. If on the list there is a tuple indexed by X_i, then $\mathcal{B}$ responds with ζ_i.
      ii. Otherwise, $\mathcal{B}$ randomly chooses a string $\zeta_i \in \{0,1\}^n$ and inserts a new tuple (X_i, ζ_i) to the list. It responds to $\mathcal{A}$ with ζ_i.
   (b) **RKGen Query.** $\mathcal{B}$ Chooses a randomly $t \in Z_q^*$ and let $k_1 = tk_2$, chooses $a, b \in Z_q^*$, let $(\frac{s+h_0+h_1}{s+h_0} = a, \frac{k_2}{s+h_0} = b)$, so $(rk_1, rk_2) = (\frac{s+h_0+h_1+k_1}{s+h_0}, \frac{k_2}{s+h_0}) = (a + tb, b)$[14]. $\mathcal{B}$ computes rk_3 as following.

$$s = x - h_0, \frac{s + h_0 + h_1}{s + h_0} = a, \frac{k_2}{s + h_0} = b, rk_3 = \frac{t}{s + h_1 + h_0}Q_2 = td_B$$

   (c) **Reencrypt Query.** The challenge $\mathcal{B}$ runs $\mathsf{ReEnc}(rk_{A\to B}, C_A, B)$ and returns the results.
2. **Challenge.** Algorithm $\mathcal{A}$ outputs two messages (m_0, m_1) of equal length on which it wants to be challenged. C chooses a random string $R \in \{0,1\}^n$ and a random element $r \in Z_p^*$, and defines $C_{ch} = (U, V) = (rQ_1, R)$. $\mathcal{B}$ gives C_{ch} as the challenge to $\mathcal{A}$. Observe that the decryption of C_{ch} is

$$V \oplus H_2(e(U, d_A)) = R \oplus H_2(e(rQ_1, \frac{1}{x}Q_2))$$

3. **Phase 2.** $\mathcal{A}$ issues more queries like in Phase 1 except natural constraints and Algorithm $\mathcal{B}$ responds as before.
4. **Guess.** After algorithm $\mathcal{A}$ outputs its guess, $\mathcal{B}$ picks a random tuple(X_i, ζ_i) from H_2list. $\mathcal{B}$ first computes $T = X_i^{1/r}$, and then returns $(T/T_0)^{1/c_0^2}$. Note that $e(P_1, P_2)^{1/x} = (T/T_0)^{1/c_0^2}$ if $T = e(Q_1, Q_2)^{1/x}$. Let H be the event that algorithm $\mathcal{A}$ issues a query for $H_2(e(rQ_1, \frac{1}{x}Q_2))$ at some point during the simulation above. Using the same methods in [5], we can prove the following two claims:

[14] s is the $\mathsf{master-key}$

Claim 1: $Pr[H]$ in the simulation above is equal to $Pr[H]$ in the real attack.
Claim 2: In the real attack we have $Pr[H] \geq 2\epsilon(k)$. Following from the above two claims, we have that $\mathcal{B}$ produces the correct answer with probability at least $2\epsilon(k)/q_2$.

Thus we prove Lemma 3.

From the above three Lemma, we prove Theorem 1.

S Proof for Theorem 19

Proof. Same as the above theorem except in the simulation the role of A and B exchanged.

T Proof for Theorem 20

Proof. We just give the intuition for this theorem. The master-key is s, and delegator's private key is $\frac{1}{s+H_1(ID)}$, the delegatee's private key is $\frac{1}{s+H_1(ID')}$, the re-encryption key is ($\frac{s+H_1(ID')+k_1}{s+H_1(ID)}$ mod p, $\frac{k_2}{s+H_1(ID)}$ mod p, $\frac{k_1}{k_2(s+H_1(ID'))}P_2$). Because the re-encryption key is uniformly distributed in Z_p^*, and the original SK IBE is secure, we can conclude that s can not be disclosed by the proxy, delegatee and delegator's colluding.