Certificateless Broadcast Proxy Re-encryption With Group-Oriented Model

Chih-Hung Wang*, Pei-Jyun Lu
Department of Computer Science and Information Engineering, National Chiayi University, Chiayi City, Taiwan 60004.

* Corresponding author. Tel.: +886-5-2717736; email: wangch@mail.ncyu.edu.tw
Manuscript submitted September 15, 2015; accepted January 5, 2016.
doi: 10.17706/ijcce.2016.5.5.358-366

Abstract: The major concept of certificateless cryptosystem is to remove the certificate in public-key cryptosystem and solve the key escrow drawback in the ID-based scheme. For the privacy protection in the public cloud environment, the proxy re-encryption scheme was developed to allow a semi-trusted proxy to transform a ciphertext under one person's public key into other ciphertext under another person's public key without decryption. Nowadays, many related applications of certificateless proxy re-encryption scheme have been proposed. However, most of certificateless proxy re-encryption schemes allow only a single receiver, which means for more receivers, re-transforming the ciphertext multiple times is needed. This paper introduces a novel notion of proxy re-encryption in which the sender can broadcast the re-encrypted ciphertext to a group of receivers through the proxy's assistance based on certificateless public key cryptography. The proposed scheme has the property of constant-size re-encryption key and ciphertext, and is particularly suitable for the application executing in the enormous cloud networking.

Key words: Certificateless cryptosystem, proxy re-encryption, broadcast encryption, cloud security.

1. Introduction

The concept of the proxy re-encryption (PRE) was first proposed by Blaze *et al*. [1] in 1998 and has been very popular in recent years. The PRE scheme supports the delegation of decryption via a semi-trusted third party named proxy. The proxy can apply the re-encryption key to transform the original ciphertext encrypted under delegator's (sender's) public key into the delegatee's (receiver's) public key without revealing the underlying plaintext. In the scheme, the proxy cannot learn any information about messages or obtain any unauthorized transformation key that can be used to transform the ciphertext to be one decryptable by the unauthorized users during the process. The PRE scheme allows the delegator to construct a key called re-encryption key and deliver it to the proxy to re-encrypt the messages for other users, and therefore they can use their own private keys to decrypt the re-encrypted ciphertexts.

The PRE scheme is now very useful in various applications such as encrypted email forwarding system [2], access control in file storage [3], law enforcement [4], cloud storage [5], etc., and it is also particularly suitable for the storage-limited devices which need outsourcing a large amount of files to the untrusted server. The proxy re-encryption has been developed in many different manners. According to the direction of transformation, PRE can be categorized into bidirectional PRE, in which the proxy can transform the ciphertext from delegator to delegatee and vice versa, and unidirectional PRE [1], [6], in which the proxy can transform the ciphertext from delegator to delegatee but cannot perform it in the opposite direction.

According to the security level of transformation, PRE can be categorized into the resistance against chosen ciphertext attacks [7]-[10] and collusion attacks [11], [12]. For the efficiency and functionality, PRE can also be built by pairing-free PRE [12], [13] and identity-based PRE [7], [9], [14]. Apart from that, some kinds of schemes belong to single-hop PRE [3] or multi-hop PRE [1], [6] according to whether or not the transformation can be applied on the ciphertext more than once. In our paper, we address the scheme on the single-hop unidirectional PRE scheme.

Since the PRE schemes can be constructed based on either the traditional public key cryptography (PKC) [1], [3], [6], [11] or identity-based public key cryptography (IBC) [9] system, they would suffer from some key management issues. The traditional PKC has encountered the difficulty of certificate management because the authentication of public key is obtained via certification generated by the certification authority (CA); therefore, the demands for certificate will cause the issues associated with revocation, storage and distribution of the certificate. Compared with the traditional PKC scheme, IBC eliminates the need of certificates, but the dependence on private key generator (PKG), who can generate users' private keys in centralization manner by using system master secret key, will cause key escrow problem. That is, PKG knows all registered users' private keys.

The concept of certificateless public key cryptography (CLPKC) introduced in 2003 [2] is a compromise between the PKC and IBC. CLPKC scheme solves the revocation issue of PKC and the key escrow of IBC, respectively. CLPKC scheme does not require the certificates and also can solve the key escrow. Although the user obtains the *partial private key* generated by PKG, it cannot retrieve user's private key since the whole private key now contains *secret information* selected by the user himself. Consequently, the certificateless proxy re-encryption scheme (CLPRE) enhances the key management mechanism that are especially suitable for the untrusted services and large-scale network environment such as cloud systems. In recent years, many researches about CLPRE have been proposed. Sur *et al*. [15] introduced the notion of CLPRE in 2010 and proposed a CCA secure CLPRE scheme based on the Libert and Quisquater's CLPKE [16]. In 2012, Xu *et al*. [5] proposed a CLPRE scheme. Most of the researches above are extended from CLPKE [16] and the original CLPKE [2], and these schemes are also classified into pairing-based [15] and pairing-free [17].

PRE provides a secure and flexible method for a sender to store and share data. A sender may encrypt his message with his own public key and then store the ciphertext in an honest-but-curious proxy. When the receiver is decided, the sender delegates a re-encryption key associated with the receiver to the proxy and then the proxy can re-encrypt the initial ciphertext to the intended receiver. Finally, the receiver can decrypt the resulting ciphertext with her/his private key. However, the original PRE scheme allows only a single receiver. That means for more receivers, re-transforming the ciphertext multiple times is needed. To solve this problem, the concept of broadcast PRE (BPRE) [18] has been proposed, in which the ciphertext can be re-encrypted to multi-recipient at one time. That is to say, the sender can delegate a re-encryption key associated with the intended receiving group to make each member in this receiving group decrypt the re-encryption ciphertext with his own private key. The issue of broadcast encryption has received lots of attention but the most efficient approaches on a BPRE presently are built by an identity-based public key cryptography. So far, it has not been developed to combine the certificateless public key cryptography and broadcast encryption to alleviate the revocation overhead of the traditional PKC and eliminate the key escrow problem in IBC. The main challenge is that the certificateless public key cryptography has two parts of keys namely the *partial private key* constructed by PKG and the *secret value* selected by the user himself. It has some technical difficulties to build an efficient model that can support constant-size ciphertext for the broadcasting encryption in certificateless public key cryptosystem.

Our Contribution. In this paper, we propose a new alternative approach of group-oriented certificateless broadcast proxy re-encryption by using asymmetric group key agreement based on Zhang *et al*. [19] and identity-based broadcast encryption based on Delerabl´ee's scheme [20]. The sender in the proposed

scheme can transform an encrypted message to a specific receiving group through the assistance of the proxy, and each intended user in the receiving group is allowed to decrypt the ciphertext without need of cooperation with others. That is, the sender can delegate the decryption right to a group of users at one time.

The rest of the paper is organized as follows. Section 2 gives elementary definitions and security notation. Section 3 presents the detailed construction of group-oriented certificateless broadcast proxy re-encryption (GCLBPRE for short). Section 4 gives the security analysis. Finally, Section 5 concludes this paper.

2. Preliminaries

We propose a formal definition of our scheme. We first provide a concise overview of bilinear pairing and related computation assumption on which our GCLBPRE scheme is based.

2.1. Bilinear Groups

Let G_1 and G_2 be two cyclic groups of the same prime order q and g is a generator of G_1. A bilinear map is defined as $e: G_1 \times G_2 \to G_T$ with the following properties:

1) Bilinear: $\forall g \in G_1, h \in G_2$ and $a,b \in Z_q^*$, $e(g^a, h^b) = e(g,h)^{ab}$.

2) Non-degenerate: $\exists g \in G_1, h \in G_2, e(g,h) \neq 1$.

3) Computable: There is an efficient algorithm to compute $e(g,h)$ for all $g \in G_1, h \in G_2$.

2.2. Security Assumptions

Definition 1 Computational Diffie-Hellman (CDH) Assumption. Given the tuple $(g, g^a, g^b) \in G^3$, where $a, b \in Z_q^*$, it is infeasible to compute g^{ab}.

Definition 2 Divisible Computational Diffie–Hellman (DCDH) Assumption. Given the tuple $(g, g^a, g^b) \in G^3$, where $a, b \in Z_q^*$, it is infeasible to compute $g^{a/b}$.

Definition 3 *k*-Bilinear Diffie–Hellman Exponent (*k*-BDHE) Assumption [19]-[21]. Given the tuple $(g,h, g^{\alpha^i}) \in G_1$ for $i = 1,2,\ldots,k,k+2,\ldots 2k$ as input, it is infeasible to compute $e(g,h)^{\alpha^{k+1}}$.

Definition 4 (IND-CLPRE-CCA). The security for the CLPRE is under chosen ciphertext attack (IND-CCA) that is secure against the Type I adversary. The attacker's *public key request* would be answered by running the Partial-Private-Key-Extract algorithm (see the details in [16], [17]). In CLPRE scheme, IND-CLPRE-CCA is based on IND-CCA; the detail definitions were described in Appendix A. of [16].

Our scheme is constructed from the modification of Yang *et al.*'s CLPRE [17] which provides IND-CCA security. The security in CLPRE needs to take account of both CLPKE and PRE securities. According to the security definitions in CLPKE, we need to consider two types of these adversaries. First, we must further strengthen the model to prevent the adversary from replacing the public key since there is no CA to make the certification. In addition, we also need to consider that the adversary equipped with the master key, in order to model the security against an eavesdropping PKG. The brief introduction can be seen in Section 4 of security analysis. The readers also can refer to [22] for details.

2.3. A Formal Description of the Group-Oriented Certificateless Broadcast Proxy Re-encryption Scheme

In the following section, we present our model for a GCLBPRE. We define our notion as follows.

Definition 5. A formal description of the GCLBPRE scheme consists of the following algorithms:

- **Setup (1^λ):** Generate a list of public parameters ***PK*** and master key ***MK*** for a security parameter 1^λ.

- **Partial-Private-Key-Extract (*PK*, *MK*, ID_i):** Given parameters ***PK***, master key ***MK*** and an identity for the user, return a partial private key (ω_i, t_i, s_i).
- **Set-Secret-Value (*PK*, ID_i):** Given parameters ***PK*** and an identity for the user, return the secret value s_{ID_i} .
- **Set-Private-Key (*PK*, t_i, s_{ID_i}):** Given parameters ***PK***, a user's partial private key and secret value, the user runs this algorithm to generate his own private key sk_i .
- **Set-Public-Key (*PK*, ω_i, s_{ID_i}):** Given parameters ***PK***, a user's partial private key and secret value, the user runs this algorithm to generate his public key pk_i .
- **Set-Member's-Decryption-Key (*PK*, pk_i):** Given ***PK*** and a users' public key pk_i, the users in the receiving group run this algorithm to generate and distribute the shares used to construct user's own group member decryption key d_{ID_i} .
- **Encrypt (*PK*, ID_A, pk_A, m):** Given parameters ***PK***, an identity ID_A and the sender's public key pk_A, this algorithm is used for the generation of a regular ciphertext of *m*.
- **Set-Re-encryption-key (*PK*, ID_A, pk_A, $\{ID_j\}_{j\in[1,n]}$, $\{pk_j\}_{j\in[1,n]}$):** Given parameters ***PK***, a sender's identity and corresponding public key pk_A, an identity of receiving group $\{ID_j\}_{j\in[1,n]}$ and the corresponding public key set $\{pk_j\}_{j\in[1,n]}$. The sender runs this algorithm to generate a re-encryption key from the user ID_A to the receiving group.
- **Re-encrypt (*PK*, $RK_{A\to[1,n]}$, C_A):** Given parameters ***PK***, re-encryption key $RK_{A\to[1,n]}$ and regular ciphertext C_A, this algorithm is executed by the proxy for the generation of re-encryption ciphertext C_R .
- **Decrypt 1 (*PK*, $\{ID_j\}_{j\in[1,n]}$, sk_j, d_{ID_j}, C_R):** Taken parameters ***PK***, receiving group users' identities and a re-encryption ciphertext C_R as inputs, this algorithm (performed by one of the group members) outputs the plaintext *m*.
- **Decrypt 2 (*PK*, sk_A, C_A):** Taken parameters ***PK***, sender ID_A's private key sk_A, and a ciphertext C_A as inputs, this algorithm (performed by the sender ID_A) outputs the plaintext *m*.

3. The Proposed Scheme–Detailed Construction

3.1. Overview

In this section, we present the first construction of group-oriented certificateless broadcast proxy re-encryption scheme. We design it for re-encrypting the ciphertext to a specific group of the receivers by using the main technique named asymmetric group key agreement, in which the sender calculates an asymmetric encryption key by aggregating all users' public keys and then uses this key to encrypt the massage. In our scheme, members can be classified into different groups according to their attributes, jurisdictions and positions. The sender then chooses the desired group to send the message.

Before expatiating the main algorithms, a brief sketch of our scheme is explained below (also see Fig. 1). First, PKG generates a list of public parameters ***PK*** and a master key ***MK***, and then executes ***Partial-Private-Key-Extract*** algorithm to generate and send a *partial-private-key* to the user. Second, the user chooses a secret value s_{ID_i} by running the algorithm ***Set-Secret-Value*** and then generates his own private key sk_i and public key pk_i by the algorithms ***Set-Private-Key*** and ***Set-Public-Key***. Third, all group members ($ID_1,\ldots,ID_n$) run the algorithm ***Set-Member's-Decryption-Key*** to generate and distribute the

shares and therefore each member can construct his own group member decryption key d_{ID_i} that will be used for the final decryption process.

After the above key establishment process, the sender ID_A can generate a *re-encryption key* $RK_{A\to[1,n]}$ associated with the receiving group by the algorithm ***Set-Re-encryption-Key*** and delegate the key to the proxy to transform the regular ciphertext (encrypted by ID_A's private key) to the ciphertext C_R through the algorithm ***Re-encrypt***. Finally, the receiver runs the algorithm ***Decrypt1*** to decrypt the resulting ciphertext with his private key.

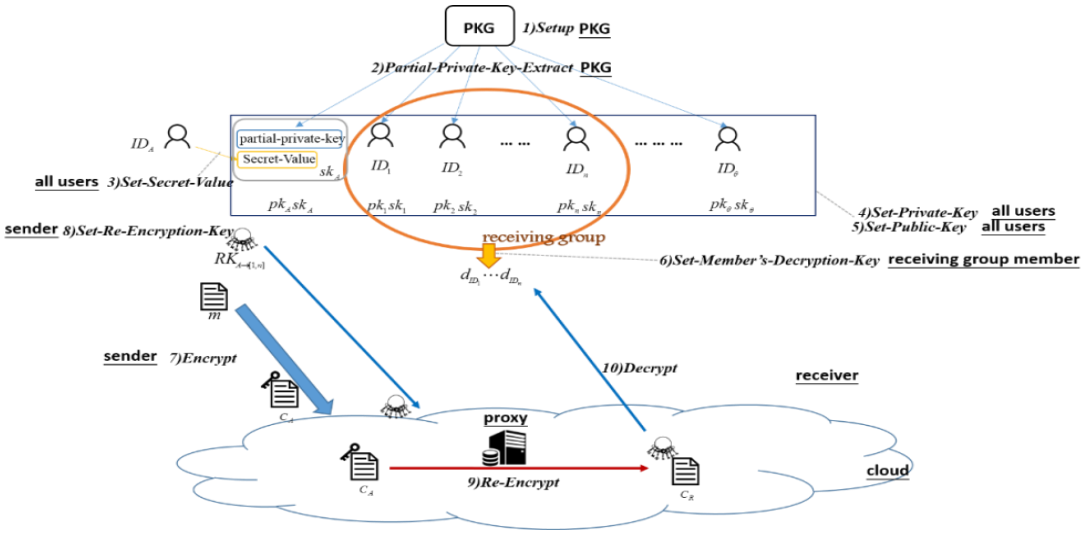

Fig. 1. Sketch of proposed scheme.

3.2. The Proposed Algorithms: GCLBPRE

The scheme consists of the following algorithms and the detailed construction is shown as follows.

Setup (1^λ): Given the security parameters 1^λ, a bilinear map group system $\beta=(q, G_1, G_2, G_T, e(,))$ is constructed where $|q|=\lambda$. The two bilinear map groups, G_1 and G_2, are of the same order q. Pick a generator $g\in G_1$ and a generator $h\in G_2$, and select $g_1,\ldots,g_\theta\in G_2$, where θ is the largest number of members that the system can support. Choose hash functions $H_1:\{0,1\}^*\times G_1\to Z_q^*$, $H_2:G_1\to\{0,1\}^\ell$ for an integer ℓ being the bit length, $H_3:\{0,1\}^*\to Z_q^*$ and $H_4:G_1\to Z_q^*$. Select $\alpha\in Z_q^*$ and $\bar{g}\in G_1$ at random as the secrets, and thus the public key is denoted by $y=g^\alpha$ and $\bar{y}=\bar{g}^\alpha$. The whole public parameters for the system are $\boldsymbol{PK}=(G_1, G_2, G_T, u, h, h^\alpha,\ldots,h^{\alpha^\theta}, g, g_1,\ldots, g_\theta, q, y, \bar{y}, \ell, H_1, H_2, H_3, H_4)$, where $u=e(\bar{g}, h)$. The master secret key ***MK*** is defined as a tuple $(\bar{g},\alpha)$.

Partial-Private-Key-Extract (*PK*, *MK*, ID_i): Given ***PK*** and an identity ID_i for the user ID_i as inputs, compute

$$s_i = g^{\frac{1}{\alpha+H_3(ID_i)}}$$

where $i=\{A,1,\ldots,n\}$ and $t_i=H_4(s_i)+\alpha H_1(ID_i,\omega_i)$. Set $\omega_i=g^{H_4(s_i)}$, and then return (ω_i, t_i, s_i) as user ID_i's partial private key.

Set-Secret-Value (*PK*, ID_i): Given ***PK*** and identity ID_i as inputs. Pick $z_i\in Z_q^*$ at random and $V_i\in G_2$.

Return $s_{ID_i}=(z_i,V_i)$ as user ID_i's secret value.

Set-Private-Key (*PK*, t_i, s_{ID_i}): Given ***PK***, user ID_i's partial private key t_i and secret value s_{ID_i} as inputs. Return user ID_i's secret key $sk_i=(t_i,s_i,z_i,V_i)$.

Set-Public-Key (*PK*, ω_i, s_{ID_i}): Given ***PK*** , user ID_i's partial private key ω_i and secret value $s_{ID_i}=(z_i,V_i)$ as input. Compute $\mu_i=g^{-z_i}$ and $\eta_i=e(g,V_i)$. Return user ID_i 's public key $pk_i=(\omega_i,\mu_i,\eta_i)$.

Set-Member-Decryption-Key (*PK*, pk_i): Given ***PK*** and pk_i as inputs, the user ID_i generates and distributes his shares to other group members and aggregates the collected shares to generate his own group member decryption key d_{ID_i}. The following two steps show how the user performs this algorithm: 1) Each user ID_i computes secret shares $\sigma_{i,j}=(V_i\cdot g_j)^{z_i}$ for $1\le j\le n$, where $n\le\theta$ (let the members of the group be $ID_1,\ldots,ID_n$) and sends them to the corresponding members in the group by an authenticated channel. The user with index *j* will receive only his share $\sigma_{i,j}$ from the user of index *i*. As the result, each member gets *n* secret shares from others as shown in Table 1, where $\sigma_{i,i}=\sigma_{i,i}=(V_i\cdot g_j)^{z_i}$ is not sent to others and only known to the user ID_i. Table 2 shows the description of keys each member needs to keep. 2) To get the group decryption key, each member in the group computes

$$d_{ID_i}=\prod_{l=1}^{n}\sigma_{l,i}$$

Table 1. Message Retrieval by Participants

Required for	ID_1	ID_2	ID_3	…	ID_n	All members
$ID_1\Rightarrow$	$\sigma_{1,1}$	$\sigma_{1,2}$	$\sigma_{1,3}$	…	$\sigma_{1,n}$	(μ_1,η_1)
$ID_2\Rightarrow$	$\sigma_{2,1}$	$\sigma_{2,2}$	$\sigma_{2,3}$	…	$\sigma_{2,n}$	(μ_2,η_2)
$ID_3\Rightarrow$	$\sigma_{3,1}$	$\sigma_{3,2}$	$\sigma_{3,3}$	…	$\sigma_{3,n}$	(μ_3,η_3)
⋮	⋮	⋮	⋮	⋱	⋮	⋮
$ID_n\Rightarrow$	$\sigma_{n,1}$	$\sigma_{n,2}$	$\sigma_{n,3}$	…	$\sigma_{n,n}$	(μ_n,η_n)
Decryption key	d_{ID_1}	d_{ID_2}	d_{ID_3}	…	d_{ID_n}	(μ,η)

Table 2. Keys for Each User Used in the Scheme

Style	Key Value(s)
Private	Partial private key (ω_i , t_i , s_i), Secret value $s_{ID_i}=(z_i,V_i)$, Private key $sk_i=(t_i,s_i,z_i,V_i)$, Group member decryption key $d_{ID_i}=\prod_{l=1}^{n}\sigma_{l,i}$
Public	Public key pk_i = (ω_i,μ_i,η_i)

Encrypt (*PK*, ID_A, pk_A, *m*): Given ***PK***, the sender ID_A and its corresponding public key pk_A as input. The sender's public key can be written as $pk_A=(\omega_A,\mu_A,\eta_A)$. To encrypt a message *m* using sender's identity ID_A and public key pk_A, the sender ID_A computes $\gamma_A=\omega_A\cdot y^{H_1(ID_A,\omega_A)}$ and $Y_A=\gamma_A{}^{H_4(\mu_A)}$, randomly picks $r\in Z_q^*$, and computes $c_1=g^r$, $c_2=m\oplus H_2(Y_A{}^r)$. The ciphertext is denoted by $C_A=(c_1,c_2)$.

Set-Re-encryption-Key (*PK*, ID_A, pk_A, $\{ID_j\}_{j\in[1,n]}$, $\{pk_j\}_{j\in[1,n]}$): The sender's public key and private

key are denoted by $pk_A=(\omega_A,\mu_A,\eta_A)$ and $sk_A=(t_A,s_A,z_A,V_A)$. The member ID_j in the receiving group has the public key $pk_j=(\omega_j,\mu_j,\eta_j)$. Assume that the receiving group members are $\{ID_1,ID_2,\ldots,ID_n\}$ where $n\le\theta$. In this phase, the sender aggregates the public keys of all receivers in the group to form a pair of group encryption keys ($\mu=\prod_{i=1}^{n}\mu_i$, $\eta=\prod_{i=1}^{n}\eta_i$).

The sender ID_A selects $k,t\in Z_q^*$ at random and computes the re-encryption key as follows:

$$rk_1=\bar{y}^{-k},rk_2=h^{k\cdot\prod_{i=1}^{n}(\alpha+H_3(ID_i))},rk_3=g^t,rk_4=\mu^t,rk_5=(t_A\cdot H_4(\mu_A)+z_A)\cdot X_{A\to[1,n]},$$

where $X_{A\to[1,n]}=H_3(K,\Gamma,ID_A,pk_A,\{ID_j\}_{j\in[1,n]},\{pk_j\}_{j\in[1,n]})$, $K=u^k$ and $\Gamma=\eta^k$. Set the re-encryption key for the receiving group $RK_{A\to[1,n]}=(rk_1,rk_2,rk_3,rk_4,rk_5)$.

Re-encrypt (*PK*, $RK_{A\to[1,n]}$, C_A): Parse C_A as (c_1,c_2), computes $c_1'=c_1^{rk_5}$ and set $c_2'=c_2$, $c_3'=rk_1$, $c_4'=rk_2$, $c_5'=rk_3$, $c_6'=rk_4$. Return the re-encryption ciphertext $C_R=(c_1',c_2',c_3',c_4',c_5',c_6')$.

Decrypt1 (*PK*, $\{ID_j\}_{j\in[1,n]}$, sk_j, d_{ID_j}, C_R): This algorithm needs to retire the $X_{A\to[1,n]}$ by using valid user's identity ID_j and the corresponding private key

$$s_j=g^{\frac{1}{\alpha+H_3(ID_j)}}$$

First of all, note that the values of K and Γ can be regarded as the common keys of the group and each group member can calculate them by solving the re-encrypted ciphertext with his private key. First, the user ID_j computes *K* as follows:

$$K=\left[e(c_3',h^{p_{j,[1,n]}(\alpha)})\cdot e(s_j,c_4')\right]^{\frac{1}{\prod_{i=1,i\ne j}^{n}H_3(ID_i)}},\text{ with }p_{j,[1,n]}(\alpha)=\frac{1}{\alpha}\cdot\left(\prod_{i=1,i\ne j}^{n}(\alpha+H_3(ID_i))-\prod_{i=1,i\ne j}^{n}(ID_i)\right).$$

Second, each valid user ID_j in the receiving group can reveal Γ by using his group member decryption key $\Gamma=e(c_5',d_{ID_j})\cdot e(c_6',g_j)$.

Finally, ID_j computes $X_{A\to[1,n]}=H_3(K,\Gamma,ID_A,pk_A,\{ID_j\}_{j\in[1,n]},\{pk_j\}_{j\in[1,n]})$, and then outputs $m=c_2'\oplus H_2(c_1'^{1/X_{A\to[1,n]}})$.

Decrypt2 (*PK*, sk_A, C_A): This is a decryption for the sender ID_A and can only be executed on the non-re-encrypted messages. Parse C_A as (c_1,c_2), pk_A as (ω_A,μ_A,η_A) and sk_A as (t_A,s_A,z_A,V_A). Computes $m=c_2\oplus H_2(c_1^{(t_A\cdot H_4(\mu_A)+z_A)})$.

4. Security Analysis

Our GCLBPRE scheme is constructed from the modification of Yang et al.'s CLPRE [17] which provides IND-CCA security. We also involve the function of broadcasting based on Delerabl'ee's IBBE [20] which reaches IND-CPA security under the GDDHE assumption. The technique of asymmetric group key agreement proposed by Zhang *et al.* [19] reaches the IND-CPA security. Apart from the above security considerations, Zhang *et al.*'s scheme also satisfies the following five security attributes: 1) known-key

security, 2) unknown key-share, 3) key-compromise impersonation, 4) perfect forward security, and 5) key control (see the details in [19] security analysis).

Due to the security description above, we consider our scheme achieving the security of IND-CPA. The proposed scheme also reaches the toleration of the Type I and Type II adversaries for the CLPKE as the definition in [2], [16]. The followings give a brief description of the two adversaries.

Type I adversary: Type I adversary A_I does not access the master key MK. However, A_I may request public key and then he will replace it with the values which he choices.

Type II adversary: Type II adversary A_{II} has accessed the master key MK, but is not allowed to replace the public key of users. A_{II} can request public key, make private key extraction and do the decryption.

5. Conclusions

We proposed a new concept of group-oriented certificateless broadcast proxy re-encryption scheme that combines the certificateless proxy re-encryption with the concept of broadcasting. Our scheme allows a sender to delegate the decryption of ciphertext to a specific group of receivers, via a re-encryption process of proxy. The proposed scheme is the first construction to re-encrypt the ciphertext for a specific group of the receivers by using the main technique named asymmetric group key agreement. Furthermore, the members in the scheme can be pre-classified into different groups according to their different properties to make the sender conveniently choose different desired groups when sending the message.

There is still insufficient for our scheme that the receiving group needs to be built in advance and then the members of the group are required to distribute their shares for the construction of the individual decryption keys. This result causes the amount of transmission between users in the group to be greatly large. How to reduce the transmission inside the group as well as to reach the goal of certificateless broadcast re-encryption with shorter ciphertext will become a major challenge and future research.

Acknowledgment

This work was supported in part by the Ministry of Science and Technology of Taiwan under the Grant MOST 104-2221-E-415-012.

References

[1] Blaze, M., Bleumer, G., & Strauss, M. (1998). Divertible protocols and atomic proxy cryptography. *Lecture Notes in Computer Science*, *1403*, 127-144.

[2] Al-Riyami, S. S., & Paterson, K. G. (2003). Certificateless public key cryptography. *Lecture Notes in Computer Science, 2894*, 452-473.

[3] Ateniese, G., Fu, K., Green, M., & Hohenberger, S. (2006). Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. on Information and System Security*, *9(1)*, 1-30.

[4] Ivan, A. A., & Dodis, Y. (2003). Proxy cryptography revisited. *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[5] Xu, L., Wu, X., & Zhang, X. (2012). CL-PRE: A certificateless proxy re-encryption scheme for secure data sharing with public cloud. *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security* (pp. 87-88).

[6] Canetti, R., & Hohenberger, S. (2007). Chosen-ciphertext secure proxy re-encryption. *Proceedings of the 14th ACM Conference on Computer and Communications Security* (pp. 185-194).

[7] Chu, C. K., & Tzeng, W. G. (2007). Identity-based proxy re-encryption without random oracles. *Information Security, Lecture Notes in Computer Science, 4779*, 189-202.

[8] Deng, R. H., Weng, J., Liu, S., & Chen, K. (2008). Chosen-ciphertext secure proxy re-encryption without pairings. *Lecture Notes in Computer Science*, *5339*, 1-17.

[9] Green, M., & Ateniese, G. (2007). Identity-based proxy re-encryption. *Lecture Notes in Computer Science, 4521*, 288-306.

[10] Shao, J., Cao, Z., & Liu, P. (2011). SCCR: a generic approach to simultaneously achieve CCA security and collusion-resistance in proxy re-encryption. *Security and Communication Networks*, *4(2)*, 122-135.

[11] Libert, B., & Vergnaud, D. (2008). Unidirectional chosen-ciphertext secure proxy re-encryption. *IEEE Transactions on Information Theory*, *57(3),* 360-379.

[12] Shao, J., & Cao, Z. (2009). CCA-secure proxy re-encryption without pairings. *Lecture Notes in Computer Science, 5443*, 357-376.

[13] Chow, S. S., Weng, J., Yang, Y., & Deng, R. H. (2010). Efficient unidirectional proxy re-encryption. *Lecture Notes in Computer Science, 6055*, 316-332.

[14] Wang, H., Shao, J., & Cao, Z. (2010). Multi-use unidirectional identity-based proxy re-encryption schemes. *Journal of Information Sciences, 180(20),* 4042-4059.

[15] Sur, C., Jung, C. D., Park, Y., & Rhee, K. H. (2010). Chosen-ciphertext secure certificateless proxy re-encryption. *Lecture Notes in Computer Science, 6109*, 214-232.

[16] Libert, B., & Quisquater, J. J. (2006). On constructing certificateless cryptosystems from identity based encryption. *Lecture Notes in Computer Science, 3958*, 474-490.

[17] Yang, K., Xu, J., & Zhang, Z. (2014). Certificateless proxy re-encryption without pairings. *Lecture Notes in Computer Science, 8565*, 67-88.

[18] Chu, C. K., Weng, J., Chow, S. S., Zhou, J., & Deng, R. H. (2009). Conditional proxy broadcast re-encryption. *Lecture Notes in Computer Science, 5594*, 327-342.

[19] Zhang, L., Wu, Q., Qin, B., Domingo-Ferrer, J., & González-Nicolás, Ú. (2011). Asymmetric group key agreement protocol for open networks and its application to broadcast encryption. *Journal of Computer Networks*, *55(15)*, 3246-3255.

[20] Delerablée, C. (2007). Identity-based broadcast encryption with constant size ciphertexts and private keys. *Lecture Notes in Computer Science, 4833*, 200-215.

[21] Boneh, D., Boyen, X., & Goh, E. J. (2005). Hierarchical identity based encryption with constant size ciphertext. *Lecture Notes in Computer Science, 3494*, 440-456.

[22] Safavi-Naini, R., Baek, J., & Susilo, W. (2005). Certificateless public key encryption without pairing. *Lecture Notes in Computer Science , 3650*, 134-148.

Chih-Hung Wang was born in Kaohsiung, Taiwan in 1968. He received the BS degree in information science from Tunghi University and MS degree in information engineering from National Chung Cheng University, Taiwan in 1991 and 1993, respectively. He received the Ph.D. degree in information engineering from National Cheng Kung University, Taiwan in 1998. He is presently an associate professor of the Department of Computer Science and Information Engineering, National Chiayi University, Taiwan. His research interests include cryptography, information security and data compression.

Pei-Jyun Lu is presently a master student of the Department of Computer Science and Information Engineering, National Chiayi University, Taiwan. Her research interests include cryptography and information security.