Proxy Re-encryption Systems for Identity-based Encryption *

Toshihiko Matsuo

NTT DATA CORPORATION
matsuotsh@nttdata.co.jp

Abstract. A proxy re-encryption system allows the proxy to transform ciphertexts encrypted under Alice's public key into the different ciphertexts that can be decrypted by Bob's secret key. In this paper, we propose new proxy re-encryption systems; one for the transformation from ciphertexts encrypted under a traditional certificate-based public key into the ciphertexts that can be decrypted by an secret key for Identity-Based Encryption, and the other one for the transformation from ciphertexts encrypted in IBE manner into the different ciphertexts that can be decrypted by the other secret key for the IBE.
Keywords: proxy re-encryption system, public key encryption, identity-based encryption.

1 Introduction

1.1 Background

A proxy re-encryption system allows the proxy to transform ciphertexts computed under Alice's public key into the different ciphertexts that can be decrypted by using Bob's secret key. This system works as follows; Alice or a trusted third party generates a re-encryption key and sets it in a proxy. On receiving Alice's ciphertexts, the proxy transforms the ciphertext by running the re-encryption algorithm with the re-encryption key, and sends the transformed ciphertext to Bob. Bob decrypts it by his secret key. As it can be seen that Alice delegates her decryption rights to Bob via proxy, we call Alice a *delegator* and Bob a *delegatee*. The proxy re-encryption system should at least satisfy the following requirements; 1) a proxy alone cannot obtain the underlying plaintext, 2) and Bob cannot obtain the underlying plaintext without the proxy cooperating. The proxy re-encryption system can be a primitive for various attractive applications, and thus it has been active research area [BBS98,J99,DI03,ZMSR04,AFGH05,GA06,CH07].

One of the most promising application is the access control system over the network storage [MO97,AFGH05]. In this system, Alice performs a content holder who stores some contents encrypted under her public key in the network storage. The proxy performs an access controller who transforms the

* This is the revised version of a paper appeared in Pairing 2007 [M07].

stored ciphertexts into the different ciphertexts that can be decrypted by Bob's secret key when Alice allows Bob to access her contents. Since the proxy can transform the stored ciphertexts without Alice's secret key, it can reduce the amount of trust in the access control server. Beside the access control system, the proxy re-encryption system can be applied to the secure e-mail forwarding system [AFGH05], the outsourced filtering of encrypted spam [AFGH05], the law enforcement [DI03] and so on.

1.2 Our motivation and contribution

Recently, *Identity-Based Encryption* (for short, IBE) has been one of the most active research area [BF01,BB04a,BB04b,GS04,W05,G06]. In the IBE system, a sender Catherine encrypts a message to an IBE receiver Alice by using Alice's identity as a public key. Providing that Alice sets her e-mail address to the public key and it includes the revocation date, Catherine can easily make sure not only that the public key belongs to Alice, but also when the public key is revoked. Therefore, the IBE system can dramatically improve the workload for public key certificate management, while it is heavy burden in the traditional certificate-based public key encryption (for short, CBE) system premised on Public Key Infrastructure (for short, PKI) service. The IBE system necessarily requires a third party called Public Key Generator (PKG) which generates all secret keys for IBE users by using its master-secret key, and thus the IBE system works where the PKG operation can be allowed. As each user has own policy, role, purpose or circumstances in the ciphertext communication, one might adopt the IBE system because of less certificate management and the other one might adopt the CBE system because of congeniality with the other applications, of disallowance to PKG operation, or of the other reason.

Then a lot of messages encrypted in the different manner circulate among the world, and this circumstance yields the demand for the proxy re-encryption systems transforming CBE ciphertexts into the different IBE ciphertexts (type 1), IBE ciphertexts into the different IBE ciphertexts (type 2), IBE ciphertexts into the different CBE ciphertexts (type 3), and CBE ciphertexts into the different CBE ciphertexts (type 4). However, there is no system for type 1 transformation and only a few systems for type 2 transformation are proposed so far [DI03,GA06]. Therefore, we propose two systems for type 1 and type 2 transformation.

Our first proposal, *hybrid proxy re-encryption system*, is the first system achieving type 1 transformation. Our second proposal for type 2 transformation, *identity-based proxy re-encryption system*, holds the following advantages compare to the previous proposals.

- Our system achieves optimal secret key size, that is, it needs no additional secret key besides the secret key of the underlying IBE system for delegatees to decrypt re-encrypted ciphertexts while it is required in [DI03].
- Our system achieves optimal ciphertext size, that is, the size of re-encrypted ciphertexts is the exactly same as that of the corresponding original cipher-

texts while it is required to extend original ciphertext size for re-encryption in [GA06]. [1]
- Our system does not need additional algorithm or process for decrypting re-encrypted ciphertexts while it is required in [GA06].
- Our system is semantically secure in the standard model while previous systems [DI03,GA06] are semantically secure in the random oracle model (our security notion is slightly weaker than that defined in [GA06]).

1.3 Organization

The rest of this paper consists of four sections. Sec. 2 gives some definitions and preliminaries to understand our study. In Sec. 3, we present the hybrid proxy re-encryption system. We propose the identity-based proxy re-encryption system in Sec. 4 and finally conclude this study in Sec. 5.

2 Preliminaries

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

2.1 Bilinear groups

Let $\mathbb{G}$ and $\mathbb{G}_1$ be multiplicative cyclic groups of prime order p, and g be a generator of $\mathbb{G}$. We say that $\mathbb{G}_1$ has an admissible bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ if the following conditions hold.

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

2.2 Assumption

Definition 1. *For randomly chosen integers $a, b, c \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, a random generator $g \stackrel{R}{\leftarrow} \mathbb{G}$, and an element $R \stackrel{R}{\leftarrow} \mathbb{G}_1$, we define the advantage of an algorithm $\mathcal{A}$ in solving the decision Bilinear Diffie-Hellman (dBDH) problem as follows:*

$$\mathtt{Adv}_{\mathbb{G}}^{dbdh}(\mathcal{A}) = \left|\Pr[\mathcal{A}(g, g^a, g^b, g^c, \hat{e}(g, g)^{abc}) = 0] - \Pr[\mathcal{A}(g, g^a, g^b, g^c, R) = 0]\right|$$

where the probability is over the random choice of generator $g \in \mathbb{G}$, the randomly chosen integers a, b, c, the random choice of $R \in \mathbb{G}_1$, and the random bits used by A. We say that the (k, t, ϵ)-dBDH assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ in solving the dBDH problem in $\mathbb{G}$ under a security parameter k.

[1] Actually, the re-encryption system achieving optimal ciphertext size is also proposed in [GA06]; however, the delegatee has to be aware of one whom the original ciphertext is sent for while she does not have to in our system.

2.3 Digital signature scheme

A digital signature scheme is made up of three algorithms, $\mathbf{KeyGen}_{\Sigma}$, **Sign**, and **Verify**, for generating keys, signing, and verifying signatures, respectively.

- $\mathbf{KeyGen}_{\Sigma}(k)$. Given a security parameter k, generate a signing key sk_{Σ} and the corresponding verification key vk_{Σ}.
- $\mathbf{Sign}(sk_{\Sigma}, M)$. Given the signing key sk_{Σ} and a message M, generate a signature σ.
- $\mathbf{Verify}(vk_{\Sigma}, M, \sigma)$. Given the verification key vk_{Σ}, the message M and its signature σ, output 1 if σ is valid, otherwise output 0.

Though the standard notion of security for a digital signature scheme is called existential unforgeability under a chosen message attack [GMR88], we introduce the slightly different notion, strong existential unforgeability under a passive attack that is defined using the following game between a challenger and an adversary $\mathcal{A}$:

- **SetUp** The challenger runs algorithm $\mathbf{KeyGen}_{\Sigma}$ to obtain a signing key sk_{Σ} and the corresponding verification key vk_{Σ}. The adversary $\mathcal{A}$ is given vk_{Σ}.
- **Message-signature pair** Suppose that q is an integer. The challenger selects messages $M_1, \ldots, M_q$ from the message domain and makes the signatures $\sigma_1, \ldots, \sigma_q$ for each message M_i $(1 \leq i \leq q)$ where $\sigma_i = \mathbf{Sign}(sk_{\Sigma}, M_i)$. The challenger gives the message-signature pair sets $\mathsf{S}_{ms} = \{(M_i, \sigma_i)\}_{1 \leq i \leq q}$ to the adversary $\mathcal{A}$.
- **Output** Eventually, A outputs a pair (M', σ') and wins the game if $(M', \sigma') \notin \mathsf{S}_{ms}$ and $\mathbf{Verify}(vk_{\Sigma}, M', \sigma') = 1$.

Definition 2. *We define $\mathcal{A}$'s advantage in the games as follows.*

$$\mathtt{Adv}^{eu}(\mathcal{A}) = \Pr[(M', \sigma') \notin \mathsf{S}_{ms} \wedge \mathbf{Verify}(vk_{\Sigma}, M', \sigma') = 1]. \tag{1}$$

We say that the digital signature scheme is (k, t, q, ϵ)-strong existentially unforgeable under a passive attack if for any t time adversary $\mathcal{A}$ that observes at most q message-signature pairs under a security parameter k, we have that $\mathtt{Adv}^{eu}(\mathcal{A}) < \epsilon$.

2.4 Certificate-based Public Key Encryption system

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

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

2.5 Identity Based Encryption system

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

- $\mathbf{SetUp}_{\mathsf{IBE}}(k)$. Given a security parameter k, generate a pair $(parms, mk)$, where $parms$ denotes the public parameters and mk is the master-secret key.
- $\mathbf{KeyGen}_{\mathsf{IBE}}(mk, parms, \mathsf{ID})$. Given the master-secret key mk and an identity ID with $parms$, generate a secret key sk_{ID} for ID.
- $\mathbf{Enc}_{\mathsf{IBE}}(\mathsf{ID}, parms, M)$. Given a message M and the identity ID with $parms$, compute the encryption of M, C_{ID}, for ID.
- $\mathbf{Dec}_{\mathsf{IBE}}(sk_{\mathsf{ID}}, parms, C_{\mathsf{ID}})$. Given the secret key sk_{ID}, decrypt the ciphertext C_{ID}.

In setup, a trusted third party, PKG, runs $\mathbf{SetUp}_{\mathsf{IBE}}$ and generates its master-secret key and public parameters. When an IBE user requests a secret key corresponding to her identity (i.e. public key), PKG generates the secret key by running $\mathbf{KeyGen}_{\mathsf{IBE}}$, and give it to the user via secure and authenticated channel. A sender encrypts a message by running $\mathbf{Enc}_{\mathsf{IBE}}$ with the receiver's identity and public parameters. The receiver decrypts a ciphertext by running $\mathbf{Dec}_{\mathsf{IBE}}$ with her secret key.

Security. IBE security [BF01] is defined by the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

- **Setup**. The challenger $\mathcal{C}$ runs the $\mathbf{SetUp}_{\mathsf{IBE}}$ algorithm and gives $\mathcal{A}$ the resulting system parameters, $parms$, keeping the master-secret key mk to itself.
- **Phase 1**. $\mathcal{A}$ adaptively queries $\mathcal{C}$ as follows: $\mathcal{A}$ requests the secret key for ID from $\mathcal{C}$. $\mathcal{C}$ generates the secret key sk_{ID} by running algorithm $\mathbf{KeyGen}_{\mathsf{IBE}}$ and returns them to $\mathcal{A}$. After some number of queries, $\mathcal{A}$ selects two equal length plaintexts $M_0, M_1 \in \mathcal{M}$ and a target identity ID^*, and sends them to $\mathcal{C}$.
- **Challenge**. Given $(M_0, M_1, \mathsf{ID}^*)$, $\mathcal{C}$ picks a random bit $d \in \{0, 1\}$ and sets the challenge ciphertext to $C_{\mathsf{ID}^*_d} = \mathbf{Enc}_{\mathsf{IBE}}(\mathsf{ID}^*, parms, M_d)$, which is sent to $\mathcal{A}$.
- **Phase 2**. $\mathcal{A}$ continues to issue queries as in Phase 1 with the restriction that $\mathcal{A}$ cannot issue secret key queries for ID^*.
- **Guess**. Finally, $\mathcal{A}$ outputs a guess $d' \in \{0, 1\}$.

The adversary $\mathcal{A}$ wins if $d' = d$. We say that the identity based encryption system is IND-ID-CPA secure if $|\Pr[d' = d] - 1/2|$ is negligible.

Definition 3. *We define $\mathcal{A}$'s advantage in an IND-ID-CPA games as follows*

$$\mathtt{Adv}^{id}_{\mathsf{IBE}}(\mathcal{A}) = 2(\Pr[d' = d] - 1/2) \tag{2}$$

We say that the an IBE system is (k, t, q, ϵ)-identity, adaptive chosen plaintext secure if for any t time IND-ID-CPA adversary $\mathcal{A}$ that makes at most q chosen secret key queries under a security parameter k we have that $\mathtt{Adv}^{id}_{\mathsf{IBE}}(\mathcal{A}) < \epsilon$. As shorthand, we say that an IBE system is (k, t, q, ϵ) IND-ID-CPA secure.

Canetti *et al.* [CHK03,CHK04] defined a weaker notion of security in which the adversary commits ahead of time to the public key it will attack. We refer to this notion as selective identity, chosen plaintext secure IBE (IND-sID-CPA). The game is exactly the same as IND-ID-CPA except that the adversary $\mathcal{A}$ discloses to the challenger the target identity ID^* before the Setup phase. The restrictions on secret key queries from Phase 2 also hold in Phase 1.

3 Hybrid proxy re-encryption system

In this section, we introduce our new proxy re-encryption system, *hybrid proxy re-encryption system*. It consists of a CBE system, an IBE system, and additional algorithms that allow ciphertexts encrypted under a CBE public key to be transformed into the different ciphertexts that can be decrypted by an IBE secret key.

3.1 Definition

There are four parties involved in a hybrid proxy re-encryption system, delegator, proxy, delegatee and its PKG. On receiving a ciphertext encrypted in the CBE manner by delegator's public key, the proxy re-encrypts it into ciphertexts that the delegatee who holds an IBE secret key can decrypt, using a re-encryption key generated by the delegator for a particular delegatee.

A hybrid proxy re-encryption consists of: 1) the four algorithms making up an IBE system $\mathbf{SetUp}_{\mathsf{IBE}}$, $\mathbf{KeyGen}_{\mathsf{IBE}}$, $\mathbf{Enc}_{\mathsf{IBE}}$, and $\mathbf{Dec}_{\mathsf{IBE}}$, 2) the three algorithms making up a CBE system $\mathbf{KeyGen}_{\mathsf{CBE}}, \mathbf{Enc}_{\mathsf{CBE}}$, and $\mathbf{Dec}_{\mathsf{CBE}}$, 3) and four algorithms for re-encryption, which are —

- $\mathbf{EGen}(sk_{\mathsf{ID}}, parms)$. Given an IBE secret key sk_{ID} for the IBE user ID with IBE public parameters $parms$, generate e_{ID} for re-encryption key generation.
- $\mathbf{KeyGen}_{\mathsf{PRO}}(sk, e_{\mathsf{ID}}, parms)$. Given a CBE secret key sk and e_{ID} with $parms$, generate a re-encryption key rk_{ID} that re-encrypts CBE ciphertexts into the IBE ciphertexts for ID.
- $\mathbf{ReEnc}(rk_{\mathsf{ID}}, parms, C_{\mathsf{PK}}, \mathsf{ID})$. Given the re-encryption key rk_{ID}, a ciphertext C_{PK} encrypted under the traditional public key, and ID with $parms$, re-encrypt ciphertext C_{PK} into C_{ID} that can be decrypted by the IBE user ID.
- $\mathbf{Check}(parms, C_{\mathsf{PK}}, pk)$. Given C_{PK} and pk with $parms$, output 0 if C_{PK} is a malformed ciphertext. Otherwise, output 1.

Let the PKG employ the digital signature scheme $(\mathbf{KeyGen}_{\Sigma}, \mathbf{Sign}, \mathbf{Verify})$ described in Sec. 2.3; however we do not describe it in the above for conciseness. When a CBE user delegates her decryption rights to an IBE user, the hybrid proxy re-encryption system works as follows.

– *SetUp:*

1. The PKG generates its signing key sk_Σ and the corresponding verification key vk_Σ by running $\mathbf{KeyGen}_\Sigma$. The PKG also generates its master-secret key mk and public parameters $parms$ by running $\mathbf{SetUp}_{\mathsf{IBE}}$. The PKG makes $(parms, vk_\Sigma)$ public, keeping (mk, sk_Σ) to itself.
2. The CBE user generates its secret key sk and the corresponding public key pk by running $\mathbf{KeyGen}_{\mathsf{CBE}}$ with the input $parms$, and makes pk public, keeping sk to itself.

– *Re-encryption key generation and deployment:*
1. When one requests the delegation from the CBE user (i.e. delegator) to the IBE user ID (i.e. delegatee),
   - If no IBE secret key has issued to the delegatee, the PKG generates sk_{ID} by running $\mathbf{KeyGen}_{\mathsf{IBE}}$, and computes e_{ID} by running **EGen**. The PKG makes a digital signature σ_e for $\mathsf{ID}||e_{\mathsf{ID}}$ by running **Sign**. Then PKG issues $(sk_{\mathsf{ID}}, \mathsf{ID}||e_{\mathsf{ID}}, \sigma_e)$ to the delegatee. The delegatee sends $(\mathsf{ID}||e_{\mathsf{ID}}, \sigma_e)$ to the delegator.
   - Otherwise, the delegatee sends previously issued $\mathsf{ID}||e_{\mathsf{ID}}$ and the corresponding signature σ_e to the delegator.
2. On receiving $(\mathsf{ID}||e_{\mathsf{ID}}, \sigma_e)$, the delegator verifies it by running **Verify** with vk_Σ.
   - If it is valid then the delegator generates a re-encryption key rk_{ID} by running $\mathbf{KeyGen}_{\mathsf{PRO}}$ with the input e_{ID}. The delegator sets rk_{ID} in the proxy.
   - Otherwise the delegator rejects.

– *Re-encryption:* Suppose that one sends a ciphertext C_{PK} to the delegatee ID via the proxy. On receiving the CBE ciphertext C_{PK}, the proxy runs the algorithm **Check** with the input $(parms, C_{\mathsf{PK}}, pk)$.
   - If **Check** outputs 0 then the proxy rejects the re-encryption request.
   - Otherwise, the proxy re-encrypts C_{PK} into C_{ID} by running **ReEnc**, and sends C_{ID} to the delegatee ID.

– *Decryption:* The delegatee decrypts C_{ID} by running $\mathbf{Dec}_{\mathsf{IBE}}$ with the IBE secret key sk_{ID}.

3.2 Security notion

In the following, each value appeared in i-th query by the adversary and in the corresponding answer is denoted with letter i. We sometimes denote the delegatee's identity in i-th query by ID_i.

Chosen plaintext security: We model chosen plaintext security for a hybrid proxy re-encryption system as a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. In this game, the adversary is allowed to adaptively choose the IBE secret key queries and re-encryption key queries. Intuitively, these queries imply the situation that: (1)the adversary compromises arbitrary IBE users and obtains their secret keys, (2)the adversary compromises arbitrary proxies and obtains the re-encryption keys, (3)and the adversary requests the re-encryption key generation

of the delegator. Since the adversary obviously wins the game if it obtains both delegatee's secret key and the corresponding re-encryption key involving the same identity, she is not allowed to ask such query. More precisely, IND-ID-CPA security is defined as follows:

Setup. The challenger $\mathcal{C}$ generates (sk_Σ, vk_Σ) by running $\mathbf{KeyGen}_\Sigma$. $\mathcal{C}$ generates $(parms, mk)$ by running $\mathbf{SetUp}_{\mathsf{IBE}}$. $\mathcal{C}$ also generates (pk, sk) by running $\mathbf{KeyGen}_{\mathsf{CBE}}$. $\mathcal{C}$ gives $(parms, pk, vk_\Sigma)$ to $\mathcal{A}$, keeping (mk, sk, sk_Σ) to itself.

Phase 1. Given $(parms, pk, vk_\Sigma)$, $\mathcal{A}$ adaptively queries the challenger $\mathcal{C}$. When $\mathcal{A}$ queries $\mathcal{C}$, it responds as follows:

- **Secret key queries**. When $\mathcal{A}$ queries $\mathcal{C}$ at a point ID_i, $\mathcal{C}$ generates a secret key sk_{ID_i} for ID_i by running $\mathbf{KeyGen}_{\mathsf{IBE}}$. $\mathcal{C}$ computes e_{ID_i} by running **EGen** with the input sk_{ID_i}. $\mathcal{C}$ generates a signature σ_{e_i} for $\mathsf{ID}_i||e_{\mathsf{ID}_i}$ by running **Sign**, and returns $(sk_{\mathsf{ID}_i}, \mathsf{ID}_i||e_{\mathsf{ID}_i}, \sigma_{e_i})$ to $\mathcal{A}$.
- **Type-1 re-encryption key queries**. When $\mathcal{A}$ queries $\mathcal{C}$ at a point ID_i, $\mathcal{C}$ generates an IBE secret key sk_{ID_i} by running $\mathbf{KeyGen}_{\mathsf{IBE}}$, and computes e_{ID_i} by running **EGen** with the input sk_{ID_i}. $\mathcal{C}$ generates a signature σ_{e_i} for $\mathsf{ID}_i||e_{\mathsf{ID}_i}$ by running **Sign**. $\mathcal{C}$ runs $\mathbf{KeyGen}_{\mathsf{PRO}}$ with the inputs e_{ID_i}, and returns the resulting re-encryption key rk_{ID_i} with $(\mathsf{ID}_i||e_{\mathsf{ID}_i}, \sigma_{e_i})$ to $\mathcal{A}$.
- **Type-2 re-encryption key queries**. Suppose that $\mathcal{A}$ queries $\mathcal{C}$ about $(\mathsf{ID}_i||e_{\mathsf{ID}_i}, \sigma_{e_i})$. If $(\mathsf{ID}_i||e_{\mathsf{ID}_i}, \sigma_{e_i})$ has already generated in the answering for secret key query, $\mathcal{C}$ rejects the query. Otherwise $\mathcal{C}$ verifies $(\mathsf{ID}_i||e_{\mathsf{ID}_i}, \sigma_{e_i})$ by running **Verify** with vk_Σ and works as follows;
  - If it is valid then $\mathcal{C}$ runs $\mathbf{KeyGen}_{\mathsf{PRO}}$ with the inputs e_{ID_i}, and returns the resulting re-encryption key rk_{ID_i}.
  - Otherwise $\mathcal{C}$ rejects the query.

Challenge. After some queries, $\mathcal{A}$ selects two equal length plaintexts $M_0, M_1 \in \mathcal{M}$ and sends them to $\mathcal{C}$. $\mathcal{C}$ picks $d \overset{R}{\leftarrow} \{0,1\}$ and computes

$$C_{\mathsf{PK}} = \mathbf{Enc}_{\mathsf{CBE}}(pk, parms, M_d).$$

$\mathcal{C}$ returns C_{PK} to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ continues to issue queries as in Phase 1, and $\mathcal{C}$ responds as before.

Guess. Finally, $\mathcal{A}$ outputs a guess $d' \in \{0,1\}$.

The adversary $\mathcal{A}$ wins if $d' = d$. The hybrid proxy re-encryption system is secure in the sense of IND-ID-CPA if $|\Pr[d' = d] - 1/2|$ is negligible.

Definition 4. *Let $\mathcal{A}$ be an adversary against the hybrid proxy re-encryption system. Define the IND-ID-CPA advantage of $\mathcal{A}$ as follows.*

$$\mathtt{Adv}^{id}_{\mathsf{hyd}}(\mathcal{A}) = 2(\Pr[d' = d] - 1/2). \tag{3}$$

We say that a hybrid proxy re-encryption system is (k, t, q, ϵ) adaptive chosen plaintext secure if for any t time IND-ID-CPA adversary $\mathcal{A}$ that makes at most q chosen queries under a security parameter k we have that $\mathtt{Adv}^{id}_{\mathsf{hyd}}(\mathcal{A}) < \epsilon$. As shorthand, we say that a hybrid proxy re-encryption system is (k, t, q, ϵ) IND-ID-CPA secure.

Note that this game encompasses the notion of semantic security for the CBE system, as well as that for the IBE system, and also the notion that a set of re-encryption keys cannot be "combined"to form new re-encryption keys for other identities. For example, if the CBE system is not semantically secure, then the adversary can win the game by simply distinguishing the challenge ciphertext.

3.3 Construction

We describe our hybrid proxy re-encryption system involving the ElGamal-type CBE system and the BB-IBE system [BB04a]. Let the PKG employ a digital signature scheme ($\mathbf{KeyGen}_{\Sigma}, \mathbf{Sign}, \mathbf{Verify}$). We describe the following algorithms making up the system:

- *The underlying IBE system (BB-IBE system):*
  $\mathbf{SetUp}_{\mathsf{IBE}}(k)$. Given a security parameter k, select a random generator $g \in \mathbb{G}$ and random elements $g_2, h \in \mathbb{G}$. Pick a random $\alpha \in \mathbb{Z}_p^*$. 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.
  $\mathbf{KeyGen}_{\mathsf{IBE}}(mk, parms, \mathsf{ID})$. Given $mk = g_2^{\alpha}$ and ID with $parms$, pick a random $u \in \mathbb{Z}_p^*$. Set

$$sk_{\mathsf{ID}} = (d_0, d_1) = \left(g_2^{\alpha}(g_1^{\mathsf{ID}}h)^u, g^u\right).$$

  $\mathbf{Enc}_{\mathsf{IBE}}(\mathsf{ID}, parms, M)$. To encrypt a message $M \in \mathbb{G}_1$ under the public key $\mathsf{ID} \in \mathbb{Z}_p^*$, pick a random $r \in \mathbb{Z}_p^*$ and compute

$$C_{\mathsf{ID}} = \left(g^r, (g_1^{\mathsf{ID}}h)^r, M\hat{e}(g_1, g_2)^r\right) \in \mathbb{G}^2 \times \mathbb{G}_1.$$

  $\mathbf{Dec}_{\mathsf{IBE}}(sk_{\mathsf{ID}}, parms, C_{\mathsf{ID}})$. Given ciphertext $C_{\mathsf{ID}} = (C_1, C_2, C_3)$ and the secret key $sk_{\mathsf{ID}} = (d_0, d_1)$ with $prams$, compute

$$M = \frac{C_3\hat{e}(d_1, C_2)}{\hat{e}(d_0, C_1)}.$$

- *The underlying CBE system (ElGamal-type CBE system):*
  $\mathbf{KeyGen}_{\mathsf{CBE}}(k, parms)$. Given a security parameter k and $parms$, pick a random $\beta, \theta, \delta \in \mathbb{Z}_p^*$. 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 key is $sk = (\theta, \beta, \delta)$.
  $\mathbf{Enc}_{\mathsf{CBE}}(pk, parms, M)$. Given $pk = (g_3, g_4, g_5)$ and a message M with $parms$, pick a random $r \in \mathbb{Z}_p^*$ and compute

$$C_{\mathsf{PK}} = \left(g_3^r, g_4^r, g_5^r, M\hat{e}(g_1, g_2)^r\right) \in \mathbb{G}^3 \times \mathbb{G}_1.$$

  $\mathbf{Dec}_{\mathsf{CBE}}(sk, parms, C_{\mathsf{PK}})$. Given $C_{\mathsf{PK}} = (C_1, C_2, C_3, C_4)$ and the secret key $sk = (\theta, \beta, \delta)$ with $parms$, compute $M = C_4/\hat{e}(C_2^{1/\beta}, g_2)$.
- *The delegation system:*

EGen$(sk_{\mathsf{ID}}, parms)$. Given $sk_{\mathsf{ID}} = (d_0, d_1) = (g_2^{\alpha}(g_1^{\mathsf{ID}}h)^u, g^u)$ for ID with $parms$, set $e_{\mathsf{ID}} = d_1 = g^u$.

KeyGen$_{\mathsf{PRO}}(sk, e_{\mathsf{ID}}, parms)$. Given $sk = (\theta, \beta, \delta)$ and $e_{\mathsf{ID}} = g^u$ for ID with $parms$, set $rk_{\mathsf{ID}} = (\theta, g^{u/\beta}, \delta)$.

ReEnc$(rk_{\mathsf{ID}}, parms, C_{\mathsf{PK}}, \mathsf{ID})$. Given a CBE ciphertext $C_{\mathsf{PK}} = (C_1, C_2, C_3, C_4)$, the re-encryption key $rk_{\mathsf{ID}} = (\theta, g^{u/\beta}, \delta)$ and ID with $parms$, re-encrypt the ciphertext C_{PK} into C_{ID} as follows.

$$C_{\mathsf{ID}} = (C_1', C_2', C_3') = \left(C_1^{1/\theta}, C_3^{1/\delta}, C_4\hat{e}(g^{u/\beta}, C_2^{\mathsf{ID}})\right) \in \mathbb{G}^2 \times \mathbb{G}_1.$$

Check$(parms, C_{\mathsf{PK}}, pk)$. Given $C_{\mathsf{PK}} = (C_1, C_2, C_3, C_4)$ and $pk = (g_3, g_4, g_5)$ with $parms$, set $v_1 = \hat{e}(C_1, g_4)$, $v_2 = \hat{e}(C_2, g_3)$, $v_3 = \hat{e}(C_2, g_5)$ and $v_4 = \hat{e}(C_3, g_4)$. If $v_1 = v_2$ and $v_3 = v_4$ then output 1, otherwise output 0.

3.4 Security analysis

We first describe why the proxy re-encrypting does not make the underlying public key cryptosystems weak.

In our system, the re-encryption key $rk_{\mathsf{ID}} = (\theta, g^{u/\beta}, \delta)$ involves the delegator's decryption key β and the second component of delegatee's IBE secret key $d_1 = g^u$. Thus it might reveal some information about β and g^u; however this does not make the underlying public key cryptosystems weak. This is because

- it is computationally hard to recover β completely from the public key and the re-encryption key if the discrete logarithm problem is hard, and
- the underlying IBE can be proved semantically secure even if the second component of the secret key d_1 is exposed. (See Lemma 1 in Appendix A).

Therefore our proxy re-encryption system is secure as long as the re-encryption key is generated and deployed appropriately, and the digital signature system is used to ensure appropriate re-encryption key generation.

The above observation yields the security notion in section 3.2. Then, it is sufficient to show our system being secure if the following theorem holds.

Theorem 1. *Suppose that the (k, t, ϵ)-dBDH assumption holds and the PKG's digital signature scheme is (k, t', q, ϵ')-strong existentially unforgeable. Then the hybrid proxy re-encryption system is (k, t'', q, ϵ'') IND-ID-CPA secure for any q, k, $\epsilon'' \leq \epsilon + \epsilon'$, and $t'' + t' < t - \Theta(\tau_e q + \tau_s q + \tau_v q)$ where τ_e is the maximum time for an exponentiation in $\mathbb{G}$, τ_s is the maximum time for running* **Sgin**, *and τ_v is the maximum time for running* **Verify**.

Proof. Let $\mathcal{A}$ be an adversary against the hybrid proxy re-encryption system in the IND-ID-CPA sense. We construct an adversary B which solves the dBDH problem in $\mathbb{G}$ by utilizing $\mathcal{A}$. Providing that $\mathcal{B}$ is given an input $(g, \Gamma_1, \Gamma_2, \Gamma_3, X) = (g, g^a, g^b, g^c, X)$, where $X = \hat{e}(g, g)^{abc}$ or $X = R \stackrel{R}{\leftarrow} \mathbb{G}_1$. We describe how $\mathcal{B}$ works in the following.

Initialization. $\mathcal{B}$ generates a blank list QAL to write down query-answer pairs for every query.

Setup. $\mathcal{B}$ selects a (k, t', q, ϵ')-strong existentially unforgeable digital signature scheme $(\mathbf{KeyGen}_{\Sigma}, \mathbf{Sign}, \mathbf{Verify})$ and generates $(sk_{\Sigma}, vk_{\Sigma})$ by running $\mathbf{KeyGen}_{\Sigma}$. To generate the system parameters, $\mathcal{B}$ picks $x, y, z, w \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and sets $g_1 = \Gamma_1$, $g_2 = \Gamma_2$, $h = g^z$ $g_3 = g^x$, $g_4 = g^y$ and $g_5 = h^w$. It gives $\mathcal{A}$ the system parameters $parms = (g, g_1, g_2, h)$, $pk = (g_3, g_4, g_5)$ and vk_{Σ}. Note that the corresponding PKG's master-secret key, which is unknown to $\mathcal{B}$, is $g_2^a = g^{ab} \in \mathbb{G}$.

Phase 1. Given pk, $parms$ and vk_{Σ}, $\mathcal{A}$ asks some queries to the challenger. When $\mathcal{A}$ queries the challenger, $\mathcal{B}$ works as follows.

- **Secret key queries**. Suppose that $\mathcal{A}$ queries the challenger at a point ID_i.
  - If $\mathsf{ID}_i \neq 0$ then $\mathcal{B}$ selects $r_i \overset{R}{\leftarrow} \mathbb{Z}_p^*$, sets

$$sk_{\mathsf{ID}_i} = (d_0, d_1) = \left(g_2^{\frac{-z}{\mathsf{ID}_i}} (g_1^{\mathsf{ID}_i} g^z)^{r_i}, g_2^{\frac{-1}{\mathsf{ID}_i}} g^{r_i}\right)$$

    and $e_{\mathsf{ID}_i} = d_1$. $\mathcal{B}$ computes $\mathbf{Sign}(sk_{\Sigma}, \mathsf{ID}_i || e_{\mathsf{ID}_i}) = \sigma_{e_i}$, and returns $(sk_{\mathsf{ID}_i}, \mathsf{ID}_i || e_{\mathsf{ID}_i}, \sigma_{e_i})$ to $\mathcal{A}$. $\mathcal{B}$ adds the query and the answer to the list QAL.
  - Otherwise $\mathcal{B}$ rejects the query.
- **Type-1 re-encryption key queries**. When $\mathcal{A}$ queries the challenger at a point ID_i, $\mathcal{B}$ selects $r_i' \overset{R}{\leftarrow} \mathbb{Z}_p^*$, sets $rk_{\mathsf{ID}_i} = (x, g_1^{r_i'}, w)$ and $e_{\mathsf{ID}_i} = g^{yr_i'}$. $\mathcal{B}$ generates a signature σ_{e_i} for $\mathsf{ID}_i || e_{\mathsf{ID}_i}$ by running $\mathbf{Sign}$. $\mathcal{B}$ returns rk_{ID_i} with $(\mathsf{ID}_i || e_{\mathsf{ID}_i}, \sigma_{e_i})$ to $\mathcal{A}$. $\mathcal{B}$ adds the query and the answer to the list QAL.
- **Type-2 re-encryption key queries**. Suppose that $\mathcal{A}$ queries the challenger about $(\mathsf{ID}_i || e_{\mathsf{ID}_i}, \sigma_{e_i})$. $\mathcal{B}$ checks the list QAL.
  - If the IBE secret key sk_{ID_i} corresponding to $(\mathsf{ID}_i || e_{\mathsf{ID}_i}, \sigma_{e_i})$ is in the list then $\mathcal{B}$ rejects the query.
  - If rk_{ID_i} corresponding to $(\mathsf{ID}_i || e_{\mathsf{ID}_i}, \sigma_{e_i})$ is in the list then $\mathcal{B}$ returns rk_{ID_i} to $\mathcal{A}$.
  - Otherwise, $\mathcal{B}$ computes $v = \mathbf{Verify}(vk_{\Sigma}, \mathsf{ID}_i || e_{\mathsf{ID}_i}, \sigma_{e_i})$.
    - If $v = 1$ then $\mathcal{B}$ halts.
    - Otherwise, $\mathcal{B}$ rejects the query.

Challenge. After some queries, $\mathcal{A}$ selects two equal length plaintexts $M_0, M_1 \in \mathcal{M}$. Given (M_0, M_1), $\mathcal{B}$ selects $d \overset{R}{\leftarrow} \{0, 1\}$ and sets

$$C_{\mathsf{PK}} = (\Gamma_3^x, \Gamma_3^y, \Gamma_3^{zw}, M_d X).$$

$\mathcal{B}$ returns C_{PK} to $\mathcal{A}$. Notice that if $X = \hat{e}(g, g)^{abc} = \hat{e}(g_1, g_2)^c$ then C_{PK} is a valid encryption of M_d. On the other hand, if X is uniform and independent in $\mathbb{G}_1$ then C_{PK} is independent of d in the adversary's view.

Phase 2. $\mathcal{A}$ continues to issue queries as in Phase 1, and $\mathcal{B}$ responds as before.

Solve. Finally, $\mathcal{A}$ outputs a guess $d' \in \{0,1\}$. $\mathcal{B}$ concludes its own game by outputting a guess as follows. If $d' = d$ then $\mathcal{B}$ outputs 1 meaning $X = \hat{e}(g,g)^{abc}$. Otherwise, it outputs 0 meaning $X = R$.

We claim that $\mathcal{B}$ generates a valid secret key and the corresponding auxiliary information for ID_i. To see this, let $\tilde{u}_i = r_i - \frac{b}{\mathsf{ID}_i}$. Then we have that

$$\begin{aligned}(d_{\mathsf{ID}_i}, e_{\mathsf{ID}_i}) = \left(g_2^{\frac{-z}{\mathsf{ID}_i}}(g_1^{\mathsf{ID}_i}g^z)^{r_i}, g_2^{\frac{-1}{\mathsf{ID}_i}}g^{r_i}\right) &= \left(\frac{g_2^a(g_1^{\mathsf{ID}_i}g^z)^{r_i}}{(g_1^{\mathsf{ID}_i}g^z)^{\frac{b}{\mathsf{ID}_i}}}, g^{r_i - \frac{b}{\mathsf{ID}_i}}\right) \\ &= \left(g_2^a(g_1^{\mathsf{ID}_i}g^z)^{r_i - \frac{b}{\mathsf{ID}_i}}, g^{r_i - \frac{b}{\mathsf{ID}_i}}\right) \\ &= \left(g_2^a(g_1^{\mathsf{ID}_i}h)^{\tilde{u_i}}, g^{\tilde{u_i}}\right)\end{aligned}$$

We also claim that $\mathcal{B}$ can perfectly simulate the re-encryption key for ID_i since it looks random and independent of any other values if the adversary does not obtain the corresponding secret key.

$\mathcal{B}$ fails to simulate the challenger if $\mathcal{B}$ halts in the Type-2 re-encryption key query. Otherwise $\mathcal{B}$ perfectly simulates the challenger. The maximum probability of that $\mathcal{B}$ halts is obviously upper-bounded by $\mathtt{Adv}^{eu}(\mathcal{A})$. Therefore, we conclude the theorem.

3.5 Application

IBE requires less certificate management of its users and allows them to set an arbitrary string as a public key. Then IBE has high congeniality with services working on platforms which manage some identities with limited resource [2], like mobile phone. Now we consider a CBE user utilizing a desktop computer and an IBE user utilizing a mobile phone. Since a mobile phone platform can not allow heavy certificate management, the CBE user has to deploy the IBE system in its desktop computer if the CBE user would like to send the mobile phone user a ciphertext; however, it requires extra cost of the CBE user.

The hybrid system allows a CBE user to send ciphertexts to an IBE user without deploying the additional encryption manner. When the CBE user send a message to an IBE user secretly, the CBE user has only to encrypt a message in the CBE manner, then send it to the proxy.

4 Identity based proxy re-encryption system

An identity-based proxy re-encryption system consists of an IBE system and additional algorithms that allow ciphertexts encrypted under one's IBE public key to be transformed into the different ciphertexts that can be decrypted by the other's IBE secret key. In this section, we describe our identity-based proxy re-encryption system.

[2] Limited memory, low power CPU, low power buttery and so on.

4.1 Definition

There are five entities involved in an identity-based proxy re-encryption system, delegator, proxy, delegatee, PKG and *Re-encryption Key Generator*, RKG. [3] In this system, each of delegator and delegatee is an IBE user. The RKG generates re-encryption keys and sets them into the proxy via secure channel while the delegator does in the hybrid proxy re-encryption system. An identity-based proxy re-encryption system consists of: 1) the four algorithms making up an IBE system $\mathbf{SetUp}_{\mathsf{IBE}}$, $\mathbf{KeyGen}_{\mathsf{IBE}}$, $\mathbf{Enc}_{\mathsf{IBE}}$, and $\mathbf{Dec}_{\mathsf{IBE}}$, 2) and five algorithms for re-encryption, which are –

EGen$(sk_{\mathsf{ID}}, parms)$. Given an IBE secret key sk_{ID} for ID with $parms$, generate e_{ID} for re-encryption key generation.
KeyGen$_{\mathsf{RKG}}(mk, parms)$. Given an IBE master-secret key mk with $parms$, generate a secret key sk_R for re-encryption.
KeyGen$_{\mathsf{PRO}}(sk_R, e_{\mathsf{ID}'}, parms, \mathsf{ID}, \mathsf{ID}')$. Given sk_R, $e_{\mathsf{ID}'}$, the delegator's identity ID and the delegatee's identity ID' with $parms$, generate a re-encryption key $rk_{\mathsf{ID}\to\mathsf{ID}'}$.
ReEnc$(rk_{\mathsf{ID}\to\mathsf{ID}'}, parms, C_{\mathsf{ID}}, \mathsf{ID}, \mathsf{ID}')$. Given the delegator's identity ID, the delegatee's identity ID', the re-encryption key $rk_{\mathsf{ID}\to\mathsf{ID}'}$, and an IBE ciphertext C_{ID} with $parms$, re-encrypt C_{ID} into the different IBE ciphertext $C_{\mathsf{ID}'}$.
Check$(parms, C_{\mathsf{ID}}, \mathsf{ID})$. Given the delegator's identity ID and an IBE ciphertext C_{ID} with $parms$, output 0 if C_{ID} is a malformed ciphertext for ID. Otherwise, output 1.

We assume that the PKG deploys the digital signature scheme described in Sec. 2.3. When one delegates her decryption rights to the other, the identity-based proxy re-encryption system works as follows.

– *SetUp:* The PKG generates
  1. its signing key sk_Σ and the corresponding verification key vk_Σ by running $\mathbf{KeyGen}_\Sigma$,
  2. its master-secret key mk and public parameters $parms$ by running $\mathbf{SetUp}_{\mathsf{IBE}}$,
  3. and a secret key sk_R for the RKG by running $\mathbf{KeyGen}_{\mathsf{RKG}}$.

  The PKG makes $(vk_\Sigma, parms)$ public and sets sk_R in the RKG, keeping (mk, sk_Σ) to itself.
– *Re-encryption key generation and deployment:* When one requests the delegation from ID (i.e. delegator) to ID' (i.e. delegatee), the RKG makes sure that ID approves the delegation to ID'. If ID denies it, the RKG rejects the request. Otherwise the RKG works as follows.
  - If no IBE secret key has issued to the delegatee ID', the PKG generates $sk_{\mathsf{ID}'}$ by running $\mathbf{KeyGen}_{\mathsf{IBE}}$, and computes $e_{\mathsf{ID}'}$ by running **EGen**. The PKG makes a digital signature σ'_e for $\mathsf{ID}'||e_{\mathsf{ID}'}$ by running **Sign**. Then PKG issues $(sk_{\mathsf{ID}'}, \mathsf{ID}'||e_{\mathsf{ID}'}, \sigma'_e)$ to the delegatee.
  - On receiving $(\mathsf{ID}'||e_{\mathsf{ID}'}, \sigma'_e)$ from the delegatee, the RKG verifies it by running **Verify** with vk_Σ.

[3] The PKG and the RKG might be operated by one entity.

* If it is valid then the RKG generates a re-encryption key $rk_{\mathsf{ID}\to\mathsf{ID}'}$ by running $\mathbf{KeyGen}_{\mathsf{PRO}}$ with the input $e_{\mathsf{ID}'}$. The RKG sets $rk_{\mathsf{ID}\to\mathsf{ID}'}$ in the proxy.
* Otherwise the RKG rejects the request.

– *Re-encryption:* Suppose that one sends a ciphertext C_{ID} to the delegatee ID'. On receiving the ciphertext C_{ID}, the proxy rejects the re-encryption request if $rk_{\mathsf{ID}\to\mathsf{ID}'}$ does not exists. Otherwise, the proxy runs the algorithm **Check** with the input $(parms, C_{\mathsf{ID}}, \mathsf{ID})$.
  - If **Check** outputs 0 then the proxy rejects the re-encryption request.
  - Otherwise, the proxy re-encrypts C_{ID} into $C_{\mathsf{ID}'}$ by running **ReEnc**, and sends $C_{\mathsf{ID}'}$ to the delegatee ID'.
– *Decryption:* The delegatee decrypts $C_{\mathsf{ID}'}$ by running $\mathbf{Dec}_{\mathsf{IBE}}$ with the secret key $sk_{\mathsf{ID}'}$.

4.2 Security notion

In the following, each value appeared in i-th query by the adversary and in the corresponding answer is denoted with letter i. We sometimes denote a delegator's identity by ID_i and a delegatee's one by ID'_i that the adversary asks in i-th query. $\mathsf{ID}_i \to \mathsf{ID}'_i$ represents the delegation from ID_i to ID'_i.

Chosen plaintext security: We model chosen plaintext security for an identity-based proxy re-encryption system as a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. In this game, the adversary is allowed to adaptively choose the secret key queries, re-encryption key queries and re-encryption queries. Intuitively, these queries imply the situation that: (1)the adversary compromises arbitrary IBE users and obtains their secret keys, (2)the adversary compromises arbitrary proxies and obtains the re-encryption keys, (3)the adversary requests the re-encryption key generation of the RKG, (4)and the adversary obtains re-encrypted ciphertexts by using proxy as an oracle. Since the adversary obviously wins the game if it obtains the secret key for the target identity, she is not allowed to ask such queries. Besides this, the adversary also wins the game if she obtains both of the delegatee's secret key and the corresponding re-encryption key involving the same identity. Therefore she is also not allowed to ask such queries.

More precisely, IND-ID-CPA security is defined as follows:

Setup. The challenger $\mathcal{C}$ selects a digital signature scheme $(\mathbf{KeyGen}_{\Sigma}, \mathbf{Sign}, \mathbf{Verify})$. $\mathcal{C}$ generates
1. $(sk_{\Sigma}, vk_{\Sigma})$ by running $\mathbf{KeyGen}_{\Sigma}$,
2. $(parms, mk)$ by running $\mathbf{SetUp}_{\mathsf{IBE}}$, and
3. sk_R by running $\mathbf{KeyGen}_{\mathsf{RKG}}$.

$\mathcal{C}$ gives $(parms, vk_{\Sigma})$ to $\mathcal{A}$, keeping (mk, sk_{Σ}, sk_R) to itself.

Phase 1. Given $(parms, vk_{\Sigma})$, $\mathcal{A}$ adaptively queries $\mathcal{C}$. When $\mathcal{A}$ queries $\mathcal{C}$, it responds as follows:

– **Secret key queries**. When $\mathcal{A}$ queries $\mathcal{C}$ at a point ID_i, $\mathcal{C}$ generates a secret key sk_{ID_i} for ID_i by running $\mathbf{KeyGen}_{\mathsf{IBE}}$. $\mathcal{C}$ computes e_{ID_i} by running **EGen** with the input sk_{ID_i}. $\mathcal{C}$ generates a signature σ_{e_i} for $\mathsf{ID}_i||e_{\mathsf{ID}_i}$ by running **Sign**, and $\mathcal{C}$ returns $(sk_{\mathsf{ID}_i}, \mathsf{ID}_i||e_{\mathsf{ID}_i}, \sigma_{e_i})$ to $\mathcal{A}$.
– **Type-1 re-encryption key queries**. When $\mathcal{A}$ queries $\mathcal{C}$ about $\mathsf{ID}_i \rightarrow \mathsf{ID}'_i$, $\mathcal{C}$ generates an IBE secret key $sk_{\mathsf{ID}'_i}$ by running $\mathbf{KeyGen}_{\mathsf{IBE}}$, and computes $e_{\mathsf{ID}'_i}$ by running **EGen** with the input $sk_{\mathsf{ID}'_i}$. $\mathcal{C}$ generates a signature $\sigma_{e'_i}$ for $\mathsf{ID}'_i||e_{\mathsf{ID}'_i}$ by running **Sign**. $\mathcal{C}$ runs $\mathbf{KeyGen}_{\mathsf{PRO}}$ with the inputs $e_{\mathsf{ID}'_i}$, and returns the resulting re-encryption key $rk_{\mathsf{ID}_i \rightarrow \mathsf{ID}'_i}$ with $(\mathsf{ID}'_i||e_{\mathsf{ID}'_i}, \sigma_{e'_i})$ to $\mathcal{A}$.
– **Type-2 re-encryption key queries**. Suppose that $\mathcal{A}$ queries $\mathcal{C}$ about $(\mathsf{ID}_i \rightarrow \mathsf{ID}'_i, \mathsf{ID}'_i||e_{\mathsf{ID}'_i}, \sigma_{e'_i})$. If $(\mathsf{ID}'_i||e_{\mathsf{ID}'_i}, \sigma_{e'_i})$ has already generated in the answering for secret key query, then $\mathcal{C}$ rejects the query. Otherwise $\mathcal{C}$ verifies $(\mathsf{ID}'_i||e_{\mathsf{ID}'_i}, \sigma_{e'_i})$ by running **Verify** with vk_{Σ} and works as follows;
  • If it is valid then $\mathcal{C}$ runs $\mathbf{KeyGen}_{\mathsf{PRO}}$ with the input $e_{\mathsf{ID}'_i}$, and returns the resulting re-encryption key $rk_{\mathsf{ID}_i \rightarrow \mathsf{ID}'_i}$.
  • Otherwise $\mathcal{C}$ rejects the query.
– **Re-encryption queries**. Suppose that $\mathcal{A}$ queries $\mathcal{C}$ about $(sk_{\mathsf{ID}'_i}, C_{\mathsf{ID}_i}, \mathsf{ID}_i \rightarrow \mathsf{ID}'_i)$. If $sk_{\mathsf{ID}'_i}$ has never issued to $\mathcal{A}$ then $\mathcal{C}$ rejects the query. Otherwise, $\mathcal{C}$ runs **Check** with the input $(parms, C_{\mathsf{ID}_i}, \mathsf{ID}_i)$.
  • If **Check** outputs 0 then $\mathcal{C}$ rejects the query.
  • Otherwise, $\mathcal{C}$ generates $e_{\mathsf{ID}'_i}$ by running **EGen** with $sk_{\mathsf{ID}'_i}$ as an input. $\mathcal{C}$ generates $rk_{\mathsf{ID}_i \rightarrow \mathsf{ID}'_i}$ by running $\mathbf{KeyGen}_{\mathsf{PRO}}$ with the input $e_{\mathsf{ID}'_i}$. $\mathcal{C}$ re-encrypts C_{ID_i} into $C_{\mathsf{ID}'_i}$ by running **ReEnc** with the input $rk_{\mathsf{ID}_i \rightarrow \mathsf{ID}'_i}$. $\mathcal{C}$ returns $C_{\mathsf{ID}'_i}$ to $\mathcal{A}$.

Challenge. After some queries, $\mathcal{A}$ selects two equal length plaintexts $M_0, M_1 \in \mathcal{M}$ and a target identity ID^* which no secret key for ID^* has issued, and sends them to $\mathcal{C}$. Given $(M_0, M_1, \mathsf{ID}^*)$, $\mathcal{C}$ selects $d \overset{R}{\leftarrow} \{0,1\}$ and computes

$$C_{\mathsf{ID}^*} = \mathbf{Enc}_{\mathsf{IBE}}(\mathsf{ID}^*, parms, M_d).$$

$\mathcal{C}$ returns C_{ID^*} to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ continues to issue queries as in Phase 1, and $\mathcal{C}$ responds as before except the following case.
– If $\mathcal{A}$ makes the secret key query at the point ID^*, then $\mathcal{C}$ rejects.
– If $\mathcal{A}$ makes the re-encryption query such that $\mathsf{ID}_i = \mathsf{ID}^*$, then $\mathcal{C}$ rejects.

Guess. Finally, $\mathcal{A}$ outputs a guess $d' \in \{0,1\}$.

The adversary $\mathcal{A}$ wins if $d' = d$. An identity-based proxy re-encryption system is secure in the sense of IND-ID-CPA if $|\Pr[d' = d] - 1/2|$ is negligible.

Definition 5. *Let $\mathcal{A}$ be an adversary against the identity-based proxy re-encryption system. Define the IND-ID-CPA advantage of $\mathcal{A}$ as follows.*

$$\mathtt{Adv}^{id}_{\mathsf{ibp}}(\mathcal{A}) = 2(\Pr[d' = d] - 1/2). \tag{4}$$

We say that an identity-based proxy re-encryption system is (k, t, q, ϵ) adaptive chosen plaintext secure if for any t time IND-ID-CPA adversary $\mathcal{A}$ that makes

at most q chosen queries under a security parameter k we have that $\mathtt{Adv}^{id}_{\mathsf{ibp}}(\mathcal{A}) < \epsilon$. *As shorthand, we say that an identity-based proxy re-encryption system is* (k, t, q, ϵ) *IND-ID-CPA secure.*

We define the selective adversary who is identical to the above adversary except that it discloses to the challenger the target identity ID^* before the setup. The restrictions on queries from Phase 2 also hold in Phase 1. We denote the selective IND-ID-CPA by IND-sID-CPA and the advantage of the selective adversary by $\mathtt{Adv}^{sid}_{\mathsf{ibp}}$. The definition is as same as that of Definition. 5.

4.3 Construction

Let the PKG deploys a digital signature scheme $(\mathbf{KeyGen}_{\Sigma}, \mathbf{Sign}, \mathbf{Verify})$. Our identity-based proxy re-encryption system involves BB-IBE system described in Sec. 3.3 and the following algorithms.

- *The delegation system:*
  EGen$(sk_{\mathsf{ID}}, parms)$. Given $sk_{\mathsf{ID}} = (d_0, d_1) = (g_2^{\alpha}(g_1^{\mathsf{ID}}h)^u, g^u)$ with $parms$, set $e_{\mathsf{ID}} = d_1$.
  KeyGen$_{\mathsf{RKG}}(mk, parms)$. Given $mk = \alpha$ with $parms$, set $sk_R = \alpha$.
  KeyGen$_{\mathsf{PRO}}(sk_R, e_{\mathsf{ID}'}, parms, \mathsf{ID}, \mathsf{ID}')$. Given $sk_R = \alpha$, $e_{\mathsf{ID}'} = g^{u'}$ with $parms$, set $rk_{\mathsf{ID}\to\mathsf{ID}'} = (\mathsf{ID} \to \mathsf{ID}', g^{u'\alpha})$.
  ReEnc$(rk_{\mathsf{ID}\to\mathsf{ID}'}, parms, C_{\mathsf{ID}}, \mathsf{ID}, \mathsf{ID}')$. Given the delegator's identity ID, the delegatee's identity ID', $rk_{\mathsf{ID}\to\mathsf{ID}'} = (\mathsf{ID} \to \mathsf{ID}', g^{u'\alpha})$, $C_{\mathsf{ID}} = (C_1, C_2, C_3)$ with $parms$, re-encrypt the ciphertext C_{ID} into $C_{\mathsf{ID}'}$ as follows.

$$C_{\mathsf{ID}'} = (C_1', C_2', C_3') = \big(C_1, C_2, C_3\hat{e}(C_1^{\mathsf{ID}'-\mathsf{ID}}, g^{u'\alpha})\big) \in \mathbb{G}^2 \times \mathbb{G}_1.$$

  Check$(parms, C_{\mathsf{ID}}, \mathsf{ID})$. Given the delegator's identity ID and $C_{\mathsf{ID}} = (C_1, C_2, C_3)$ with $parms$, compute $v_0 = \hat{e}(C_1, g_1^{\mathsf{ID}}h)$ and $v_1 = \hat{e}(C_2, g)$. If $v_0 = v_1$ then output 1. Otherwise output 0.

In this system, we let the PKG set $mk = \alpha$ while $mk = g_2^{\alpha}$ described in Sec. 3.3.

Remark: We consider the case that a malicious player modifies a target ciphertext $C_{\mathsf{ID}^*} = (g^{r^*}, (g_1^{\mathsf{ID}^*}h)^{r^*}, M^*\hat{e}(g_1, g_2)^{r^*})$ into the different ciphertext for ID (for short C_{ID}) such that she can derive some information about M^* from the underlying message of C_{ID} by utilizing the proxy as an oracle. The algorithm **Check** prevents such modification where $\mathsf{ID} \neq \mathsf{ID}^*$ because passing through the check implies that C_{ID} is the form of $C_{\mathsf{ID}} = (g^r, (g_1^{\mathsf{ID}}h)^r, M\hat{e}(g_1, g_2)^r)$, and it is obviously hard to make such modification since the underlying BB-IBE is semantically secure.

4.4 Security analysis

In this section, we show that the proposed system is semantically secure.

Theorem 2. *Suppose that the (k, t, ϵ)-dBDH assumption holds and the PKG's digital signature scheme is (k, t', q, ϵ')-strong existentially unforgeable. Then the identity-based proxy re-encryption system is (k, t'', q, ϵ'') IND-sID-CPA secure for any q, k, $\epsilon'' \leq \epsilon + \epsilon'$, and $t'' + t' < t - \Theta(\tau_e q + \tau_s q + \tau_v q)$ where τ_e is the maximum time for an exponentiation in $\mathbb{G}$, τ_s is the maximum time for running* **Sgin**, *and τ_v is the maximum time for running* **Verify**.

Proof. Let $\mathcal{A}$ be an adversary against the identity-based proxy re-encryption system in the IND-sID-CPA sense. We construct an adversary $\mathcal{B}$ which solves the dBDH problem in $\mathbb{G}$ by utilizing $\mathcal{A}$. Providing that $\mathcal{B}$ is given an input $(g, \Gamma_1, \Gamma_2, \Gamma_3, X) = (g, g^a, g^b, g^c, X)$, where $X = \hat{e}(g, g)^{abc}$ or $X = R \overset{R}{\leftarrow} \mathbb{G}_1$. We describe how $\mathcal{B}$ works in the following.

Initialization. The selective identity game begins with $\mathcal{A}$ first outputting a target identity ID^*. $\mathcal{B}$ generates two blank lists SKL and RKL.

Setup. $\mathcal{B}$ selects a (k, t', q, ϵ')-strong existentially unforgeable digital signature scheme $(\mathbf{KeyGen}_\Sigma, \mathbf{Sign}, \mathbf{Verify})$ and generates (sk_Σ, vk_Σ) by running $\mathbf{KeyGen}_\Sigma$. To generate the system parameters, algorithm $\mathcal{B}$ picks $z \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and sets $g_1 = \Gamma_1$, $g_2 = \Gamma_2$, $h = g_1^{-\mathsf{ID}^*} g^z$, and $parms = (g, g_1, g_2, h)$. $\mathcal{B}$ gives $(parms, vk_\Sigma)$ to $\mathcal{A}$. Note that the corresponding PKG's master-secret key, which is unknown to $\mathcal{B}$, is $g_2^a = g^{ab} \in \mathbb{G}$.

Phase 1. Given $parms$ and vk_Σ, $\mathcal{A}$ asks some queries to the challenger. When $\mathcal{A}$ queries the challenger, $\mathcal{B}$ works as follows.

- **Secret key queries**. Suppose that $\mathcal{A}$ queries the challenger at a point ID_i. If $\mathsf{ID}_i = \mathsf{ID}^*$ then $\mathcal{B}$ rejects the query. Otherwise, $\mathcal{B}$ selects $r_i \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and sets

$$sk_{\mathsf{ID}_i} = (d_0, d_1) = \left(g_2^{\frac{-z}{\mathsf{ID}_i - \mathsf{ID}^*}} (g_1^{\mathsf{ID}_i - \mathsf{ID}^*} g^z)^{r_i}, g_2^{\frac{-1}{\mathsf{ID}_i - \mathsf{ID}^*}} g^{r_i}\right)$$

  and $e_{\mathsf{ID}_i} = d_1$. $\mathcal{B}$ computes $\mathbf{Sign}(sk_\Sigma, \mathsf{ID}_i || e_{\mathsf{ID}_i}) = \sigma_{e_i}$, and returns $(sk_{\mathsf{ID}_i}, \mathsf{ID}_i || e_{\mathsf{ID}_i}, \sigma_{e_i})$ to $\mathcal{A}$. $\mathcal{B}$ adds the query and the corresponding answer to the list SKL.
- **Type-1 re-encryption key queries**. When $\mathcal{A}$ queries the challenger about $\mathsf{ID}_i \rightarrow \mathsf{ID}'_i$, $\mathcal{B}$ selects $r'_i \overset{R}{\leftarrow} \mathbb{Z}_p^*$, sets $rk_{\mathsf{ID}_i \rightarrow \mathsf{ID}'_i} = (\mathsf{ID}_i \rightarrow \mathsf{ID}'_i, g_1^{r'_i})$ and $e_{\mathsf{ID}'_i} = g^{r'_i}$. $\mathcal{B}$ generates a signature $\sigma_{e'_i}$ for $\mathsf{ID}'_i || e_{\mathsf{ID}'_i}$ by running **Sign**. $\mathcal{B}$ returns $rk_{\mathsf{ID}_i \rightarrow \mathsf{ID}'_i}$ with $(\mathsf{ID}'_i || e_{\mathsf{ID}'_i}, \sigma_{e'_i})$ to $\mathcal{A}$. $\mathcal{B}$ adds $(rk_{\mathsf{ID}_i \rightarrow \mathsf{ID}'_i}, \mathsf{ID}'_i || e_{\mathsf{ID}'_i}, \sigma_{e'_i})$ to the list RKL.
- **Type-2 re-encryption key queries**. Suppose that $\mathcal{A}$ queries the challenger about $(\mathsf{ID}_i \rightarrow \mathsf{ID}'_i, \mathsf{ID}'_i || e_{\mathsf{ID}'_i}, \sigma_{e'_i})$. If the IBE secret key $sk_{\mathsf{ID}'_i}$ corresponding to $(\mathsf{ID}'_i || e_{\mathsf{ID}'_i}, \sigma_{e'_i})$ is in the list SKL then $\mathcal{B}$ rejects the query. Otherwise,
  - If $rk_{\mathsf{ID}_i \rightarrow \mathsf{ID}'_i}$ corresponding to $(\mathsf{ID}'_i || e_{\mathsf{ID}'_i}, \sigma_{e'_i})$ is in the list RKL then $\mathcal{B}$ returns $rk_{\mathsf{ID}_i \rightarrow \mathsf{ID}'_i}$ to $\mathcal{A}$.
  - Otherwise, $\mathcal{B}$ computes $v = \mathbf{Verify}(vk_\Sigma, \mathsf{ID}'_i || e_{\mathsf{ID}'_i}, \sigma_{e'_i})$.
    - If $v = 1$ then $\mathcal{B}$ halts.

$*$ Otherwise, $\mathcal{B}$ rejects the query.

– **Re-encryption queries**. Suppose that $\mathcal{A}$ queries $\mathcal{C}$ about $(sk_{\mathsf{ID}'_i}, C_{\mathsf{ID}_i}, \mathsf{ID}_i \rightarrow \mathsf{ID}'_i)$ where $C_{\mathsf{ID}_i} = (C_1, C_2, C_3)$. If $sk_{\mathsf{ID}'_i}$ is not in the list SKL or $\mathsf{ID}_i = \mathsf{ID}^*$ then $\mathcal{B}$ rejects the query. Otherwise $\mathcal{B}$ computes $v_0 = \hat{e}(C_1, g_1^{\mathsf{ID}_i} h)$ and $v_1 = \hat{e}(C_2, g)$.
  - If $v_0 \neq v_1$ then $\mathcal{B}$ rejects the query.
  - Otherwise, $\mathcal{B}$ generates a secret key $sk_{\mathsf{ID}_i} = (d_0, d_1)$ as the way of that at **Secret key queries**. $\mathcal{B}$ decrypts C_{ID_i} to obtain the plaintext M_i by using sk_{ID_i}. $\mathcal{B}$ sets $C_{\mathsf{ID}'_i} = (C_1, C_2, M_i \hat{e}(C_1, d'_0)/\hat{e}(C_2, d'_1))$ where d'_0 and d'_1 are components of $sk_{\mathsf{ID}'_i}$. $\mathcal{B}$ returns $C_{\mathsf{ID}'_i}$ to $\mathcal{A}$.

Challenge. After some queries, $\mathcal{A}$ selects two equal length plaintexts $M_0, M_1 \in \mathcal{M}$. Given (M_0, M_1), $\mathcal{B}$ selects $d \stackrel{R}{\leftarrow} \{0,1\}$ and sets

$$C_{\mathsf{ID}^*} = (\Gamma_3, \Gamma_3^z, M_d X).$$

$\mathcal{B}$ returns C_{ID^*} to $\mathcal{A}$. Notice that if $X = \hat{e}(g,g)^{abc} = \hat{e}(g_1, g_2)^c$ then C_{ID^*} is a valid encryption of M_d. On the other hand, if X is uniform and independent in $\mathbb{G}_1$ then C_{ID^*} is independent of d in the adversary's view.

Phase 2. $\mathcal{A}$ continues to issue queries as in Phase 1, and $\mathcal{B}$ responds as before.

Solve. Finally, $\mathcal{A}$ outputs a guess $d' \in \{0,1\}$. $\mathcal{B}$ concludes its own game by outputting a guess as follows. If $d' = d$ then $\mathcal{B}$ outputs 1 meaning $X = \hat{e}(g,g)^{abc}$. Otherwise, it outputs 0 meaning $X = R$.

We claim that $\mathcal{B}$ generates the valid secret key sk_{ID_i} for ID_i. To see this, let $\tilde{u}_i = r_i - \frac{b}{\mathsf{ID}_i - \mathsf{ID}^*}$. Then we have that

$$\begin{aligned}
sk_{\mathsf{ID}_i} = (d_0, d_1) &= \left(g_2^{\frac{-z}{\mathsf{ID}_i - \mathsf{ID}^*}} (g_1^{\mathsf{ID}_i - \mathsf{ID}^*} g^z)^{r_i}, g_2^{\frac{-1}{\mathsf{ID}_i - \mathsf{ID}^*}} g^{r_i} \right) \\
&= \left(\frac{g_2^a (g_1^{\mathsf{ID}_i - \mathsf{ID}^*} g^z)^{r_i}}{(g_1^{\mathsf{ID}_i - \mathsf{ID}^*} g^z)^{\frac{b}{\mathsf{ID}_i - \mathsf{ID}^*}}}, g^{r_i - \frac{b}{\mathsf{ID}_i - \mathsf{ID}^*}} \right) \\
&= \left(g_2^a (g_1^{\mathsf{ID}_i - \mathsf{ID}^*} g^z)^{r_i - \frac{b}{\mathsf{ID}_i - \mathsf{ID}^*}}, g^{r_i - \frac{b}{\mathsf{ID}_i - \mathsf{ID}^*}} \right) \\
&= \left(g_2^a (g_1^{\mathsf{ID}_i} h)^{\tilde{u}_i}, g^{\tilde{u}_i} \right)
\end{aligned}$$

We also claim that $\mathcal{B}$ can perfectly simulate the re-encryption key for ID_i since it looks random and independent of any other values if the adversary does not obtain the corresponding IBE secret key for ID_i.

$\mathcal{B}$ fails to simulate the challenger if $\mathcal{B}$ halts in the Type-2 re-encryption key query. Otherwise $\mathcal{B}$ perfectly simulates the challenger. The maximum probability of that $\mathcal{B}$ halts is obviously upper-bounded by $\mathtt{Adv}^{eu}(\mathcal{A})$. Therefore, we conclude the theorem.

4.5 Toward the chosen ciphertext security

Green and Ateniese [GA06] proposed the semantically secure identity-based proxy re-encryption system and constructed the CCA-secure system based on

the former system, applying CHK conversion [CHK04] to it. It might be able to construct the CCA-secure system based on our proposed system by using the same technique. It is the further study.

5 Conclusion

In this study, we proposed two proxy re-encryption systems; one for the decryption right delegation from a CBE user to IBE users, and the other one for the delegation among IBE users. The former is the first "hybrid "proxy re-encryption system, and the latter has some advantage over the previously proposed identity-based systems. We introduced the security notion and proved that both our systems are semantically secure based on the dBDH assumption, in the standard model. We presented neither a hybrid system nor an identity-based system secure in the CCA sense. This is the further study.

6 Acknowledgements

We would like to thank Dan Boneh, Eu-Jin Goh and the other anonymous reviewers for giving helpful suggestions.

References

[AFGH05] G. Ateniese, K. Fu, M. Green and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage ", In *Proceedings of the 12th Annual Network and Distributed System Security Symposium - NDSS'05*, pages 83-107, 2005.

[BB04a] D. Boneh and X. Boyen, "Efficient selective-id secure identity based encryption without random oracles ", In *Advances in Cryptology - EUROCRYPT'04*, *Lecture Notes in Computer Science*, LNCS 3027, pages 223-238. Springer-Verlag, 2004.

[BB04b] D. Boneh and X. Boyen, "Secure identity based encryption without random oracles ", In *Advances in Cryptology - CRYPTO'04*, *Lecture Notes in Computer Science*, LNCS 3152, pages 443-459. Springer-Verlag, 2004.

[BBS98] M. Blaze, G. Bleumer and M. Strauss, "Divertible protocols and atomic proxy cryptography ", In *Advances in Cryptology - EUROCRYPT'98*, *Lecture Notes in Computer Science*, LNCS 1403, pages 127-144. Springer-Verlag, 1998.

[BF01] D. Boneh and M K. Franklin, "Identity-based encryption from the Weil pairing ", In *Advances in Cryptology - CRYPTO'01*, *Lecture Notes in Computer Science*, LNCS 2139, pages 213-229. Springer-Verlag, 2001.

[CH07] R. Canetti, S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption ", http://eprint.iacr.org/2007/171

[CHK03] R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme ", In *Advances in Cryptology - EUROCRYPT'03*, *Lecture Notes in Computer Science*, LNCS 2656, pages 255-271. Springer-Verlag, 2003.

[CHK04] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption ", In *Advances in Cryptology - EUROCRYPT'04*, *Lecture Notes in Computer Science*, LNCS 3027, pages 207-222. Springer-Verlag, 2004.

[DI03] Y. Dodis and A. Ivan, "Proxy cryptography revisited ", In *Proceedings of the 10th Annual Network and Distributed System Security Symposium - NDSS'03*, 2003.

[G06] C. Gentry, "Practical identity-based encryption without random oracles ", In *Advances in Cryptology - EUROCRYPT'06*, *Lecture Notes in Computer Science*, LNCS 4004, pages 445-464. Springer-Verlag, 2006.

[GMR88] S. Goldwasser, S. Micali and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks ", SIAM J. Computing, 17(2), pages 281-301, 1988.

[GA06] M. Green and G. Ateniese, "Identity-Based Proxy Re-Encryption ", http://eprint.iacr.org/2006/473

[GS04] C. Gentry and A. Silverberg, "Hierarchical id-based cryptography ", In *Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security*, pages 548-566. Springer-Verlag, 2002.

[J99] M. Jakobsson, "On quorum controlled asymmetric proxy re-encryption ", In *Proceedings of Public Key Cryptography - PKC'99*, *Lecture Notes in Computer Science*, LNCS 1560, pages 112-121. Springer-Verlag, 1999.

[M07] T. Matsuo, "Proxy Re-encryption Systems for Identity-based Encryption ", In *Proceedings of Pairing-Based Cryptography - Pairing'07*, *Lecture Notes in Computer Science*, LNCS 4575, pages 247-267. Springer-Verlag, 2007.

[MO97] M. Mambo and E. Okamoto, "Proxy cryptosystems: Delegation of the power to decrypt ciphertexts ", In *IEICE Trans. Fund. Electronics Communications and Computer Science,* , E80-A/1:54-63,1997.

[S84] A. Shamir, "Identity-based cryptosystems and signature schemes ", In *Advances in Cryptology - CRYPTO'84*, *Lecture Notes in Computer Science*, LNCS 196, pages 47-52. Springer-Verlag, 1985.

[W05] B. Waters, "Efficient identity-based encryption without random oracles ", In *Advances in Cryptology - EUROCRYPT'05*, *Lecture Notes in Computer Science*, LNCS 3494, pages 114-127. Springer-Verlag, 2005.

[ZMSR04] L. Zbou, M. A. Marsh, F. B. Schneider and A. Redz, "Distributed blinding for ElGamal re-encryption ", Technical Report 2004-1924, Cornell Computer Science Department, 2004.

A Lemma 1

Boneh and Boyen [BB04a] proved BB-IBE system being semantically secure if both components of secret key remains secret; however we can prove that the disclosure of second component of BB-IBE secret key does not make BB-IBE system weak. Here we define the security under such disclosure by the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Initialization. The adversary $\mathcal{A}$ selects a target identity ID^* and gives it to the challenger $\mathcal{C}$.

Setup. Suppose that BB-IBE system consists of four algorithms, $\mathbf{SetUp}_{\mathsf{IBE}}$, $\mathbf{KeyGen}_{\mathsf{IBE}}$, $\mathbf{Enc}_{\mathsf{IBE}}$, and $\mathbf{Dec}_{\mathsf{IBE}}$. $\mathcal{C}$ runs the $\mathbf{SetUp}_{\mathsf{IBE}}$ algorithm and generates the system parameters $parms$ and the master-secret key mk. $\mathcal{C}$ gives $parms$ to $\mathcal{A}$, keeping mk to itself.

Phase 1. $\mathcal{A}$ adaptively queries $\mathcal{C}$ as follows.

- **Secret key queries.** $\mathcal{A}$ requests the secret key for ID from $\mathcal{C}$. $\mathcal{C}$ generates the secret key sk_{ID} by running algorithm $\mathbf{KeyGen}_{\mathsf{IBE}}$ and returns it to $\mathcal{A}$ if $\mathsf{ID}_i \neq \mathsf{ID}^*$. Otherwise, $\mathcal{C}$ rejects the query.
- **Second component queries.** $\mathcal{A}$ requests the second component of the secret key for ID from $\mathcal{C}$. $\mathcal{C}$ generates the secret key sk_{ID} by running algorithm $\mathbf{KeyGen}_{\mathsf{IBE}}$ and returns the second component of sk_{ID}.

After some number of queries, $\mathcal{A}$ selects two equal length plaintexts $M_0, M_1 \in \mathcal{M}$, and sends them to $\mathcal{C}$.

Challenge. Given (M_0, M_1), $\mathcal{C}$ picks a random bit $d \in \{0,1\}$ and sets the challenge ciphertext to $C_{\mathsf{ID}^*} = \mathbf{Enc}_{\mathsf{IBE}}(\mathsf{ID}^*, parms, M_d)$, which is sent to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ continues to issue queries as in Phase 1.

Guess. Finally, $\mathcal{A}$ outputs a guess $d' \in \{0,1\}$.

The adversary $\mathcal{A}$ wins if $d' = d$. We say that the BB-IBE system is IND-sID-CPA variant secure if $|\Pr[d' = d] - 1/2|$ is negligible.

Definition 6. *We define $\mathcal{A}$'s advantage in an IND-sID-CPA variant games as follows*

$$\mathtt{Adv}^{vsid}_{\mathsf{BB-IBE}}(\mathcal{A}) = 2(\Pr[d' = d] - 1/2) \tag{5}$$

We say that the BB-IBE system is (k, t, q, ϵ) IND-sID-CPA variant secure if for any t time IND-sID-CPA variant adversary $\mathcal{A}$ that makes at most q chosen secret key queries under a security parameter k we have that $\mathtt{Adv}^{vsid}_{\mathsf{BB-IBE}}(\mathcal{A}) < \epsilon$.

Lemma 1. *Suppose that the BB-IBE system is (k, t, q, ϵ) selective-identity, adaptive chosen plaintext (IND-sID-CPA) secure, then, for any q, k, and $t' < t - \Theta(\tau q)$, BB-IBE system is (k, t', q, ϵ) IND-sID-CPA variant secure where τ is the maximum time for an exponentiation in $\mathbb{G}$.*

Proof. Let $\mathcal{A}$ be an IND-sID-CPA variant adversary. We construct an original IND-sID-CPA adversary $\mathcal{B}$ specified in Sec. 2.5 by utilizing $\mathcal{A}$. We describe how $\mathcal{B}$ works in the following.

Initialization. When $\mathcal{A}$ selects the target identity ID^*, $\mathcal{B}$ forwards it to its challenger.

Setup. On receiving public parameters $parms = (g, g_1, g_2, h)$ from the challenger, $\mathcal{B}$ forwards it to $\mathcal{A}$.

Phase 1.

- **Secret key queries.** When $\mathcal{A}$ requests the secret key for ID, $\mathcal{B}$ forwards the request to the challenger. $\mathcal{B}$ obtains the corresponding secret key and forwards it to $\mathcal{A}$.
- **Second component queries.** When $\mathcal{A}$ requests the second component of secret key for ID, $\mathcal{B}$ selects $r \xleftarrow{R} \mathbb{Z}_p$ and gives g^r to $\mathcal{A}$.

Challenge. When $\mathcal{A}$ selects (M_0, M_1), $\mathcal{B}$ forwards it to the challenger. $\mathcal{B}$ obtains the challenge ciphertext and forwards it to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ continues to issue queries as in Phase 1. $\mathcal{B}$ responds the queries as in Phase 1.

Guess. If $\mathcal{A}$ outputs a guess $d' \in \{0,1\}$, $\mathcal{B}$ outputs d'.

It is obvious that $\mathcal{B}$ simulates the challengher for $\mathcal{A}$ perfectly, and wins the game whenever $\mathcal{A}$ wins. We conclude the proof.