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

Xu an Wang, Xiaoyuan Yang

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

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. They predicated that proxy re-encryption and re-signature will play an important role in our life. In 2007, Matsuo proposed the concept of four types of re-encryption schemes: CBE to IBE(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 KGC for proxy re-encryption in identity based setting. We find that if we can introduce the KGC 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:

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, KGC can decrypt every re-encrypted ciphertext for IBE users.We ask question like this: can the malicious KGC not decrypt the re-encryption ciphertext? Surprisingly, the answer is affirmative.We construct such a scheme and prove its security in the standard model. So we give the conclusion that key escrow problem is not unavoidable in re-encryption from CBE to IBE.

2. We propose a proxy re-encryption scheme from IBE to CBE. To the best of our knowledge, this is the first type 3 scheme. We give the security model for proxy re-encryption scheme from IBE to CBE and prove our scheme's security in this model without random oracle.

3. One feature of proxy re-encryption schemes in [29] is that they are all based on BB1 identity based encryption.We ask question like this: can we construct proxy re-encryption schemes based on BB2 identity based encryption? We give affirmative answer to this question. We construct an IBE to IBE proxy re-encryption scheme based on BB2 with the help of KGC and prove its security in the standard model.

4. In [30] there was a conclusion that it is hard to construct proxy re-encryption scheme based on BF and SK IBE. When considering

KGC in the proxy key generation, we can construct a proxy re-encryption scheme based on SK IBE. Interestingly, this proxy re-encryption can achieve IND-ID-CCA2 secure, which makes it is a relative efficient proxy re-encryption scheme with pairing which can achieve CCA2 secure in the literature. But this scheme can not resist DDos attack [38]. We also prove our scheme's security.

5. At last, we give some observations on the difficulty of constructing proxy re-encryption based on BF identity based encryption. Our technique can no longer be used to the BF identity based encryption.

Thus, we almost solve the problem of constructing proxy re-encryption scheme in identity based setting, and we note that our technique maybe can also be used to construct proxy re-signature scheme in identity based setting, which is our further work.

Table of Contents

1 How to Solve Key Escrow Problem in Proxy Re-encryption from CBE to IBE

1.1 Introduction

The concept of proxy re-cryptography comes from the work of Blaze, Bleumer, and Strauss in 1998. The goal of proxy re-encryptiohn 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 re-encryption schemes and discussed its several potential applications.They predicated that re-encryption will play an important role in our life. Since then, many excellent schemes have been proposed,including re-encryption schemes in certificate based setting [11, 23, 27, 28],re-encryption schemes in identity based setting [12, 17, 29, 34]and re-encryption schemes in hybrid setting [29]. Now the IEEE P1363.3 standard working group is setting up a standard with pairing including re-encryption [31].

[**Related Work**]In 2007, Matsuo proposed a new type of re-encryption scheme which can re-encrypt the ciphertext in the certificate based encryption(CBE) setting to one that can be decrypted in identity based settingIBE [29].This scheme sets up an example for constructing re-encryption schemes between CBE and IBE.Now their scheme is being standardized by IEEEP1363.3 working group [31].

[**Our Motivation**]We extend their research in re-encryption from CBE to IBE. As we all know, in IBE setting, KGC can decrypt every user's ciphertext and the key escrow problem seems unavoidable for IBE.There are many good papers on this topic [1, 18]. So we consider the key escrow problem in re-encryption too, we find that the re-encryption scheme in [29]is inherited suffering from this problem. Is this unavoidable for re-encryption from CBE to IBE?

[**Our Contribution**]Our results show that the answer is negative! Actually, this result lies in the difference between IBE and Re-encryption from CBE to IBE. In IBE, KGC allocates private keys for users but In Re-encryption, there is another semi-trusted party "proxy", like the idea in certificateless public cryptography [1, 19], the IBE users can have their own secret key during the re-encryption process.Depending on this secret key, the delegatee can decrypt the re-encrypted ciphertext while KGC no longer can!

We organize our paper as following. In section 2,we revisit the re-encryption scheme from CBE to IBE in [29].In section 3, we propose our new re-encryption scheme from CBE to IBE and show why it solves the key escrow problem.In section 4, we prove our new scheme's security.We give our concluding remarks in section 5.

1.2 Revisit the Re-encryption Scheme from CBE to IBE

The hybrid proxy re-encryption scheme involving the ElGamal-type CBE scheme and the BB-IBE scheme.

– The underlying IBE scheme (BB-IBE scheme):

1. $\mathbf{SetUp_{IBE}(k)}$.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 $g_1 = g^\alpha$,$mk = g_2^\alpha$, and $parms = (g, g_1, g_2, h)$. Let mk be the master-secret key and let $parms$ be the public parameters.
2. $\mathbf{KeyGen_{IBE}(mk, parms, ID)}$. Given $mk = g_2^\alpha$ and ID with $parms$, pick a random $u \in Z_p^*$. Set $sk_{ID} = (d_0, d_1) = (g_2^\alpha(g_1^{ID}h)^u, g^u)$.
3. $\mathbf{Enc_{IBE}(ID, parms, M)}$. 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 $C_{ID} = (g^r, (g_1^{ID}h)^r, Me(g_1, g_2)^r) \in G^2 \times G_1$.
4. $\mathbf{Dec_{IBE}(sk_{ID}, parms, C_{ID})}$. Given ciphertext $C_{ID} = (C_1, C_2, C_3)$ and the secret key $sk_{ID} = (d_0, d_1)$ with $prams$, compute $M = C_3e(d_1, C_2)/e(d_0, C_1)$.

– The underlying CBE scheme (ElGamal-type CBE scheme):
  1. $\mathbf{KeyGen_{CBE}(k, parms)}$. 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. $\mathbf{Enc_{CBE}(pk, parms, M)}$. Given $pk = (g_3, g_4, g_5)$ and a message M with $parms$, pick a random $r \in Z_p^*$ and compute $C_{PK} = (g_3^r, g_4^r, g_5^r, Me(g_1, g_2)^r$ $\in G^3 \times G_1$.
  3. $\mathbf{Dec_{CBE}(sk, parms, C_{PK})}$. 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. $\mathbf{EGen(sk_{ID}, parms)}$. 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. $\mathbf{KeyGen_{PRO}(sk, e_{ID}, parms)}$. Given $sk = (\theta, \beta, \delta)$ and $e_{ID} = g^u$ for ID with $parms$, set $rk_{ID} = (\theta, g^{u/\beta}, \delta)$.
  3. $\mathbf{ReEnc(rk_{ID}, parms, C_{PK}, ID)}$. Given a CBE ciphertext $C_{PK} = (C_1, C_2, C_3, C_4)$, the re-encryption key $rk_{ID} = (\theta, g^{u/\beta}, \delta)$ and ID with $parms$, re-encrypt the ciphertext C_{PK} into C_{ID}as follows. $C_{ID} = (C_1', C_2', C_3') = (C_1^{1/\theta}, C_3^{1/\delta}, C_4e(g^{u/\beta}, C_2^{ID})) \in G^2 \times G_1$.
  4. $\mathbf{Check(parms, C_{PK}, pk)}$. 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.

In this scheme,KGC 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-key generation, he certainly can decrypt the re-encryption ciphertext if the delegatee can!

1.3 Our New Re-encryption Scheme from CBE to IBE Which Can Resist Malicious KGC Attack

Our scheme shares the same underlying CBE scheme (ElGamal-type CBE scheme) as [29] scheme. The difference lies in the underlying IBE scheme (BB-IBE scheme) and delegation scheme.

– The underlying IBE scheme (BB-IBE scheme):

1. $\mathbf{SetUp_{IBE}(k)}$.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 $g_1 = g^\alpha$,$mk = g_2^\alpha$, and $parms = (g, g_1, g_2, h)$. Let mk be the master-secret key and let $parms$ be the public parameters.
2. $\mathbf{KeyGen_{IBE}(mk, parms, ID)}$. Given $mk = g_2^\alpha$ and ID with $parms$, pick a random $u \in Z_p^*$. Set $sk_{ID} = (d_0, d_1) = (g_2^\alpha (g_1^{ID} h)^u, g^u)$.
3. $\mathbf{Enc_{IBE}(ID, parms, M)}$. 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 $C_{ID} = (g^r, (g_1^{ID} h)^r, Me(g_1, g_2)^r) \in G^2 \times G_1$.
4. $\mathbf{Dec1_{IBE}(sk_{ID}, parms, C_{ID})}$. Given normal ciphertext $C_{ID} = (C_1, C_2, C_3)$ and the secret key $sk_{ID} = (d_0, d_1)$ with $prams$, compute $M = \frac{C_3 e(d_1, C_2)}{e(d_0, C_1)}$.
5. $\mathbf{Dec2_{IBE}(sk_{ID}, parms, C_{ID})}$. Given re-encryption ciphertext $C_{ID} = (C_1, C_2, C_3)$, $sk_{ID} = (d_0, d_1, k)$, $prams$, compute $M = (\frac{C_3 C_4^k e(d_1, C_2^k)}{e(d_0, C_1^k)})^{\frac{1}{k}}$.

– The underlying CBE scheme (ElGamal-type CBE scheme):
  1. $\mathbf{KeyGen_{CBE}(k, parms)}$. Given a security parameter k, $parms$, pick a random $\theta, \beta, \delta \in Z_p$. Set $g_3 = g^\theta$,$g_4 = g_1^\beta$ and $g_5 = h^\delta$. The public key is $pk = (g_3, g_4, g_5)$. The secret random key is $sk = (\theta, \beta, \delta)$.
  2. $\mathbf{Enc_{CBE}(pk, parms, M)}$. Given $pk = (g_3, g_4, g_5)$ and a message M with $parms$, pick a random $r \in Z_p^*$ and compute $C_{PK} = (g_3^r, g_4^r, g_5^r, Me(g_1, g_2)^r)$ $\in G^3 \times G_1$.
  3. $\mathbf{Dec_{CBE}(sk, parms, C_{PK})}$. 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. $\mathbf{EGen(sk_{ID}, parms)}$. Given $sk_{ID} = (d_0, d_1) = (g_2^\alpha (g_1^{ID} h)^u, g^u)$ for ID with $parms$, 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 $k = H(pk, ID, r)$ set $(d_0', d_1') = (d_0, d_1^k) = (g_2^\alpha (g_1^{ID} h)^u, g^{ku})$,set $e_{ID} = d_1' = g^{ku}$.The user's real private key is $sk_{ID} = (d_0', d_1', k)$.
  2. $\mathbf{KeyGen_{PRO}(sk, e_{ID}, parms)}$. The delegator given input $e_{ID} = g^{uk}, sk = (\theta, \beta, \delta), , parms$, he chooses randomly $t \in Z_p^*$ and set it as the $trankey$ and output $rk_{ID} = (1/t\theta, g^{ku/\beta}, 1/\delta)$.
  3. $\mathbf{Ciphertext - Transformation(C_{PK}, Trankey)}$. Given a CBE ciphertext $C_{PK} = (C_1, C_2, C_3, C_4)$, the delegator transforms $C_{PK} = (C_1, C_2, C_3, C_4)$ into $C_{PK}' = (C_1^t, C_2, C_3, C_4)$ and sends (C_{PK}', g_3^t) to the proxy.
  4. $\mathbf{ReEnc(rk_{ID}, parms, C_{PK}, ID)}$. Given a CBE ciphertext $C_{PK}' = (C_1^t, C_2, C_3, C_4)$, the re-encryption key $rk_{ID} = (1/t\theta, g^{ku/\beta}, 1/\delta)$ and ID with $parms$, re-encrypt the ciphertext C_{PK}into C_{ID} as follows. $C_{ID} = (C_1', C_2', C_3', C_4') = (C_1^{t/t\theta}, C_3^{1/\delta}, e(g^{ku/\beta}, C_2^{ID}), C_4) \in G^2 \times G_1^2$.
  5. $\mathbf{Check(parms, C_{PK}, pk)}$. Given $C_{PK} = (C_1, C_2, C_3, C_4)$ and $pk = (g_3, g_4, g_5)$ with $parms$, set $v_1 = e(C_1^t, g_4)$, $v_2 = e(C_2, g_3^t)$, $v_3 = e(C_2, g_5)$ and $v_4 = e(C_3, g_4)$. If $v_1 = v_2$ and $v_3 = v_4$ then output 1, otherwise output 0.

We verify correctness of our scheme. Following the $Dec2_{IBE}(sk_{ID}, parms, C_{ID})$ scheme, we have

$$\begin{aligned}(\frac{C_3C_4^k e(d_1,C_2^k)}{e(d_0,C_1^k)})^{\frac{1}{k}} &= (\frac{e(g^{ku/\beta},C_2^{ID})M^k e(g_1,g_2)^k e(g^u,h^{kr})}{e(g_2^{\alpha}(g_1^{ID}h)^u,g^{rk})})^{\frac{1}{k}} \\ &= (\frac{M^k e(g_1,g_2)^k e(g^{uk},(g_1^{ID}h)^r)}{e(g_2^{\alpha}(g_1^{ID}h)^u,g^{rk})})^{\frac{1}{k}} \\ &= (\frac{M^k e(g_1,g_2)^k}{e(g_2^{\alpha},g^{rk})})^{\frac{1}{k}} \\ &= (M^k)^{\frac{1}{k}} \\ &= M\end{aligned}$$

Although our scheme can resolve the key escrow problem in proxy re-encryption from CBE to IBE, there are still some issues we must consider.

Remark 1. In our scheme, the decryption algorithm has two different procedure for two level ciphertext. But how can the decryption algorithm distinguish them? We give a very simple solution. The proxy can sign the re-encryption ciphertext. Assuming the proxy has private ,public key and signature algorithm(sk, vk, Σ),then the proxy can sign the re-encryption ciphertext as $\Sigma_{sk}(c)$, thus everyone can verify the ciphertext and distinguish it from normal ciphertext.

Remark 2. In our scheme, every IBE user has a self generated private key k.It's this k that can make our scheme resist KGC decrypting every user's ciphertext.We can see that even if KGC and proxy collude, he yet still can not decrypt the ciphertext.

1.4 Security Models for Re-encryption from CBE to IBE Which Can Resist Malicious KGC Attack

In this section,we first give our security model for re-encryption schemes from CBE to IBE. We then give the security proof for our scheme in this new model.As [29], we just prove our scheme's IND-ID-CPA security. For achieving CCA2 security, we can fellow the technique in [17]. We can see the re-encryption scheme from CBE to IBE in figure 1.

First we define the following oracles, which can be invoked multiple times in any order, subject to the constraints list in the various definition:

- **Uncorrupted user's key generation (O_{keygen}):** Obtain a new key pair as $(pk, sk) \leftarrow KeyGen_{CBE}(1^k)$. A is given pk .
- **Corrupted user's key generation ($O_{corkeygen}$):** Obtain a new key pair as $(pk, sk) \leftarrow KeyGen_{CBE}(1^k)$. Obtain $sk_{ID} \leftarrow KeyGen_{IBE}(mk, parms, ID)$.$A$ is given $(pk, sk), sk_{ID}$.
- **Re-encryption key generation ($O_{rekeygen}$)**:On input (pk, ID) by the adversary, where pk was generated before by $KeyGen$ and ID is a user in IBE setting, return the re-encryption key $rk_{ID} = KeyGen_{PRO}(sk, e_{ID}, parms)$ where sk is the secret keys that correspond to pk and e_{ID} is the delegatee's input for re-encryption key generation purpose.

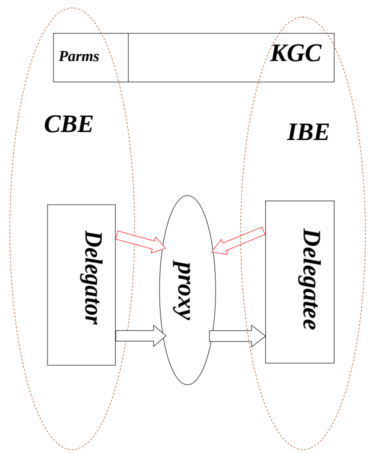

Fig. 1. Proxy re-encryption from CBE to IBE

- **Encryption oracle**$O_{enc_{IBE},enc_{CBE}}$**:** For IBE users, to encrypt a message $M \in G_1$ under the public key $ID \in Z_p^*$, return $Enc_{IBE}(ID, parms, M)$. For CBE users, given pk and a message M with $parms$, return $Enc_{CBE}(pk, parms, M)$.
- **Re-encryption** O_{renc}**:** Output the re-encrypted ciphertext $ReEnc(rk_{ID}, parms, C_{PK}, ID)$.

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 delegation partner, KGC, 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 re-encryption from CBE to IBE, KGC knows every IBE's normal secret key, so for every level 1 normal ciphertext, KGC can decrypt every normal ciphertext. Thus we consider the case that proxy and/or delegator are corrupted. We can see the intuition from the top left corner in figure 2. In this case, we consider the case that malicious CBE users and malicious proxy colludes.

Definition 1. (IBE-Level1-IND-ID-CPA) *A PRE scheme from CBE to IBE is level1-IND-ID-CPA secure if the probability*

$Pr[sk_{ID^\star} \leftarrow O_{keygen}(\lambda), \{(pk_x, sk_x) \leftarrow O_{corkeygen}(\lambda)\}, \{sk_{ID_x} \leftarrow O_{corkeygen}(\lambda)\},$
$\{(pk_h, sk_h) \leftarrow O_{keygen}(\lambda)\}, \{sk_{ID_h} \leftarrow O_{keygen}(\lambda)\},$
$\{R_{hx} \leftarrow O_{rekeygen}(sk_h, ID_x)\}, \{R_{xh} \leftarrow O_{rekeygen}(sk_x, ID_h)\},$
$\{R_{h\star} \leftarrow O_{rekeygen(sk_h, ID^\star)}\},$
$(m_0, m_1, St) \leftarrow A^{O_{renc}, O_{enc_{IBE}}}_{O_{enc_{CBE}}}(ID^\star, \{(pk_x, sk_x)\}, \{sk_{ID_x}\}, \{(pk_h, sk_h)\}, \{R_{xh}\},$
$\{R_{hx}\}),$

$d^\star \xleftarrow{R} \{0,1\}, C^\star = enc_{IBE}(m_{d^\star}, ID^\star), d' \leftarrow A^{O_{renc}, O_{enc_{IBE}}, O_{enc_{CBE}}}(C^\star, St)$:
$d' = d^\star]$
is negligibly close to $1/2$ *for any PPT adversary* A. *In our notation,* St *is a state information maintained by* A *while* $ID^\star$ *is the target user's pubic and private key pair, the challenger also chooses other keys for corrupt and honest parties. For other honest parties, keys are subscripted by* h *or* h' *and we subscript corrupt keys by* x *or* x'. *In the game,* A *is said to have advantage* ϵ *if this probability, taken over random choices of* A *and all oracles, is at least* $1/2 + \epsilon$.

In re-encryption from CBE to IBE, even KGC knows every IBE's normal secret key, but he does not the local secret key k, so malicious may no longer learn the re-encryption ciphertext. But for the delegator, he certainly can decrypt the ciphertext which will be re-encrypted. Thus we consider only the case that proxy and/or KGC are corrupted, We must point out this model is not considered in the previous literature.

We can see the intuition from the top right corner in figure 2.In this case, we consider the malicious KGC and malicious proxy colluding. The goal of this paper is to construct such a scheme resisting malicious KGC attack.

Definition 2. (IBE-Level2-IND-ID-CPA) *A PRE scheme from CBE to IBE is level2-IND-ID-CPA secure if the probability*
$Pr[(parms, master-key) \leftarrow O_{KGCsetup}(\lambda), sk_{ID^\star} \leftarrow O_{keygen}(\lambda), (pk^\star, sk^\star) \leftarrow$
$O_{keygen}(\lambda),$
$\{(pk_x, sk_x) \leftarrow O_{corkeygen}(\lambda)\}, \{sk_{ID_x} \leftarrow O_{corkeygen}(\lambda)\},$
$\{(pk_h, sk_h) \leftarrow O_{keygen}(\lambda)\}, \{sk_{ID_h} \leftarrow O_{keygen}(\lambda)\},$
$\{R_{\star h} \leftarrow O_{rekeygen(sk^\star, ID_h)}\}, \{R_{\star x} \leftarrow O_{rekeygen(sk^\star, ID_x)}\},$
$\{R_{hx} \leftarrow O_{rekeygen}(sk_h, ID_x)\}, \{R_{xh} \leftarrow O_{rekeygen}(sk_x, ID_h)\},$
$(m_0, m_1, St) \leftarrow A^{O_{renc}, O_{enc_{IBE}}}_{O_{enc_{CBE}}}(ID^\star, \{(pk_x, sk_x)\}, \{sk_{ID_x}\}, \{pk_h\}, \{R_{xh}\}, \{R_{hx}\},$
$\{R_{\star h}\}, \{R_{\star x}\}, \{(parms, master-key)\}),$
$d^\star \xleftarrow{R} \{0,1\}, C^\star = renc(m_{d^\star}, pk^\star, ID^\star), d' \leftarrow A^{O_{renc}, O_{enc_{IBE}}}_{O_{enc_{CBE}}}(C^\star, St) : d' = d^\star]$

is negligibly close to $1/2$ *for any PPT adversary* A. *In the above game, any query to oracle* O_{renc} *which makes the output is* $C^\star$ *is returned with* $\perp$.*In our notation,* St *is a state information maintained by* A *while* $ID^\star$ *is the target IBE user, the challenger also chooses other keys for corrupt and honest parties. For other honest parties, keys are subscripted by* h *or* h' *and we subscript corrupt keys by* x *or* x'. *In the game,* A *is said to have advantage* ϵ *if this probability, taken over random choices of* A *and all oracles, is at least* $1/2 + \epsilon$.

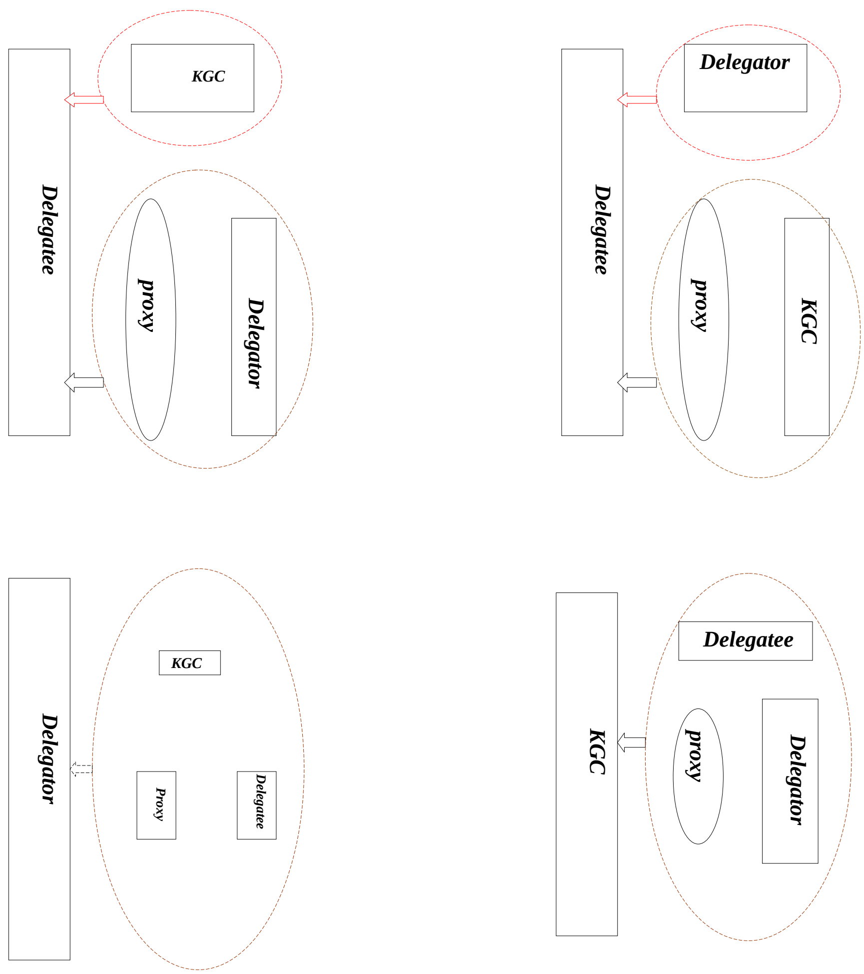

Fig. 2. Security models for internal adversaries

Delegator Security.

In re-encryption from CBE and IBE, the delegator is a CBE user. The re-encryption scheme can not influence CBE 's security. In this case, we consider the delegatee, proxy and KGC are all colluding. We must point out this model is not considered in previous literature. We can see the intuition from the down left corner in figure 2.

Definition 3. (CBE-IND-CPA) *A PRE scheme from CBE to IBE is IND-CPA secure for CBE if the probability*
$Pr[(parms, master-key) \leftarrow O_{KGCsetup}(\lambda), (pk^\star, sk^\star) \leftarrow O_{keygen}(\lambda),$
$\{(pk_x, sk_x) \leftarrow O_{corkeygen}(\lambda)\}, \{sk_{ID_x} \leftarrow O_{corkeygen}(\lambda)\},$
$\{(pk_h, sk_h) \leftarrow O_{keygen}(\lambda)\}, \{sk_{ID_h} \leftarrow O_{keygen}(\lambda)\},$
$\{R_{\star h} \leftarrow O_{rekeygen(sk^\star, ID_h)}\}, \{R_{\star x} \leftarrow O_{rekeygen(sk^\star, ID_x)}\},$
$\{R_{hx} \leftarrow O_{rekeygen}(sk_h, ID_x)\}, \{R_{xh} \leftarrow O_{rekeygen}(sk_x, ID_h)\},$
$(m_0, m_1, St) \leftarrow A^{O_{renc}, O_{enc_{IBE}}}_{O_{enc_{CBE}}}(pk^\star, \{(pk_x, sk_x)\}, \{sk_{ID_x}\}, \{(pk_h, sk_h)\}, \{R_{xh}\},$
$\{R_{hx}\}, \{R_{\star h}\}, \{R_{\star x}\}, \{(parms, master-key)\}),$
$d^\star \xleftarrow{R} \{0,1\}, C^\star = enc_{CBE}(m_{d^\star}, pk^\star), d' \leftarrow A^{O_{renc}, O_{enc_{IBE}}}_{O_{enc_{CBE}}}(C^\star, St) : d' = d^\star]$
is negligibly close to $1/2$ *for any PPT adversary* A*. In our notation,* St *is a state information maintained by* A *while* $(pk^\star, sk^\star)$ *is the target user's pubic and private key pair, the challenger also chooses other keys for corrupt and honest parties. For other honest parties, keys are subscripted by* h *or* h' *and we subscript corrupt keys by* x *or* x'*. In the game,* A *is said to have advantage* ϵ *if this probability, taken over random choices of* A *and all oracles, is at least* $1/2 + \epsilon$.

KGC Security.

In re-encryption from CBE and IBE, KGC's master secret key can not leverage even the delegator, the delegatee and proxy colludes.We must point out this model is not considered in previous literature. We can see the intuition from the down right corner in figure 2.

Definition 4. (KGC-OW) *A PRE scheme from CBE to IBE is secure for KGC if the output*
$Exp[\{(pk_x, sk_x) \leftarrow O_{corkeygen}(\lambda)\}, \{sk_{ID_x} \leftarrow O_{corkeygen}(\lambda)\},$
$\{(pk_h, sk_h) \leftarrow O_{keygen}(\lambda)\}, \{sk_{ID_h} \leftarrow O_{keygen}(\lambda)\},$
$\{R_{xx'} \leftarrow O_{rekeygen(sk_x, ID_{x'})}\}, \{R_{x'x} \leftarrow O_{rekeygen(sk_{x'}, ID_x)}\},$
$\{R_{hx} \leftarrow O_{rekeygen}(sk_h, ID_x)\}, \{R_{xh} \leftarrow O_{rekeygen}(sk_x, ID_h)\},$
$mk \leftarrow A^{O_{renc}, O_{enc_{IBE}}}_{enc_{CBE}}(\{(pk_x, sk_x)\}, \{sk_{ID_x}\}, \{(pk_h, sk_h)\}, \{R_{xh}\}, \{R_{hx}\}, \{R_{xx'}\},$
$\{R_{x'x}\}, \{parms\})$

is not the real $master-key$ *for any PPT adversary* A*.The challenger also chooses other keys for corrupt and honest parties. For other honest parties, keys are subscripted by* h *or* h' *and we subscript corrupt keys by* x *or* x'*.*

1.5 Security Analysis

In this section, we will give our scheme's security results based on the models defined in Sec 1.4. We can see these results in figure 2. In this figure, the entity on the left denotes the target, the three right entities denote the internal adversary. Entities in the circle denote colluders. Red circle means the colluders in this circle can break the security of the target, while the brown circle means not. We give the results below:

- For delegatee's IBE-level1-IND-sID-CPA security, KGC alone can break it, while the proxy and delegator's colluding can not.
- For delegatee's IBE-level2-IND-ID-CPA security, delegator alone can break it, while the proxy and KGC's colluding can not.
- For delegator's CBE-IND-CPA security, the proxy, KGC and delegatee's colluding can not break it.
- For KGC's OW security, even if allowing the proxy, delegator and delegatee collude any way, they can not break the KGC's OW security, that is, they can not get the $master-key$.

Theorem 1. *Suppose the DBDH assumption holds, then our scheme is IBE-Level1-IND-sID-CPA secure for the proxy and delegator's colluding.*

Proof. Suppose A can attack our scheme, we construct an algorithm B solves the DBDH problem in G. On input (g, g^a, g^b, g^c, T), algorithm 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 B works by interacting with A in a selective identity game as follows:

1. **Initialization**. The selective identity game begins with A first outputting an identity ID^* that it intends to attack.
2. **Setup**.To generate the system's parameters, algorithm B picks $\alpha' \in Z_p$ at random and defines $h = g_1^{-ID^*} g^{\alpha'} \in G$. It gives A the parameters $params = (g, g_1, g_2, h)$. Note that the corresponding $master-key$, which is unknown to B, is $g_2^a = g^{ab} \in G^*$. 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 A the public key $pk_i = (g_{i_1}, g_{i_2}, g_{i_3})$.
3. **Phase 1**
   - *"A issues up to private key queries on ID_i."* B selects randomly $r_i \in {Z_p}^*$ and $k' \in Z_p$, sets $sk_{ID_i} = (d_0, d_1, d_2) = (g_2^{\frac{-\alpha'}{ID_i - ID^*}} (g_1^{(ID_i - ID^*)} g^{\alpha'})^{r_i}, g_2^{\frac{-1}{ID_i - ID^*}} g^{r_i}, k')$. We claim sk_{ID_i} is a valid random private key for ID_i. To see this, let $\widetilde{r_i} = r_i - \frac{b}{ID - ID^*}$. Then we have that
     $d_0 = g_2^{\frac{-\alpha'}{ID_i - ID^*}} (g_1^{(ID_i - ID^*)} g^{\alpha})^{r_i} = g_2^{\alpha} (g_1^{(ID_i - ID^*)} g^{\alpha})^{r_i - \frac{b}{ID - ID*}} = g_2^a (g_1^{ID_i} h)^{\widetilde{r_i}}$.
     $d_1 = g_2^{\frac{-1}{ID_i - ID^*}} g^{r_i} = g^{\widetilde{r_i}}$.
   - *"A issues up to private key queries on pk_i".* B returns (x_i, y_i, z_i).
   - *"A issues up to rekey generation queries on (pk_j, ID_i)".* The challenge B computes $rk_{pk \to id} = (k'/x_j, (g_2^{\frac{-1}{ID_i - ID^*}} g^{r_i})^{\frac{k'}{y_j}}, k'/z_j)$ and returns it to A.

- *"A issues up to rekey generation queries on* (pk_j, ID^*)*"*. The challenge B randomly choose a $k' \in Z_p$, and computes $rk_{pk_j \to ID^*} = (k'/x_j, (g^{u'})^{k'/y_j}, k'/z_j)$ where u' is a randomly choose from Z_p^* and returns it to A.
- *"A issues up to re-encryption queries on* (C, pk_j, ID_i) *or* (C, pk_j, ID^*)*"* The challenge B runs $ReEnc(rk_{pk_j \to ID_i}, C, pk_j, ID_i)$ or $ReEnc(rk_{pk_j \to ID^*}, C, pk_j, ID^*)$ and return the results.

4. **Challenge** When A decides that Phase1 is over, it outputs two messages $M_0, M_1 \in G$. Algorithm 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** A issues queries as he does in Phase 1 excepts natural constraints.
6. **Guess** Finally, A outputs a guess $b' \in \{0,1\}$. Algorithm B concludes its own game by outputting a guess as follows. If $b = b'$, then 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 A's advantage for breaking the scheme is same as B's advantage for solving DBDH problem. ∎

Theorem 2. *Our scheme is IBE-Level2-IND-ID-CPA secure for the proxy and KGC's colluding.*

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

1. **Setup**.To generate the system's parameters, the challenger B picks $\alpha \in Z_p$, it randomly choose $x \in Z_q^*$, computes $h = g^x$ and computes $g_1 = g^\alpha$, it randomly choose $y \in Z_q^*$ and computes $g_2 = g^y$, it also computes $master-key = g_2^\alpha$. It gives $params = (g, g_1, g_2, h)$ to A.
2. **Phase 1**
   - *"A issues up to master-key query "*. The challenger B returns (α, g_2^α).
   - *"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 $sk_{ID} = (d_0, d_1, d_2) = (g_2^\alpha (g_1^{ID} h)^u, g^u, k')$.
   - *"A issues up to private key queries on* pk*"*. B returns (θ, β, δ).
   - *"A issues up to rekey generation queries on* (pk, ID)*"*. The challenge 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 A.
   - *"A issues up to re-encryption queries on* (C, pk, ID)*"*. The challenge B runs $ReEnc(rk_{pk \to ID}, C, pk, ID)$ and return the results.
3. **Challenge** When A decides that Phase1 is over, it outputs two messages $M_0, M_1 \in G$ and the attack identity ID^*, Algorithm B picks g^u as the ID^*'s second item of its private key, he picks a random bit b and $r, k^* \in Z_p^*$ responds with the ciphertext $C = (g^r, h^r, e(g^{k^* u}, g_1^{IDr}), M_b \cdot e(g_2, (g^{r)^\alpha)}))$. Hence if k^* is the real secret key of ID^*, then C is a valid encryption of M_b under ID^*. Otherwise, C is independent of b in the adversary's view.
4. **Phase2** A issues queries as he does in Phase 1 except natural constraints.
5. **Guess** Finally, A outputs a guess $b' \in \{0,1\}$. Algorithm B concludes its own game by outputting a guess as follows. If $b = b'$, then B outputs 1. Otherwise it outputs 0.

Thus the maximal probability of A successes is $1/p$, which is negligible. ∎

Theorem 3. *Our scheme is CBE-IND-CPA secure for the proxy, KGC and delegatee's colluding except the case of the target CBE ciphertext has been re-encrypted by the proxy.*

Proof. In this case, the KGC and delegatee's colluding just likes [29]'s proxy re-encryption scheme from CBE to IBE, the proof is the same as [29]. ∎

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

Proof. Suppose the target CBE ciphertext is $C'_{PK} = (C_1^t, C_2, C_3, C_4)$ and has been re-encrypted by proxy to be $C_{ID} = (C'_1, C'_2, C'_3, C'_4) = (C_1^{t/t\theta}, C_3^{1/\delta}, e(g^{ku/\beta}, C_2^{ID}), C_4)$, the KGC can decrypt the ciphertext as following. Because $C'_1 = g^r$, he can compute $w = g^{r\alpha}$, so he can get the plaintext by

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

Thus we prove this theorem. ∎

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

Proof. We just give the intuition for this theorem. When considering the proxy, delegatee and delegator's colluding, the KGC only interact with delegatee, that is, its IBE users. And we know the BB_1 identity based encryption is secure under DBDH assumption. That's imply the attacker can not recover the KGC's $master - key$. Thus we prove this theorem. ∎

1.6 Conclusion

In 2007, Matsuo proposed a new type of re-encryption scheme which can re-encrypt the ciphertext in the certificate based encryption(CBE) setting to one that can be decrypted in identity based setting [29].Now this scheme is being standardized by IEEEP1363.3 working group [31]. In this paper, we further extend their research. One feature of their scheme is that it inherits the key escrow problem from IBE, that is, KGC can decrypt every re-encrypted ciphertext for IBE users.We ask question like this: can the malicious KGC not decrypt the re-encryption ciphertext? Surprisingly, the answer is affirmative.We construct such a scheme and prove its security. So we give our conclusion that key escrow problem is not unavoidable in re-encryption from CBE to IBE.

2 Proxy Re-encryption Scheme from IBE to CBE

2.1 Introduction

The concept of proxy re-cryptography comes from the work of Blaze, Bleumer, and Strauss in 1998. The goal of proxy re-encryptiohn 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 proxy re-encryption schemes and discussed its several potential applications. Since then, many excellent schemes have been proposed,including proxy re-encryption schemes in certificate based setting [11, 23, 27, 28],re-encryption schemes in identity based setting [12,17,29,34]and proxy re-encryption schemes in hybrid setting [29]. Now the IEEE P1363.3 standard working group is setting up a standard with pairing including proxy re-encryption [31].

[**Related Work**]In 2007, Matsuo proposed a new type of proxy re-encryption scheme which can re-encrypt the ciphertext in the certificate based encryption(CBE) setting to one that can be decrypted in identity based setting [29].This scheme sets up an example for constructing proxy re-encryption schemes between CBE and IBE. Now their scheme is being standardized by IEEEP1363.3 working group [31].

[**Our Motivation and Contribution**]We follow the research in [29], that is, can we construct a re-encryption scheme from IBE to CBE? We answer this question affirmatively. Surprisingly, if we consider the help of KGC when generating re-encryption key in Matsuo's proxy re-encryption from CBE to IBE, we find that it is easy to construct a proxy re-encryption scheme from IBE to CBE. We believe that introducing the KGC in re-encryption is not unreasonable. As we all know, the KGC plays an important role in IBE. Specifically, the KGC can know every IBE user's private key and thus can decrypt every IBE user's ciphertext. So it's reasonable to introduce KGC for re-encryption key generating in proxy re-encryption in IBE setting.

We organize our paper as following. In section 2, we revisit the proxy re-encryption from CBE to IBE proposed in [29]. In section 3, we propose our proxy re-encryption scheme from IBE to CBE .In section 4, we give the security model for our scheme. and we prove our scheme's security in the model.We give our conclusion in section 5.

2.2 Revisit the Proxy Re-encryption Scheme from CBE to IBE

The proxy re-encryption scheme from CBE to IBE involves the ElGamal-type CBE scheme and the BB-IBE scheme.

- The underlying CBE scheme (ElGamal-type CBE scheme):
  1. $\mathbf{KeyGen_{CBE}(k, parms)}$. 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. $\mathbf{Enc_{CBE}(pk, parms, M)}$. Given $pk = (g_3, g_4, g_5)$ and a message M with $parms$, pick a random $r \in Z_p^*$ and compute $C_{PK} = (g_3^r, g_4^r, g_5^r, Me(g_1, g_2)^r \in G^3 \times G_1$.
3. $\mathbf{Dec_{CBE}(sk, parms, C_{PK})}$. 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 underlying IBE scheme (BB-IBE scheme):
1. $\mathbf{SetUp_{IBE}(k)}$.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 $g_1 = g^\alpha$,$mk = g_2^\alpha$, and $parms = (g, g_1, g_2, h)$. Let mk be the master-secret key and let $parms$ be the public parameters.
2. $\mathbf{KeyGen_{IBE}(mk, parms, ID)}$. Given $mk = g_2^\alpha$ and ID with $parms$, pick a random $u \in Z_p^*$. Set $sk_{ID} = (d_0, d_1) = (g_2^\alpha(g_1^{ID}h)^u, g^u)$.
3. $\mathbf{Enc_{IBE}(ID, parms, M)}$. 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 $C_{ID} = (g^r, (g_1^{ID}h)^r, Me(g_1, g_2)^r) \in G^2 \times G_1$.
4. $\mathbf{Dec_{IBE}(sk_{ID}, parms, C_{ID})}$. Given ciphertext $C_{ID} = (C_1, C_2, C_3)$ and the secret key $sk_{ID} = (d_0, d_1)$ with $prams$, compute $M = C_3e(d_1, C_2)/e(d_0, C_1)$.

– The delegation scheme:
1. $\mathbf{EGen(sk_{ID}, parms)}$. 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. $\mathbf{KeyGen_{PRO}(sk, e_{ID}, parms)}$. Given $sk = (\theta, \beta, \delta)$ and $e_{ID} = g^u$ for ID with $parms$, set $rk_{ID} = (\theta, g^{u/\beta}, \delta)$.
3. $\mathbf{ReEnc(rk_{ID}, parms, C_{PK}, ID)}$. Given a CBE ciphertext $C_{PK} = (C_1, C_2, C_3, C_4)$, the re-encryption key $rk_{ID} = (\theta, g^{u/\beta}, \delta)$ and ID with $parms$, re-encrypt the ciphertext C_{PK} into C_{ID}as follows. $C_{ID} = (C_1', C_2', C_3') = (C_1^{1/\theta}, C_3^{1/\delta}, C_4e(g^{u/\beta}, C_2^{ID})) \in G^2 \times G_1$.
4. $\mathbf{Check(parms, C_{PK}, pk)}$. 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.

2.3 Our Proposed Proxy Re-encryption Scheme from IBE to CBE

The proxy re-encryption scheme from IBE to CBE involving the ElGamal-type CBE scheme and the BB1-IBE scheme.

– The underlying IBE scheme (BB1-IBE scheme):
1. $\mathbf{SetUp_{IBE}(k)}$.Given a security parameter k, select a random generator $g \in G$, choose randomly $t_1, t_2 \in Z_q^*$ and computes $g_2 = g^{t_1}, h = g^{t_2}$. Pick a random $\alpha \in Z_p^*$. Set $g_1 = g^\alpha$,$mk = (g_2^\alpha, \alpha, t_1, t_2)$, and $parms = (g, g_1, g_2, h)$. Let (mk, α) be the master-secret key and let $parms$ be the public parameters.
2. $\mathbf{KeyGen_{IBE}(mk, parms, ID)}$. Given $mk = g_2^\alpha$ and ID with $parms$, pick a random $u \in Z_p^*$. Set $sk_{ID} = (d_0, d_1) = (g_2^\alpha(g_1^{ID}h)^u, g^u)$.

3. $\mathbf{Enc_{IBE}}(\mathbf{ID}, \mathbf{parms}, \mathbf{M})$. 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 $C_{ID} = (g^r, (g_1^{ID}h)^r, Me(g_1, g_2)^r) \in G^2 \times G_1$.
4. $\mathbf{Dec_{IBE}}(\mathbf{sk_{ID}}, \mathbf{parms}, \mathbf{C_{ID}})$. Given ciphertext $C_{ID} = (C_1, C_2, C_3)$ and the secret key $sk_{ID} = (d_0, d_1)$ with $prams$, compute $M = C_3e(d_1, C_2)/e(d_0, C_1)$.

– The underlying CBE scheme (ElGamal-type CBE scheme):

1. $\mathbf{KeyGen_{CBE}}(\mathbf{k}, \mathbf{parms})$. Given a security parameter k , $parms$, pick a random $\theta \in Z_p$. Set $g_3 = g_1{}^{\theta}$. The public key is $pk = g_3$. The secret random key is $sk = \theta$.
2. $\mathbf{Enc_{CBE}}(\mathbf{pk}, \mathbf{parms}, \mathbf{M})$. Given $pk = g_3$ and a message M with $parms$, pick a random $r \in Z_p^*$ and compute $C_{PK} = (g_3^r, Me(g_1, g_2)^r) \in G \times G_1$.
3. $\mathbf{Dec1_{CBE}}(\mathbf{sk}, \mathbf{parms}, \mathbf{C_{PK}})$. Given $C_{PK} = (C_1, C_2)$ and the secret key $sk = \theta$ with $parms$, compute $M = C_2/e(C_1^{1/\theta}, g_2)$.
4. $\mathbf{Dec2_{CBE}}(\mathbf{sk}, \mathbf{parms}, \mathbf{C_{PK}})$. Given a normal ciphertext $C_{PK} = (C'_1, C'_2)$ and the secret key $sk = k_2\theta$ with $parms$, compute $M = C'_2/e(C_1'^{1/k_2\theta}, g_2)$.

– The delegation scheme:

1. $\mathbf{ReKeyGen_{PRO}}(\mathbf{ID}, \mathbf{pk})$. The KGC chooses a collision resistent hash function $H : \{0, 1\}^{3|p|} \rightarrow Z_p^*$ and a random seed $n \in Z_p^*$, and computes $k_1 = H(ID, pk, n)$. The KGC computes $\frac{\alpha+k_1}{ID\alpha+t_2}, w = g_2^{k_1}$ and sends it to the proxy. The delegatee choose a randomly k_2, computes $k_2\theta$ and sends it to the proxy. He preserves k_2 for decryption. The proxy sets the re-encryption key $rk = (\frac{\alpha+k_1}{ID\alpha+t_2}, k_2\theta, w)$.We note that the KGC chooses a different k for every different user pair (ID, pk).
2. $\mathbf{ReEnc}(\mathbf{rk_{ID,pk}}, \mathbf{parms}, \mathbf{C_{ID}}, \mathbf{pk})$. Given a IBE ciphertext $C_{ID} = (C_1, C_2, C_3) = (g^r, (g_1^{ID}h)^r, Me(g_1, g_2)^r)$, first run "Check" algorithm, if return "invalid" then "Abort", otherwise, do the following: Given re-encryption key $rk = (\frac{\alpha+k_1}{ID\alpha+t_2}, k_2\theta, w)$, the proxy re-encrypt the ciphertext C_{ID} into C_{pk} as following. $C_{pk} = (C'_1, C'_2) = (C_2^{\frac{\alpha+k_1}{ID\alpha+t_2}\cdot k_2\theta}, C_3e(C_1, w)) \in G \times G_1$.
3. $\mathbf{Check}(\mathbf{parms}, \mathbf{C_{ID}})$. Given $C_{ID} = (C_1, C_2, C_3)$ with $parms$, set $v_1 = e(C_1, g_1^{ID}h)$, $v_2 = e(C_2, g)$. If $v_1 = v_2$ then output "Valid", otherwise output "Invalid".

We can verify its correctness as the following

$$
\begin{aligned}
\frac{C_3e(C_1, w)}{e((C_2^{\frac{\alpha+k_1}{ID\alpha+t_2}\cdot k_2\theta})^{\frac{1}{k_2\theta}}, g_2)} &= \frac{Me(g_1, g_2)^r e(g^r, w)}{e(((g_1^{ID}h)^{r\cdot\frac{\alpha+k_1}{ID\alpha+t_2}\cdot k_2\theta})^{\frac{1}{k_2\theta}}, g_2)} \\
&= \frac{Me(g_1, g_2)^r e(g^r, g_2^{k_1})}{e((g_1^{ID}h)^{r\cdot\frac{\alpha+k_1}{ID\alpha+t_2}}, g_2)} \\
&= \frac{Me(g_1, g_2)^r e(g^r, g_2^{k_1})}{e(g^{(\alpha+k_1)r}, g_2)}
\end{aligned}
$$

$$= \frac{Me(g_1,g_2)^r e(g^r, g_2^{k_1})}{e(g^{\alpha r}, g_2)e(g^{k_1 r}, g_2)}$$
$$= \frac{Me(g_1,g_2)^r e(g^r, g_2^{k_1})}{e(g_1,g_2)^r e(g^r, g_2^{k_1})}$$
$$= M$$

Remark 3. In our scheme, we must note that the KGC computes a different k for every different user pair (ID, pk). Otherwise, if the adversary know $\frac{\alpha+k_1}{ID\alpha+t_2}$ for three different ID_1, ID_2, ID_3 but one k and pk, he can compute α, t_2, which is not secure of course.

2.4 Security Models for Proxy Re-encryption from IBE to CBE

First we define the following oracles, which can be invoked multiple times in any order, subject to the constraints list in the various definition:

- **Uncorrupted user's key generation (O_{keygen}):** Obtain a new key pair as $(pk, sk) \leftarrow KeyGen_{CBE}(1^k)$. A is given pk .
- **Corrupted user's key generation ($O_{corkeygen}$):** Obtain $sk_{ID} \leftarrow KeyGen_{IBE}(mk, parms, ID)$. Obtain a new key pair as $(pk, sk) \leftarrow Key\ Gen_{CBE}(1^k)$. A is given $sk_{ID}, (pk, sk)$.
- **Re-encryption key generation ($O_{rekeygen}$):**On input (ID, pk) by the adversary, where pk was generated before by $KeyGen$ and ID is a user in IBE setting, return the re-encryption key $ReKeyGen_{PRO}(ID, pk)$.
- **Encryption oracle($O_{enc_{IBE}, enc_{CBE}}$):** For IBE users, to encrypt a message $M \in G_1$ under the public key $ID \in Z_p^*$, return $Enc_{IBE}(ID, parms, M)$. For CBE users, given pk and a message M with $parms$, return $Enc_{CBE}(pk, parms, M)$.
- **Re-encryption** (O_{renc})**:** Output the re-encrypted ciphertext $ReEnc(rk_{ID,pk}, parms, C_{ID}, pk)$.

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 delegation partner, KGC, or some collusion between them (Internal Security). Generally speaking, internal adversaries are more powerful than external adversaries.We give the security models as following.
Delegator Security.

Definition 5. (IBE-IND-ID-CPA) *A PRE scheme from IBE to CBE is IBE-IND-ID-CPA secure if the probability*
$Pr[sk_{ID^\star} \leftarrow O_{keygen}(\lambda), \{(pk_x, sk_x) \leftarrow O_{corkeygen}(\lambda)\}, \{sk_{ID_x} \leftarrow O_{corkeygen}(\lambda)\},$
$\{(pk_h, sk_h) \leftarrow O_{keygen}(\lambda)\}, \{sk_{ID_h} \leftarrow O_{keygen}(\lambda)\},$
$\{R_{hx} \leftarrow O_{rekeygen}(ID_h, sk_x)\}, \{R_{xh} \leftarrow O_{rekeygen}(ID_x, sk_h)\},$
$\{R_{\star h} \leftarrow O_{rekeygen(ID^\star, sk_h)}\}, \{R_{\star x} \leftarrow O_{rekeygen(ID^\star, sk_x)}\}$
$(m_0, m_1, St) \leftarrow A^{O_{renc}, O_{enc_{IBE}}}_{O_{enc_{CBE}}}(ID^\star, \{(pk_x, sk_x)\}, \{sk_{ID_x}\}, \{(pk_h, sk_h)\}, \{R_{xh}\},$

$\{R_{hx}\}, \{R_{\star h}\}, \{R_{\star x}\}),$
$d^\star \xleftarrow{R} \{0,1\}, C^\star = enc_{IBE}(m_{d^\star}, ID^\star), d' \leftarrow A^{O_{renc}, O_{enc_{IBE}}, O_{enc_{CBE}}}(C^\star, St):$
$d' = d^\star]$
is negligibly close to 1/2 for any PPT adversary A. In the above game, any query to oracle O_{renc} which makes the output is $C^\star$ is returned with $\perp$. In our notation, St is a state information maintained by A while $sk^\star$ is the target user's pubic and private key pair, the challenger also chooses other keys for corrupt and honest parties. For other honest parties, keys are subscripted by h or h' and we subscript corrupt keys by x or x'. In the game, A is said to have advantage ϵ if this probability, taken over random choices of A and all oracles, is at least $1/2 + \epsilon$.

Delegatee Security.

Definition 6. (CBE-IND-CPA) *A PRE scheme from IBE to CBE is CBE-IND-CPA secure for CBE if the probability*
$Pr[(parms, master - key) \leftarrow O_{KGCsetup}(\lambda), (pk^\star, sk^\star) \leftarrow O_{keygen}(\lambda),$
$\{(pk_x, sk_x) \leftarrow O_{corkeygen}(\lambda)\}, \{sk_{ID_x} \leftarrow O_{corkeygen}(\lambda)\},$
$\{(pk_h, sk_h) \leftarrow O_{keygen}(\lambda)\}, \{sk_{ID_h} \leftarrow O_{keygen}(\lambda)\},$
$\{R_{h\star} \leftarrow O_{rekeygen(ID_h, sk^\star)}\}, \{R_{x\star} \leftarrow O_{rekeygen(ID_x, sk^\star)}\},$
$\{R_{hx} \leftarrow O_{rekeygen}(ID_h, sk_x)\}, \{R_{xh} \leftarrow O_{rekeygen}(ID_x, sk_h)\},$
$(m_0, m_1, St) \leftarrow A^{O_{renc}, O_{enc_{IBE}}}_{O_{enc_{CBE}}}(pk^\star, \{(pk_x, sk_x)\}, \{sk_{ID_x}\}, \{(pk_h, sk_h)\}, \{R_{xh}\},$
$\{R_{hx}\}, \{R_{\star h}\}, \{R_{\star x}\}, \{(parms, master - key)\}),$
$d^\star \xleftarrow{R} \{0,1\}, C^\star = enc_{CBE}(m_{d^\star}, pk^\star), d' \leftarrow A^{O_{renc}, O_{enc_{IBE}}, O_{enc_{CBE}}}(C^\star, St):$
$d' = d^\star]$
is negligibly close to 1/2 for any PPT adversary A. In the above game, any query to oracle O_{renc} which makes the output is $C^\star$ is returned with $\perp$. In our notation, St is a state information maintained by A while $(pk^\star, sk^\star)$ is the target user's pubic and private key pair, the challenger also chooses other keys for corrupt and honest parties. For other honest parties, keys are subscripted by h or h' and we subscript corrupt keys by x or x'. In the game, A is said to have advantage ϵ if this probability, taken over random choices of A and all oracles, is at least $1/2 + \epsilon$.

KGC Security.

In proxy re-encryption from IBE to CBE, KGC's master key can not leverage even if the delegator, the delegatee and proxy collude.

Definition 7. (KGC-OW) *A PRE scheme from IBE to CBE is secure for KGC if the*
$Pr[\{(pk_x, sk_x) \leftarrow O_{corkeygen}(\lambda)\}, \{sk_{ID_x} \leftarrow O_{corkeygen}(\lambda)\},$
$\{(pk_h, sk_h) \leftarrow O_{keygen}(\lambda)\}, \{sk_{ID_h} \leftarrow O_{keygen}(\lambda)\},$
$\{R_{xx'} \leftarrow O_{rekeygen(ID_{x'}, sk_x)}\}, \{R_{x'x} \leftarrow O_{rekeygen(ID_x, sk_{x'})}\},$
$\{R_{hx} \leftarrow O_{rekeygen}(ID_h, sk_x)\}, \{R_{xh} \leftarrow O_{rekeygen}(ID_x, sk_h)\},$

$mk' \leftarrow A^{O_{renc}, O_{enc_{IBE}}}_{enc_{CBE}}(St, \{(pk_x, sk_x)\}, \{sk_{ID_x}\}, \{(pk_h, sk_h)\}, \{R_{xh}\}, \{R_{hx}\}, \{R_{xx'}\}, \{R_{x'x}\}, \{parms\}) : mk = mk']$

is is negligibly close to 0 for any PPT adversary A. In our notation, St is a state information maintained by A, For the honest parties, keys are subscripted by h or h′ and we subscript corrupt keys by x or x′.

2.5 Security Analysis

In this section, we will give our scheme's security results based on the models defined in the above section. We give the results below:

- For delegator's IBE-IND-sID-CPA security, the proxy and delegatee's colluding can not break it.
- For delegatee's CBE-IND-CPA security, the KGC , delegator and proxy's colluding can not break it.
- For KGC's OW security, even if allowing the proxy, delegator and delegatee collude in any way, they can not break the KGC's OW security, that is, they can not get the $master - key$.

Now let's prove these security results.

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

Proof. Suppose A can attack our scheme, we construct an algorithm B solves the mDBDH problem in G. On input $(g, g^a, g^{a^2}, g^b, g^c, T)$, algorithm 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 B works by interacting with A in a selective identity game as follows:

1. **Initialization**. The selective identity game begins with A first outputting an identity ID^* that it intends to attack.
2. **Setup**.To generate the system's parameters, algorithm B picks $\alpha' \in Z_p$ at random and defines $h = g_1^{-ID^*} g^{\alpha'} \in G$. It gives A the parameters $params = (g, g_1, g_2, h)$. Note that the corresponding $master - key$, which is unknown to B, is $g_2^a = g^{ab} \in G^*$. B picks random $x_i, y_i, z_i \in Z_p$, computes $g_{i_1} = g^{x_i}$. it gives A the public key $pk_i = g_{i_1}$.
3. **Phase 1**
   - *"A issues up to private key queries on ID_i".* B selects randomly $r_i \in Z_p{}^*$ and $k' \in Z_p$, sets $sk_{ID_i} = (d_0, d_1) = (g_2^{\frac{-\alpha'}{ID_i - ID^*}} (g_1^{(ID_i - ID^*)} g^{\alpha})^{r_i}$, $g_2^{\frac{-1}{ID_i - ID^*}} g^{r_i})$. We claim sk_{ID_i} is a valid random private key for ID_i. To see this, let $\widetilde{r_i} = r_i - \frac{b}{ID - ID^*}$. Then we have that $d_0 = g_2^{\frac{-\alpha'}{ID_i - ID^*}} (g_1^{(ID_i - ID^*)} g^{\alpha})^{r_i} = g_2^{\alpha} (g_1^{(ID_i - ID^*)} g^{\alpha})^{r_i - \frac{b}{ID - ID*}} = g_2^a (g_1^{ID_i} h)^{\widetilde{r_i}}$. $d_1 = g_2^{\frac{-1}{ID_i - ID^*}} g^{r_i} = g^{\widetilde{r_i}}$.
   - *"A issues up to private key queries on pk_i".* B returns x_i.

- *"A issues up to rekey generation queries on (ID, pk_i)"*. The challenge B chooses a randomly $x \in Z_p^*$, sets $rk_{ID,pk1} = x$ and returns it to A. he computes $rk_{ID,pk3} = w = \frac{(g_4^{(ID-ID^*)x} g_1^{\alpha' x}}{g_4}$ and $rk_{ID,pk2} = k' x_i$ where k' chosen randomly from Z_p^*, sends them to the proxy. We have
$$(g_1^{ID} h)^x = g_1 g^{k_1}$$
$$g_1^{k_1} = (\frac{(g_1^{ID} h)^x}{g_1})^{\alpha} = \frac{(g_1^{ID-ID^*} g^{\alpha'})^{\alpha x}}{g_1^{\alpha}} = \frac{(g_4^{(ID-ID^*)x} g_1^{\alpha' x}}{g_4} = w$$
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.
- *"A issues up to re-encryption queries on (C_{ID}, ID, pk_i)"*. The challenge B runs $ReEnc(rk_{ID \to pk_i}, C_{ID}, ID, pk_i)$ and return the results.

4. **Challenge** When A decides that Phase1 is over, it outputs two messages $M_0, M_1 \in G$. Algorithm B picks a random bit b and responds with the ciphertext $C = (g^c, (g^{\alpha\prime})^c, M_b \cdot T)$. 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** A issues queries as he does in Phase 1 except natural constraints.
6. **Guess** Finally, A outputs a guess $b' \in \{0,1\}$. Algorithm B concludes its own game by outputting a guess as follows. If $b = b'$, then 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 A's advantage for breaking the scheme is same as B's advantage for solving mDBDH problem. ∎

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

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 lies in $k_2\theta$. But even if the proxy, delegator and KGC's colluding, they can only get $k_2\theta$ where k_2 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 level1 ciphertext or for CBE level2 ciphertext. ∎

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

Proof. We just give the intuition for this theorem. When considering the proxy, delegatee and delegator's colluding, the KGC only interact with delegator and proxy. The re-encryption key $rk = (\frac{\alpha+k_1}{ID\alpha+t_2}, k_2\theta, w)$ is distributed same as $(x, k, \frac{(g_4^{(ID-ID^*)x} g_1^{\alpha' x}}{g_4})$ where x and k are 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 KGC's $master - key$. Thus our scheme is KGC-OW secure for the proxy, delegatee and delegator's colluding. ∎

2.6 Conclusion

In 2007, Matsuo proposed a new type of re-encryption scheme which can re-encrypt the ciphertext in the certificate based encryption(CBE) setting to one that can be decrypted in identity based setting [29](IBE). In this paper, we try to solve a problem left by [29], that is, can we construct a proxy re-encryption scheme from IBE to CBE? We answer this question affirmatively, we propose the first proxy re-encryption scheme from IBE to CBE with the help of KGC. We also give the security model for proxy re-encryption scheme from IBE to CBE and prove our scheme's security in this model.

3 Proxy Re-encryption Scheme Based on BB2 Identity Based Encryption

3.1 Introduction

The concept of proxy re-cryptography 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 proxy re-encryption schemes and discussed their several potential applications especially in distributed secure storage.They predicated that proxy re-encryption will play an important role in our life [2]. Since then, many excellent schemes have been proposed, including proxy re-encryption schemes in certificate based setting [11, 23, 27, 28],proxy re-encryption schemes in identity based setting [12,17,29,34]and proxy re-encryption schemes in hybrid setting [29]. Now the IEEE P1363.3 standard working group is setting up a standard with pairing including proxy re-encryption [31].

[**Related Work**]In 2007, Matsuo proposed the concept of four types of proxy re-encryption 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]. One feature of their schemes is that they are all based on BB1 identity based encryption [6]. They excluded constructing proxy re-encryption schemes based on BB2 identity based encryption [6] for technique reasons [30].

[**Our Motivation**]We extend their research in proxy re-encryption from IBE to IBE. We follow the framework proposed by Boyen [8], that is, the IBE framework can be divided into three categories. The first kind is "Full Domain Hash" framework [5]; the second is "Exponent Inversion" framework, including the second scheme BB2 in [6]; the third is "Commutative Blinding" framework, including the first scheme BB1 in [6]. This framework is the most flexible which has been used to construct group signature, ring signature and many other useful applications. Also Matsuo's re-encryption schemes lie in this framework. Recently, "Exponent Inversion" framework has found applications in fuzzy IBE, delegation IBE and hierarchical IBE, which makes it much more flexible than previous thought [8]. So we reconsider the problem of constructing proxy re-encryption based on BB2 identity based encryption. Surprisingly, if we consider the help of KGC, then it is easy to construct proxy re-encryption based on BB2 identity based encryption.

[**Our Contribution**]We construct a proxy re-encryption scheme based on BB2 identity based encryption with the help of KGC. As we all know, the KGC plays an important role in IBE. Specifically, the KGC can know every IBE user's private key and thus can decrypt every IBE user's ciphertext. So it's reasonable to give a position to KGC in proxy re-encryption in IBE setting. In our proxy re-encryption scheme, the re-encryption key is generated by the KGC only. The assumption of our scheme is the KGC must be trusted completely, which is a shortcoming. We hope we can reduce this trust in our further research.

We organize our paper as following. In section 2,we revisit the BB2 identity based encryption in [6].In section 3, we propose our new re-encryption scheme from IBE to IBE with the help of KGC. In section 4, we give the security model for our scheme and prove its security in the standard model. We give our conclusion in section 5.

3.2 Revisit the BB2 Identity Based Encryption

Let G be a bilinear group of prime order p and g be a generator of 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 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 $params$ and the secret $master - key$ are given by $params = (g, g^x, g^y)$, $master - key = (x, y)$
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 $K = g^{\frac{1}{(ID+x+ry)}} \in G$,
   (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(**$params, ID, M$**):** 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

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

This scheme is an efficient identity based encryption and proved to be IND-sID-CPA secure in the standard model. In 2006, Gentry proposed a practical identity based encryption based on this scheme which can achieve IND-ID-CCA2 with tight security proof [20]. Thus this scheme plays an important role in identity based encryption.

3.3 Our Proxy Re-encryption Scheme Based on BB2 Identity Based Encryption

1. **ReKeyGen**$_{ID \rightarrow ID'}$**:** The KGC chooses a collision resistent hash function $H : \{0,1\}^{3|p|} \rightarrow Z_p^*$ and a random seed $t \in Z_p^*$, and computes

$k = H(ID, ID', t)$. He computes $rk_{ID\to ID'} = (\frac{ID'+x+k}{ID+x}, w = g^{\frac{k}{ID'+x+r'y}})$ and sends them to the proxy as the proxy re-encryption key. We note that the KGC chooses a different k for every different user pair (ID, ID').

2. **ReEnc** $(rk_{ID\to ID'}, parms, C_{ID}, ID')$**:**. On input the ciphertext $C_{ID} = (C_1, C_2, C_3) = (g^{s\cdot ID}X^s, Y^s, e(g,g)^s \cdot M)$, the proxy first run Check, if it returns "Invalid", then reject, else computes $C_{ID'} = (C_1', C_2', C_3') = ({C_1}^{rk_{ID\to ID'}}, C_2, C_3 e(C_1, w))$, and sends it to the delegatee.
3. **Check:**. On input a ciphertext $C_{ID} = (C_1, C_2, C_3)$, the proxy computes $v_1 = e(C_1, Y)$ and $v_2 = e(C_2, g^{ID}X)$, if $v_1 = v_2$, then return "Valid", else return "Invalid".

First we verify our scheme's correctness as following.

$$\begin{aligned}
\frac{C_3'}{e(C_1' C_2'^{r'}, K)} &= \frac{C_3 e(C_1, w)}{e({C_1}^{rk_{ID\to ID'}} C_2^{r'}, g^{1/(ID'+x+r'y)})} \\
&= \frac{C_3 e(C_1, w)}{e((g^{s\cdot ID}X^s)^{\frac{ID'+x+k}{ID+x}} Y^{sr'}, g^{1/(ID'+x+r'y)})} \\
&= \frac{C_3 e(C_1, w)}{e(g^{s(ID'+x+ry)} Y^{sr'}, g^{1/(ID'+x+r'y)})} \\
&= \frac{C_3 e(C_1, w)}{e(g^{s(ID'+x+k+r'y)}, g^{1/(ID'+x+r'y)})} \\
&= \frac{e(g,g)^s \cdot M \cdot e(C_1, w)}{e(g^{s(ID'+x+k+r'y)}, g^{1/(ID'+x+r'y)})} \\
&= \frac{e(g,g)^s \cdot M \cdot e(C_1, g^{\frac{k}{ID'+x+r'y}})}{e(g^{s(ID'+x+k+r'y)}, g^{1/(ID'+x+r'y)})} \\
&= \frac{e(g,g)^s \cdot M}{e(g^{s(ID'+x+r'y)}, g^{1/(ID'+x+r'y)})} \\
&= M
\end{aligned}$$

3.4 Security Models

First we define the following oracles, which can be invoked multiple times in any order, subject to the constraints list in the various definition:

- **Corrupted user's key generation** $(O_{corkeygen})$**:** Obtain $sk_{ID} \leftarrow KeyGen_{IBE}(mk, parms, ID)$.$A$ is given sk_{ID}.
- **Re-encryption key generation** $(O_{rekeygen})$:On input (ID, ID') by the adversary, where pk was generated before by $KeyGen$ and ID is a user in IBE setting, return the re-encryption key $rk_{ID\to ID'} = KeyGen_{PRO}$ $(sk, e_{ID}, parms)$ where sk is the secret keys that correspond to pk and e_{ID} is the delegatee's input for re-encryption key generation purpose.
- **Encryption oracle**$O_{enc_{IBE}}$**:** For IBE users, to encrypt a message $M \in G_1$ under the public key $ID \in Z_p^*$, return $Enc_{IBE}(ID, parms, M)$.
- **Re-encryption** O_{renc}**:** Output the re-encrypted ciphertext $ReEnc$ $(rk_{ID\to ID'}, parms,\ C_{ID}, ID')$.

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 delegation partner, KGC, 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.

We consider the case that proxy and delegator are corrupted.

Definition 8. (Delegatee-IBE-IND-ID-CPA) *A PRE scheme from IBE to IBE is Delegatee-IBE-IND-ID-CPA secure if the probability*
$Pr[sk_{ID^\star} \leftarrow O_{keygen}(\lambda), \{sk_{ID_x} \leftarrow O_{corkeygen}(\lambda)\},$
$\{sk_{ID_h} \leftarrow O_{keygen}(\lambda)\},$
$\{R_{hx} \leftarrow O_{rekeygen}(sk_{ID_h}, ID_x)\}, \{R_{xh} \leftarrow O_{rekeygen}(sk_{ID_x}, ID_h)\},$
$\{R_{h\star} \leftarrow O_{rekeygen(sk_{ID_h}, ID^\star)}\}, \{R_{x\star} \leftarrow O_{rekeygen(sk_{ID_x}, ID^\star)}\},$
$(m_0, m_1, St) \leftarrow A_{O_{renc}, O_{enc_{IBE}}(ID^\star, \{sk_{ID_x}\}, \{R_{xh}\}}, \{R_{hx}\}, \{R_{h\star}\}, \{R_{x\star}\}),$

$d^\star \xleftarrow{R} \{0,1\}, C^\star = enc_{IBE}(m_{d^\star}, ID^\star), d' \leftarrow A^{O_{renc}, O_{enc_{IBE}}}(C^\star, St) : d' = d^\star]$
is negligibly close to $1/2$ *for any PPT adversary* A*. In our notation,* St *is a state information maintained by* A *while* $sk_{ID^\star}$ *is the target user's private key, the challenger also chooses other keys for corrupt and honest parties. For other honest parties, keys are subscripted by* h *or* h' *and we subscript corrupt keys by* x *or* x'*. In the game,* A *is said to have advantage* ϵ *if this probability, taken over random choices of* A *and all oracles, is at least* $1/2+\epsilon$.

Delegator Security.

We consider the case that proxy and delegatee are corrupted.

Definition 9. (Delegator-IBE-IND-ID-CPA) *A PRE scheme from IBE to IBE is Delegator-IBE-IND-ID-CPA secure if the probability*
$Pr[sk_{ID^\star} \leftarrow O_{keygen}(\lambda), \{sk_{ID_x} \leftarrow O_{corkeygen}(\lambda)\}, \{sk_{ID_h} \leftarrow O_{keygen}(\lambda)\},$
$\{R_{hx} \leftarrow O_{rekeygen}(sk_{ID_h}, ID_x)\}, \{R_{xh} \leftarrow O_{rekeygen}(sk_{ID_x}, ID_h)\},$
$\{R_{\star h} \leftarrow O_{rekeygen(sk_{ID^\star}, ID_h)}\}, \{R_{\star x} \leftarrow O_{rekeygen(sk_{ID^\star}, ID_x)}\},$
$(m_0, m_1, St) \leftarrow A^{O_{renc}, O_{enc_{IBE}}}(ID^\star, \{sk_{ID_x}\}, \{R_{xh}\}, \{R_{hx}\}, \{R_{\star h}\}, \{R_{\star x}\}),$

$d^\star \xleftarrow{R} \{0,1\}, C^\star = enc_{IBE}(m_{d^\star}, ID^\star), d' \leftarrow A^{O_{renc}, O_{enc_{IBE}}}(C^\star, St) : d' = d^\star]$
is negligibly close to $1/2$ *for any PPT adversary* A*. In our notation,* St *is a state information maintained by* A *while* $sk_{ID^\star}$ *is the target user's private key, the challenger also chooses other keys for corrupt and honest parties. For other honest parties, keys are subscripted by* h *or* h' *and we subscript corrupt keys by* x *or* x'*. In the game,* A *is said to have advantage* ϵ *if this probability, taken over random choices of* A *and all oracles, is at least* $1/2+\epsilon$.

KGC Security.

In proxy re-encryption from IBE and IBE, KGC's master key can not leverage even if the delegator, the delegatee and proxy collude.

Definition 10. (KGC-OW) *A PRE scheme from CBE to IBE is KGC-OW secure if the output*
$Exp[\{sk_{ID_x} \leftarrow O_{corkeygen}(\lambda)\},$
$\{sk_{ID_h} \leftarrow O_{keygen}(\lambda)\},$
$\{R_{xx'} \leftarrow O_{rekeygen(sk_{ID_x}, ID_{x'})}\}, \{R_{x'x} \leftarrow O_{rekeygen(sk_{ID'_x}, ID_x)}\},$
$\{R_{hx} \leftarrow O_{rekeygen}(sk_{ID_h}, ID_x)\}, \{R_{xh} \leftarrow O_{rekeygen}(sk_{ID_x}, ID_h)\},$
$mk \leftarrow A^{O_{renc}, O_{enc_{IBE}}}(\{sk_{ID_x}\}, \{R_{xh}\}, \{R_{hx}\}, \{R_{xx'}\}, \{R_{x'x}\}, \{parms\})$
is not the real $master-key$ *for any PPT adversary* A*.The challenger also chooses other keys for corrupt and honest parties. For other honest parties, keys are subscripted by* h *or* h' *and we subscript corrupt keys by* x *or* x'*.*

3.5 Security Analysis

In this section, we will give our scheme's security results:
- For delegatee's IBE-IND-sID-CPA security, KGC alone can break it, while the proxy and delegator's colluding can not.
- For delegator's IBE-IND-sID-CPA security, KGC alone can break it, while the proxy and delegatee's colluding can not.
- For KGC's OW security, even if allowing the proxy, delegator and delegatee collude any way, they can not break the KGC's OW security, that is, they can not get the $master-key$.

Theorem 9. *Suppose Decision q-BDHI assumption holds in* G*, then our scheme is Delegator-IBE-IND-sID-CPA secure for the proxy and delegatee's colluding.*

Proof. Suppose A has advantage in attacking the proxy re-encryption IBE system.We build an algorithm B that uses A to solve the Decision $q-BDHI$ problem in G.Algorithm B is given as input a random $(q+2)$-tuple $(g, g^{\alpha}, g^{(\alpha^2)}, ..., g^{(\alpha^q)}, T) \in (G^*)^{q+1} \times G_1$ 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 G_1). Algorithm B's goal is to output 1 if $T = e(g,g)^{1/\alpha}$ and 0 otherwise. Algorithm B works by interacting with A in a selective identity game as follows:

1. **Preparation.** Algorithm B builds a generator $h \in G^*$ 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:
   (a) 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.
   (b) 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 $u = h^{\alpha}$.

(c) Check that $h \in G^*$.Indeed if we had $h = 1$ in G this would mean that $w_j = -\alpha$ for some j easily identifiable w_j ,at which point B would be able to solve the challenge directly.We thus assume that all $w_j \neq -\alpha$.

(d) Observe that for any $i = 1, ..., q-1$, it is easy for 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 $h^{1/(\alpha+w_i)} = \mathrm{g}^{\ f_i(\alpha)} = \prod_{i=0}^{q-2} (g^{(\alpha^i)})^{d_i}$.

(e) Next B computes

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

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.

2. **Initialization.** The selective identity game begins with A first outputting an identity $ID^* \in Z_p^*$ that it intends to attack.
3. **Setup** To generate the system parameters,algorithm B does the following:
   (a) Pick random $a, b \in Z_p^*$ under the constraint that $ab = ID^*$.
   (b) Compute $X = u^{-a} h^{-ab} = h^{-a(\alpha+b)}$ and $Y = u = h^{\alpha}$.
   (c) Publish $params = (h, X, Y)$ as the public parameters. Note that X, Y are independent of ID^* 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 B does not know the value of x or y,but does know the value of $x + ay = -ab = -ID^*$.
4. **Phase 1**.
   - *"A issues up to $q_s < q$ private key queries"*.
   Consider the i-th query for the private key corresponding to public key $ID_i \neq ID^*$. 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 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) 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 A's view. Algorithm B gives A the private key $(r, h_i^{1/(r-\alpha)})$. For completeness, we note that B can construct the private key for ID_i with $r = a$ as $(r, h^{1/ID_i - ID^*})$. Hence,the r in the private key given to 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 $ID_i = ID^*$ 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^*.

- *"A issues up to rekey generation queries on* (ID_i, ID_j)*".*
The challenger B chooses a randomly $U \in Z_p^*$ and sets $\frac{ID_j+x+k}{ID_i+x} = U$, he also computes

$$w = \frac{(h^{\frac{1}{\alpha+w_j}})^{\frac{ID_i \cdot U - ID_j + (U-1)(aw_j - ab)}{r_j - a}}}{h^{(U-1)\cdot\frac{a}{r_j-a}}}$$

we can see $w = h^{\frac{k}{ID_j+x+r_jy}}$, because the following:

$$(h^{ID}X)^U = (h^{ID}X)^{\frac{ID_j+x+k}{ID_i+x}} = h^{ID'} \cdot X \cdot h^k$$

$$h^k = \frac{(h^{ID}X)^U}{h^{ID'}X} = h^{(ID_i+x)U-(ID_j+x)}$$

$$w = h^{\frac{k}{ID_j+x+r_jy}} = h^{\frac{(ID_i+x)U-(ID_j+x)}{(r_j-a)(\alpha+w_j)}}$$

$$= h^{\frac{(ID_i \cdot U - ID_j)+(U-1)(-a(\alpha+b))}{(r_j-a)(\alpha+w_j)}}$$

$$h^{\frac{(ID_i \cdot U - ID_j)+(U-1)(-a(\alpha+b))}{(r_j-a)(\alpha+w_j)}+(U-1)\frac{a}{r_j-a}} = h^{\frac{(ID_i \cdot U - ID_j)+(U-1)(-a(\alpha+b))+(U-1)a(\alpha+w_i)}{(r_j-a)(\alpha+w_j)}}$$

$$= h^{\frac{(ID_i \cdot U - ID_j)+(U-1)(-a\alpha-ab+a\alpha+aw_j)}{(r_j-a)(\alpha+w_j)}}$$

$$= h^{\frac{(ID_i \cdot U - ID_j)+(U-1)(-ab+aw_j)}{(r_j-a)(\alpha+w_j)}}$$

$$= (h^{\frac{1}{\alpha+w_j}})^{\frac{ID_i \cdot U - ID_j + (U-1)(aw_j - ab)}{r_j - a}}$$

$$w = \frac{(h^{\frac{1}{\alpha+w_j}})^{\frac{ID_i \cdot U - ID_j + (U-1)(aw_j - ab)}{r_j - a}}}{h^{(U-1)\cdot\frac{a}{r_j-a}}}$$

thus our simulation is a perfect simulation. Because U 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.

- *"A issues up to rekey generation queries on* (ID^*, ID)*".*
Same as the above.
- *"A issues up to re-encryption queries on* (C_{ID_i}, ID_i, ID_j)*".*
The challenge B runs $ReEnc(rk_{ID_i \to ID_j}, C_{ID_i}, ID_j)$ and returns the results

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

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

It follows that CT is a valid encryption of M_b under ID^* , 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 CT is independent of the bit b.
6. **Phase2.** A issues more private key queries, for a total of at most $q_s < q$. Algorithm B responds as before. A issues more other queries like in Phase1 except natural constraints and Algorithm B responds as before.
7. **Guess.** Finally, A outputs a guess $b' \in \{0,1\}$. If $b = b'$ then 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 A's advantage for breaking the scheme is same as B's advantage for solving q-BDHI problem. This completes the proof. ∎

Theorem 10. *Suppose the q-BDHI assumption holds, then our scheme is Delegatee-IBE-IND-sID-CPA secure for the proxy and delegator's colluding.*

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

Theorem 11. *Suppose the q-BDHI assumption holds, then our scheme is KGC-OW secure for the proxy, delegatee and delegator's colluding.*

Proof. We just the the intuition for this theorem. The master-key is (x, y), and delegator's private key is $(r_i, g^{\frac{1}{ID_i+x+r_i y}})$, the delegatee's private key is $(r_j, g^{\frac{1}{ID_j+x+r_j y}})$, the proxy re-encryption key is $\frac{ID'+x+k}{ID+x}, w = g^{\frac{k}{ID'+x+r'y}}$. Because the proxy re-encryption key is uniformly distributed in Z_p^*, and the original BB2 IBE is secure, we can conclude that (x, y) can not be disclosed by the proxy, delegatee and delegator's colluding. ∎

3.6 Conclusion

In 2007, Matsuo proposed the concept of four types of re-encryption 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]. In this paper, we further extend their research. One feature of their schemes is that they are all based on BB1 identity based encryption [6].We ask question like this: can we construct proxy re-encryption schemes based on BB2 identity based encryption [6]? We give affirmative answer to this question. We construct an IBE to IBE proxy re-encryption scheme based on BB2 with the help of KGC and prove its security in the standard model.

4 Proxy Re-encryption Scheme Based on SK Identity Based Encryption

4.1 Introduction

The concept of proxy re-cryptography 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 2007, Matsuo proposed the concept of four types of proxy re-encryption 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]. One feature of their schemes is that they are all based on BB1 identity based encryption [6]. We reconsider the problem of constructing proxy re-encryption based on SK identity based encryption. Surprisingly, if we consider the help of KGC, then it is easy to construct proxy re-encryption based on SK identity based encryption. Interestingly , our proxy re-encryption scheme even can achieve CCA2 secure, which makes it is unique.

We organize our paper as following. In section 2,we revisit the SK identity based encryption in [9, 13, 35]. In section 3, we propose our new proxy re-encryption scheme from IBE to IBE based on SK identity based encryption. In section 4, we give the security models for our scheme and prove its security in the model. We give our conclusion in section 5.

4.2 Revisit 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.
   - 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 $P_1 = \varphi(P_2)$.
   - Pick a random $s \in Z_q^*$ and compute $P_{pub} = sP_1$.
   - 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.
   - Pick a random $\sigma \in \{0,1\}^n$ and compute $r = H_3(\sigma, m)$.
   - Compute $Q_A = H_1(ID_A)P_1 + P_{pub}$, $g^r = e(P_1, P_2)^r$.

- 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:
   - Compute $g' = e(U, d_A)$ and $\sigma' = V \oplus H_2(g')$.
   - Compute $m' = W \oplus H_4(\sigma)$ and $r' = H_3(\sigma', m')$.
   - if $U \neq r'(H_1(ID_A)P_1 + P_{pub})$, output $\perp$, else return m' as the plaintext.

4.3 Our Proposed Proxy Re-encryption Scheme Based On SK Identity Based Encryption

We modify the underlying SK identity based encryption for the proxy re-encryption purpose, our proposed proxy re-encryption scheme based on SK identity based encryption are as following:

1. **Setup.** Same as the original scheme.
2. **Extract.** Same as the original scheme.
3. **ReKeyGen$_{ID \to ID'}$:** The KGC chooses a collision resistent hash function $H_5 : \{0,1\}^{3|p|} \to Z_p^*$ and a random seed $t \in Z_p^*$, and computes $k = H_5(ID, ID', t)$. He computes $rk_{ID \to ID'} = (\frac{s+H_1(ID')+k}{s+H_1(ID)}, w = \frac{k}{s+H_1(ID')}P_2)$ and $s = kP_1$. He sends $rk_{ID \to ID'}$ to the proxy as the proxy re-encryption key via authenticated channel. He also sends $s = kP_1$ to the delegatee via authenticated channel for "Verify" purpose. We note that the KGC chooses a different k for every different user pair (ID, ID').
4. **Encrypt1.** Given a plaintext $m \in M$, ID_A and M_{pk}, the following steps are performed.
   - Pick a random $\sigma \in \{0,1\}^n$ and compute $r = H_3(\sigma, m)$.
   - Compute $Q_{ID} = H_1(ID)P_1 + P_{pub}$, $g^r = e(P_1, P_2)^r$.
   - Set the ciphertext to $C = (rP_1, rQ_{ID}, \sigma \oplus H_2(g^r), m \oplus H_4(\sigma))$.
5. **ReEnc** $(rk_{ID \to ID'}, parms, C_{ID}, ID')$**:**. On input the ciphertext $C_{ID} = (C_1, C_2, C_3, C_4) = (rP_1, rQ_{ID}, \sigma \oplus H_2(g^r), m \oplus H_4(\sigma))$, the proxy computes $C_{ID'} = (C_1', C_2', C_3', C_4') = (rk_{ID \to ID'}C_2, e(C_1, w), C_3, C_4)$, and sends it to the delegatee.
6. **Decrypt1.** Given a ciphertext $C_{ID'} = (C_1', C_2', C_3', C_4')$, follows the steps:
   - Compute $g' = \frac{e(C_1', d_{ID'})}{C_2'}$ and $\sigma' = C_3' \oplus H_2(g')$.
   - Compute $m' = C_4' \oplus H_4(\sigma)$ and $r' = H_3(\sigma', m')$.
7. **Verify.** If $C_1' \neq r'(H_1(ID')P_1 + P_{pub} + s)$, output $\perp$, else return m' as the plaintext.

First we verify our scheme's correctness as following.

$$g' = \frac{e(C_1', d_{ID'})}{C_2'} = \frac{e(rk_{ID \to ID'}C_2, \frac{1}{s+H_1(ID')}P_2)}{e(C_1, w)}$$
$$= \frac{e(rk_{ID \to ID'}C_2, \frac{1}{s+H_1(ID')}P_2)}{e(C_1, \frac{k}{s+H_1(ID')}P_2)}$$

$$
\begin{aligned}
&= \frac{e(\frac{s+H_1(ID')+k}{s+H_1(ID)} rQ_{ID}, \frac{1}{s+H_1(ID')} P_2)}{e(C_1, \frac{k}{s+H_1(ID')} P_2)} \\
&= \frac{e(\frac{s+H_1(ID')+k}{s+H_1(ID)} r(H_1(ID)+s)P_1, \frac{1}{s+H_1(ID')} P_2)}{e(rP_1, \frac{k}{s+H_1(ID')} P_2)} \\
&= \frac{e(rP_1, P_2)e(rkP_1, \frac{1}{s+H_1(ID')} P_2)}{e(rP_1, \frac{k}{s+H_1(ID')} P_2)} \\
&= e(P_1, P_2)^r \\
&= g^r
\end{aligned}
$$

$$\sigma' = C_3' \oplus H_2(g') = \sigma \oplus H_2(g^r) \oplus H_2(g^r) = \sigma$$

$$m' = m, r' = r$$

$$C_1' = rk_{ID \to ID'} C_2 = r(H_1(ID') + k + s)P_1 = r'(H_1(ID')P_1 + P_{pub} + s)$$

Thus our scheme is a correct proxy re-encryption scheme. Note that in our scheme, the delegatee must receive s for every delegation pair (ID, ID'), the purpose of this step is for "Verify" the ciphertext.

4.4 Security Models and Security Analysis

The security models are same as in Section 3.4, they follow the security model for proxy re-encryption scheme from IBE to IBE.
Interestingly, our proxy re-encryption scheme even can achieve **IND-ID-CCA2 secure** while all the above proxy re-encryption scheme can only achieve **IND-sID-CPA secure**. We also note this scheme is the **most efficient** scheme for proxy re-encryption with pairing which can achieve CCA2 secure in literature, which makes it is **so unique**! But unfortunately, this scheme cannot resist DDos attack introduced in [38].

In this section, we will give our scheme's security results:

- For delegatee's IBE-IND-ID-CCA2 security, KGC alone can break it, while the proxy and delegator's colluding can not.
- For delegator's IBE-IND-ID-CCA2 security, KGC alone can break it, while the proxy and delegatee's colluding can not.
- For KGC's OW security, even if allowing the proxy, delegator and delegatee collude any way, they can not break the KGC's OW security, that is, they can not get the $master - key$.

Theorem 12. *Suppose q-BDHI assumption holds in G, then our scheme is Delegator-IBE-IND-ID-CCA2 secure for the proxy and delegatee's colluding.*

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 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 A or indirectly by an extraction query,a decryption query or the challenge operation).Then there exists an IND-CCA adversary B which runs in time $O(time(A)+q_D\cdot(T+\Gamma_1))$ 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 G_1.*

$PRE-BasicPub^{hy}$ is specified by seven algorithms: **KeyGen**, **ReKeyGen**, **Encrypt**, **ReEnc**, **Decrypt1**, **Decrypt2**, **Verify**.

KeyGen: *Given a security parameter k, the parameter generator follows the steps.*

1. *Identical with step 1 in Setup algorithm of $SK-PRE-IBE$.*
2. *The KGC 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 $\frac{1}{h_0'+s}P$.*
3. *Pick three cryptographic hash functions: $H_2 : G_T \rightarrow \{0,1\}^n$, $H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow Z_q^*$ and $H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$ 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 $e(h_0'P_1 + P_{pub}, d_B) = e(P_1, P_2)$.

ReKeyGen: *The KGC chooses a collision resistent hash function $H_5 : \{0,1\}^{3|p|} \rightarrow Z_p^*$ and a random seed $t \in Z_p^*$, and computes $k = H_5(h_0, h_0', t)$. He computes $rk_{A\rightarrow B} = (\frac{s+h_0'+k}{s+h_0}, w = \frac{k}{s+H_0'}P_2)$ and $s = kP_1$. He sends $rk_{A\rightarrow B}$ to the proxy as the proxy re-encryption key via authenticated channel. He also sends $s = kP_1$ to the delegatee via authenticated channel for "Verify" purpose.*

Encrypt: *Given a plaintext $m \in M$ and the public key K_{pubA} and K_{pubB},*

1. *Pick a random $\sigma \in \{0,1\}^n$ and compute $r = H(\sigma, m)$,and $g^r = e(P_1, P_2)^r$.*
2. *For the delegator, set the ciphertext to $C = (rP_1, r(h_0P_1 + P_{pub}), \sigma \oplus H_2(g^r), m \oplus H(\sigma))$.*
3. *For the delegatee, set the ciphertext to $C = (rP_1, r(h_0'P_1 + P_{pub}), \sigma \oplus H_2(g^r), m \oplus H(\sigma))$.*

ReEnc:. *On input the ciphertext $C_A = (C_1, C_2, C_3, C_4) = (rP_1, rQ_{ID}, \sigma \oplus H_2(g^r), m \oplus H_4(\sigma))$, the proxy computes $C_B = (C_1', C_2', C_3', C_4') = (rk_{A\rightarrow B}C_2, e(C_1, w), C_3, C_4)$, and sends it to the delegatee.*

Decrypt1: *For the delegator, given a ciphertext $C_A = (U, V, W)$, K_{pubA}, and the private key d_A,*

1. *Compute $g' = e(U, d_A)$ and $\sigma' = V \oplus H(g')$,*

2. *Compute $m' = W \oplus H_4(\sigma')$ and $r' = H_3(\sigma', m')$,*
3. *If $U \neq r'(h_0P_1 + P_{pub})$,reject the ciphertext, else return m' as the plaintext.*

Decrypt2. *For the delegatee, given a ciphertext $C_B = (C_1', C_2', C_3', C_4')$:*

- *Compute $g' = \frac{e(C_1', d_B)}{C_2'}$ and $\sigma' = C_3' \oplus H_2(g')$.*
- *Compute $m' = C_4' \oplus H_4(\sigma)$ and $r' = H_3(\sigma', m')$.*

Verify. *For the delegatee, if $C_1' \neq r'(h_0'P_1 + P_{pub} + s)$, output $\perp$, else return m' as the plaintext.*

Proof. The proof for this lemma is similar as lemma1 in [13].

Lemma 2. *Let H_3, H_4 be random oracles. Let A be an IND-CCA adversary against $PRE-BasicPub^{hy}$ defined in Lemma1 with advantage $\epsilon(k)$. Suppose 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 B against the following* **PRE-BasicPub** *scheme,which is specified by six algorithms:* **KeyGen**, **ReKeyGen**, **Encrypt**, **ReEnc**, **Decrypt1**, **Decrypt2**.

keygen: *Given a security parameter k, the parameter generator follows the steps.*

1. *Identical with step 1 in algorithm keygen of $PRE - BasicPub^{hy}$.*
2. *Identical with step 2 in algorithm keygen of $PRE - BasicPub^{hy}$.*
3. *Pick a cryptographic hash function $H_2 : G_T \rightarrow \{0,1\}^n$ 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 $e(h_0'P_1 + P_{pub}, d_B) = e(P_1, P_2)$.

ReKeyGen: *Identical with* **ReKeyGen** *of $PRE - BasicPub^{hy}$ except no s generation.*

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 $g^r = e(P_1, P_2)^r$.*

ReEnc: *Identical with* **ReEnc** *of $PRE - BasicPub^{hy}$.*

Decrypt1: *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 $m = V \oplus H_2(g')$.*

Decrypt2: *Identical with* **Decrypt2** *of $PRE - BasicPub^{hy}$.*

with advantage $\epsilon_1(k)$ and running time $t_1(k)$ where

$$\epsilon_1(k) \geq \frac{1}{2(q_3 + q_4)}[(\epsilon(k) + 1)(1 - \frac{2}{q})^{q_D} - 1]$$

$$t_1(k) \leq t(k) + O((q_3 + q_4) \cdot (n + logq)).$$

Proof. The proof for this lemma is similar as lemma2 in [13], actually this is the Fujisaki-Okamoto transformation [16].

Lemma 3. *Let H be a random oracle. Suppose there exists an IND-CPA adversary Adv against the* **PRE-BasicPub** *defined in Lemma2 which has advantage $\epsilon(k)$ and queries H at most q_2 times. Then there exists an algorithm C to solve the q_1-BDHI problem with advantage at least $2\epsilon(k)/q_2$ and running time $O(time(Adv)+q_1^2\cdot T_2)$ where T_2 is the time of a multiplication operation in G_2.*

Proof. Algorithm C is given as input a random q_1-BDHI instance $(q, G_1, G_2, G_T, \varphi, P_1, P_2, xP_2, x^2P_2, ..., x^{q_1}P_2)$ where x is a random element from Z_q^*. Algorithm C finds $e(P_1, P_2)^{\frac{1}{x}}$ by interacting with Adv as follows: Algorithm C 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, C solves the $q_1 - BDHI$ problem directly. Otherwise C 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 $1 \leq i < q_1$.
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 C passes Adv the public key $K_{pubA} = (q, G_1, G_2, G_T, \varphi, e, n, Q_1, Q_2, xQ_1 - h_0Q_1, h_0, (h_1 + h_0, \frac{1}{h_1+x}Q_2), ..., (h_i + h_0, \frac{1}{h_i+x}Q_2), ..., (h_{q_1-1} + h_0 , \frac{1}{h_{q_1-1}+x}Q_2), H_2)$(ie. setting $P_{pub} = xQ_1 - h_0Q_1$), and the private key is $d_A = \frac{1}{x}Q_2$, which C does not know. H_2 is a random oracle controlled by C. Note that $e((h_i + h_0)Q_1 + p_{pub}, d_A) = e(Q_1, Q_2)$. Hence K_{pubA} is a valid public key of A in **BasicPub**.
7. Now C passes Adv the public key $K_{pubB} = (q, G_1, G_2, G_T, \varphi, e, n, Q_1, Q_2, xQ_1 - h_0Q_1, h_0' = h_1 + h_0, ..., (h_i + h_0, \frac{1}{h_i+x}Q_2), ..., (h_{q_1-1} + h_0 , \frac{1}{h_{q_1-1}+x}Q_2), H_2)$(ie. setting $P_{pub} = xQ_1 - h_0Q_1$), and the private key is $d_B = \frac{1}{h_1+x}Q_2$, which C knows. H_2 is a random oracle controlled by C. Note that $e((h_i + h_0)Q_1 + p_{pub}, d_B) = e(Q_1, Q_2)$. Hence K_{pub} is a valid public key of B in**BasicPub**.

Now B starts to respond to queries as follows.

1. **Phase1**
   H_2-query(X_i). At any time algorithm Adv 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, C does the following operations:

(a) If on the list there is a tuple indexed by X_i, then B responds with ζ_i.

(b) Otherwise, C 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 A with ζ_i.

ReKeyGenration query. C Choose a randomly $a \in Z_q^*$, set $\frac{s+h_0'+k}{s+h_0} = a$. C computes $w = a(h_0 - h_0')d_B + (a-1)Q_2$. He sets $rk_{A\to B} = (a, w)$ is of the right form. Because the following

$$\begin{aligned} \frac{s+h_0'+k}{s+h_0} &= a \\ s &= x - h_0 \\ e(a((h_0+s)Q_1 - (h_0'+s)Q_1, d_B) &= e(w, Q_1) \\ w &= \frac{(ah_0 + as - h_0' - s)}{s+h_0'}Q_2 \\ w - (a-1)Q_2 &= \frac{ah_0 + as - h_0' - s - (a-1)(s+h_0')}{s+h_0'}Q_2 \\ &= \frac{ah_0 - h_0' - (a-1)h_0'}{s+h_0'}Q_2 \\ &= \frac{a(h_0 - h_0')}{s+h_0'}Q_2 \\ &= a(h_0 - h_0')d_B \\ w &= a(h_0 - h_0')d_B + (a-1)Q_2 \end{aligned}$$

ReEncryption query. The challenge C runs $ReEnc(rk_{A\to B}, C_A, B)$ and returns the results.

2. **Challenge.** Algorithm Adv 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 $C_{ch} = (U, V) = (rQ_1, R)$. B gives C_{ch} as the challenge to Adv. 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. **Phase2.** Adv issues more queries like in Phase1 except natural constraints and Algorithm C responds as before.
4. **Guess.** After algorithm Adv outputs its guess, C picks a random tuple(X_i, ζ_i) from H_2list. C 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 Adv 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:

Claim1: $Pr[H]$in the simulation above is equal to $Pr[H]$ in the real attack.

Claim2: In the real attack we have $Pr[H] \geq 2\epsilon(k)$. Following from the above two claims, we have that C produces the correct answer with probability at least $2\epsilon(k)/q_2$.

This completes the proof of this theorem. ∎

Theorem 13. *Suppose q-BDHI assumption holds in G, then our scheme is Delegatee-IBE-IND-ID-CCA2 secure for the proxy and delegator's colluding.*

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

Theorem 14. *Suppose the q-BDHI assumption holds, then our scheme is KGC-OW secure for the proxy, delegatee and delegator's colluding.*

Proof. We just the 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 proxy re-encryption key is $\frac{s+H_1(ID')+k}{s+H_1(ID)}$, $w = \frac{k}{s+H_1(ID')}P_2$. Because the proxy 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. ∎

4.5 Conclusion

In 2007, Matsuo proposed the concept of four types of proxy re-encryption 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]. One feature of their schemes is that they are all based on BB1 identity based encryption [6]. They excluded constructing proxy re-encryption schemes based on SK identity based encryption [6] for technique reasons [30]. We reconsider the problem of constructing proxy re-encryption based on SK identity based encryption. Surprisingly, if we consider the help of KGC, then it is easy to construct proxy re-encryption based on SK identity based encryption. Interestingly , our proxy re-encryption scheme even can achieve CCA2 secure, which makes it is unique.

5 Some Observation on Constructing Proxy Re-encryption Scheme Based on BF Identity Based Encryption

5.1 Revisit BF Identity Based Encryption

We now revisit the BF identity based encryption scheme.
We assume that recipient identities are represented as bit strings of arbitrary length, and that the messages to be encrypted are bit strings of some fixed length l.
Let g be the respective generator of a bilinear group G of prime order p, and let $e : G \times G \to G_t$ be a bilinear map taking its arguments in G. Additionally, we require the availability of four cryptographic hash functions viewed as random oracles:

1. a function $H_1 : \{0,1\}^* \to G$ for hashing the recipient identity;
2. a function $H_2 : G_t \to \{0,1\}^l$ for xor-ing with the session key;
3. a function $H_3 : \{0,1\}^l \times \{0,1\}^l \to Z_p$ for deriving a blinding coefficient;
4. a function $H_4 : \{0,1\}^l \to \{0,1\}^l$ for xor-ing with the plaintext.

The BF-IBE system then consists of the following algorithms:

1. **Setup:** To generate IBE system parameters, select a random integer $w \in Z_p$, and set $g_{pub} = g^w$. The public system parameters *params* and the master secret key *masterk* are given by:
$$params = (g, g_{pub}) \in G^2, masterk = w \in Z_p.$$
2. **Extract** To generate a private key d_{ID} for an identity $ID \in \{0,1\}^*$, using the master key w, the trusted authority computes $h_{ID} = H_1(ID)$ and then $d_{ID} = (h_{ID})^w$ in G. The private key is the group element:
$$d_{ID} \in G$$
3. **Encrypt:** To encrypt a message $M \in \{0,1\}^l$ for a recipient of identity $ID \in \{0,1\}^*$, the sender picks a random $s \in \{0,1\}^l$,derives $r = H_3(s, M)$, computes $h_{ID} = H_1(ID)$ and $y_{ID} = e(h_{ID}, g_{pub})$, and outputs:
$$C = (g^r, s \oplus H_2(y_{ID}^r), M \oplus H_4(s)) \in G \times \{0,1\}^l \times \{0,1\}^l$$
4. **Decrypt:** To decrypt a given ciphertext $C = (u, v, w)$ using the private key d_{ID} , the recipient successively computes:
$$v \oplus H_2(e(u, d_{ID})) = s, w \oplus H_4(s) = M, H_3(s, M) = r$$
Then, it verifies that $g^r = u$, and rejects the ciphertext if the equality is not satisfied. Otherwise, it outputs $M \in \{0,1\}^l$ as the decryption of C.

5.2 Our Observation

We note that the ciphertexts for ID and ID' are

$$C = (g^r, s \oplus H_2(y^r_{ID}), M \oplus H_4(s))$$
$$C = (g^r, s \oplus H_2(y^r_{ID'}), M \oplus H_4(s))$$

We must at least construct the re-encryption ciphertext of the form $C = (X, g^r, (s \oplus H_2(y^r_{ID})) \oplus H_2(y^r_{ID}) \oplus H_2(y^r_{ID'}) \oplus Y, M \oplus H_4(s))$ But we note $H_2(y^r_{ID}) \oplus H_2(y^r_{ID'}) \oplus Y = H_2(e(h^w_{ID}, g^r) \oplus H_2(e(h^w_{ID'}, g^r) \oplus Y$. and it is impossible to transform $e(h^w_{ID}, g^r)$ to $e(h^w_{ID'}, g^r) \cdot Z$ for h_{ID} and $h_{ID'}$ are two different points on the elliptic curve. That means, our technique can no longer apply. But there are maybe existing other ways to construct proxy re-encryption based on BF identity based encryption, we leave it as an open problem.

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, and 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, and 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 and 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. X.Boyen. An introduction to Identity Based Encryption. In *International Lecture Series on Pairings*, Brisbane, 2007.
9. 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.
10. 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.

11. 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.
12. C.Chu and W.Tzeng. Identity-based proxy re-encryption without random oracles. In *ISC 2007,*LNCS 4779, pp. 189–202.Springer–Verlag,2007.
13. L.Chen, Z.Cheng. Security Proof of Sakai-Kasahara's Identity-Based Encryption Scheme. http://eprint.iacr.org/2005/226.pdf, 2005.
14. Chandrasekar S.,Ambika K.,Pandu Rangan C.Signcryption with Proxy Re-encryption. http://eprint.iacr.org/2008/276.pdf,2008.
15. Y. Dodis and A. Ivan. Proxy cryptography revisited.In *NDSS '03*,2003.
16. 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.
17. 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.
18. V.Goyal. Reducing Trust in Identity Based Cryptosystems.In *Advances in Cryptology - CRYPTO 2007,* LNCS 4622, pp. 430–447.Springer–Verlag,2007.
19. C.Gentry.Certificate-based encryption and the certificate revocation problem.In *Advances in Cryptology - EUCRYPT 2003,* LNCS 2656, pp. 272–293. Springer–Verlag, 2003.
20. C.Gentry.Practical Identity-Based Encryption without Random Oracles. In In *Advances in Cryptology - EUCRYPT 2006,* LNCS 4004, pp. 445–464. Springer–Verlag, 2006.
21. 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.
22. S. Hohenberger. Advances in Signatures, Encryption, and E-Cash from Bilinear Groups. Ph.D. Thesis, MIT, May 2006.
23. S. Hohenberger, G. N. Rothblum, a. shelat, V. Vaikuntanathan. Securely Obfuscating Re-encryption.In *TCC'07,*LNCS 4392, pp. 233–252.Springer–Verlag,2007.
24. M. Jakobsson. On quorum controlled asymmetric proxy re-encryption.In *, PKC '99*,pages 112–121,Springer–Verlag,1999.
25. Varad Kirtane, C.Pandu Rangan, RSA-TBOS Signcryption with Proxy Re-encryption.http://eprint.iacr.org/2008/324.pdf,2008.
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 *First 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-encryption scheme.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. C.Ma ,J.Ao,and J.Li. Group-based Proxy Re-encryption scheme http://eprint.iacr.org/2007/274.pdf,2007.
33. C.Ma,J.Ao.Revisit of Group-based Unidirectional Proxy Re-encryption Scheme.http://eprint.iacr.org/2008/325.pdf,2008.
34. 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.
35. R.Sakai and M.Kasahara. ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report2003/054. 2003.
36. Q Tang, P Hartel, W Jonker. Inter-domain Identity-based Proxy Re-encryption. http://eprints.eemcs.utwente.nl/12259/01/.
37. Q Tang. Type-based Proxy Re-encryption and its Construction. http://eprints.eemcs.utwente.nl/13024/01/.
38. X.Wang, W.Wu, X.Yang.On DDos Attack against Proxy in Re-encryption and Re-signature. http://eprint.iacr.org/2008/354.pdf,2008.
39. X.Wang. On the Insecurity of Proxy Re-encryption from IBE to IBE in P1363.3/D1. http://eprint.iacr.org/2008/387.pdf,2008.
40. L. d. Zhou, M. A. Marsh, F. B. Schneider, and A. Redz. Distributed blinding for ElGamal re-encryption.TR 1924, Cornell CS Dept.,2004.