Non-transferable proxy re-encryption for multiple groups

Ei Mon Cho and Lwin San*

Graduate School of Science and Engineering,
Saitama University,
Saitama, Japan
Email: ei.m.c.749@ms.saitama-u.ac.jp
Email: lwinsan.07@gmail.com
*Corresponding author

Takeshi Koshiba

Faculty of Education and Integrated Arts and Sciences,
Waseda University,
Tokyo, Japan
Email: tkoshiba@waseda.jp

Abstract: In proxy re-encryption (PRE) scheme, the message is sent by a delegator to a delegatee with help of the trusted third party proxy without knowing the existing plaintext. It is known that non-transferable PRE schemes solve some problems of the naive PRE schemes. Some non-transferable PRE scheme is extended to group-based schemes, where the proxy diverts a ciphertext for a group into another group, but the existing schemes have the proxy colluding problem. To resolve the proxy colluding problem for group-based PRE schemes, we propose a non-transferable PRE scheme for multiple groups. In our scheme, there are three sub-processes, which are based on a non-transferable PRE scheme and a group signature. We show that our scheme provides the security for a delegator Alice, a delegatee Bob in the same group with Alice, and another delegatee Charlie in a different group from Alice.

Keywords: proxy re-encryption; PRE; non-transferability; group signature; multiple groups.

Reference to this paper should be made as follows: Cho, E.M., San, L. and Koshiba, T. (2018) 'Non-transferable proxy re-encryption for multiple groups', *Int. J. Space-Based and Situated Computing*, Vol. 8, No. 1, pp.20–29.

Biographical notes: Ei Mon Cho received her Master degree in Computer Science from the University of Computer Studies, Yangon in 2011. She is currently a PhD student in Saitama University. Her interests is cryptography, security, cloud computing and big data analysis.

Lwin San received his Master's of Engineering from the Saitama University. His current research interests are the proxy re-encryption, cloud computing and cryptography.

Takeshi Koshiba received his PhD degree in Computer Science from the Tokyo Institute of Technology in 2001. He was a Professor at the Saitama University until March 2017. He has been a Professor at the Faculty of Education and Integrated Arts and Sciences, Waseda University since April 2017. His current interests include theory of cryptography, quantum computation and computational complexity theory.

This paper is a revised and expanded version of a paper entitled 'Secure non-transferable proxy re-encryption for group membership and non-membership' presented at the 8th International Workshop on Trustworthy Computing and Security/TwCSec-2017, Ryerson University, Toronto, Canada, 24–26 August 2017.

1 Introduction

Proxy re-encryption (PRE) is a public key cryptosystem which allows a semi-trusted proxy to transform a ciphertext encrypted under one key into another ciphertext of the same plaintext under another key, without revealing any information of the plaintext. The concept of the proxy cryptosystem which is also called proxy decryption, is introduced by Mambo and Okamoto (1997). A proxy decryptor decrypts the encrypted ciphertext by Alice's public key on behalf of Alice. In 1998, Blaze et al. (1998) proposed the notion of PRE which is similar to the proxy cryptosystem. In PRE, the proxy converts Alice's ciphertext

to a new ciphertext for Bob without exposing the plaintext. Thus, the proxy should be key independent to avoid compromising the private keys of the sender (Alice) and the receiver (Bob).

The scheme is only useful when the mutual trust relationship exists between Alice and Bob. In the PRE scheme, a proxy with re-encryption keys can change a ciphertext for Alice (a delegator) into another ciphertext of the same plaintext for Bob (a delegatee). The proxy cannot get any information about the plaintext or the private key (Blaze et al., 1998; Canetti and Hohenberger, 2007; Guo and Hu, 2012; Phong et al., 2016).

The PRE scheme can be classified into two types: unidirectional PRE schemes and bidirectional PRE schemes. In the unidirectional PRE (Ivan and Dodis 2003), Alice can delegate to Bob without having to delegate to Alice without any secret key of Bob. It only requires the secret key of Alice and the public key of Bob to discover a re-encryption key from Alice to Bob. It does not allow Bob to Alice (the reverse direction).

In 2010, Matsuda et al. (2010) proposed a bidirectional PRE scheme without bilinear maps. The re-encryption key to converting ciphertext from Alice to Bob can be also used to translate from Bob to Alice. Canetti and Hohenberger (Canetti and Hohenberger, 2007) proposed the CCA (chosen ciphertext attack) secure bidirectional multi-hop PRE scheme in the standard model by using the bilinear maps. A bidirectional single-hop PRE scheme without bilinear maps is considered by Deng et al. in 2008 and then it is also CCA secure in random oracle model.

Bidirectional PRE schemes have attracted much attention from the cryptography community (Ateniese et al., 2006; Blaze et al., 1998; Guo et al., 2015; Ivan and Dodis, 2003; Libert and Vergnaud, 2008; Niu et al., 2009; Wang et al., 2010) because they have many interesting and useful applications, such as the e-mail forwarding, the encrypted files distribution, the digital rights management (Nuez et al., 2015) and the cloud data sharing (Fan and Liu, 2016).

In 2007, Matsuo (2007) proposed two Identity-based Re-encryption schemes where one is the transformation from ciphertexts encrypted based on a traditional certificate-based public key into the ciphertexts (CBE-IBE) and the other one is the transformation from ciphertexts encrypted in IBE manner into the different ciphertexts (IBE-IBE). However, Wang et al. (2010) gave two types of attacks, which show that the identity based PRE schemes (CBE-IBE and IBE-IBE) are insecure.

There is another type of PRE scheme which is called group-based PRE. Group communication becomes popular in many applications. Generally speaking, two groups are supposed: a sender group and a receiver group. Any member from the sender group can encrypt a message and send to the designated receiver group. Any member from the designated receiver group can decrypt the ciphertext. However, there have some problems such as PRE or forwarding message. For example, by dividing tasks among a group A and another group B, an encrypted message from the group A should be allowed to decrypt by the group B. Such kind of scenarios is supported by the group-based PRE scheme, which was first proposed by Ma and Ao (2009). Their scheme is bidirectional proxy re-encryption scheme. In their scheme, a message is sent from the group A to the group B, any member from the group B can decrypt the ciphertext. And the proxy allows the reverse direction. Because re-encryption key is generated by using the private key of the group A and the group B.

In 2006, Ateniese (2006) proposed a PRE scheme which supports unidirectional PRE and uses the delegator's private key for protecting the collusion of a proxy and a delegatee. However, this scheme is lacking the non-transferable property. This problem was first addressed by Libert in 2008. The proxy and delegatee are quite difficult to protect from colluding. Libert et al. solved the problem instead of preventing the collision of proxy and delegatee, by using traceable PRE. However, it cannot prevent the re-delegation of the proxy. Wang et al. (2010) proposed 'identity-based PRE scheme' to solve the problem of proxy colluding, in 2010. Their major advantages are that the proxy and the delegates cannot delegate the decryption right to the others without the permission of the public key generator (PKG). However, PKG in their scheme can decrypt both of the original ciphertext and the re-encrypted ciphertext. This means that the transferable problem still cannot be solved.

In 2015, Wang at al. (2015) proposed a new scheme for protecting critical information systems which is based Cramer-Shoup encryption scheme. But, it can not achieve the delegator's IND-CCA security for the proxy and the delegatee.

All above schemes do not provide non-transferable property or collusion property for the proxy re-encryption scheme. For example, the proxy or Bob can collude to get the decryption key to anyone. In addition, the re-encryption key for the proxy is generated by the trusted private key generator (PKG). However, this kind of schemes has the key escrow problem which PKG is a malicious and PKG can decrypt the original ciphertext or re-encrypted ciphertext. Furthermore, PKG can generate many re-encryption keys for adversary without accessing any right from the Alice. This problem is called PKG despotism problem. In 2012, He et al. (2012) proposed the non-transferable PRE scheme which is suitable for the key escrow problem and the PKG despotism problem. Their scheme is based on the certificateless cryptography.

1.1 Motivation

PRE has many practical applications such as email forwarding, health care cloud system and so on. For example, health care cloud system is a cloud computing service using for storing, maintaining and backing up personal health information of the patient. This system acts as a third party between physicians and patients. Therefore, this system needs to secure for patient's health records and their biometric data. By using PRE techniques, the health care cloud system can be secured as in Wang et al. (2017).

Furthermore, Group discussion of active learning system (ALS) (Yamamoto, 2016) can improve by using PRE.

In particular, our PRE can be used in secure file sharing system. In distributed file sharing system, the third party is difficult to be trusted from the confidential point of view. Therefore, the distributed file users are desired to apply the encryption methods for the confidentiality. The proxy can distribute the encrypted file without using the information of original data. With group or non-group members, our scheme supports to decrypt by using the authority of the sender. When a receiver accesses to the proxy to request forwarding the ciphertext, the proxy re-encrypt the message without learning any information from the original ciphertext or key.

1.2 Comparison with previous works

We compare some existing PRE schemes with our proposed scheme as the following properties (as shown in Table 1). Our scheme provides the non-transferable property for both the members in the same group or the outsider. Here, we enumerate several properties in PRE schemes.

Table 1 Comparison with previous work

Property	*He et al.*	*Ma et al.*	*Ours*
Unidirectional	Yes	No	Yes
Bidirectional	No	Yes	No
Proxy-Invisible	Yes	No	Yes
Collusion-resistance	Yes	No	Yes
Non-Transferable	Yes	No	Yes
Non-Transitive	Yes	Yes	Yes
Multi types of groups	No	No	Yes

- Unidirectional: 'delegator Alice to delegatee Bob' – does not allow reverse re-encryption 'Bob to Alice'.
- Bidirectional: the re-encryption scheme is reversible. The re-encryption key can be used to translate message from proxy to delegatee, as well as from delegateee to the proxy.
- Proxy-invisible: although the delegator needs to know the existence of proxy for re-encryption step, delegatee does not need to know that. We have to determine the proxy is invisible for delegator or invisible for delegatee.
- Collusion-resistance: the delegatee and proxy's collusion can receive the output from delegator. But they cannot recover the secrete key of the delegator. •
- Non-transferable: even though the proxy and the delegatee Bob devise re-encryption for other person, the re-encryption key of $rk_{A \to C}$ is impossible to produce without the permission.
- Non-transitive: the proxy cannot generate the reencryption key $rk_{A \to C}$ based on the re-encryption key of Alice to Bob ($rk_{A \to B}$) and re-encryption key of Bob to Charlie ($rk_{B \to C}$).
- Multi types of groups: the proxy allows to participate any kind of delegatee wich belongs to the delegator's group or outside of the delegator's group. For example, Alice is a delegator from a group A, Bob is a delegatee from the group A and Charlie is from another group B. In such kind of situation, PRE scheme provides to reencrypt whether delegator and delegatee are group members or outsiders.

1.3 Our contribution

Our new PRE scheme takes the background idea of non-transferable PRE scheme (He et al., 2012) to support the non-transferable property. We suppose that there are three kinds of participants: a delegator *i* (Alice), a delegatee *j* (Bob) who is the same group with Alice and another delegatee *j* (Charlie) who is from the outside of the group. Therefore, we need to think about the non-transferable property for both the same group or the outsider. For the communication between Alice and Bob, we borrow the idea of Ma and Ao (2009) for group signature properties. Alice does not need to send her certificate to the proxy or Bob. By using delegatee ID from Alice's and Bob's public keys, the proxy can generate the re-encryption key. At the point of the communication of Alice to Charlie, Alice has to send her signature (certificate) of the designated ciphertext to Charlie. When Charlie sends his public key and the proof of Alice signature, the proxy can generate the re-encryption key (as shown in Figure 1). The characteristics of our proposed scheme are as follow:

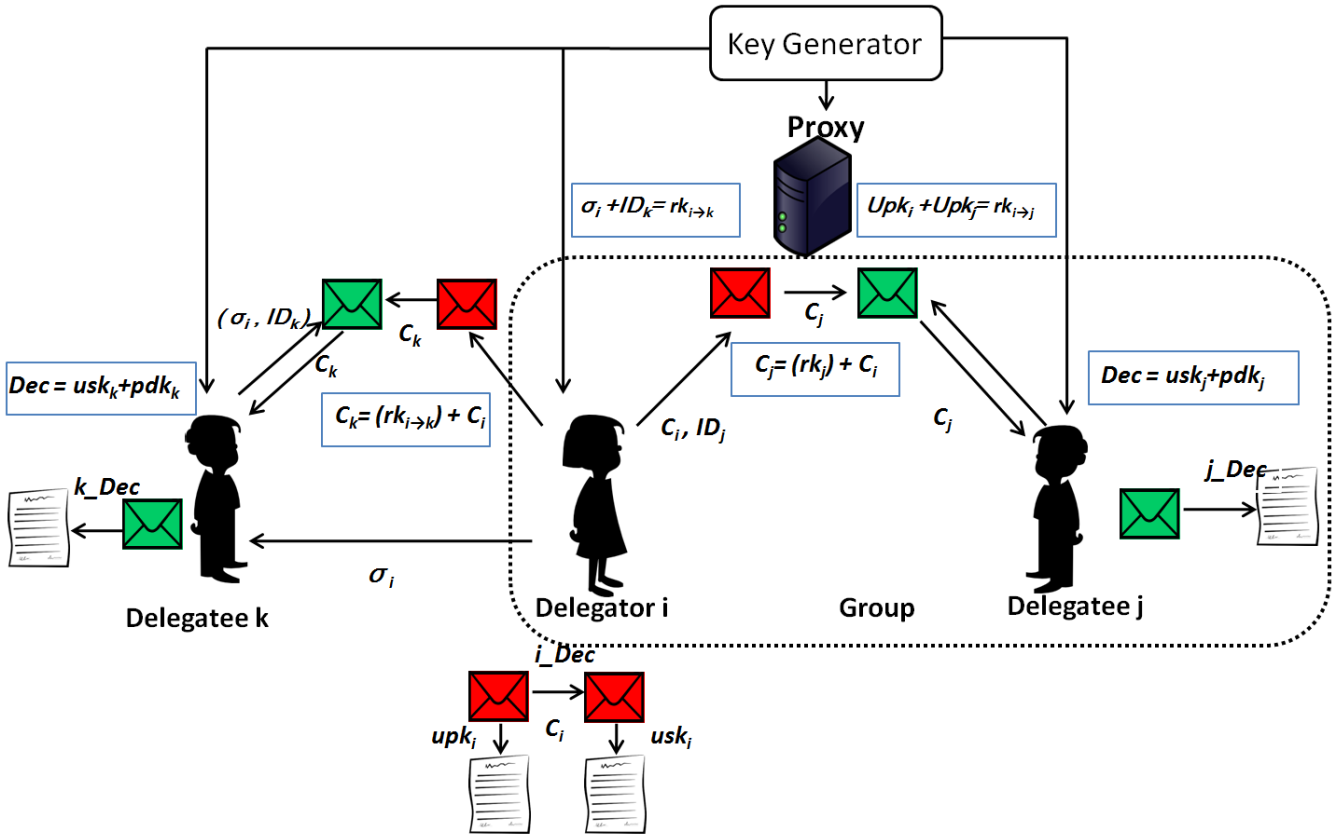
- Any kind of delegatee/delegator can participate in our PRE scheme. We support both group or non-group members of the delegator.
- No one can decrypt the original ciphertext without getting any permission from the delegator.
- Proxy cannot re-encrypt the ciphertexts without knowing the public key of any delegatee.

We organise the paper as follows. In Section 2, we prepare preliminaries. Section 3 provides the construction of our scheme. Section 4 shows the security proof of our scheme. In Section 5, we give a conclusion.

2 Preliminaries

In this section, we will present some primitives that will be used in our scheme.

Figure 1 Our PRE scheme (see online version for colours)



2.1 Bilinear map

Definition 3.1. Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative cyclic groups of prime order p, and g be generator of $\mathbb{G}$. We say that $\mathbb{G}_T$ has an admissible bilinear map e: $\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$, , if the following conditions hold.

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

2.2 Assumptions

Definition 3.2 (computational Diffie-Hellman assumption). Given g^a and g^b for some $a, b \in Z_q^*$, compute $g^{ab} \in \mathbb{G}_1$. A $(\mathcal{T}, \varepsilon)$-CDH attacker in $\mathbb{G}_1$ is a probabilistic machine Ω running time $\mathcal{T}$ such that

$$Succ_{cdh}^{\mathbb{G}_1}(\Omega) = \Pr\left[\Omega\left(g, g^a, g^b\right) = g^{ab}\right] \geq \varepsilon$$

where the probability is taken over the random values a and b. The CDH problem is $(\mathcal{T}, \varepsilon)$ intractable if there is no $(\mathcal{T}, \varepsilon)$-attacker in $\mathbb{G}_1$. CDH assumption states that it is the case for polynomial t and non-negligible e.

Definition 3.3 (transacted decision augmented bilinear Diffie-Hellman exponent assumption). The security of our proposed scheme is based on a complexity assumption named Truncated q-ABDHE (He et al., 2012) which has been proposed by (Libert et al., 2008). Let $e:(\mathbb{G}\times\mathbb{G})\to\mathbb{G}_T$ be a bilinear map, where $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of large prime order p. Given a vector of $q + 3$ elements:

$$\left(g', g'^{(\alpha^{q+2})}, g, g^{\alpha}, \ldots, g^{(\alpha^q)}\right) \in \mathbb{G}^{q+3}$$

and an element $Z \in \mathbb{G}_T$ as input, output 0 if $Z = e\left(g^{\alpha^{q+2}}, g'\right)$ and output 1 otherwise. And algorithm B has advantage ε in solving truncated q-ABDHE if:

$$\left|\Pr\left[B\left(g', g'^{(\alpha^{q+2})}, g, g^{\alpha}, \ldots, g^{(\alpha^q)}, e\left(g^{(\alpha^{q+1})}, g'\right)\right) = 0\right]\right.$$
$$\left. - \Pr\left[B\left(g', g'^{(\alpha^{q+2})}, g, g^{\alpha}, \ldots, g^{(\alpha^q)}, Z\right) = 0\right]\right| \geq \varepsilon$$

where the probability is over the random choice of generators g, g' in $\mathbb{G}$, the random choice of α in $\mathbb{Z}_p$, the random choice of $Z \in \mathbb{G}_T$ and the random its consumed by B.

2.3 Syntax of our PRE scheme

Our proposed scheme belongs to 13 algorithms. 'Delegator i' (Alice) owns a message, 'Delegatee j' (Bob) is the same group with 'Delegator i' (Alice) and 'Delegatee k' (Charlie) is the receiver of the message without same group of 'Delegator i' (Alice).

- Setup: from the input of the security parameter 1^k, the public parameters *mpk* and the master secret key *msk* are generated.
- Key generation:
  1 Set-secret-value. The algorithm generates a secret value which is only known to the user himself.
  2 Partial-private-key. On input a user's identity *ID* and *msk*, the algorithm generates a partial private key for the user.
  3 Set-private-key. On input the partial private key and the secret value, the algorithm outputs the whole private key for the user.
  4 Set-public-key. On input a user's identity *ID* and a secret value, the algorithm generates a public key.
- Private key correctness check: the algorithm checks the correctness of the private key.
- Encryption: the encryption algorithm takes a public key upk_A of the delegator Alice and a message *m* as input, outputs a ciphertext C_A encrypted under upk_A.
- Alice-decryption (delegator i): the decryption algorithm takes a private key usk_i of the delegator *i* and a ciphertext C_i as input, outputs the message *m*.
- Bob-re-encryption key generation: the algorithm verifies the delegator $j's$ signature and the public key. The re-encryption key generation algorithm outputs a re-encryption key rk_j and other relational values.
- Bob-partial-decryption-key generation: the algorithm checks the correctness of the reencryption key, and generates a partial decryption key.
- Bob-re-encryption: the re-encryption algorithm takes re-encryption key rk_j and ciphertext C_i as input, outputs a re-encrypted ciphertext C_j under upk_j.
- Bob-decryption (delegatee *j*): the decryption algorithm takes the private key usk_j of delegatee *j*, the partial decryption key and the ciphertext C_i as input, outputs message *m*.
- Charlie-re-encryption key generation: the algorithm verifies the delegator $k's$ ID and the public key. The re-encryption key generation algorithm outputs a re-encryption key rk_k and other relational values.
- Charlie-partial-decryption-key generation: the algorithm checks the correctness of the reencryption key, and generates a partial decryption key.
- Charlie-re-encryption: the re-encryption algorithm takes re-encryption key rk_k and ciphertext C_i as input, outputs a re-encrypted ciphertext C_k under upk_k.
- Charlie-decryption (delegatee *k*): the decryption algorithm takes private key usk_k of delegatee *k*, the partial decryption key and ciphertext C_i as input, outputs the message *m*.

2.4 Security model

The security against *IND-ID-CCA* is required in many applications. To achieve it, we used the technique of He et al. (2012) with a little modification.

Definition 3.4 (IND-ID-CCA secure). We consider the following game between an adversary $\underline{A}$ and a challenger *C*. Chosen ciphertext security for PRE systems is defined via the following game between an adversary *A* and a challenger *C*.

- Phase 1: *C* generates the public parameter and sends it to *A*. *A* generates queries $q_i, \ldots, q_m$, with query q_i being one of the following:

  Public key extraction / $(\text{pkextract}, ID_i)$: for user ID_i.
  Encryption / $(\text{encrypt}, ID_i, m_j)$.
  Partial Private KeyExtraction / $(\text{pskextract}, ID_i)$.
  Re-encryption KeyExtraction / $(\text{rkextract}, ID_i, IDi', \sigma_i)$: or delegator ID_i and delegatee ID_i' by using signature of ID_i.
  Decryption / $(\text{decrypt}, ID_i, c_i)$.
  Re-encryption / $(\text{re-encrypt}, ID_i, ID_{i'}, c_i, \sigma_i)$: by using signature of ID_i.

- Challenge: *A* submits two plaintexts $M_0, M_i, \in M$ and an identity (ID_j, ID_j'). C selects a random bit $b \in \{0, 1\}$, sets $C = Encrypt(params, ID_j, M_b)$, and sends *C* to the adversary as its challenge ciphertext.
- Phase 2: *A* is restricted from the following queries.

  $(Encrypt, ID_j, M_0)$ *and* $(Encrypt, ID_j', M_1)$.
  $(Decrypt, ID_j, C_j)$.
  Any pair of queries $(reextract, ID_j, ID_j')$ *and* $(decrypt, ID_j', c_j')$ *where cj is the re-encrypted ciphertext using* $rk_{j \to j'}$.
  Guess : *A outputs* $b' \in \{0,1\}$ *as a guess of b.*
  The advantage ofadversary A can be defined as
  $Adv_A^{cpa}(\lambda) = \left|\Pr[b' = b] - 1/2\right|$

The PRE scheme is *IND-ID-CCA* secure for the adversary *A* if this advantage is negligible.

3 Construction of the scheme

We now construct our PRE scheme which is based on He et al.'s (2012) PRE scheme using different type of setting for different kind of delegatee. There are a delegator *i* (Alice) who owns a message, a delegatee *j* (Bob) who is the same group with the delegator *i* (Alice) and a delegatee *k* (Charlie) who is from the different group of delegator *i* (Alice).

3.1 Setup

We assume $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order *p* such that *p* is an *n*-bit prime, and $e: \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be the bilinear map. $H_1: \{0,1\}^* \to \mathbb{Z}_p$, $H': \mathbb{G}_T \to \mathbb{Z}_p$ are secure hash functions. The PKG selects four random generators $h_1, h_2, h_3, g \in \mathbb{G}$ and randomly chooses $\alpha \in \mathbb{Z}_p$. It sets $g_1 = g^{\alpha}$. Define the message $M \in \mathbb{G}_T$.. The public parameters *mpk* and master secret key *msk* are given by *mpk* = (g, g_1, h_1, h_2, h_3, H_1, H, H', M) and *msk* = α.

3.2 Key generation

On input the public key/master secret key pair (*mpk*, *msk*) and an identity $ID_i \in \{0, 1\}^n$ of entity *i*, the PKG computes $id_i = H_1(ID_i)$. If $id_i = \alpha$, it aborts. Otherwise, the protocol proceeds as follow:

- Set-secret-value. Entity *i* selects $r_i \in \mathbb{Z}_p$ at random. r_i is $i's$ secret value.
- Partial-Private-Key-Extract.
  1. A sends $R = h_1^{r_i}$ to PKG, and gives PKG the following zero-knowledge proof of knowledge: $PK\{r_i : R = h_1^{r_i}\}$
  2. PKG randomly selects $r_i', r_{i,2}, r_{i,3} \in \mathbb{Z}_p$ and computes $h_i' = (Rg^{r_i'})^{1/(\alpha - id_i)}$, $h_{i,2} = (h_2 g^{-r_{i,2}})^{1/(\alpha - id_i)}$, $h_{i,3} = (h_3 g^{-r_{i,3}})^{1/(\alpha - id_i)}$ and sends $i's$ partial private key $r_i', h_i', r_{i,2}, h_{i,2}, r_{i,3}, h_{i,3}$ to *i*.
- Set-Private-Key. *i* computes $r_{i,1} = r_i' / r_i$, $h_{i,1} = (h_i')^{1/r_i} = (h_1 g^{-r_{i,1}})^{1/(\alpha - id_i)}$. Then, $i's$ private key can be denoted as $usk_i = (r_i, r_{i,1}, h_{i,1}, r_{i,2}, h_{i,2}, r_{i,3}, h_{i,3})$. Similarly, the delegatee $j's$ private key is denoted as $usk_j = (r_j, r_{j,1}, h_{j,1}, r_{j,2}, h_{j,2}, r_{j,3}, h_{j,3})$.
- Set-Public-Key. A publishes her public key $upk_i = (P_{i,1}, P_{i,2})$, where $p_{i,1} = g_1^{r_i}$, and $p_{i,2} = g^{r_i id_i}$. Anyone can verify the validity of upk_i by checking if the equalities $e(g^{id_i}, p_{i,1}) = e(g_1, p_{i,2})$, and $e(h_1^{r_i}, g_1) = e(h_1, p_{i,1})$ hold (i.e., $h_1^{r_i}$ can be obtained from PKG).

3.3 Private key correctness check

On input (*mpk*, usk_{ID}) and an identity $ID \in \{0, 1\}^n$, *i* computes $id_i = H_1(ID_i)$ and checks whether $e(h_{i,t}, g_1 / g^{id_i}) = e(h_t g^{-ri,t}, g)$ for t = 1, 2, 3. If correct, output 1. Otherwise, output 0.

3.4 Encryption

[**Enc**(m, upk_i) → *Ciphertext* C_i]

To encrypt a message $m \in \mathbb{G}_T$ using public key, the sender checks whether the equalities $e(g^{id_i}, p_{i,1}) = e(g_1, p_{i,2})$ and $e(h_1^{r_i}, g_1) = e(h_1, p_{i,1})$ hold. If not, output $\perp$ and abort encryption. Otherwise, the sender generates a unique randomly-selected secret parameter $s \in Z_p$, and computes $id_i = H_1(ID_i)$. Finally, sender outputs the ciphertext C where $C = (C_1, C_2, C_3, C_4, C_5, C_6) = (p_{A,1}{}^s p_{A,2}{}^{-s}, e(g, g)^s, m.e(g, h_1)^{-s}, e(g, g)^{H'(m)}, g^{s\beta + H'(m)}, e(g, h_3^{s\beta}))$. We set $\beta = H(C_1, C_2, C_3, C_4)$.

3.5 Alice-descryption (delegator i)

[**Dec**(usk_i C_i) → *Message* m]

To decrypt a ciphertext $C = (C_1, C_2, C_3, C_4, C_5, C_6)$ using secret key usk_i, delegator Alice computes $\beta = H(C_1, C_2, C_3, C_4)$ and tests whether $e(C_5, g) = C_2^{\beta} C_4$ and $C_6 = e(C_1, h_{i,2} h_{i,3}^{\beta})^{\frac{1}{r_i}} \cdot C_2^{r_{i,2} + r_{i,3}B}$. If it is not equal, outputs $\perp$. Else computes $m = C_3 \cdot e(C_1, h_{i,1})^{\frac{1}{r_i}} \cdot C_2^{r_{i,1}}$. If $e(g, g)^{H'(m)} = C_4$ holds, return *m*, otherwise return $\perp$.

3.6 Bob-re-encryption key generation (delegate-j)

[**ReKeyGen**(ID_j, upk_j) → rk_j]

- Delegatee *j* (Bob) belongs to the same group of delegator *i* (Alice). Delegatee *j* (Bob) is only allowed to decrypt the messages intended for delegator *i* (Alice) during some specific time period *t*. To achieve this property, the delegator *i* (Alice) generates a random value at $a_t \in \mathbb{Z}_p$ for each time period *t*, where $t > 1$. at will be invalid after the period *t*. Delegatee *j* (Bob) sign the message and send ID_j and upk_j to the PKG secure channel. Delegator Sign:

  Choose $z \in \mathbb{Z}_p$ and compute $U = g^z$

  Compute $V = H_1(ID_j, U)$

  Compute $W = g^{\alpha r_{i+V}}$

  The signature of j is $\sigma = (U, W)$

- PKG verifies the signature of delegatee *j* (Bob) to identify whether delegatee *j* (Bob) is is from the same group of delegator *i* (Alice) or not. PKG Verify:

Compute $V = H_I\left(ID_j, U\right)$.

Accept the signature if

$e(h_1, W) = e(h_1^{r_i}, g^{\alpha})e(h_1, g)^V$.

If verification is success, PKG generates a unique randomly-selected secret parameter $y \in \mathbb{Z}_p$ and computes re-encryption key $rk_{i\to j} = (\frac{\alpha - id_j}{\alpha - id_i} + a_t y) \bmod p$, $i_1 = (h_1^{r_i} g_i^{-r'})^y$, $j_1 = (h_1^{rj} g^{-r_j'})^{\frac{a_t y}{(\alpha - id_j)}}$, $j_2 = h_1^{a+y}$ and sends $rk_{i\to j}$, i_1, j_1, j_2 to delegator *i*.

3.7 Bob-partial-decryption-key generation

[**PartialDecKeyGen** *check*(rk_j) → pdk_j]

- Delegatee *j* (Bob) sends h_j' to delegator *i* (Alice) via a secure and authenticated channel.
- Delegator *i* (Alice) checks whether $e(h_1, j_1) = e(j_2, h' - j)$ to ensure j_1 is a valid value which will help delegatee for decryption later. If correct, output 1, otherwise, output 0.
- Delegator *i* (Alice) checks whether $h_i'^{(id_i - id_j)} \cdot i_1^{a_t} \cdot (h_1^{r_i} g^{-r_i'}) = (h_1^{r_i} g^{-r_i'})^{rk_{i\to j}}$ to ensure that $rk_{i\to j}$ is a re-encryption key generated properly for delegation from her to delegator *j* (Bob).
- Delegator *i* (Alice) sends the re-encryption key $rk_{i\to j}$ (Bob) to Proxy via an authenticated channel.
- Delegator *i* (Alice) computes $h_k'^{\frac{1}{r_i}}$ and $B_1^{\frac{1}{r_k}}$ and sends them to delegator *j* (Bob) as partial decryption key.

3.8 Bob-re-encryption (delegator j)

[**ReEnc**(rk_j, C_i, upk_j) → Ciphertext C_j]

Proxy computes $\beta = H(C_1, C_2, C_3, C_4)$ and tests whether $e(C_5, g) = C_2^{\beta} C_4$. If it is not equal, output ⊥. Else computes $C' = C_1^{rk_{i\to j}} = g^{r_i s(\alpha - id_i)(\frac{\alpha - id_j}{\alpha - id_j} + a_t y)}$ and sends (C_1', C_2, C_3, C_4, C_5) to delegatee *j* (Bob).

3.9 Bob-decryption (delegatee j)

[**Dec**(usk_j, pdk_j, C_j) → Message *m*]

Delegatee *j* (Bob) computes $\beta = H(C_1, C_2, C_3, C_4)$ and tests whether $e(C_5, g) = C_2^{\beta} C_4$. If it is not equal, output ⊥. Else delegatee *j* (Bob) computes

$$
\begin{aligned}
& C_3 \frac{e\left(C_2', h_j^{(1/r_i),(1/r_j)}\right) C_2^{r_{j,1}}}{e\left(C_1, j_1^{(1/r_i),(1/r_j)}\right)} \\
&= C_3 \left[\frac{e\left(g^{r_i s(\alpha - id_i)\left(\frac{\alpha - id_j}{\alpha - id_i}\right)}, \left(h_1 g^{-r_{j,1}}\right)^{\frac{1}{(\alpha - id_i)^{r_i}}} \right) \left(e(g,g)^s\right)_{j,1}^{r}}{e\left(g^{r_i s(\alpha - id_i), \left(h_1 g^{-r_{j,1}}\right)^{\frac{a_t y}{(\alpha - id_j)^{r_i}}}} \right)} \right] \\
&= C_3 e\left(g^{s(\alpha - id_i)\left(\frac{\alpha - id_j}{\alpha - id_i}\right)}, \left(h_1 g^{-r_{j,1}}\right)^{\frac{1}{(\alpha - id_j)}} \right) e(g,g)^{s * r_{j,1}} \\
&= m \cdot e\left(g, h_1\right)^{-s} e\left(g^{g(\alpha - id_j)}, \left(h_1 g^{-r_{j,1}}\right)^{\frac{1}{(\alpha - id_j)}} \right) e(g,g)^{s r_{j,1}} \\
&= m
\end{aligned}
$$

If $e(g, g)^{H'(m)} = C_4$ holds, return *m*; otherwise return ⊥.

3.10 Charlie-re-encryption key generation (delegatee k)

[**ReKeyGen**(ID_k, upk_k) → rk_k]

- Delegatee *k* (Charlie) is only allowed to decrypt messages intended for delegator *i* (Alice) during some specific time period *t*. To achieve this property, the delegator *i* (Alice) generates a random value $a_i \in \mathbb{Z}_p$ for each time period *t*, where $i \geq 1$. a_i will be invalid after the period *t*. Delegator $_i$ (Alice) signs delegatee's *k* (Charlie) identity ID_k, and sends the signature σ, ID_k, a_i to PKG via a secure channel. Delegator Sign:

  Choose $z \in \mathbb{Z}_p$ and compute $U = g^z$.

  Compute $V = H_I\left(ID_k, U\right)$.

- PKG verifies the signature of delegatee *k* (Charlie) is from the outside of delegator *i* (Alice)'s group. PKG verify:

  Compute $V = H_I\left(ID_k, U\right)$.

  Accept the signature σ if $e(h_1, W) = e(h_1^{r_i}, g^{\alpha})e(h_1, g)^V$.

- If verification passes, PKG generates a unique randomly-selected secret parameter $y \in \mathbb{Z}_p$, and computes re-encryption key $rk_{i\to k} = (\frac{\alpha - id_k}{\alpha - id_i} + a_t y) \bmod p$, $i_1 = (h_1^{r_i} g_i^{-r'})^y$, $k_1 = (h_1^{r_k} g^{-r_k'})^{\frac{a_t y}{(\alpha - id_k)}}$, $k_2 = h_1^{a+y}$ and sends $rk_i \to k_1, k_2$ to delegator *i*.

3.11 Charlie-partial-decryption-key generation (delegatee k).

[**PartialDecKeyGen** *check*(rk_k) → pdk_k]

- Delegatee *k* (Charlie) sends h_k' to delegator *i* (Alice) via a secure and authenticated channel.

- Delegator *i* (Alice) checks whether $e(h_1, k_1) = e(k_2, h' - k)$ to ensure k_1 is a valid value which will help delegatee for decryption later. If correct, output 1, otherwise, output 0.
- Delegator *i* (Alice) checks whether $h_i'^{(id_i - id_k)} \cdot i_1^{a_t} \cdot (h_1^{r_i} g^{-r_i'})^{rk_{i\to k}}$ to ensure that $rk_{i\to k}$ is a re-encryption key generated properly for delegation from her to delegatee *k*.
- Delegator *i* (Alice) sends the re-encryption key $rk_{i\to k}$ to Proxy via an authenticated channel.
- Delegator *i* (Alice) computes $h_k'^{\frac{1}{r_i}}$ and $B_1^{\frac{1}{r_k}}$ and sends them to delegatee *k* (Alice) as partial decryption key.

3.12 Charlie-re-encryption (delegatee k)

[**ReEnc**(rk_k, C_i, $uprk_k$) → Ciphertext C_k]

Proxy computes $\beta = H(C_1, C_2, C_3, C_4)$ and tests whether $e(C_5, g) = C_2^{\beta} C_4$. If it is not equal, output ⊥. Else computes $C' = C_1^{rk_{i\to k}} = g^{r_i s(\alpha - id_i)(\frac{\alpha - id_k}{\alpha - id_i} + a_t y)}$ and sends (C_1', C_2, C_3, C_4, C_5) to delegatee *k*.

3.13 Charlie-decryption (delegatee k)

[**Dec**(usk_k, pdk_k, C_k) → Message *m*]

Delegatee *k* (Charlie) computes $\beta = H(C_1, C_2, C_3, C_4)$ and tests whether $e(C_5, g) = C_2^{\beta} C_4$. If it is not equal, output ⊥. Else delegatee *k* (Charlie) computes

$$C_3 \frac{e\left(C_2', h_k'^{(1/r_i),(1/r_k)}\right) C_2^{r_{k,1}}}{e\left(C_1, j_1^{(1/r_i),(1/r_j)}\right)}$$

$$= C_3 \left[\frac{e\left(g^{r_i s(\alpha - id_i)\left(\frac{\alpha - id_k}{\alpha - id_i}\right)}, \left(h_1 g^{-r_{k,1}}\right)^{\frac{1}{(\alpha - id_k)^{r_i}}} \right) \left(e(g,g)^s\right)^r_{k,1}}{e\left(g^{r_i s(\alpha - id_i)}, \left(h_1 g^{-r_{j,1}}\right)^{\frac{a_t y}{(\alpha - id_{k+})^{r_i}}} \right)} \right]$$

$$= C_3 e\left(g^{s(\alpha - id_i)\left(\frac{\alpha - id_k}{\alpha - id_i}\right)}, \left(h_1 g^{-r_{k,1}}\right)^{\frac{1}{(\alpha - id_k)}} \right) e(g,g)^{s * r_{k,1}}$$

$$= m \cdot e\left(g, h_1\right)^{-s} e\left(g^{g(\alpha - id_k)}, \left(h_1 g^{-r_{j,1}}\right)^{\frac{1}{(\alpha - id_k)}} \right) e(g,g)^{s r_{k,1}}$$

$$= m$$

If $e(g,g)^{H'(m)} = C_4$ holds, return m; otherwise return ⊥.

According to the above steps, we conclude that the proposed scheme 'non-transferable PRE' with group or non-group membership.

4 Security

As mentioned in Section 2, the security model under *IND-ID-CCA* can be proved by the following.

Theorem 1: Let the truncated decision (t, ε, q)- ABDHE assumption holds for (G, G_T, e). Then, the above non-transferable re-encryption scheme is *IND-ID-CCA* secure even when the proxy and the delegatees *k* [non-group member] are colluding.

Proof: We assume *A* be a polynomial time adversary who breaks the *IND-ID-CCA*. There is an algorithm *B* solves the truncated decision *q*-ABDHE problem as follows. *B* inputs $(g', g_{q+2}, g, g_1, \ldots, g_q, Z)$ where *Z* is a random elements.

Setup: *B* generates three random polynomials $f_1(x) \in Z_p[x]$, $f_2(x) \in Z_p[x]$, $f_3(x) \in Z_p[x]$. It sets $f_1(x) \in Z_p[x]$, $f_2(x) \in Z_p[x]$, $f_3(x) \in Z_p[x]$. It sets $g^{f_1(\alpha)}$, $h_2 = g^{f_2(\alpha)}$ and $h_3 = g^{f_3(\alpha)}$ by computing them from (g, g_1, …, g_q). It sends the public parameters (g, g_1, h_1, h_2, h_3) to *A*. *B* has a list *L* to store the entry $\langle ID_Q, ph_Q, psh_Q, sh_Q \rangle$ of every user.

Phase 1: *A* issues the following queries:

- (pkextract, ID_Q): public key extraction for user ID_Q
- (encrypt, ID_Q, m_Q): encryption of plaintext for user ID_Q
- (pskextract, ID_Q): partial private key extraction for user ID_Q
- (rkextract, ID_Q, $ID_{Q'}$, σ_Q): re-encryption key extraction for delegator ID_Q and delegatee ID_Q, by using signature of ID_Q.
- (decrypt, ID_Q, c_Q): decryption of ciphertext for ID_Q
- (re-encrypt, ID_Q, $ID_{Q'}$, c_Q, is σ_Q): re-encryption of ciphertext for ID_Q to $ID_{Q'}$, by using using signature of ID_Q.

Note: *A* cannot issue private key extraction queries PKG only knows the partial private key.

- (pkextract, ID_Q): if $ID_Q = \alpha$, *B* uses α to solve the truncated decision *q*-ABDHE. Otherwise, let $F_{Q,2}(x) = (f_2(x) - f_2(ID_Q)) / (x - ID_Q)$ and $F_{Q,3}(x) = (f_3(x) - f_3(ID_Q)) / (x - ID_Q)$ be two (q – 1)-degree polynomials. *B* sets the partial private key for ID_Q to be $(r_Q', h_Q', r_{Q,2}, h_{Q,2}, r_{Q,3}, h_{Q,3})$ which is $(f_1(ID_Q), (Rg^{-r_Q'})^{1/(\alpha - ID_Q)}, f_2(ID_Q), g^{F_{Q,2}(\alpha)}, f_3(ID_Q), g^{F_{Q,3}(\alpha)})$ respectively. For $i = 2, 3$. $g^{F_{Q,i}(\alpha)} = g^{(f_i(\alpha) - f_i(ID_Q))/(\alpha - ID_Q)} = (h_i g^{-r_{Q,i}})^{1/(\alpha - ID_Q)}$. Next, *B* computes $r_{Q,1} = r_Q' / r_Q$ and $h_{Q,1} = (h_Q')^{1/r_Q}$ to complete

the private key for ID_Q. The private key for ID_Q thus becomes (r_Q, $r_{Q,1}$, $h_{Q,1}$, $r_{Q,2}$, $h_{Q,2}$, $r_{Q,3}$, $h_{Q,3}$). Note that this is a valid private key for ID_Q since for i = 1, 2, 3, $h_{Q,i} = (h_i, g^{-rQ,hi})^{(1/\alpha - ID_Q)}$ as required. B then computes the public key for ID_Q as $(g_1^{rQ}, (g^{rQ})^{ID_Q})$, stores all these information into L and returns the public key to A.

- (encrypt, ID_Q, m_Q): if $ID_Q = \alpha$, B uses α to solve the truncated decision q-ABDHE. If ID_Q is in L, B simply extracts the public key in the corresponding entry. Otherwise, B generates the public key, partial private key and private key for ID_Q as in the above, stores them into L, encrypts m_Q by performing the usual encryption algorithm with the public key concerned and returns the ciphertext of m_Q to A.
- (pskextract, ID_Q): if $ID_{Q'} = \alpha$ or $ID_{Q=\alpha}$, B uses α to solve the truncated decision q-ABDHE. If ID_Q is in L, B simply returns the partial private key in the corresponding entry. Otherwise, B generates the public key, partial private key and private key for ID_Q as in the above, stores them into L and returns the partial private key to A.
- (rkextract, ID_Q, $ID_{Q'}$, σ_Q): if $ID_{Q=\alpha}$ or $ID_{Q'} = \alpha$, B uses α to solve the truncated decision q-ABDHE. If ID_Q or $ID_{Q'}$ or both are in L, B extracts the partial private key(s) in the corresponding entry (entries). Otherwise, B generates the public key, partial private key and private key for the identity not in L as in the above, stores them into L and uses the partial private keys concerned for further processing as follows. B computes the re-encryption key $rk_{Q \to Q'}$ using the usual re-encryption key calculation algorithm except that B generates the random value σ_Q on behalf of ID_Q and α is replaced by a random value (since B does not know the value of α). Note that although $rk_{Q \to Q'}$ is an invalid re-encryption key, A cannot verify its correctness since it does not possess the private key of ID_Q and it cannot query it from B either. B does not allow to query again.
- (decrypt, ID_Q, c_Q): if $ID_Q = \alpha$, B uses α to solve the truncated decision q-ABDHE. If ID_Q is in L, B simply uses the private key in the corresponding entry to decrypt c_Q by performing the usual delegator decryption algorithm. Otherwise, B generates the public key, partial private key and private key for ID_Q as in the above, stores them into L and uses the private key concerned decrypt c_Q by performing the usual delegator decryption algorithm.
- (re-encrypt, ID_Q, $ID_{Q'}$, c_Q, σ_Q): if $ID_Q = \alpha$ or $ID_Q = \alpha$, B uses α to solve the truncated decision q-ABDHE. If ID_Q or $ID_{Q'}$, or both are in L, B extracts the public and private keys in the corresponding entry (entries). Otherwise, B generates the public key, partial private key and private key for the identity not in L as in the above, stores them into L and uses the public and private keys concerned for further processing as follows. B decrypts c_Q using the private key for ID_Q by performing the usual delegator decryption algorithm and then encrypts the plaintext obtained using the public key for $ID_{Q'}$ by performing the usual encryption algorithm. This ensures that the re-encrypted ciphertext is decryptable by the private key for $ID_{Q'}$.

At the end of Phase 1, A outputs (ID_A, M_0, M_1) where A may have queried anything about ID_A but must not have queried (encrypt, ID_A, M_0) and (encrypt, ID_A, M_1) before. If $ID_A = \alpha$, B uses α to solve the truncated decision q-ABDHE. If ID_A is in L, B simply extracts the partial private key in the corresponding entry. Otherwise, B computes a partial private key $(r'_A, h'_A, r_{A,2}, h_{A,2}, r_{A,3}, h_{A,3})$ for ID_A as in the above. Next B generates bit $c \in \{0, 1\}$. Let $f_4(x) = x^{q+2}$ and let $F_{4,A}(x) = (f_4(x) - f_4(ID_A)) / (x - ID_A)$ be a polynomial of degree $q + 1$. B continues to set $u = g'^{(f_4(\alpha) - f_4(ID_A))r_A}$, $v = Z \times e(g', \prod_{i=0}^{q} g^{F_{4,A,i}\alpha^i})$ and $w = M_c / e(u, h_A, 1)^{1/r_A} v^{r_A}$, and $t = e(g, g)^{H'(M_c)}$ where $F_{4,A,i}$ is the coefficient of x^i in $F_{4,A(x)}$. After setting $\beta = H(u, v, w, t)$, B sets $y = e(u, h_{A,2} h_{A,3}^{\beta})^{1/r_A} v^{r_{A,2} + r_{A,3}^{\beta}}$, $z = g'^{\beta F_{4,A}(\alpha)} g^{H'(M_c)}$. B sends $c_A = (u, v, w, t, z, y)$ to A as the challenge ciphertext.

Phase 2: this phase proceeds as in Phase 1. However A is restricted from issuing the following queries:

1 (encrypt, ID_A, M_0) and (encrypt, ID_A, M_1).

2 (decrypt, ID_A, c_A).

3 Any pair of queries (rkextract, ID_A, ID_A) and (decrypt, ID_A, c_A) where c_A is the re-encrypted ciphertext using $rk_{A \to A'}$.

At the end of Phase 2, the adversary A outputs guesses $c'_{0,1}$. If $c' \in \{0, 1\}$, B outputs 0 (indicating that $Z = e(g_{q+1}, g')$). Otherwise, B outputs 1.

5 Conclusions

We attempt to solve the proxy colluding problem for multiple type of group communication. Most of the authors are made their attention on the PRE scheme which is only one user to another user. In this paper, we extend the notion of non-transferable PRE scheme which is used for one or more group communication. In our scheme, the proxy and the delegatees cannot collude because they are unable to generate re-encryption key for re-delegating decryption right without the original delegator's help. To the members

of different two groups, they can decrypt the ciphertext with the help of the proxy by using the delegator's certificate. This new feature will more effective for the distributed group communication.

In future, we plan to be applied at the lightweight device such as wireless sensor networks (WANs) (Jaballah et al., 2015) and the machine type communication (MTC) (Zhang et al., 2014) which make more effective by using our multiple group PRE scheme. We also leave the open problems of finding the secure PRE scheme which is not using the certificate.

Acknowledgements

This work is supported in part by JSPS Grant-in-Aids for Scientific Research (A) JP16H01705 and for Scientific Research (B) JP17H01695.

References

Ateniese, G., Fu, K., Green, M. and Hohenberger, S. (2006) 'Improved PRE schemes with application to secure distributed storage', *ACM Transactions on Information and System Security*, Vol. 9, No. 1, pp.1–30.

Blaze, M., Bleumer, G. and Strauss, M. (1998) 'Divertible protocols and atomic proxy cryptography', in *EUROCRYPT '98*, *Lecture Notes in Computer Science*, Vol. 1403, pp.127–144, Springer, Berlin.

Canetti, R. and Hohenberger, S. (2007) 'Chosen-ciphertext secure proxy re-encryption', in *Proc. ACM Conference on Computer and Communication Security*, pp.185–1946.

Cho, E-M., San, L. and Koshiba, T. (2017) 'Secure non-transferable proxy re-encryption for group membership and non-membership', in Barolli, L. et al. (Eds.): *NBiS 2017*, *Lecture Notes on Data Engineering and Communications Technologies*, Vol. 7, pp.876–887, Springer, Cham, Switzerland.

Deng, R.H., Weng, J., Liu, S. and Chen, K. (2008) 'Chosen-ciphertext secure proxy re-encryption without pairings', in Franklin, M.K., Hui, L.C.K. and Wong, D.S. (Eds.): *CANS, Lecture Notes in Computer Science*, Vol. 5339, pp.1–7, Springer, Berlin; Heidelberg; New York, NY.

Fan, X. and Liu, F. (2016) 'Various proxy re-encryption schemes from lattices', *IACR Cryptology ePrint Archive 2016/278* [online] https://eprint.iacr.org/ 2016/278.pdf.

Guo, H., Zhang, Z. and Xu, J. (2015) 'Non-transferable proxy re-encryption', *IACR Cryptology ePrint Archive 2015/1216* [online] https://eprint.iacr.org/2015/ 1216.pdf.

Guo, L. and Hu, L. (2012) 'Efficient bidirectional proxy re-encryption with direct chosen-ciphertext security', *Computers and Mathematics with Applications*, Vol. 63, No. 1, pp.151–157.

He, Y., Chim, T.W., Hui, L.C.K. and Yiu, S-M. (2012) 'Non-transferable proxy re-encryption scheme', in *Proc. 5th International Conference on New Technologies, Mobility and Security, IEEE*, pp.1–4.

Ivan, A. and Dodis, Y. (2003) 'Proxy cryptography revisited', in *Proc. NDSS'03, The Internet Society*.

Jaballah, W.B., Mosbah, M., Youssef, H. and Zemmari, A. (2015) 'Lightweight secure group communications for resource constrained devices', in *International Journal of Space-Based and Situated Computing 2015*, Vol. 5, No. 4, pp.187–200.

Libert, B. and Vergnaud, D. (2008)'Tracing malicious proxies in proxy re-encryption', in *Pairing 2008*, Vol. 5209, pp.332353, Lecture Notes in Computer Science, Springer, Berlin; New York.

Ma, C. and Ao, J. (2009) 'Group-based proxy re-encryption scheme', in ICIC 2009, Vol. 5754, pp.1025–1034, Lecture Notes in Computer Science, Springer, Berlin; New York.

Mambo, M. and Okamoto, E. (1997) 'Proxy cryptosystems: delegation of the power to decrypt ciphertexts', *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E80A, No. 1, pp.54–63.

Matsuda, T., Nishimaki, R. and Tanaka, K. (2010) 'CCA proxy re-encryption without bilinear maps in the standard model', in *PKC 2010*, Vol. 6056, pp.261–278, Lecture Notes in Computer Science, Springer.

Matsuo, T. (2007) 'Proxy re-encryption systems for identity- based encryption', in *Pairing 2007*, Vol. 4575, pp.247267, Lecture Notes in Computer Science, Springer, Berlin; New York.

Niu, K., Wang, X.A. and Zhang, M. (2009) 'How to solve key escrow problem in proxy re-encryption from CBE to IBE', in *Proc. 1st International Workshop on Database Technology and Applications, IEEE Computer Society*, pp.95–98.

Nuez, D., Agudo, I. and Lopez, J. (2015) 'A parametric family of attack models for proxy re-encryption', in *Proc. IEEE 28th Computer Security Foundations Symposium, CSF 2015*, pp.290–301.

Phong, L., Wang, L., Aono, Y., Nguyen, M. and Boyen, X. (2016) 'Proxy re-encryption schemes with key privacy from LWE'. *IACR Cryptology ePrint Archive 2016/327* [online] https://eprint.iacr.org/ 2016/327.pdf.

Wang, L., Wang, L., Mambo, M. and Okamoto, E. (2010) 'New identity-based proxy re-encryption schemes to prevent collusion attacks', in *Pairing 2010*, Vol. 6487, pp.327346, Lecture Notes in Compuer Science, Springer.

Wang, X.A. and Yang, X. (2010) 'On the insecurity of an identity based proxy re-encryption scheme', *in Fundamenta Informaticae*, Vol. 98, Nos. 2–3, pp.277–281.

Wang, X.A., Ma, J. and Yang, X.Y. (2015) 'A new proxy re-encryption scheme for protecting critical information systems', in *Journal of Ambient Intelligence and Humanized Computing*, Vol. 6, No. 6, pp.699–711.

Wang, X.A., Ma, J., Xhafa, F., Zhang, M. and Luo, X. (2017) 'Cost-effective secure E-health cloud system using identity based cryptographic techniques', in *Journal of Future Generation Computer Systems*, Vol. 67, pp.242–254.

Yamamoto, N. (2016) 'An improved group discussion system for active learning using smartphone and its experimental evaluation', in *International Journal of Space-Based and Situated Computing*, Vol. 6, No. 4, pp.221–227.

Zhang, Y., Chen, J., Li, H., Cao, J. and Lai, C. (2014) 'Group-based authentication and key agreement for machine-type communication', in *International Journal of Grid and Utility Computing*, Vol. 5, No. 2, pp.87–95.