A New Efficient Certificateless Multi-Receiver Public Key Encryption Scheme

Jun Zhu[1,2], Lin-Lin Chen[3], Xian Zhu[4] and Ling Xie[5]

[1] **College of Computer Science, Nanjing University of Science and Technology Zijin College Nanjing, Jiangsu 210000, China**

[2] **College of Computer and Information Engineering, Hohai University Nanjing, Jiangsu 210000, China**

[3] **College of Computer Science, Nanjing University of Science and Technology Zijin College Nanjing, Jiangsu 210000, China**

[4] **College of Computer Science, Nanjing University of Science and Technology Zijin College Nanjing, Jiangsu 210000, China**

[5] **College of Computer Science, Nanjing University of Science and Technology Zijin College Nanjing, Jiangsu 210000, China**

Abstract

Multi-receiver public key encryption is important in insecurity and open network environment and has been applied in many scenarios such as first pay television system, streaming media services and so on. To avoid costly management of certificate and settle the matter of key escrow, we combine multi-receiver public key encryption with certificateless cryptography, and then present the notion, security model as well as a concrete scheme for certificateless multi-receiver encryption. Our new ideal scheme is efficient and only needs one (or none if pre-computation has been considered) pairing computation in the step of encryption. Meanwhile, we prove the IND-CCA security of our scheme under the intractability of CDHI and Gap-BDH problem. The efficient scheme is able to be generally applied in a variety of scenarios especially in broadcast communication.

Keywords: *Public Key Encryption, Certificateless Cryptography, Multi-Receiver, Random Oracle, Bilinear Map.*

1. Introduction

When a message sender wants to communicate with n users each of whom keeps a public key pk_i and a private key sk_i ($i = 1,\ldots, n$), he could encrypt messages M_i under pk_i and then send the resulting ciphertexts C_i to the corresponding user. This structure is called multi-plaintext, multi-receiver public key encryption [1-3]. The other case is that only one message M needs to be encrypted, which is similar to broadcast encryption [4-5]. Conversely, this structure is called single-plaintext, multi-receiver public key encryption (SMRE).

A naive or natural way to build a SMRE scheme is that the sender performs n times encryption operations for M under each user's public key and gets a ciphertext list (C_1,..., C_n). Nevertheless, this method is inefficient and expensive on computational cost and bandwidth requirement. Thus, the technique of "randomness re-use" has been subsequently presented by Kurosawa [3]. Using this technique, the length of ciphertexts and the computational cost can be almost half of that in the naive method.

However, as suggested by Baek et al. [6], just applying this technique is not enough to obtain an efficient SMRE scheme. For example, if the most widely used identity-based encryption scheme in [7] is utilized to construct a SMRE scheme, it requires at least n bilinear pairing computations. Aiming to solve this problem, Baek et al. [6] presented a detailed definition as well as security model for multi-receiver identity-based encryption and constructed a concrete scheme.

This paper is aimed at combining multi-receiver encryption with certificateless public key cryptography (CLPKC) which was first presented in [8]. In the structure of CLPKC, there is no problem about certificate management or key escrow. This unique charm makes CLPKC has a great vogue [9-13]. In recent years, an increasing number of scholars have devoted themselves to study multi-receiver encryption in CLPKC [14-17]. It is interesting and important to find a practical certificateless SMRE scheme.

Based on the construction in [6], this paper introduces the notion, security model and a secure and efficient scheme for certificateless multi-receiver public key encryption. Furthermore, the security proof is given based on the assumption that CDHI and Gap-BDH problem are infeasible. The new scheme is efficient and only needs one (or none if pre-computation has been considered) pairing computation in the operation of encryption. The ideal

IJCSI International Journal of Computer Science Issues, Volume 13, Issue 6, November 2016
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.20943/01201606.17

scheme can be widely applied in the insecurity and open network environment.

2. Preliminaries

Suppose G_1 and G_2 are groups of order q. The generator of G_1 is P. A bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_2$ satisfies the conditions as below:

(1) Bilinear: $\hat{e}(xM, yN) = \hat{e}(M, N)^{xy} = \hat{e}(xyM, N)$ for M, $N \in G_1$ and $x, y \in Z_q^*$.

(2) Non-degenerate: $\hat{e}(P, P) \neq 1$.

(3) The map is computable.

Bilinear Diffie-Hellman Problem (BDHP): Given (P, P^x, P^y, P^z) with randomly chosen x, y, $z \in Z_q^*$, BDHP aims to calculate $\hat{e}(P, P)^{xyz}$.

Gap-Bilinear Diffie-Hellman Problem (Gap-BDHP): Provided (P, P^x, P^y, P^z) for randomly chosen x, y, $z \in Z_q^*$, Gap-BDHP reminded by Bilinear Decisional Diffie-Hellman oracle is to compute $\hat{e}(P, P)^{xyz}$.

Computational Diffie-Hellman Inversion Problem (CDHIP): CDHIP is to calculate P^y by supplied (P, P^x, P^{xy}) with randomly chosen $x, y \in Z_q^*$.

3. Definitions for Certificateless Multi-Receiver Public Key Encryption

3.1 Description of Schemes

Definition 1 (CL-SMRE): A certificateless multi-receiver public key encryption scheme has following steps:

Setup: Input a security parameter *sp*, Key Generation Center (KGC) generates system parameters p and a master key *ms*.

PPK-Ext: Input p, *ms* as well as an identity *ID*, KGC can obtain the partial private key D_{ID} after running this procedure.

SV-Set: Given p and *ID* as inputs, the owner of *ID* selects a secret value X_{ID} by running this algorithm.

SK-Set: A user with identity *ID* calculates its private key S_{ID} by inputting p, D_{ID} and X_{ID}.

PK-Set: This step is run by the user to generate its public key P_{ID} after inputting p and X_{ID}.

Encrypt: Input p, multiple identities $(ID_1, ..., ID_n)$ of the receivers with their public key $(P_{ID_1}, ..., P_{ID_n})$ and a message M, the sender is responsible for creating the ciphertext C.

Decrypt: The owner of S_{ID_i} is in charge of performing this procedure by inputting p, C and S_{ID_i}, aiming to recover the message M or output $\perp$ indicating a decryption failure.

3.2 Security Model for CL-SMRE Schemes

A general adversary $\mathcal{A}$'s actions and the challenger $\mathcal{C}$'s responses in our security model are presented as follows:

(1) Partial Private Key Extraction query

$\mathcal{A}$ may ask any identity *ID*'s partial private key. The challenger $\mathcal{C}$ responds with D_{ID} by running algorithm PPK-Ext.

(2) Public Key query

$\mathcal{A}$ may ask any identity *ID*'s public key. The challenger $\mathcal{C}$ runns algorithm PK-Set to calculate P_{ID}.

(3) Replace Public Key request

For any entity, $\mathcal{A}$ can repetitively replace P_{ID} with arbitrary value P'_{ID}. Hereafter, P'_{ID} is then utilized by $\mathcal{C}$ in any case.

(4) Private Key query

$\mathcal{A}$ may ask any identity *ID*'s private key. $\mathcal{C}$ can respond with S_{ID} by running algorithm SK-Set.

(5) Decryption query

$\mathcal{A}$ could ask a decryption of a ciphertext C. To recover the plaintext, the challenger responds through the algorithm Decrypt on input the ciphertext and the private key corresponded to identity's current P_{ID}.

As in [8], two types of adversaries exist in CLPKC. $\mathcal{A}_I$ could replace any entity's public key but cannot get master key. In our security model, $\mathcal{A}_I$ could put forward any one of above five requests. Several natural restrictions on the behaviors of $\mathcal{A}_I$ are:

(1) $\mathcal{A}_I$ is banned from requesting private keys of target multiple identities $(ID_1^*, ..., ID_n^*)$.

(2) If an identity *ID*'s public key has been changed, $\mathcal{A}_I$ cannot query its private key any more.

(3) It's forbidden to request partial private keys of target multiple identities $(ID_1^*, ..., ID_n^*)$ and meanwhile substitute their public keys.

(4) The decryption query should not be requested on challenge ciphertext C^* which is encrypted under the challenge identity ID_i^*'s $P_{ID_i^*}$.

The other type adversary $\mathcal{A}_{II}$ is aware of master key but cannot replace any entity's public key. $\mathcal{A}_{II}$ can request public keys, private keys and decryption queries but must keep appointments as follows:

(1) $\mathcal{A}_{II}$ is banned from replacing public keys in any case.

(2) It's forbidden to request private keys of target multiple identities $(ID_1^*, ..., ID_n^*)$.

IJCSI
www.IJCSI.org

IJCSI International Journal of Computer Science Issues, Volume 13, Issue 6, November 2016
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.20943/01201606.17

(3) The decryption query should not be requested on challenge ciphertext C^* which is encrypted under the challenge identity ID_i^*'s $P_{ID_i^*}$.

For convenience, we adopt the concept of "selective identity attack" in [18] and assume that two types of attackers in our security model output target multiple identities in the initial phase. Although the assumption causes that our security is not as strong as the model in [7], we can demonstrate the IND-CCA security of our scheme under the model in [7], for the similar reason in [6], we omit it here.

Definition 2 (IND-sMID-CCA): A certificateless multi-receiver encryption scheme is IND-sMID-CCA secure when no adversary could win the game below with a non-negligible advantage:

Phase 1: $\mathcal{A}$ confirms ($ID_1^*,\ldots,ID_n^*$) as target multiple identities.

Phase 2: $\mathcal{C}$ obtains a master key ms and public parameters p through running Setup algorithm. When adversary is $\mathcal{A}_I$, $\mathcal{C}$ keeps ms secret. On the other hand, $\mathcal{C}$ shares ms with $\mathcal{A}_{II}$.

Phase 3: $\mathcal{A}$ puts forward some of above five requests which must be subject to the restrictions defined above.

Phase 4: $\mathcal{A}$ determines two plaintexts (m_0, m_1) with the same length. $\mathcal{C}$ selects one of them randomly and denotes with m_h. Afterwards, a ciphertext C^* which is the encryption of m_h under target multiple identities' current public key is delivered to $\mathcal{A}$.

Phase 5: $\mathcal{A}$ continues to issue requests as in Phase 3. Moreover, it is banned from asking decryption query for C^*.

Phase 6: Finally, a guess $h' \in \{0, 1\}$ is output by $\mathcal{A}$.

The adversary's advantage is defined as Adv ($\mathcal{A}$) = 2(Pr [$h = h'$] – 1/2).

The notion of IND-sMID-CPA is like **Definition 2** except that $\mathcal{A}$ is forbidden to put forward decryption queries.

4. Concrete Construction of CL-SMRE Scheme and Security Analysis

Firstly, we give a basic scheme which will be proved IND-sMID-CPA secure in the random oracle model, and then in order to enhance security, we modify our basic scheme to a full scheme to provide chosen ciphertext security.

4.1 Basic Scheme

Setup: Input a security parameter sp, KGC first generates bilinear parameters $< G_1, G_2, \hat{e}>$ in which the order of G_1 and G_2 are both q. Select m from Z_q^* and elements P, Q from G_1 respectively at random and define $P' = mP$. The master key is $ms=m$ and the system parameters are $p = <q, H_1, H_2, G_1, G_2, \hat{e}, P, Q, P'>$ where $H_1:\{0,1\}^* \to G_1$, $H_2: G_2 \times G_1 \to \{0,1\}^n$ are hash functions.

PPK-Ext: This procedure calculates an identifier ID's partial private key $D_{ID} = mH_1(ID) \in G_1$.

SV-Set: Given p and ID as inputs, the owner of ID selects X_{ID} from Z_q^* randomly as its secret value.

SK-Set: A user with identity ID calculates its private key $S_{ID}=(X_{ID}, D_{ID}) \in Z_q^* \times G_1$ after inputting p, D_{ID} and X_{ID}.

PK-Set: This step generates the public key $P_{ID}= X_{ID}P$ after inputting p and X_{ID}.

Encrypt: To encrypt a message M with public key $(P_{ID_1},\ldots,P_{ID_n})$, the sender chooses random value r_1 and r_2 from Z_q^*, computes $C = <U, V_1,\ldots, V_n, W_1, \ldots, W_n, X, \mathcal{L}> = <r_1P, r_1H_1(ID_1)+r_1Q, \ldots, r_1H_1(ID_n)+r_1Q, r_2P_{ID_1}, \ldots, r_2P_{ID_n}, M \oplus H_2(\hat{e}(P', r_1Q) \,||\, r_2P), \mathcal{L}>$ in which $\mathcal{L}$ indicates how V_i and W_i are contacted with every receiver.

Decrypt: With the help of $\mathcal{L}$, the owner of S_{ID_i} finds corresponding V_i and W_i and computes the plaintext $M = X \oplus H_2(\frac{\hat{e}(P',V_i)}{\hat{e}(U,D_{ID_i})} \,||\, W_iX_{ID_i}^{-1})$.

We can verity the consistency of decryption algorithm as below:

$$X \oplus H_2(\frac{\hat{e}(P',V_i)}{\hat{e}(U,D_{ID_i})} \,||\, W_iX_{ID_i}^{-1})$$

$$=M \oplus H_2(\hat{e}(P',r_1Q)||r_2P) \oplus H_2(\frac{\hat{e}(mP, r_1H_1(ID_i)+r_1Q)}{\hat{e}(r_1P, mH_1(ID_i))} \,||\, r_2P_{ID_i}X_{ID_i}^{-1})$$

$$=M \oplus H_2(\hat{e}(mP,r_1Q)||r_2P) \oplus H_2(\frac{\hat{e}(mP,r_1Q).\hat{e}(mP,r_1H_1(ID_i))}{\hat{e}(r_1P,mH_1(ID_i))} \,||\, r_2.(X_{ID_i}P).X_{ID_i}^{-1})$$

$$= M \oplus H_2(\hat{e}(mP,r_1Q)||r_2P) \oplus H_2(\hat{e}(mP,r_1Q) \,||\, r_2P)$$

$$=M$$

4.2 Security Analysis of Basic Scheme

Theorem 1: When an IND-sMID-CPA adversary $\mathcal{A}_I$ can attack the Basic scheme with advantage ϵ (H_1, H_2 are random oracles), then there is an algorithm $\mathcal{B}$ who can solve BDHP with a non-negligible advantage.

Proof: Suppose that $\mathcal{B}$ has (P, xP, yP, zP) as an instance of the BDHP.

Phase 1: $\mathcal{A}_I$ confirms ($ID_1^*,\ldots,ID_n^*$) as target multiple identities.

Phase 2: $\mathcal{B}$ sets public parameters $p=<q, H_1, H_2, G_1, G_2, \hat{e}, P, Q, P'>$ where $Q = yP$, $P'= zP$ and H_1, H_2 are hash functions under $\mathcal{B}$'s control:

IJCSI International Journal of Computer Science Issues, Volume 13, Issue 6, November 2016
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.20943/01201606.17

(1) H_1 queries on ID_j
$\mathcal{B}$ keeps an h_1 list of tuples (ID_j, l_j, L_j).
1) If (ID_j, l_j, L_j) has existed in h_1 list, $\mathcal{B}$ responds with L_j.
2) Else if $ID_j = ID_i^*$ for some $i \in [1, n]$, $\mathcal{B}$ selects l_j from Z_q^* randomly, computes $L_j = l_j P - Q$ as answer and adds the corresponding tuple to h_1 list.
3) Else $\mathcal{B}$ selects l_j from Z_q^* at random, computes $L_j = l_j P$ as answer and adds corresponding tuple to h_1 list.
(2) H_2 queries on X_j
$\mathcal{B}$ keeps an h_2 list of tuples (X_j, Y_j). If (X_j, Y_j) has existed, $\mathcal{B}$ responds with Y_j. Otherwise, $\mathcal{B}$ chooses $Y_j \in \{0,1\}^n$ at random, responds with Y_j and inserts (X_j, Y_j) to h_2 list.
Phase 3: $\mathcal{B}$ answers several queries put forward by $\mathcal{A}_I$ as follows.
(1) Partial Private Key Extraction query on ID_j
When (ID_j, l_j, L_j) has existed in h_1 list, $\mathcal{B}$ computes $D_{ID_j} = l_j (zP)$ as answer. Otherwise, $\mathcal{B}$ issues an H_1 query on ID_j.
(2) Public Key query on ID_j
$\mathcal{B}$ keeps a public key list of tuples (ID_j, X_j, X_jP). If ID_j's corresponding tuple has existed in the list, $\mathcal{B}$ responds with $P_{ID_j} = X_jP$. On the other hand, $\mathcal{B}$ picks a random X_j from Z_q^*, inserts (ID_j, X_j, X_jP) to the list and answers with X_jP.
(3) Replace Public Key request on ID_j
$\mathcal{B}$ records the situation and then the current value $P_{ID}^{'}$ is utilized by $\mathcal{B}$ in any case.
(4) Private Key query on ID_j
Suppose that $ID_j \neq ID_i^*$ ($i = 1,\dots, n$) and ID_j's public key is not been changed. $\mathcal{B}$ first issues H_1 query and public key query on ID_j and then calculates $S_{IDj} = (l_j (zP), X_j)$.
Phase 4: After $\mathcal{B}$ has selected the message m_h, he first picks random r^* from Z_q^* and R^* from $\{0, 1\}^n$, searches h_1 list to get l_j and public key list to obtain P_{ID_j} corresponding to ID_i^* ($i = 1,\dots, n$) and then computes l_jxP. $\mathcal{B}$ responds with $C^* = (xP, l_1xP,\dots, l_nxP, r^* P_{ID_1}, \dots, r^* P_{ID_n}, R^*)$.
Phase 5: As in Phase 3, $\mathcal{B}$ continues to answer $\mathcal{A}_I$'s queries.
Phase 6: A guess h' is output by $\mathcal{A}_I$.
Analysis: C^* is valid since $l_ixP = l_ixP\text{-}xQ+xQ = x(l_iP\text{-}Q)+xQ = xH_1(ID_i^*)+xQ$ ($i = 1,\dots, n$). If H_2 is modelled as a random oracle, $\mathcal{A}_I$ has advantage only if $e(P', r_1Q) = e(zP, xyP) = e(P, P)^{xyz}$ is an input of h_2 list. Thereafter $\mathcal{B}$ can solve BDHP.

Theorem 2: When an IND-sMID-CPA adversary $\mathcal{A}_{II}$ can attack the Basic scheme with advantage ϵ (H_1, H_2 are random oracles), then there is an algorithm $\mathcal{B}$ who can solve CDHIP with a non-negligible advantage.

Proof: Suppose that $\mathcal{B}$ is given (P, xP, xyP) as an instance of the CDHIP.
Phase 1: $\mathcal{A}_{II}$ confirms ($ID_1^*, \dots, ID_n^*$) as target multiple identities.
Phase 2: $\mathcal{B}$ selects a random m from Z_q^* and delivers m to $\mathcal{A}_{II}$ as master key. $\mathcal{B}$ sets public parameters $p=<q, H_1, H_2, G_1, G_2, \hat{e}, P, Q, P'>$ where $P' = mP$, Q is randomly selected from G_1 and H_1, H_2 are hash functions under $\mathcal{B}$'s control:
(1) H_1 queries on ID_j
$\mathcal{B}$ keeps an h_1 list of tuples (ID_j, L_j).
1) If (ID_j, L_j) has existed in h_1 list, $\mathcal{B}$ responds with L_j.
2) Else $\mathcal{B}$ selects random L_j from G_1 as answer and adds corresponding tuple to h_1 list.
(2) H_2 queries on X_j
$\mathcal{B}$ keeps an h_2 list of tuples (X_j, Y_j). If (X_j, Y_j) has existed, $\mathcal{B}$ responds with Y_j. Otherwise, $\mathcal{B}$ chooses $Y_j \in \{0,1\}^n$ at random, responds with Y_j and inserts (X_j, Y_j) to h_2 list.
Phase 3: $\mathcal{B}$ answers several queries put forward by $\mathcal{A}_{II}$.
(1) Public Key query on ID_j
$\mathcal{B}$ keeps a public key list of tuples (ID_j, X_j, T_j).
1) If ID_j's corresponding tuple has existed in the list, $\mathcal{B}$ responds with T_j.
2) Else if $ID_j = ID_i^*$ ($i \in [1, n]$), $\mathcal{B}$ picks a random X_j from Z_q^*, computes $T_j = xPX_j$, inserts (ID_j, X_j, T_j) to the list and answers with T_j.
3) Else $\mathcal{B}$ picks a random X_j from Z_q^*, calculates $T_j = x_j P$, inserts (ID_j, X_j, T_j) to the list and returns T_j as answer.
(2) Private Key query on ID_j
Suppose that $ID_j \neq ID_i^*$ ($i = 1,\dots, n$). $\mathcal{B}$ first issues H_1 query and public key query on ID_j and then calculates $S_{IDj} = (mL_j, X_j)$.
Phase 4: $\mathcal{B}$ first picks random r^* from Z_q^* and R^* from $\{0, 1\}^n$, searches h_1 list to get L_j and public key list to obtain X_j corresponding to ID_i^* ($i = 1,\dots, n$) and then computes $x_j xyP$. $\mathcal{B}$ responds with $C^* = (r^*P, r^*L_1+r^*Q,\dots, r^*L_n+r^*Q, x_1xyP,\dots, x_nxyP, R^*)$.
Phase 5: As in Phase 3, $\mathcal{B}$ continues to answer $\mathcal{A}_{II}$'s queries.
Phase 6: A guess h' is output by $\mathcal{A}_{II}$.
Analysis: If H_2 is modelled as a random oracle, $\mathcal{A}_{II}$ has advantage only if yP is an input of h_2 list. Thereafter $\mathcal{B}$ can solve CDHIP.

4.3 Full Scheme

The full scheme then can be depicted as follows.
Setup: Input a security parameter sp, KGC first generates bilinear parameters $< G_1, G_2, \hat{e}>$ in which the order of G_1

and G_2 are both q. Select m from Z_q^* and elements P, Q from G_1 respectively at random and define $P'= mP$. The master key is $ms=m$ and the system parameters are $p = <q, H_1, H_2, H_3, H_4, G_1, G_2, \hat{e}, P, Q, P'>$ where $H_1:\{0,1\}^*\rightarrow G_1$, $H_2: G_2\times G_1\rightarrow\{0,1\}^n$, $H_3: \{0,1\}^n\rightarrow\{0,1\}^n$, $H_4: G_1\times\ldots\times G_1\times\{0,1\}^n\times\{0,1\}^n\rightarrow\{0,1\}^k$ are hash functions.

PPK-Ext, SV-Set, SK-Set, PK-Set: These steps are the same as them in Section 4.1.

Encrypt: To encrypt a message M with public key $(P_{ID_1},\ldots,P_{ID_n})$, the sender chooses random value r_1, r_2 from Z_q^* and $R\in\{0,1\}^n$, computes $C = <U, V_1,\ldots, V_n, W_1, \ldots, W_n, Z_1, Z_2, \mathcal{L}, \sigma> = <r_1P, r_1H_1(ID_1)+r_1Q, \ldots, r_1H_1(ID_n)+r_1Q, r_2P_{ID_1}, \ldots, r_2P_{ID_n}, R\oplus H_2(\hat{e}(P', r_1Q) \| r_2P), M\oplus H_3(R), \mathcal{L}, H_4(R, M, V_1,\ldots, V_n, W_1, \ldots, W_n, Z_1, Z_2, \mathcal{L})>$.

Decrypt: With the help of $\mathcal{L}$, the owner of S_{ID_i} finds corresponding V_i and W_i and computes $R = Z_1 \oplus H_2(\frac{\hat{e}(P',V_i)}{\hat{e}(U,D_{ID_i})} \| W_i X_{ID_i}^{-1})$, $M= Z_2\oplus H_3(R)$, $\sigma' = H_4(R, M, V_1,\ldots, V_n, W_1, \ldots, W_n, Z_1, Z_2, \mathcal{L})>$. This algorithm will output M when $\sigma'=\sigma$, otherwise, it returns $\perp$.

4.4 Security Analysis of Full scheme

Theorem 3: When an IND-sMID-CCA adversary $\mathcal{A}_I$ can attack the Full scheme with advantage ϵ (H_i (i=1,2,3,4) are random oracles), then there is an algorithm $\mathcal{B}$ who can work out Gap-BDHP with a non-negligible advantage.

Theorem 4: When an IND-sMID-CCA adversary $\mathcal{A}_{II}$ can attack the Full scheme with advantage ϵ (H_i (i=1,2,3,4) are random oracles), then there is an algorithm $\mathcal{B}$ who can work out CDHIP with a non-negligible advantage.

To prove the above two theorems, we present two lemmas. Theorem 3 can be deduced from Lemma 1 and Theorem 1, and Theorem 4 can be deduced from Lemma 2 and Theorem 2.

Lemma 1: When an IND-sMID-CCA adversary $\mathcal{A}_I$ can attack the Full scheme with advantage ϵ with the help of BDDH oracle (H_i (i=1,2,3,4) are random oracles), then there is an IND-sMID-CPA adversary $\mathcal{B}_I$ who can attack the Basic scheme with a non-negligible advantage.

Proof: The simulation is as below.

Phase 1: $\mathcal{A}_I$ outputs target multiple identities (ID_1^*,…, ID_n^*). $\mathcal{B}_I$ then passes (ID_1^*,…, ID_n^*) to its challenger as its own challenged identities.

Phase 2: Once receiving the common parameter $<q, H_1, H_2, G_1, G_2, \hat{e}, P, Q, P'>$ from the challenger, $\mathcal{B}_I$ then passes $<q, H_1, H_2, H_3, H_4, G_1, G_2, \hat{e}, P, Q, P'>$ to $\mathcal{A}_I$, where H_3 and H_4 are in the possession of $\mathcal{B}_I$.

(1) H_1 and H_2 queries

Upon receiving such queries from $\mathcal{A}_I$, $\mathcal{B}_I$ passes the queries to the challenger. The answers responded by the challenger will be returned to $\mathcal{A}_I$ and recorded by $\mathcal{B}_I$.

(2) H_3 *and* H_4 queries

Upon receiving such queries, $\mathcal{B}_I$ first picks a value randomly, and then inserts the value to the corresponding list.

Phase 3: $\mathcal{B}_I$ responds to several queries put forward by $\mathcal{A}_I$:

(1) Partial Private Key Extraction query

(2) Public Key query

(3) Private Key query

Once receiving above queries, $\mathcal{B}_I$ passes the query to the challenger. The answer responded by the challenger will be returned to $\mathcal{A}_I$ and recorded by $\mathcal{B}_I$.

(4) Replace Public Key request on ID_j

$\mathcal{B}_I$ records the situation and then passes the same request to the challenger.

(5) Decryption query

$\mathcal{A}_I$ supplies identities ID_j (j= 1,…, n) and a ciphertext C= $<U, V_1,\ldots, V_n, W_1, \ldots, W_n, Z_1, Z_2, \mathcal{L}, \sigma>$. $\mathcal{B}_I$ responds with $\perp$ when $((R, M, V_1,\ldots, V_n, W_1,\ldots, W_n, Z_1, Z_2, \mathcal{L}), \sigma)$ doesn't exist in h_4 list. On the other hand, $\mathcal{B}_I$ does:

1) Compute $H_3(R)$ and verify whether $H_3(R)\oplus M = Z_2$. If not, return $\perp$.

2) Compute $R\oplus Z_1$, then look up h_2 list to find whether it has a tuple $((x, y), R\oplus Z_1)$. If not, return $\perp$.

3) Verify whether (P, U, Q, P', x) is a BDH tuple with the help of BDDH oracle. If not, return $\perp$.

4) Check whether $\hat{e}(y, P_{ID_j}) = \hat{e}(W_j, P)$. If not, return $\perp$.

5) Return M as plaintext.

Phase 4: On receiving the challenged ciphertext C' from the challenger, $\mathcal{B}_I$ sets $C^*= (C', X^*, Y^*)$ for $\mathcal{A}_I$ where $X^*\in\{0,1\}^n$, $Y^*\in\{0,1\}^k$ are randomly picked by $\mathcal{B}_I$.

Phase 5: As in Phase 3, $\mathcal{B}_I$ continues to answer $\mathcal{A}_I$'s queries.

Phase 6: A guess is output by $\mathcal{A}_I$.

Analysis: Once $\mathcal{A}_I$ works out the guess h', $\mathcal{B}_I$ uses h_3 and h_4 list to find an element $w\in\{0,1\}^n$ satisfying $H_3(w)\oplus M_{h'}= X^*$ and $H_4(w, M_{h'}, C', X^*)= Y^*$, then the element w is the answer to solve the challenger's problem, and hence $\mathcal{B}_I$ can attack the Basic scheme.

Lemma 2: When an IND-sMID-CCA adversary $\mathcal{A}_{II}$ can attack the Full scheme with advantage ϵ(H_i (i=1,2,3,4) are random oracles), then there is an IND-sMID-CPA adversary $\mathcal{B}_{II}$ who can attack the Basic scheme with a non-negligible advantage.

Proof: The simulation is as below.

Phase 1: $\mathcal{A}_{II}$ determines target multiple identities (ID_1^* ,…, ID_n^*). $\mathcal{B}_{II}$ then passes (ID_1^* ,…, ID_n^*) to its challenger as its own challenged identities.

Phase 2: Once receiving the master key m and the common parameter $<q, H_1, H_2, G_1, G_2, \hat{e}, P, Q, P'>$ from the challenger, $\mathcal{B}_{II}$ then passes $<q, H_1, H_2, H_3, H_4, G_1, G_2, \hat{e}, P, Q, P'>$ and m to $\mathcal{A}_{II}$, where H_3 and H_4 are in the possession of $\mathcal{B}_{II}$.

(1) H_1 and H_2 queries

Upon receiving such queries from $\mathcal{A}_{II}$, $\mathcal{B}_{II}$ passes the queries to the challenger. The answers responded by the challenger will be returned to $\mathcal{A}_{II}$ and recorded by $\mathcal{B}_{II}$.

(2) H_3 *and* H_4 queries

Upon receiving such queries, $\mathcal{B}_{II}$ first picks a value randomly, and then inserts the value to the corresponding list respectively.

Phase 3: $\mathcal{B}_{II}$ answers several queries put forward by $\mathcal{A}_{II}$:

(1) Public Key query

(2) Private Key query

Upon receiving above queries, $\mathcal{B}_{II}$ passes the query to the challenger. The answer responded by the challenger will be returned to $\mathcal{A}_{II}$ and recorded by $\mathcal{B}_{II}$.

(3) Decryption query

$\mathcal{A}_{II}$ supplies identities ID_j (j= 1,…, n) and a ciphertext C= $<U, V_1,..., V_n, W_1,..., W_n, Z_1, Z_2, \mathcal{L}, \sigma>$. B_{II} responds with $\perp$ when $((R, M, V_1,..., V_n, W_1,..., W_n, Z_1, Z_2, \mathcal{L}), \sigma)$ doesn't exist in h_4 list. On the other hand, $\mathcal{B}_{II}$ does:

1) Compute $H_3(R)$ and verify whether $H_3(R) \oplus M = Z_2$. If not, return $\perp$.

2) Compute $R \oplus Z_1$, then look up h_2 list to find whether it has a tuple$((x, y), R \oplus Z_1)$. If not, return $\perp$.

3) Check whether $\hat{e}(U, Q)^m = x$. If not, return $\perp$.

4) Check whether $\hat{e}(y, P_{ID_j}) = \hat{e}(W_j, P)$. If not, return $\perp$.

5) Return M as plaintext.

Phase 4: On receiving the challenged ciphertext C' from the challenger, $\mathcal{B}_{II}$ sets $C^* = (C', X^*, Y^*)$ for $\mathcal{A}_{II}$ where $X^* \in \{0,1\}^n$, $Y^* \in \{0,1\}^k$ are randomly picked by $\mathcal{B}_{II}$.

Phase 5: As in Phase 3, $\mathcal{B}_{II}$ continues to answer $\mathcal{A}_{II}$'s queries.

Phase 6: A guess is output by $\mathcal{A}_{II}$.

Analysis: Once $\mathcal{A}_{II}$ works out the guess h', $\mathcal{B}_{II}$ uses h_3 and h_4 list to find an element $w \in \{0,1\}^n$ satisfying $H_3(w) \oplus M_{h'} = X^*$ and $H_4(w, M_{h'}, C', X^*) = Y^*$, then the element w is the answer to solve the challenger's problem, and hence $\mathcal{B}_{II}$ can attack the Basic scheme.

5. Performance Analysis

Aiming to analyze the performance, we compare the computational cost of Encrypt algorithm and the length of ciphertext in our full scheme with those in another construction in which a message is encrypted n times with the help of CC's typical CLPKE scheme [9]. The results of comparison are shown in **Table 1**, where E represents exponentiation operation, S represents multiplication operation and P represents the most time-consuming operation——pairing.

Table 1: Performance Analysis

Schemes	***Encrypt***	***The Length of Ciphertext***
Scheme constructed from CC's construction [9]	nP+2nS+ nE	3n
Our full scheme	1P+(2n+3)S	2n+3

6. Conclusions

In this paper, we studied multi-receiver encryption in the area of CLPKC and introduced the notion and security model of CL-SMRE schemes. We also presented a concrete construction for a secure and efficient CL-SMRE scheme. The scheme only needs one (or none if pre-computation has been considered) pairing computation in Encrypt algorithm. Furthermore, we proved the security of our scheme under the assumption that CDHIP and Gap-BDHP are difficult. Though the security model is not strong enough where the adversary outputs target multiple identities in the initial phase, we suggest that our scheme can reach to CCA secure under the strong security model in [7]. The ideal scheme presented in this paper has effective and practical applications to guarantee confidentiality in group communications in the insecurity and open network environment.

One shortage of our scheme is that the length of the ciphertexts is not short enough. So for further works, we expect to find a CL-SMRE scheme with shorter ciphertexts and seek more applications for designing encryption schemes.

Acknowledgments

This research was supported by Natural Science Foundation of the Colleges and Universities in Jiangsu Province (No. 16KJB520019, No. 15KJB520017) and Scientific Research Foundation of Nanjing University of Science and Technology Zijin College.

References

[1] M. Bellare, A. Boldyreva, and S. Micali, "Public-key Encryption in a Multi-User Setting: Security Proofs and Improvements", in EUROCRYPT'00, 2000, Vol. 1807, pp. 259-274.

[2] M. Bellare, A. Boldyreva, and D. Pointcheval, "Multi-Recepient Encryption Schemes: Security Notions and Randomness Re-Use", in PKC 2003, 2003, Vol. 2567, pp. 85-99.

[3] K. Kurosawa, "Multi-Recepient Public-Key Encryption with Shortened Ciphertext", in PKC 2002, 2002, Vol. 2274, pp. 48-63.

[4] Y. Dodis, and N. Fazio, "Public Key Broadcast Encryption for Stateless Receivers", Lecture Notes in Computer Science, Vol. 2696, 2003, pp. 61-80.

[5] Y. Dodis, and N. Fazio, "Public Key Trace and Revoke Scheme Secure against Adaptive Chosen Ciphertext Attack", Lecture Notes in Computer Science, Vol. 2567, 2010, pp. 100-115.

[6] J. Baek, R. Safavinaini, and W. Susilo, "Efficient Multi-receiver Identity-Based Encryption and Its Application to Broadcast Encryption", Lecture Notes in Computer Science, Vol. 3386, No. 3, 2005, pp. 380-397.

[7] D. Boneh, and M. Franklin, "Identity-Based Encryption from the Weil Pairing", Siam Journal on Computing, Vol. 32, No. 3, 2003, pp. 213-229.

[8] S. S. Al-Riyami, and K. G. Paterson, "Certificateless Public Key Cryptography", Lecture Notes in Computer Science, Vol. 2894, No. 2, 2003, pp. 452-473.

[9] Z. H. Cheng, and R. Comley, "Efficient Certificateless Public Key Encryption", Cryptology ePrint Archive, Vol. 249, 2005, pp. 1-25.

[10] Y. X. Sun, F. T. Zhang, and J. Baek, "Strongly Secure Certificateless Public Key Encryption without Pairing", Lecture Notes in Computer Science, Vol. 4856, 2007, pp. 194-208.

[11] A. W. Dent, B. Libert, and K. G. Paterson, "Certificateless Encryption Schemes Strongly Secure in the Standard Model", Lecture Notes in Computer Science, Vol. 4939, 2008, pp. 344-359.

[12] K. Bentahar, P. Farshim, J. Malone-Lee, and N. P. Smart, "Generic Constructions of Identity-based and Certificateless KEMs", Journal of Cryptology, Vol. 21, No. 2, 2008, pp. 178-199.

[13] W. J. Yang, F. T. Zhang, and L. M. Shen, "Efficient Certificateless Encryption Withstanding Attacks from Malicious KGC without using Random Oracles", Security and Communication Networks, Vol. 7, No. 2, 2014, pp. 445-454.

[14] L. J. Pang, H. X. Li, L. C. Jiao, and Y. M. Wang, "Design and Analysis of a Provable Secure Multi-Recipient Public Key Encryption Scheme", Journal of Software, Vol. 20, No. 10, 2009, pp. 2907-2914.

[15] Y. X. Sun, H. Li, and X. Q. Li, "Certificateless Signcryption KEM to Multiple Recipients", Journal of Electronics & Information Technology, Vol. 32, No. 9, 2010, pp. 2249-2252.

[16] S. K. Zeng, M. X. He, and M. W. Tang, "Deniable Ring Authentication Based on Multi-receiver Encryption", Journal of Xihua University (Natural Science), Vol. 34, No. 2, 2015, pp. 1-5.

[17] Y. L. Qin, X. P. Wu, and W. Hu, "Efficient Certificateless Multi-receiver Anonymous Signcryption Scheme", Journal on Communications, Vol. 37, No. 6, 2016, pp. 129-136.

[18] R. Canetti, S. Halevi, and J. Katz, "A Forward-Secure Public-Key Encryption Scheme", Lecture Notes in Computer Science, Vol. 2656, 2003, pp. 255-271.

Jun Zhu, a PhD student who will receive PhD degree in the major of computer science and technology, Hohai University, China. In June 2010 she has got her master degree of computer application at Nanjing Normal University. Since September 2010, she has worked in Nanjing University of Science and Technology Zijin College and is a lecturer and the director of the major of software engineering. She has published 11 papers and one patent related to certificateless cryptology, meanwhile, she has been supported by two provincial scientific research projects in 2016. Her research topics are related to cryptology, information security and intelligent information processing. **Corresponding Author**.

Linlin Chen, a PhD student who will receive PhD degree in the major of computer science and technology, Nanjing University of Science and Technology, China, and has got her master degree at Suzhou University in 2006. Now she works in Nanjing University of Science and Technology Zijin College and is the dean of the college of computer science. She has published several papers related to software architecture and data mining.

Xian Zhu, has got her master degree of computer application at Nanjing Normal University in 2009 and now works in Nanjing University of Science and Technology, China. She has been supported by natural science foundation of the colleges and universities in Jiangsu province in 2015. Her research topics are related to pattern recognition and biological information.

Ling Xie, a PhD student who will receive PhD degree in quantum communication, Nanjing University, China. In 2006, she got her master degree at Nanjing University of Technology and began to work in Nanjing University of Science and Technology Zijin College. She has published several papers and one patent related to automation control and quantum communication.