Breaking and Repairing Proxy Re-encryption from IBE to IBE in Pairing'07

Xu an Wang, Xiaoyuan Yang

Key Laboratory of Information and Network Security
Engneering College of Chinese Armed Police Force, Xi'an 710086, P.R. China
wangxahq@yahoo.com.cn

Abstract. Proxy re-encryption, introduced by Blaze et al. in 1998, allows a proxy to transform a ciphertext computed under Alice's public key into one that can be opened under Bob's decryption key [3]. In ACNS'07, Green et al. proposed the first identity based proxy re-encryption schemes [13]. Later in Pairing'07, Matsuo proposed another few more proxy re-encryption schemes in identity based setting [18]. They proposed the concept of the four types of proxy re-encryption: IBE to IBE, IBE to CBE, CBE to IBE and CBE to CBE, and constructed two proxy re-encryption schemes: proxy re-encryption from CBE to IBE, proxy re-encryption from IBE to IBE. Now both of the schemes are being standardized by P1363.3 workgroup [17]. In this paper, we show that the proxy re-encryption scheme from IBE to IBE is not secure. Specially, we give two attacks to this scheme. The first attack shows that the proxy himself only can re-encrypt any IBE user's ciphertext to be the delegatee's ciphertext, which beyond the ability supposed to give proxy, that is, just re-encrypting the delegator's ciphertext to be delegatee's ciphertext. The second attack shows that, if the proxy colludes with any delegatee, the proxy and this delegatee can derive any other IBE user's secret key, thus can decrypt any other IBE user's ciphertext. We propose an IND-Pr-sID-CPA secure and an IND-Pr-ID-CCA secure identity based proxy re-encryption scheme based on BB_1 IBE. We prove the former scheme's security in the standard model and the later scheme's security in the random oracle model. Compared with other identity based proxy re-encryption scheme, there two schemes are more efficient.

1 Introduction

The concept of proxy re-encryption comes from the work of Blaze et al. 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 its several potential applications[1,2]. In ACNS'07, Green et al. proposed the first identity based proxy re-encryption schemes [13]. In ISC'07, Chu et al. proposed the first IND-CCA2 proxy re-encryption schemes in the standard model, they constructed their scheme based on Waters' IBE. But unfortunately Shao et al. found a flaw in their scheme and they fixed this flaw by proposing an improved scheme [19]. In Pairing'07, Matsuo proposed another few more proxy re-encryption schemes in identity based setting [18]. Interestingly, they proposed the concept of four types of proxy re-encryption: IBE to IBE, IBE to CBE, CBE to CBE and CBE to IBE, which can help the ciphertext circulate smoothly in the network. They constructed two proxy re-encryption schemes: one is the hybrid proxy re-encryption from CBE to IBE, the other is the proxy re-encryption from IBE to IBE. Meanwhile, both of the schemes are now being standardized by P1363.3 workgroup [17]. Until very recently, Tang et al. extend the concept of identity based proxy re-encryption, they proposed a concept of inter-domain identity based proxy re-encryption which aimed to constructing proxy re-encryption scheme between different domain in identity based setting [20].

1.1 Related Works

In ACNS'07, Green et al. proposed the first identity based proxy re-encryption schemes [13]. They defined the algorithms and security models for identity based proxy re-encryption, and constructed their scheme by using a variant of the efficient Dodis/Ivan key splitting approach to settings with a bilinear map. The re-encryption key in their scheme is of the form $(H_1(Alice)^{-s} \cdot H(X), IBE_{Bob}(X))$. When the proxy re-encrypt, it does some transformations and sends $IBE_{Bob}(X)$ to the delegatee. And then the delegatee decrypt $IBE_{Bob}(X)$ to recover X and use this X to recover the original message. They excluded generating re-encryption key involving PKG for two reasons: From a theoretical point of view, having the PKG, generating the proxy keys makes the problem of finding IB-PRE schemes quite unchallenging; From a practical point of view, having the PKG involved in the generation of proxy keys constitute a considerable bottleneck in many applications, it would force the PKG to be on-line and available even during the generation of proxy keys (other than IBE keys).

In ISC'07, Chu et al proposed the first IND-CCA2 secure proxy re-encryption in the standard model based on Waters' IBE [10]. They follow the paradigm proposed in [13] (We denote it as Green's paradigm). unfortunately Shao et al. found their scheme can not achieve IND-CCA2 secure and they fixed this flaw by proposing an improved scheme [19]. However, both of these schemes are not efficient due to the structure of Waters' IBE and Green's paradigm.

In Pairing'07, Matsuo proposed four types of proxy re-encryption: IBE to IBE, CBE to IBE, IBE to CBE and CBE to CBE. They constructed a hybrid proxy re-encryption scheme from CBE to IBE and a proxy re-encryption scheme from IBE to IBE. They extend the algorithms for proxy re-encryption scheme proposed in [13]. There are four parties involved in their proxy re-encryption schemes: delegator, delegatee, proxy and PKG. That is, PKG now can involve in re-encryption key generation. As a result, their schemes are much more efficient than schemes in [13]. But their schemes can only achieve IND-sID-CPA secure in the standard model.

Until very recently, Tang et al. proposed the new concept of inter-domain identity based proxy re-encryption. They concern on constructing proxy re-encryption between different domain in identity based setting. They follow the way of generating re-encryption key in [18], that is, allowing involving PKG during re-encryption key generating process. They follow Green's paradigm but based on Boneh-Frankin IBE. Also their scheme can only achieve IND-sID-CPA secure.

1.2 Our Contribution

We find that Matsuo's proxy re-encryption from IBE to IBE is not secure. For this scheme is now being standardized by IEEE P1363.3 workgroup, we think it is necessary to rethink this primitive's security carefully. We give two attacks to this scheme, The first attack shows that the proxy himself only can re-encrypt any IBE user's ciphertext to be the delegatee's ciphertext, which beyond the ability supposed to give proxy, that is, just re-encrypting ID's ciphertext to be ID''s ciphertext. The second attack shows that, if the proxy colludes with any delegatee, the proxy and this delegatee can derive any IBE user's secret key, thus can decrypt any IBE user's ciphertext.

Furthermore, we propose two new proxy re-encryption schemes from IBE to IBE based on BB_1 IBE. Like Matsuo's scheme, our scheme no longer follows Green's paradigm. Different from other PRE schemes, PKG alone generating re-encryption key in our scheme. That is, we

completely run away from the principle proposed in [13]. But we think this is not unreasonable. Involving PKG in re-encryption key generating process always make PRE much more efficient. Furthermore, many practical IBE systems let their PKG be online 24/7/365. Even if all the above reasons are not convincing, we think at least our construction can be seen as an alterative way to construct PRE.

1.3 Roadmap

We organize our paper as following. In Section 2, we give our definition and security model for identity based proxy re-encryption. In Section 3, we review the proxy re-encryption scheme from IBE to IBE in Pairing'07, we give two attacks to this scheme, one is presented in Section 3.2, the other is presented in Section 3.3. In Section 4, we give our proposed IND-Pr-sID-CPA secure identity based proxy re-encryption scheme and prove its security in the standard model. In Section 4, we give our proposed IND-Pr-sID-CPA secure identity based proxy re-encryption scheme and prove its security in the standard model. In Section 5, we give our proposed IND-Pr-ID-CCA secure identity based proxy re-encryption scheme and prove its security in the random oracle model. We give the comparison results in Section 6. We conclude our paper in Section 7

2 Definition and Security Model for Identity Based Proxy Re-encryption

In this section, we give our definition and security model for identity based proxy re-encryption scheme, which is based on [13,20].

Definition 1. *An identity based proxy re-encryption scheme is tuple of algorithms* (Setup, KeyGen, Encrypt, Decrypt, RKGen, Reencrypt):

- Setup(1^k). *On input a security parameter, the algorithm outputs both the master public parameters which are distributed to users, and the master secret key (msk) which is kept private.*
- KeyGen(params, msk, id). *On input an identity* $id \in \{0,1\}^*$ *and the master secret key, outputs a decryption key* sk_{id} *corresponding to that identity.*
- Encrypt(params, id, m). *On input a set of public parameters, an identity* $id \in \{0,1\}^*$ *and a plaintext* $m \in M$*, output* c_{id}*, the encryption of* m *under the specified identity.*
- RKGen(params, msk, sk_{id_1}, sk_{id_2}, id_1, id_2). *On input secret keys* msk, sk_{id_1}, sk_{id_2}, *and identities* $id \in \{0,1\}^*$*, PKG, the delegator and the delegatee interactively generat the re-encryption key* $rk_{id_1 \to id_2}$*, the algorithm output it.*
- Reencrypt(params, $rk_{id_1 \to id_2}$, c_{id_1}). *On input a ciphertext* c_{id_1} *under identity* id_1*, and a re-encryption key* $rk_{id_1 \to id_2}$*, outputs a re-encrypted ciphertext* c_{id_2}.
- Decrypt(params, sk_{id}, c_{id}). *Decrypts the ciphertext* c_{id} *using the secret key* sk_{id}*, and outputs* m *or* $\perp$.

Correctness. Intuitively, an IB-PRE is correct if the Decrypt algorithm always outputs the expected decryption of a properly generated ciphertext. Slightly more formally, let $c_{id_1} \leftarrow Encrypt(params, id_1, m)$ be a properly generated ciphertext, Then $\forall m \in \mathcal{M}, \forall id_1, id_2 \in \{0,1\}^*$, where $sk_{id_1} = KeyGen(msk, id_1)$, $sk_{id_2} = KeyGen(msk, id_2)$, $rk_{id_1 \to id_2} \leftarrow RKGen(params, sk_{id_1}, id_1, id_2)$, the following propositions hold:

- Decrypt(params, sk_{id_1}, c_{id_1})$= m$
- Decrypt(params, sk_{id_2}, Reencrypt(params, $rk_{id_1 \to id_2}$, c_{id_1}))$=m$

Definition 2. *Let $\mathcal{S}$ be an IB-PRE scheme defined as a tuple of algorithms* (Setup, KeyGen, Encrypt, Decrypt, RKGen, Reencrypt). *For* ATK $\in$ (CPA, CCA), *security is defined according to the following game.*
$Exp^{A,IND-PrID-ATK,i}$

1. Select. *Choose* $i \leftarrow \{0,1\}$.
2. Setup. *Run* Setup(1^k) *to get* $(params, msk)$, *and give* params *to* $\mathcal{A}$.
3. Find phase. $\mathcal{A}$ *makes the following queries. At the conclusion of this phase* $\mathcal{A}$ *will select* $id^* \in \{0,1\}^*$ *and* $(m_0, m_1) \in \mathcal{M}^2$.
   (a) *For* $\mathcal{A}$*'s queries of the form* $(extract, id)$, *return* $sk_{id} = KeyGen(params, msk, id)$ *to* $\mathcal{A}$.
   (b) *For* $\mathcal{A}$*'s queries of the form* $(rkextract, id_1, id_2)$, *where* $id_1 \neq id_2$, *return* $rk_{id_1 \rightarrow id_2}$=*RKGen (params, msk, KeyGen(params, msk,* id_1*), KeyGen(params, msk,* id_2*),* id_1, id_2*)* *to* $\mathcal{A}$.
   (c) *For* $\mathcal{A}$*'s queries of the form* $(decrypt, id, c)$, *if* ATK=CCA *then return* m=Decrypt(params, KeyGen(params, msk, *id*), *c*) *to* $\mathcal{A}$. *Otherwise, if* ATK=CPA, *return* $\perp$ *to* $\mathcal{A}$.
   (d) *For* $\mathcal{A}$*'s queries of the form* $(reencrypt, id_1, id_2, c)$, *if* ATK=CCA *then derive a re-encryption key* $rk_{id_1 \rightarrow id_2} = RKGen(params, msk, KeyGen(params, msk, id_1), KeyGen(params, msk, id_2), id_1, id_2)$, *and return* $c' = Reencrypt(params, rk_{id_1 \rightarrow id_2}, id_1, id_2, c)$ *to* $\mathcal{A}$. *If* $ATK = CPA$, *return* $\perp$ *to* $\mathcal{A}$.

   Note that $\mathcal{A}$ *is not permitted to choose* id^* *such that trivial decryption is possible using keys extracted during this phase (e.g. , by using extracted re-encryption keys to translate from* id^* *to some identity for which* $\mathcal{A}$ *holds a decryption key).*
4. Choice and Challenge. *When* $\mathcal{A}$ *presents* $(choice, id^*, m_0, m_1)$, *compute* c^*= *Encrypt(params,* id^*, m_i*) and give* c^* *to* $\mathcal{A}$.
5. Guess stage. $\mathcal{A}$ *continues to make queries as in the find stage, with the following restrictions. Let* $(\mathcal{C}, ID)$ *be a set of ciphertext/identity pairs, initially containing the single pair* (c^*, id^*). *For all* $c \in \mathcal{C}$ *and for all* rk *given to* $\mathcal{A}$, *let* $\mathcal{C}'$ *be the set of all possible values derived via calls to* Reencrypt*:*
   (a) $\mathcal{A}$ *is not permitted to issue any query of the form (decrypt, id, c) where* $(c, id) \in (\mathcal{C} \cap \mathcal{C}')$.
   (b) $\mathcal{A}$ *is not permitted to issue any queries* $(extract, id)$ *or (rkextract,* id_1, id_2*) that would permit trivial decryption of any ciphertext in* (C, C').
   (c) $\mathcal{A}$ *is not permitted to issue any query of the form (reencrypt,* id_1, id_2, *c) where* $\mathcal{A}$ *possesses the keys to trivially decrypt ciphertexts under* id_2 *and* $(c, id_1) \in (\mathcal{C} \cap \mathcal{C}')$.
   (d) $\mathcal{A}$ *is not permitted to issue any query of the form (reencrypt,* id_1, id_2, *c) where* $\mathcal{A}$ *possesses the keys to trivially decrypt ciphertexts under* id_2 *and* $(c, id_1) \in (\mathcal{C} \cap C')$. *On successful execution of any re-encrypt query, let* c' *be the result and add the pair* (c', id_2) *to the set* C.

   At the conclusion of this stage, $\mathcal{A}$ *outputs* i', *where* $i' \in \{0,1\}$.

The outcome of the game is determined as follows: If $i' = i$ then $\mathcal{A}$ wins the game. Let $(ext, rk, dec, renc)$ be the oracles for the Find phase, and $(ext, rk, dec, renc)$ be the same oracles modified for the GUESS stage. Define $Exp_A^{IND-PrID-ATK}$ as following:

$$
\begin{aligned}
i \leftarrow \{0,1\}; (params, msk) &\leftarrow Setup(1^k);\\
(id^*, m_0, m_1, t) &\leftarrow \mathcal{A}^{ext(\cdot), rk(\cdot), dec(\cdot), renc(\cdot)}(params);\\
c^* &\leftarrow Encrypt(params, id^*, m_i);\\
i' &\leftarrow \mathcal{A}^{ext'(\cdot), rk'(\cdot), dec'(\cdot), renc'(\cdot)}(params, c^*, t);
\end{aligned}
$$

Let $Adv_A^{IND-PrID-ATK}$=$Pr(i'=i)-1/2$. If for all probabilistic polynomial time algorithms A, $Adv_A^{IND-PrID-CPA} \leq v(k)$, we say that the Identity-Based Proxy Re-encryption scheme $\mathcal{S}$ is IND-Pr-ID-CPA-secure. We say that the Identity-Based Proxy Re-encryption scheme $\mathcal{S}$ is IND-PrID-CCA-secure if for all probabilistic polynomial time algorithms $\mathcal{A}$, $Adv_A^{IND-Pr-ID-CCA} \leq v(k)$.

3 Review Proxy Re-encryption Scheme from IBE to IBE in Pairing'07 and Attacks on It

In this section, we first review the proxy re-encryption scheme from IBE to IBE in Pairing'07 [18], then we give two attacks to it.

3.1 Review Proxy Re-encryption from IBE to IBE in Pairing'07

The proxy re-encryption scheme from IBE to IBE is based on the BB_1-IBE scheme.

- The underlying IBE scheme (BB_1-IBE scheme):
  Let G be a bilinear group of prime order p (the security parameter determines the size of G). Let $e: G \times G \rightarrow G_1$ be the bilinear map. For now, we assume public keys (ID) is element in Z_p^*. We later extend the construction to public keys over $\{0,1\}^*$ by first hashing ID using a collision resistant hash $H: \{0,1\}^* \rightarrow Z_p$. We also assume messages to be encrypted are elements in G. The IBE system works as follows:
  1. $\mathbf{SetUp_{IBE}(k)}$. Given a security parameter k, select a random generator $g \in G$ and random elements $g_2, h \in G$. Pick a random $\alpha \in Z_p^*$. Set $g_1 = g^\alpha$,$mk = g_2^\alpha$, and $params = (g, g_1, g_2, h)$. Let mk be the master- secret key and let $params$ be the public parameters.
  2. $\mathbf{KeyGen_{IBE}(mk, params, ID)}$. Given $mk = g_2^\alpha$ and ID with $params$, the PKG 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. $\mathbf{Enc_{IBE}(ID, params, M)}$. To encrypt a message $M \in G_1$ under the public key $ID \in Z_p^*$, pick a random $r \in Z_p^*$ and compute $C_{ID} = (g^r, (g_1^{ID} h)^r, Me(g_1, g_2)^r)$.
  4. $\mathbf{Dec_{IBE}(sk_{ID}, params, C_{ID})}$. Given ciphertext $C_{ID} = (C_1, C_2, C_3)$ and the secret key $sk_{ID} = (d_0, d_1)$ with $params$, compute $M = \frac{C_3 e(d_1, C_2)}{e(d_0, C_1)}$.
- The delegation scheme:
  1. $\mathbf{EGen(sk_{ID}, params)}$. Given $sk_{ID} = (d_0, d_1) = (g_2^\alpha (g_1^{ID} h)^u, g^u)$ for ID with $params$, set $e_{ID} = d_1 = g^u$.
  2. $\mathbf{KeyGen_{PKG}(mk, params)}$. Given $mk = \alpha$ with $params$, set $sk_R = \alpha$.
  3. $\mathbf{KeyGen_{PRO}(sk_R, e_{ID'}, params, ID, ID')}$. Given $sk_R = \alpha, e_{ID'} = g^{u'}$ with $params$, set $rk_{ID \rightarrow ID'} = (ID \rightarrow ID', g^{u'\alpha})$.
  4. $\mathbf{Check(params, C_{ID}, ID)}$. Given the delegator's identity ID and $C_{ID} = (C_1, C_2, C_3)$ with $params$, compute $v_0 = e(C_1, g_1^{ID} h)$ and $v_1 = (C_2, g)$. If $v_0 = v_1$ then output 1. Otherwise output 0.
  5. $\mathbf{ReEnc(rk_{ID \rightarrow ID'}, params, C_{ID}, ID')}$. Given identities ID, ID', $rk_{ID \rightarrow ID'} = (ID \rightarrow ID', g^{u'\alpha})$,$C_{ID} = (C_1, C_2, C_3)$ with $params$, the proxy re-encrypt the ciphertext C_{ID} into $C_{ID'}$ as follows. First it runs "Check", if output 0, then return "Reject". Else it computes $C_{ID'} = (C_1', C_2', C_3') = (C_1, C_2, C_3 e(C_1^{ID'-ID}, g^{u'\alpha}))$.

Now this scheme is being standardized by IEEEP1363.3 workgroup [17].

Remark 1. In this scheme, the $EGen(sk_{ID}, params)$, $KeyGen_{PKG}(mk, params)$,$KeyGen_{PRO}(sk_R$, $e_{ID'}$, params, ID, ID') algorithms can be replaced with one algorithm RKGen(params, mk, sk_{ID}, $sk_{ID'}$, ID, ID'), which outputs $rk_{ID\rightarrow ID'} = (ID \rightarrow ID', g^{u'\alpha})$. Then the algorithms will be consistent with Definition 1.

3.2 Attack I

Suppose there is another IBE user ID'' with a ciphertext $C_{ID''} = (C_1'', C_2'', C_3'') = (g^{r'}, (g_1^{ID''}h)^{r'}, M'e(g_1, g_2)^{r'})$ which has not agreed to the delegation with ID', but the proxy can re-encrypt ID'''s ciphertext to be ID''s valid ciphertext. Thus ID' can decrypt ID'''s ciphertext, the attack is as following.

1. First the proxy runs "Check". Because $C_{ID''} = (C_1'', C_2'', C_3'')$ is a valid ciphertext for ID'', thus the proxy can go through.
2. Second the proxy runs "ReEnc". Given the delegator's identity ID'', the delegatee's identity ID', $rk_{ID''\rightarrow ID'} = (ID'' \rightarrow ID', g^{u'\alpha})$,$C_{ID''} = (C_1'', C_2'', C_3'')$ with $params$, re-encrypt the ciphertext $C_{ID''}$ into $C_{ID'}$ as follows. $C_{ID'} = (C_1', C_2', C_3') = (C_1'', C_2'', C_3''e(C_1''^{ID''-ID'}, g^{u'\alpha}))$. And this ciphertext is a valid ciphertext for ID' because

$$\frac{C_3'e(d_1, C_2')}{e(d_0, C_1')} = \frac{M' \cdot e(g_1, g_2)^{r'}e(g^{r'(ID'-ID'')}, g^{u'\alpha})e(g^{u'}, (g_1^{ID''}h)^{r'})}{e(g_2^{\alpha}(g_1^{ID'}h)^{u'}, g^{r'})}$$
$$= \frac{M' \cdot e(g_1, g_2)^{r'}e((g_1^{ID'}h)^{r'}, g^{u'})}{e(g_1, g_2)^{r'}e((g_1^{ID'}h)^{r'}, g^{u'})} = M'$$

Thus ID' can decrypt every ID'''s ciphertext if it colludes with the proxy.

Assume the challenge ciphertext is $C^*_{ID''} = (C_1''^*, C_2''^*, C_3''^*)$, the adversaries are the proxy and ID'. Note that ID'' does not agree to the delegation with ID', but ID does agree to the delegation with ID'. The proxy can re-encrypt $C^*_{ID''} = (C_1''^*, C_2''^*, C_3''^*)$ with $rk = (ID' - ID'', g^{u'\alpha})$ where $g^{u'\alpha}$ is selected from $rk_{ID\rightarrow ID'} = (ID \rightarrow ID', g^{u'\alpha})$. And get $C^*_{ID'}$, then ID' decrypt $C^*_{ID'}$ to get m, which breaks IND-sID-CPA security.

Remark 2. The first attack shows that the proxy himself only can re-encrypt any IBE user's ciphertext to be the delegatee's ciphertext, which beyond the ability supposed to give proxy, that is, just re-encrypting the delegator's ciphertext to be delegatee's ciphertext.

3.3 Attack II

Suppose the delegatee ID' colludes with the proxy. the proxy's re-encryption key is $g^{u'\alpha}$ and the delegatee's private key is $(g_2^{\alpha}(g_1^{ID'}h)^{u'}, g^{u'})$. Then the proxy and this delegatee can computes valid private keys for other ID.

$$\frac{g_2^{\alpha}(g_1^{ID'}h)^{u'}}{g^{u'\alpha ID'}} \cdot g^{u'\alpha ID} = g_2^{\alpha}(g_1^{ID}h)^{u'}$$

and we can see $(g_2^{\alpha}(g_1^{ID}h)^{u'}, g^{u'})$ is a valid private key for ID. Thus the proxy and this delegatee can decrypt every other IBE user's ciphertext.

Remark 3. The second attack shows that, if the proxy colludes with any delegatee, the proxy and this delegatee can derive any other IBE user's secret key, thus can decrypt any other IBE user's ciphertext.

4 Selective Identity Chosen Plaintext Secure IBPRE

In this section, we propose an IND-sID-CPA secure identity based proxy re-encryption scheme and prove its security in the standard model.

4.1 Our Proposed IND-ID-sID-CPA Secure IBPRE Scheme

- The underlying IBE scheme: The same as BB_1-IBE presented in Section 3.1 except the KGC adds (ID, u) to the `user-key-list` and preserves this list in the **KeyGen** algorithm.
- The delegation scheme:
  1. $\mathbf{KeyGen_{PRO}}(\mathbf{mk}, \mathbf{params}, \mathbf{ID}, \mathbf{ID'})$. The PKG searches in the `user-key-list` for ID', if find no item of (ID', u'), then return "Reject", otherwise it chooses a collision resistent hash function $H : \{0,1\}^{2|p|} \rightarrow Z_p^*$, and computes $k = H(ID, u')$,$w = g_1^k$. The PKG sets $rk_{ID \rightarrow ID'} = (ID \rightarrow ID', \frac{u'+k}{\alpha ID + t_2}, w)$ and sends it to the proxy via secure channel. We must note that the PKG computes a different k for every different user pair (ID, u').
  2. $\mathbf{Check}(\mathbf{params}, \mathbf{C_{ID}}, \mathbf{ID})$. Given the delegator's identity ID and $C_{ID} = (C_1, C_2, C_3)$ with $params$, compute $v_0 = e(C_1, g_1^{ID} h)$ and $v_1 = (C_2, g)$. If $v_0 = v_1$ then output 1. Otherwise output 0.
  3. $\mathbf{ReEnc}(\mathbf{rk_{ID \rightarrow ID'}}, \mathbf{params}, \mathbf{C_{ID}}, \mathbf{ID'})$. Given the identities ID, ID', $rk_{ID \rightarrow ID'} = (ID \rightarrow ID', \frac{u'+k}{\alpha ID + t_2}, w)$,$C_{ID} = (C_1, C_2, C_3)$ with $params$, the proxy re-encrypt the ciphertext C_{ID} into $C_{ID'}$ as follows. First it runs "Check", if output 0, then return "Reject". Else computes $C_{ID'} = (C_1', C_2', C_3') = (C_1, C_2, \frac{C_3 e(C_2^{rk_{ID \rightarrow ID'}}, g_1^{(ID'-ID)})}{e(w^{(ID'-ID)}, C_1)})$.

  We can verify its correctness as following

$$\frac{C_3' e(d_1, C_2')}{e(d_0, C_1')} = \frac{M e(g_1, g_2)^r e((g_1^{ID} h)^{r \frac{u'+k}{\alpha ID + t_2}}, g_1^{(ID'-ID)}) e(g^{u'}, (g_1^{ID} h)^r)}{e(g_1^{k(ID'-ID)}, g^r) e(g_2^{\alpha} (g_1^{ID'} h)^{u'}, g^r)}$$
$$= \frac{M e(g^{(u'+k)r}, g_1^{(ID'-ID)}) e(g^{u'}, (g_1^{ID} h)^r)}{e(g_1^{k(ID'-ID)}, g^r) e((g_1^{ID'} h)^{u'}, g^r)} = M$$

Remark 4. In our scheme, we must note that the PKG computes a different k for every different pair (ID, u'). Otherwise, if the adversary knows $\frac{u'+k}{\alpha ID + t_2}$ for four different ID_1, ID_2, ID_3, ID_4 but the same k, u', α, t_2 , he can compute (α, t_2, u', k), which is not secure at all.

Remark 5. Maybe one argue that in our scheme the re-encryption key can not be easy revoked. We can solve this problem by giving a temporary proxy re-encryption scheme. The IBE user can get his private key corresponding to the public key of the form "$ID \mid TimePeriod$", he will get a different $g^{u'}$ for every different "$ID \mid TimePeriod$" , thus PKG only generates different re-encryption keys for different "$TimePeriod$". Thus the proxy key can be revoked periodically.

4.2 Security Analysis

Theorem 1. *Suppose the mDBDH assumption holds, then our scheme proposed in Section 4.1 is* IND-Pr-sID-CPA *secure.*

Proof. First we explain what is the mDBDH assumption. The mDBDH assumption is: on input $(g, g^a, g^{a^2}, g^b, g^c, T)$, the probability of distinguishing $T = e(g,g)^{abc}$ and $T \neq e(g,g)^{abc}$ is negligible. We believe solving mDBDH problem is hard.

Suppose $\mathcal{A}$ can attack our scheme, we construct an algorithm $\mathcal{B}$ solves the mDBDH problem in G. On input $(g, g^a, g^{a^2}, g^b, g^c, T)$, algorithm $\mathcal{B}$'s goal is to output 1 if $T = e(g,g)^{abc}$ and 0 otherwise. Let $g_1 = g^a, g_2 = g^b, g_3 = g^c, g_4 = g^{a^2}$. Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective identity game as follows:

1. **Initialization**. The selective identity game begins with $\mathcal{A}$ first outputting an identity ID^* that it intends to attack.
2. **Setup**.To generate the system's parameters, algorithm $\mathcal{B}$ picks $\alpha' \in Z_p$ at random and defines $h = g_1^{-ID^*} g^{\alpha'} \in G$. It gives $\mathcal{A}$ the parameters $params = (g, g_1, g_2, h)$. Note that the corresponding $master - key$, which is unknown to $\mathcal{B}$, is $g_2^a = g^{ab} \in G^*$.
3. **Phase 1**
   - *"$\mathcal{A}$ issues up to private key queries on ID_i".* $\mathcal{B}$ selects randomly $r_i \in {Z_p}^*$, 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

     $$d_0 = g_2^{\frac{-\alpha'}{ID_i - ID^*}} (g_1^{(ID_i - ID^*)} g^{\alpha'})^{r_i} = g_2^a (g_1^{(ID_i - ID^*)} g^{\alpha'})^{r_i - \frac{b}{ID - ID*}} = g_2^a (g_1^{ID_i} h)^{\widetilde{r_i}}.$$

     $$d_1 = g_2^{\frac{-1}{ID_i - ID^*}} g^{r_i} = g^{\widetilde{r_i}}.$$

   - *"$\mathcal{A}$ issues up to rekey generation queries on (ID, ID')".* The challenge $\mathcal{B}$ chooses a randomly $x \in Z_p^*$, sets $rk_{ID \to ID'} = x$ and returns it to $\mathcal{A}$. He also searches in the `user-key-list` for ID', if find no item of (ID', u'), then return "Reject", otherwise he computes $w = \frac{g_4^{(ID - ID^*)x} g_1^{\alpha' x}}{g_1^{u'}}$ and sends it to the proxy. We have

     $$g^k = \frac{(g_1^{ID} h)^x}{g^{u'}}$$

     $$g_1^k = (\frac{(g_1^{ID} h)^x}{g^{u'}})^a = \frac{(g_1^{ID - ID^*} g^{\alpha'})^{ax}}{g^{u'a}} = \frac{(g_1^a)^{(ID - ID^*)x} g_1^{\alpha' x}}{g_1^{u'}} = \frac{(g_4)^{(ID - ID^*)x} g_1^{\alpha' x}}{g_1^{u'}} = w$$

     For the delegatee and the proxy or the delegator and the proxy, they can verify $e(g^k, g_1) = e(w, g)$ is always satisfied. Thus our simulation is a perfect simulation. But the delegator and delegatee cannot get any useful information from x.
   - *"$\mathcal{A}$ issues up to re-encryption queries on (C_{ID}, ID, ID')".* The challenge $\mathcal{B}$ runs $ReEnc(rk_{ID \to ID'}, C_{ID}, ID, ID')$ and returns the results.
4. **Challenge** When $\mathcal{A}$ decides that Phase1 is over, it outputs two messages $M_0, M_1 \in G$. Algorithm $\mathcal{B}$ picks a random bit b and responds with the ciphertext $C = (g^c, (g^{\alpha'})^c, M_b \cdot T)$. Hence if $T = e(g,g)^{abc} = e(g_1, g_2)^c$, then C is a valid encryption of M_b under ID^*. Otherwise, C is independent of b in the adversary's view.
5. **Phase2** $\mathcal{A}$ issues queries as he does in Phase 1 except natural constraints.
6. **Guess** Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. Algorithm $\mathcal{B}$ concludes its own game by outputting a guess as follows. If $b = b'$, then $\mathcal{B}$ outputs 1 meaning $T = e(g,g)^{abc}$. Otherwise it outputs 0 meaning $T \neq e(g,g)^{abc}$.

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

5 Toward Chosen Ciphertext Security

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

5.1 Our Proposed IND-Pr-ID-CCA Secure IBPRE Scheme

Let $\mathbb{G}$ be a bilinear group of prime order p (the security parameter determines the size of G). Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ be the bilinear map. Identities are represented using distinct arbitrary bit strings in $\{0,1\}^*$. The messages (or session keys) are bit strings in $\{0,1\}^l$ of some fixed length l. We require the availability of three hash functions viewed as random oracles:

- A hash function $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$;
- A hash function $H_2 : G_1 \times \{0,1\}^l :\to G$;
- A hash function $H_3 : G_1 :\to \{0,1\}^l$;
- A hash function $H_4 : \{0,1\}^* \times G \times G \times G \times \{0,1\}^l :\to G$;
- A hash function $H_5 : \{0,1\}^* \times Z_p \to Z_p$;

1. **SetUp**. To generate IBE system parameters, first select three integers α, β, and $\gamma \in Z_p$ at random. Set $g_1 = g^\alpha$, $g_2 = g^\beta$ and $h = g^\gamma$ in G, and compute $v_0 = e(g,g)^{\alpha\beta}$. The public system parameters *params* and the *masterkey* are given by: $params = (g, g_1, g_3, v_0)$, $masterkey = (\alpha, \beta, \gamma)$. Strictly speaking, the generator need not be kept secret, but since it will be used exclusively by the authority, it can be retained in *masterkey* rather than published in *params*.
2. **Extract**. To generate a private key d_{ID} for an identity $ID \in \{0,1\}^*$, using the *masterkey*, the trusted authority picks a random $u \in Z_p$ and outputs: $d_{ID} = (d_0, d_1) = (g_2^\alpha (g_1^{H_1(ID)} h)^u, g^u)$. The PKG adding (ID, u) to the `user-key-list` and preserving this list.
3. **Encrypt**. To encrypt a message $M \in \{0,1\}^l$ for a recipient $\{0,1\}^*$, the sender computes $s = H_2(\sigma, M)$, $k = v_0^s \in \mathbb{G}_t$, $C_1 = g^s$, $C_2 = h^s g_1^{H_1(ID)s}$, $C_3 = \sigma \cdot k$, $C_4 = M \oplus H_3(\sigma)$, $C_5 = H_4(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4)^s$, and then outputs $C = (C_1, C_2, C_3, C_4, C_5)$.
4. **ReKeyGen**. The PKG searches in the `user-key-list` for ID', if find no item of (ID', u'), then return "Reject", otherwise it computes $k' = H_5(ID, u')$,$w = g^{k'}$. The PKG sets $rk_{ID \to ID'} = (ID \to ID', \frac{\alpha H_1(ID') + t_2 + k'}{\alpha H_1(ID) + t_2}, w)$ and sends it to the proxy via secure channel. We must note that the PKG computes a different k for every different user pair (ID, u').
5. **ReEnc**. Given the identities ID, ID', $rk_{ID \to ID'} = (ID \to ID', \frac{\alpha H_1(ID') + t_2 + k'}{\alpha H_1(ID) + t_2}, w)$, $C_{ID} = (C_1, C_2, C_3, C_4, C_5)$ with *params*, the proxy re-encrypts the ciphertext C_{ID} into $C_{ID'}$ as follows.
   (a) First it computes $v_0 = e(C_5, g)$ and $v_1 = e(H_4(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4), C_1)$. If $v_0 \neq v_1$, the ciphertext is rejected.
   (b) Else computes $C_{ID'} = (C_1', C_2', C_3', C_4') = (C_1, \frac{C_2^{rk_{ID \to ID'}}}{w}, C_3, C_4)$.
6. **Decrypt**.

(a) To decrypt a normal ciphertext $C = (C_1, C_2, C_3, C_4, C_5)$ using the private key $d_{ID} = (d_0, d_1)$, the recipient computes $k = e(C_1, d_0)/e(C_2, d_1)$. It computes $v_0 = e(C_5, g)$ and $v_1 = e(H_4(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4), C_1)$. If $v_0 \neq v_1$, the ciphertext is rejected. It then computes $\sigma = C_3/k$, $M = H_4(\sigma) \oplus C_4$. It computes $s' = H_2(\sigma, M)$ and verifies that $C_1 = g^{s'}$, $C_2 = h^{s'} g_1^{H_1(ID)s'}$, if either check fails, returns $\perp$, otherwise returns M.
(b) To decrypt a normal ciphertext $C = (C_1, C_2, C_3, C_4)$ using the private key $d_{ID} = (d_0, d_1)$, the recipient computes $k = e(C_1, d_0)/e(C_2, d_1)$. It then computes $\sigma = C_3/k$, $M = H_3(\sigma) \oplus C_4$. It computes $s' = H_2(\sigma, M)$ and verifies that $C_1 = g^{s'}$, $C_2 = h^{s'} g_1^{H_1(ID)s'}$, if either check fails, returns $\perp$, otherwise returns M.

5.2 Security Analysis

Theorem 2. *Suppose the DBDH assumption holds, then our scheme proposed in Section 5.1 is* IND-Pr-ID-CCA *secure.*

Proof. Let $\mathcal{A}$ be a p.p.t. algorithm that has non-negligible advantage in attacking the scheme proposed in Section 5.1. We use $\mathcal{A}$ in order to construct a second algorithm $\mathcal{B}$ which has non-negligible advantage at solving the DBDH problem in $\mathbb{G}$. Algorithm $\mathcal{B}$ accepts as input a properly-distributed tuple $(\mathbb{G}_1 = g, g^a, g^b, g^c, R) \in G_1^4 \times G_1$ and outputs 1 if $R = e(g, g)^{abc}$. We now describe the algorithm $\mathcal{B}$, which interacts with algorithm $\mathcal{A}$ as following.
Oracle Queries. $\mathcal{B}$ simulates the random oracles H_1, H_2, H_3, H_4, H_5 as follows.

1. $H_1 : \{0,1\}^* \rightarrow Z_q^*$. On receipt of a new query for $ID \neq ID^*$, return $t \leftarrow_R Z_q^*$ and record (ID, t); On receipt of a new query for ID^*, select randomly $T \in Z_q^*$, return T and record (ID^*, T).
2. $H_2 : G_1 \times \{0,1\}^l :\rightarrow Z_q^*$. On a new query (σ, M), return $s \leftarrow_R G$ and record $(\sigma, M, s, C_1, C_3, C_4)$.
3. $H_3 : G_1 :\rightarrow \{0,1\}^l$. On receipt of a new query σ , select $p \leftarrow_R \{0,1\}^l$ and return p. Record the tuple (σ, p).
4. $H_4 : \{0,1\}^* \times G \times G \times G \times \{0,1\}^l :\rightarrow G$. On receipt of a new query $(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4)$, select $z \in Z_q^*$ and return $g^z \in G$, record $(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4, z, g^z)$.
5. $H_5 : \{0,1\}^* \times Z_p \rightarrow Z_p$. On receipt of a new query (ID, u'), return $y \leftarrow_R Z_p$ and record (ID, u', y).

Our simulation proceeds as follows:

1. **Setup.** $\mathcal{B}$ generates the scheme's master parameter as following. First it lets $g_1 = g^a, g_2 = g^b, g_3 = g^c$, algorithm $\mathcal{B}$ picks $\alpha' \in Z_p$ at random and defines $h = g_1^{-T} g^{\alpha'} \in G$. $\mathcal{B}$ lets $params = (G_1, H_1, H_2, H_3, H_4, H_5, g, g_1, g_2, g_3, h)$ and gives $params$ to $\mathcal{A}$.
2. **Find/Guess.** During the Find stage, there are no restrictions on which queries $\mathcal{A}$ may issue. The scheme permits only a single consecutive re-encryption, therefore, during the GUESS stage, $\mathcal{A}$ is restricted from issuing the following queries:
   (a) $(extract, ID^*)$ where id^* is the challenge identity.
   (b) $(decrypt, ID^*, c^*)$ wherec c^* is the challenge ciphertext.
   (c) Any pair of queries $(rkextract, ID^*, ID_i)$, $(decrypt, ID_i, c_i)$ where $c_i = Reencrypt(rk_{ID^* \rightarrow ID^i}, c^*)$.
   In the Guess stage, let ID^* be the target identity, and parse the challenge ciphertext c^* as $(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$. In both phases, $\mathcal{B}$ responds to $\mathcal{A}$'s queries as follows

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

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

$$d_1 = g_2^{\frac{-1}{H_1(ID_i)-T}}g^{r_i} = g^{\widetilde{r_i}}.$$

– On $(rkextract, ID, ID')$, The challenge $\mathcal{B}$ chooses a randomly $x \in Z_p^*$, sets $rk_{ID\rightarrow ID'} = x$ and returns it to $\mathcal{A}$. He also searches in the `user-key-list` for ID', if find no item of (ID', u'), then return "Reject", otherwise he computes $w = \frac{(g^{H_1(ID)}h)^x}{(g^{H_1(ID)}h)}$ and sends it to the proxy. We have

$$w = \frac{(g^{H_1(ID)}h)^x}{(g^{H_1(ID')}h)} = \frac{(g^{H_1(ID)}h)^{\frac{\alpha H_1(ID')+t_2+k'}{\alpha H_1(ID)+t_2}}}{(g^{H_1(ID')}h)} = g^{k'}$$

Thus our simulation is a perfect simulation. But the delegator and delegatee cannot get any useful information from x.

– On $(decrypt, ID, c)$ where (in the Guess stage) $(ID, c) \neq (ID^*, c^*)$, check whether c is a level-1 (non re-encrypted) or level-2 (re-encrypted) ciphertext. In the Guess stage, parse c^* as $(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$.
For a level-1 ciphertext, $\mathcal{B}$ parses c as $(C_1, C_2, C_3, C_4, C_5)$ and:
(a) Looks up the value $(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4)$ in the H_4 table, to obtain the tuple $(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4, z, g^z)$. If $(ID \parallel C_1 \parallel C_2 \parallel C_3 \parallel C_4)$ is not in the table, or if (in the Guess stage) $C_5 = C_5^*$, then $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$.
(b) Looks up the value (C_1, C_3, C_4) in the H_2 table, to obtain the tuple . Checks that $S = g^{zs}$. If not, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$. Later we will show that, when ID and ID' run encryption, the probability of $(\sigma, M, s, C_1, C_3, C_4)$ is same for ID and ID' is negligible.
(c) Computes $k = e(C_1, d_0)/e(C_2, d_1)$, checks that $\sigma = C_3/k$. If not, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$.
(d) Checks that $C_4 = H_5(\sigma) \oplus M$. If not, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$.
(e) Otherwise, $\mathcal{B}$ returns M to $\mathcal{A}$.
For a level-2 ciphertext, $\mathcal{B}$ parses c as (C_1, C_2, C_3, C_4) and:
(a) Looks up the value (C_1, C_3, C_4) in the H_2 table, to obtain the tuple $(\sigma, M, s, C_1, C_3, C_4)$. Computes $k = e(C_1, d_0)/e(C_2, d_1)$, checks that $\sigma = C_3/k$. If not, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$. Checks that $C_2 = h^s g_1^{H_1(ID)s}$. If so, return M. Otherwise, return $\perp$.

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

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

3. **Choice and Challenge.** At the end of the Find phase, $\mathcal{A}$ outputs (ID^*, M_0, M_1). $\mathcal{B}$ forms the challenge ciphertext as follows:
(a) Choose $\sigma \in G_1$ and $p \in \{0,1\}^n$ randomly, and insert (σ, p) in H_3 table.
(b) Insert $(\sigma, M_b, \cdot, g_3, \sigma \cdot R, M_b \oplus p)$ to H_2 table.
(c) Choose $z \in Z_p$ randomly, and insert $((g_3, g_3^{\alpha'}, \sigma \cdot R, M_b \oplus p), z, g^z)$ in the H_4 table. .

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

4. **Forgeries and Abort conditions**
   (a) Forgeries. The adversary may forge C_5 on (C_1, C_2, C_3, C_4), but from the security of BLS short signature [5], this probability is negligible.
   (b) Abort conditions. When ID and ID' run encryption, the probability of $(\sigma, M, s, C_1, C_3, C_4)$ is same for ID and ID' is negligible, because $\sigma \in \{0,1\}^l$, the maximal probability is $1/2^l$.

6 Comparison

Scheme	Sec	Sec-Mod	Assum	Enc	Check	Reenc	Dec		Ciph-Len	
							1stCiph	2-ndCiph	1stCiph	2-ndCiph
GA06a[13]	IND-Pr-ID-CPA	RO	DBDH	$1t_e + 1t_p$	0	$1t_p$	$1t_p$	$2t_p$	$1\|G\| + 1\|G_e\|$	$2\|G\| + 2\|G_e\|$
GA06b[13]	IND-Pr-ID-CCA	RO	DBDH	$1t_p + 1t_e$	$2t_p$	$2t_e$ $+2t_p$	$2t_e + 2t_p$	$1t_e + 2t_p$	$1\|G\| + 1\|G_T\|$ $+1\|G_e\| + \|m\|$	$1\|G\| + 1\|G_e\|$ $+2\|m\| + \|id\|$
CT07[10]	IND-Pr-ID-CPA	Std.	DBDH	$3t_e + 1t_p$ $+1t_s$	$1t_v$	$2t_e$	$2t_e + 3t_p$ $+1t_v$	$2t_e + 10t_p$ $+1t_v$	$3\|G\| + \|G_T\|$ $+\|vk\| + \|s\|$	$9\|G\| + 2\|G_T\|$ $+\|vk\| + \|s\|$
SXC08[19]	IND-Pr-ID-CCA	Std.	DBDH	$3t_e + 1t_p$ $+1t_s$	$1t_v$	$2t_e$ $+1t_s$	$2t_e + 3t_p$ $+1t_v$	$2t_e + 10t_p$ $+2t_v$	$3\|G\| + \|G_T\|$ $+1\|vk\| + 1\|s\|$	$9\|G\| + 2\|G_T\|$ $+2\|vk\| + 2\|s\|$
Ours 4.1	IND-Pr-sID-CPA	Std.	mDBDH	$2t_e + 1t_p$	$2t_p$	$3t_e$	$2t_p$	$2t_p$	$2\|G\| + \|G_T\|$	$2\|G\| + \|G_T\|$
Ours 5.1	IND-Pr-ID-CCA	RO	DBDH	$2t_e + 1t_{me}$	$2t_p$	$1t_e$	$2t_p + 1t_e$ $+1t_{me}$	$2t_p + 1t_e$ $+1t_{me}$	$4\|G\| + \|m\|$	$3\|G\| + \|m\|$

Table 1. Efficiency Comparison[1]

In this section, we give some comparison with other identity based proxy re-encryption schemes [10,13,19]. We denote security as Sec, security model as Sec-Mod, assumption as Assum, enc as encryption, re-encryption as Reenc, decryption as Dec, ciphertext as Ciph and ciphertext length as Ciph-Len. t_p, t_e and t_{me} represent the computational cost of a bilinear pairing, an exponentiation and a multi-exponentiation respectively, while t_{s} and t_{v} represent the computational cost of a one-time signature signing and verification respectively. $|\mathbb{G}|$, $|\mathbb{Z}_q|$, $|\mathbb{G}_e|$ and $|\mathbb{G}_T|$ denote the bit-length of an element in groups $\mathbb{G}$, $\mathbb{Z}_q$, $\mathbb{G}_e$ and $\mathbb{G}_T$ respectively. Here $\mathbb{G}$ and $\mathbb{Z}_q$ denote the groups used in our scheme, while $\mathbb{G}_e$ and $\mathbb{G}_T$ are the bilinear groups used in GA07, CT07, SXC08 schemes, i.e., the bilinear pairing is $e : \mathbb{G}_e \times \mathbb{G}_e \rightarrow \mathbb{G}_T$. Finally, $|vk|$ and $|s|$ denote the bit length of the one-time signature's public key and a one-time signature respectively.

From Table 1, we can know that our scheme is much more efficient than CT07 and SXC08 scheme, and almost as efficient as GA06a scheme. But for the proxy, our scheme is much more efficient than GA06a scheme, we think this is important for resisting DDos attack against the proxy.

[1] GA06 and SXC08 are multi-hop IBPRE but we just consider their single-hop variant.

7 Conclusion

In 2007, Matsuo proposed two types of proxy re-encryption schemes which can re-encrypt the ciphertext from CBE to IBE and IBE to IBE [18]. Now these schemes are being standardized by IEEEP1363.3 workgroup[17]. In this paper, we show that their proxy re-encryption scheme from IBE to IBE is not secure. We also propose an IND-sID-CPA IBPRE scheme and an IND-ID-CCA IBPRE scheme and prove their security. Although some excellent work has been done in IBPRE [9,10,13,14,15,16,18,19,20], there are still many open problems need to be solved.

References

1. G. Ateniese, S. Hohenberger, Proxy re-signatures: new definitions, algorithms, and applications.In *ACM CCS'05*, pp. 310–319.ACM Press,2005.
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 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.
5. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil Pairing. In *Advances in Cryptology - ASIACRYPT 2001,* pp. 514–532. Springer–Verlag, 2001.
6. 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.
7. 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.
8. X.Boyen. The BB_1 Identity-Based Cryptosystem A Standard for Encryption and Key Encapsulation. http://grouper.ieee.org/ groups/1363/IBC/submissions/Boyen-bb1-ieee.pdf.
9. 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.
10. C.Chu and W.Tzeng. Identity-based proxy re-encryption without random oracles. In *ISC 2007,*LNCS 4779, pp. 189–202.Springer–Verlag,2007.
11. S. Hohenberger. Advances in Signatures, Encryption, and E-Cash from Bilinear Groups. Ph.D. Thesis, MIT, May 2006.
12. 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.
13. 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.
14. S. Hohenberger, G. N. Rothblum, a. shelat, V. Vaikuntanathan. Securely Obfuscating Re-encryption.In *TCC'07,*LNCS 4392, pp. 233–252.Springer–Verlag,2007.
15. 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.
16. B. Libert and D. Vergnaud, Tracing Malicious Proxies in Proxy Re-Encryption.In *Second International Conference on Pairing-Based Cryptography - Pairing 2008*,Springer–Verlag,2008.
17. L.Martin(editor). P1363.3(TM)/D1, Draft Standard for Identity-based Public Cryptography Using Pairings, May 2008.
18. 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.
19. 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.
20. Q Tang, P Hartel, W Jonker. Inter-domain Identity-based Proxy Re-encryption. To appear in Inscrypt'08, also available at http://eprints.eemcs.utwente.nl/12259/01/.