Towards a Secure Certificateless Proxy Re-Encryption Scheme*

Hui Guo, Zhenfeng Zhang, Jiang Zhang, and Cheng Chen

Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing, China.
{guohui,zfzhang,zhangjiang,chencheng}@tca.iscas.ac.cn

Abstract. Proxy re-encryption (PRE) is an attractive paradigm, which gives good solutions to the problem of delegation of decryption rights. In proxy re-encryption, a semi-trusted proxy translates a ciphertext for Alice into a ciphertext of the same plaintext for Bob, without learning any information of the underlying message. As far as we know, previous PRE schemes are mainly in traditional public key infrastructure or identity-based cryptography, thus they suffer from certificate management problem or key escrow problem in practice. In order to solve these practical problems, we aim at constructing certificateless proxy re-encryption (CL-PRE) schemes.

In this paper, we first introduce a security definition against (replayable) chosen ciphertext attack (CCA) for certificateless proxy re-encryption. In our security model, the adversary is allowed to adaptively corrupt users (in a specific pattern). Then, we give some evidence that it is not easy to construct a secure CL-PRE. Actually, we present an attack to the chosen plaintext secure CL-PRE scheme proposed by Xu et al. [1]. We also show a novel generic construction for certificateless public key encryption (CL-PKE) can not be trivially adapted to CL-PRE by giving an attack to this generic construction. Finally, we present an efficient CL-PRE scheme and prove its security in the random oracle model based on well-known assumptions.

1 Introduction

Proxy re-encryption (PRE) was first proposed by Blaze, Bleumer and Strauss [2] in 1998, which allows the proxy to transform a ciphertext for Alice into a ciphertext of the same message for Bob. During the transformation, the proxy learns nothing about the underlying message. Having the proxy transform ciphertext and simultaneously keeping the message private from the proxy is the main goal for proxy re-encryption.

According to the direction of transformation, PRE schemes can be classified into two types: unidirectional and bidirectional schemes. In a unidirectional PRE scheme, the proxy can only transform the ciphertext from Alice to Bob; while in a bidirectional one, the proxy can transform in both directions. Essentially, we can construct a bidirectional

* The work is supported by the National Basic Research Program of China (No. 2013CB338003), and the National Natural Science Foundation of China (No.61170278, 91118006).

W. Susilo and R. Reyhanitabar (Eds.): ProvSec 2013, LNCS 8209, pp. 330–346, 2013.

PRE scheme by simply combining two unidirectional PRE schemes. In this paper, we only restrict our attention to unidirectional PRE schemes.

Proxy re-encryption has many applications, such as email forwarding [3], distributed files systems [4] and revocation systems [5]. Below we take Personal Health Record (PHR) sharing [6] as an example and explain the importance of constructing CL-PRE schemes.

A telemedical system involves patients, doctors and electronic medical records servers. Patients outsource their personal health records, which include various medical data, such as surgery, family history, laboratory test results, to be stored at the electronic medical server. Since patients do not hope to expose the records to those electronic medical records servers or unauthorized parties, they usually choose to encrypt their personal health records before outsourcing. When a telemedical consultation occurs, the electronic medical records server re-encrypts related personal health records to the involved doctors. During the process, the patient would not like to expose his secret key to either the server or any doctor. Proxy re-encryption provides a good solution to this problem.

When we examine the existing schemes, we find the schemes are inappropriate in the telemedical system. Schemes in [4,3,7] are all of traditional PKI-supported PRE. Since the amount of patients and doctors are huge, public key management will be the most costly and cumbersome part that reduces the efficiency of the system. Schemes in [8,9] are of identity-based proxy re-encryption (IB-PRE) and schemes in [10,11] are of attribute-based proxy re-encryption (AB-PRE). In IB-PRE or AB-PRE schemes, a trusted third party computes all private keys and is able to read all messages in the system, which is contrary to the *Health Insurance Portability and Accountability Act (HIPPA)* privacy rules. To avoid the expensive certificates in PKI and the key escrow problem inherited from IBE or ABE, we resort to certificateless public key cryptography (CL-PKC).

CL-PKC was introduced by Al-Riyami and Paterson [12] in 2003. The concept is to enjoy the advantage of identity-based public key cryptography without suffering from the key escrow problem. In CL-PKC, a sender needs both the receiver's identity and public key to encrypt a message. However, the public key here needs no certificate, which is different from the public key used in traditional PKI-supported cryptography. When decrypting, the receiver needs two parts to recover the message: one is called the partial private key corresponding to his identity which is generated by the key generation center (KGC); the other is the secret value related to the public key produced by himself. Therefore, the KGC cannot recover ciphertexts in the system in that the KGC has no information about the secret values chosen by users. We construct CL-PRE schemes for the telemedical system to enjoy both the efficiency and security provided by CL-PKC.

1.1 Related Work

Certificateless Public Key Cryptography. Since the notion CL-PKC was introduced in 2003, a variety of certificateless public key encryption (CL-PKE) schemes have been proposed. In 2005, Baek et al. [13] proposed the first CL-PKE scheme without pairing in the random oracle model. The formulation for the certificateless encryption is different from Al-Riyami and Paterson [12]: a user has to receive the partial private

key before producing their public key. In 2006, Libert et al. [14] and Chow et al's [15] proposed the generic construction of certificateless encryption respectively. Libert et al. presented the generic composition idea : given a CPA IBE scheme and a CPA public key encryption (PKE) scheme, a CCA CL-PKE scheme can be obtained in the random oracle model. Chow et al. presented a generic construction for security-mediated certificateless encryption which provides instant revocation. In 2007, Lai et al. [16] proposed two variants of Baek et al. scheme [13]. CL-PKE schemes to strengthen the scheme of Baek et al. [13], respectively. Sun et al. modified the scheme and enabled the Type I adversary to replace the public key associated with the target identity, but still disallowed the adversary to extract the partial private key of the target identity. While in Lai et al.'s scheme, the user engages in a protocol with the KGC when computing their full public and private keys, to allow the Type I adversary to extract the partial private key of the target identity. Both of the two schemes are secure against chosen ciphertext attacks in the random oracle model.

Proxy Re-Encryption. In 1998, Blaze et al. [2] proposed the concept of proxy re-encryption and constructed a bidirectional scheme, which is semantically secure in the random oracle model. In 2007, Canetti and Hohenberger [3] presented the first bidirectional scheme which is replayable chosen ciphertext secure in the standard model. In 2008, Libert and Vergnaud [7] proposed the first unidirectional single-hop PRE scheme, which is replayable CCA-secure in the standard model. In 2010, Chow et al. [17] proposed an efficient unidirectional PRE scheme without pairings.

The above schemes are in traditional public key infrastructure, which cannot avoid the certificate management problem. In 2007, Green and Ateniese [9] introduced the concept of identity based proxy re-encryption (IB-PRE) and proposed the first IB-PRE scheme in the random oracle model. In the same year, Chu and Tzeng [8] presented the first CCA secure IB-PRE scheme in the standard model. In 2010, Luo et al. [11] proposed an AB-PRE scheme.

IB-PRE and AB-PRE solve the certificate management problem, but bring in the key escrow problem. In order to solve this problem, we focus on realizing a secure CL-PRE scheme.

1.2 Our Contribution

In this paper, we introduce the syntax of CL-PRE and formulate a replayable CCA (RCCA) security model for CL-PRE. Firstly, our model considers both the Type I adversary and the Type II adversary. The Type I adversary represents attacks from outsiders with the ability to replace user's public key on his will. The Type II adversary stands for the honest but curious PKG who has access to the master secret key. Secondly, in our security model of CL-PRE, the Type I adversary has the ability to set up the dishonest user's public key or replace honest user's public key. Thirdly, our model allows the Type I adversary to adaptively corrupt honest users in a specific way. For example, it can replace the public key and query the partial private key of the honest user.

Then, we give some discussions on constructing RCCA secure CL-PRE schemes. First we present an attack to Xu et al.'s scheme [1], which was claimed to be secure

against chosen plaintext attack (CPA) in the random oracle model. Unfortunately, we show their scheme is insecure by giving a CPA attack. Secondly, we show a novel generic construction of CL-PRE which is adapted from the generic construction of CL-PKE in [14] is vulnerable to the Type I adversary under adaptively chosen ciphertext attacks. Both evidences show that it is difficult to construct a secure (especially RCCA secure) CL-PRE scheme.

Next, we present a RCCA secure CL-PRE scheme. The idea is to construct a CL-PRE scheme based on Sun et al.'s CL-PKE scheme [18] (which is the modification of Baek et al.'s CL-PKE scheme [13]). Firstly, we extend Sun et al.'s scheme into the pairing based setting. Secondly, in order to allow the adversary to extract challenger's partial private key (which is not allowed in Sun et al's sheme) and reach the RCCA security, we generate each entity's public key and private key by engaging a protocol with the KGC, similar to Lai et al.'s scheme in [16]. In the re-encryption key generation process, the delegator computes re-encryption keys on input his own private key and the public key of the delegatee. Finally, we present the security proof of the scheme in the random oracle model. As far as we know, the proposed scheme is the first CL-PRE scheme that is RCCA secure against both Type I and Type II adversaries.

2 Preliminaries

In this section, we recall the complexity assumption required in our scheme. In our paper, we use λ to denote the security parameter.

2.1 Bilinear Maps and Assumptions

In this section, we recall the definitions of the bilinear groups [19,20] and q-wDBDHI assumption based on the bilinear groups. We write $\mathbb{G} = \langle g \rangle$ to denote that g generates the group $\mathbb{G}$. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order p, a map e: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is said to be a bilinear map if it satisfies the following conditions:

1. for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. e is non-degenerate (i.e. if $\mathbb{G} = \langle g \rangle$, then $\mathbb{G}_T = \langle e(g, g) \rangle$).
3. e is efficiently computable.

Let $\mathbb{G}$, $\mathbb{G}_T$ and e be bilinear groups defined as above, we recall the following hardness assumptions over the groups.

CDH Assumption. For an algorithm $\mathcal{B}$, define its advantage as

$$\mathrm{Adv}_{\mathcal{B}}^{\mathrm{CDH}}(\lambda) = |\Pr[\mathcal{B}(g, g^a, g^b) = g^{ab}]|$$

where $a, b \leftarrow \mathbb{Z}_p^*$ are randomly chosen. We say that the CDH (Computational Diffie-Hellman) assumption holds, if for any probabilistic polynomial time (PPT) algorithm $\mathcal{B}$, its advantage $\mathrm{Adv}_{\mathcal{B}}^{\mathrm{CDH}}(\lambda)$ is negligible in λ.

q-wDBDHI Assumption. For an algorithm $\mathcal{B}$, define its advantage as

$$\mathrm{Adv}_{\mathcal{B}}^{q\text{-wDBDHI}}(\lambda) = |\Pr[\mathcal{B}(g, g^a, \ldots, g^{a^q}, g^b, e(g, g)^{b/a}) = 1]$$

$$- \Pr[\mathcal{B}(g, g^a, \ldots, g^{a^q}, g^b, e(g,g)^z) = 1]|$$

where $a, b, z \leftarrow \mathbb{Z}_p^*$ are randomly chosen. We say that the q-wDBDHI (q-weak Decision Bilinear Diffie-Hellman Inversion) assumption holds, if for any PPT algorithm $\mathcal{B}$, its advantage $\mathrm{Adv}_{\mathcal{B}}^{q\text{-wDBDHI}}(\lambda)$ is negligible in λ.

In our scheme, we use the 1-wDBDHI assumption (i.e., $q = 1$), which is slightly stronger than the DBDH assumption. We note that 1-wDBDHI assumption is also used in several other interesting cryptographic constructions [7,21].

2.2 Target-Collision Resistant Hash Function

Let $\mathcal{F} = (\mathrm{TCR}_s)_{s \in S}$ be a family of hash functions for security parameter λ and with seed $s \in S$. For an algorithm $\mathcal{A}$, define its advantage as

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{TCR}}(\lambda) = \Pr[\mathrm{TCR}_s(x) = \mathrm{TCR}_s(x') \wedge x \neq x' | s \leftarrow S,$$

$$x \leftarrow X, x' \leftarrow \mathcal{A}(\mathrm{TCR}_s, x)].$$

We define hash function family TCR is target collision resistant if for any PPT algorithm $\mathcal{A}$, its advantage $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{TCR}}(\lambda)$ is negligible in λ.

3 Certificateless Proxy Re-Encryption

In this section, we present the syntax of CL-PRE. A certificateless proxy re-encryption scheme consists of the following algorithms:

$\mathrm{Setup}(\lambda)$: This is a PPT algorithm run by the KGC, which takes a security parameter λ as input, outputs a list of public parameter $param$ and a randomly chosen master secret key msk.

$\mathrm{UserKeyGen}(param, ID)$: This is a PPT algorithm run by the user, which takes a list of public parameters $param$ as inputs, outputs a secret key sk and a public key pk.

$\mathrm{PKeyExt}(param, msk, ID, pk)$: This is a PPT algorithm run by the KGC, which takes a list of public parameters $param, msk$, a user's identity ID and pk as inputs, outputs a partial private key psk and a partial public key ppk.

$\mathrm{KeyGen}(param, ID, psk, ppk, sk, pk)$: This is a PPT algorithm run by the user, which takes a list of public parameters $param, ID$, psk, ppk, sk and pk as inputs, outputs the user's public key and private key (PK, SK).

$\mathrm{ReKeyGen}(param, ID_i, SK_i, PK_i, ID_j, PK_j)$: This is a PPT algorithm run by the user, which takes a list of public parameters $param$, a user's ID_i, SK_i, PK_i and another user's ID_j, PK_j as inputs, outputs the re-encryption key $rk_{i \to j}$ or an error symbol $\perp$.

$\mathrm{Enc}_1(param, m, ID, PK)$: This is a PPT algorithm run by the sender, which takes a list of parameters $param$, a message m, a receiver's ID and PK as inputs, outputs a 1st level ciphertext C which can not be re-encrypted.

$\mathrm{Enc}_2(param, m, ID, PK)$: This is a PPT algorithm run by the sender, which takes a list of parameters $param$, a message m, a receiver's ID and PK as inputs, outputs a 2nd level ciphertext C which can be re-encrypted.

$\mathrm{ReEnc}(param, C, ID_i, ID_j, rk_{i\to j})$: This is a PPT algorithm run by the proxy, which takes a list of public parameters $param$, users' identity ID_i and ID_j and a 2nd level ciphertext C under ID_i as inputs, outputs a 1st level ciphertext C' of ID_j or an error symbol $\perp$.

$\mathrm{Dec}_1(param, C, SK)$: This is a deterministic algorithm run by the recipient, which takes a list of public parameters $param$, a 1st level ciphertext C and SK as inputs, outputs the plaintext m or an error symbol $\perp$.

$\mathrm{Dec}_2(param, C, SK)$: This is a deterministic algorithm run by the recipient which takes a list of public parameters $param$, a 2nd level ciphertext C and SK as inputs, outputs the plaintext m or an error symbol $\perp$.

Correctness. For any public parameters $param$ generated by $\mathrm{Setup}(\lambda)$, for any message $m \in \{0,1\}^{l_0}$, in which l_0 denotes the length of the message, if SK_i and SK_j are corresponding with PK_i and PK_j, the above algorithms should satisfy the following requirements:

- $\mathrm{Dec}_2(param, \mathrm{Enc}_2(param, m, ID_i, PK_i), SK_i) = m$.
- $\mathrm{Dec}_1(param, \mathrm{Enc}_1(param, m, ID_i, PK_i), SK_i) = m$.
- If $rk_{i\to j} = \mathrm{ReKeyGen}(param, ID_i, SK_i, ID_j, PK_j)$, $C'_j = \mathrm{ReEnc}(param, \mathrm{Enc}_2(param, m, ID_i, PK_i), ID_i, ID_j, rk_{i\to j})$, then $\mathrm{Dec}_1(param, SK_j, C'_j) = m$.

3.1 Security Model

In CL-PKC, adversaries are divided into two types: the Type I adversary, who can replace user's public key on his choice; and the Type II adversary, holding the master secret key of the KGC. The Type I adversary describes the outsider's attack, while the Type II adversary stands for the curious but honest KGC, who can generate all the partial keys with the master secret key. To protect data privacy, we require that the adversary cannot gain any protected information unless holding both the partial private key and the secret value at the same time. When we take the two types of adversaries into consideration in the security models of CL-PRE, the circumstances seem to become more complex.

Unlike previous model in [7], we consider the model where the adversary can adaptively choose public keys for malicious users. In addition, we allow the Type I adversary to (partially) adaptively corrupt users, different from the previous model [7].

To capture the RCCA security notion for single-hop unidirectional CL-PRE schemes, we consider the security of ciphertexts at both levels against the Type I adversary and the Type II adversary separately. $\mathcal{A}$ denotes a Type I adversary or a Type II adversary. We associate to a CL-PRE adversary $\mathcal{A}$ the following CL-PRE RCCA experiment with parameters $(\mathcal{O}', \delta)$, where $\mathcal{O}'$ is a set of oracles provided to $\mathcal{A}$, and $\delta \in \{1, 2\}$ specifies which level ciphertext that $\mathcal{A}$ attacks. Both parameters will be instantiated in Definition 1, 2, 3 and 4.

Experiment $\mathbf{Exp}^{\mathrm{clpre,rcca}}_{\Pi_s,\mathcal{A}}(\lambda)$
$param \leftarrow \mathrm{Setup}(\lambda)$,
$(m_0, m_1, ID^*) \leftarrow \mathcal{A}^{\mathcal{O}'}(param)$,
$d^* \leftarrow \{0,1\}$,
$C^* = \mathrm{Enc}_\delta(m_{d^*}, ID^*)$,
$d' \leftarrow \mathcal{A}^{\mathcal{O}'}(param, C^*)$
If $d' = d^*$ return 1, else return 0

The advantage of $\mathcal{A}$ is defined as $\mathrm{Adv}^{\mathrm{clpre,rcca}}_{\Pi_s,\mathcal{A}}(\lambda) = |\Pr[\mathrm{Exp}^{\mathrm{clpre,rcca}}_{\Pi_s,\mathcal{A}}(\lambda) = 1] - \frac{1}{2}|$.

Security against the Type I Adversary. First, we consider the RCCA security notion against the Type I adversary at the 2nd level ciphertext. Before setting up the oracles, the challenger creates two lists: the HU list and the L list. The HU list is a list of honest users' identities. When a user is corrupted, the challenger removes the user's identity from the HU list. The L list is a list of $\langle ID, PK, \widehat{PK}\rangle$, where $ID \in HU$, PK and $\widehat{PK}$ denote the original public key and the current public key of ID. The list is to record whether the public key of a specific identity has been replaced. The challenger sets $\widehat{PK} = PK$ initially. In Definition 1 and Definition 2, the Type I adversary is provided with the following oracles:

- Honest key generation $\mathcal{O}_{\text{hkg}}$: on input ID, compute $(sk, pk) \leftarrow \mathrm{UserKeyGen}(ID)$, $(psk, ppk) \leftarrow \mathrm{PKeyExt}(ID, pk)$ and $(PK, SK) \leftarrow \mathrm{KeyGen}(ID, ppk, psk)$. Return PK.
- Delegation $\mathcal{O}_{\text{deleg}}$: on input $(ID_i, ID_j, \widehat{PK_j})$, where $\widehat{PK_j}$ may be an arbitrary public key supplied by $\mathcal{A}$, compute the re-encryption key $rk_{i\to j} = \mathrm{ReKeyGen}(ID_i, SK_i, PK_i, ID_j, \widehat{PK_j})$. Return $rk_{i\to j}$.
- Re-encryption $\mathcal{O}_{\text{renc}}$: on input $(ID_i, ID_j, \widehat{PK_j}; C)$, where $\widehat{PK_j}$ may be an arbitrary public key supplied by $\mathcal{A}$, compute the re-encrypted ciphertext $C' = \mathrm{ReEnc}(\mathrm{ReKeygen}(ID_i, ID_j, \widehat{PK_j}); C)$. Return C'.
- First level decryption $\mathcal{O}_{\text{1-dec}}$: on input a pair $(ID; C)$, compute the plaintext $m = \mathrm{Dec}_1(ID; C)$. Return m.
- Second level decryption $\mathcal{O}_{\text{2-dec}}$: on input a pair $(ID; C)$, compute the plaintext $m = \mathrm{Dec}_2(ID; C)$. Return m.
- Partial key extract oracle $\mathcal{O}_{\text{pex}}$: on input a pair (ID, PK), compute $(ppk, psk) = \mathrm{PKeyExt}(ID_i, pk)$ (pk can be extracted from PK). If $ID \in HU$ and $\widehat{PK} \neq PK$, the challenger updates $HU = HU \backslash ID$. Return (ppk, psk).
- Public key replace oracle $\mathcal{O}_{\text{pkr}}$: on input a pair $(ID, \widehat{PK})$, replace the user's public key with $\widehat{PK}$ and set $\langle ID, PK, \widehat{PK}\rangle$ on the L list.

Now, let's consider the RCCA security against the Type I adversary at the 2nd level.

Definition 1 (RCCA Security against the Type I Adversary at the 2nd Level Ciphertext). *For any single-hop unidirectional CL-PRE scheme Π_s, we instantiate the CL-PRE RCCA experiment with the Type I adversary $\mathcal{A}_I$, $\mathcal{O}' =$*

$\{\mathcal{O}_{\text{hkg}}, \mathcal{O}_{\text{deleg}}, \mathcal{O}_{\text{renc}}, \mathcal{O}_{\text{1-dec}}, \mathcal{O}_{\text{2-dec}}, \mathcal{O}_{\text{pex}}, \mathcal{O}_{\text{pkr}}\}$ *and* $\delta = 2$. *Suppose the challenger ciphertext* C^* *is generated under* ID^* *and* $\widehat{PK^*}$, *where* $\widehat{PK^*}$ *denotes the current public key of* ID^*. *We require that* $ID^* \in HU$ *and* $|m_0| = |m_1|$. *If* C^* *denotes the challenge ciphertext,* $\mathcal{A}_I$ *can never make following queries:*

- *Delegation query* $\mathcal{O}_{\text{deleg}}(ID^*, ID_x)$, *if* $ID_x \notin HU$.
- *Decryption query* $\mathcal{O}_{\text{2-dec}}(ID^*, C^*)$, *if* $\widehat{PK^*} = PK^*$.
- *Re-encryption query* $\mathcal{O}_{\text{renc}}(ID^*, ID_x, C^*)$, *if* $ID_x \notin HU$ *and* $\widehat{PK^*} = PK^*$.
- *Decryption query* $\mathcal{O}_{\text{1-dec}}(ID', C')$, *if* $\text{Dec}_1(ID', C') \in \{m_0, m_1\}$.

We say Π_s *is secure against (replayable) chosen ciphertext attacks at the 2nd level if for any polynomial time adversary* $\mathcal{A}_I$, *the advantage function* $\text{Adv}^{\text{clpre,2-rcca}}_{\Pi_s, \mathcal{A}_I}(\lambda)$ *is negligible in* λ.

When we consider the security notion at the 1st level ciphertext, we remove the restriction of re-encryption key queries. There is no reason to keep any re-encryption keys from the adversary, even those from the target entity to corrupted entities. Since $\mathcal{A}$ can do arbitrary re-encryption with re-encryption keys, $\mathcal{O}_{\text{renc}}$ is unnecessary. Then, we formulate the security definition as follows:

Definition 2 (RCCA Security against the Type I adversary at the 1st Level Ciphertext). *For any single-hop unidirectional CL-PRE scheme* Π_s, *we instantiate the CL-PRE RCCA experiment with the Type I adversary* $\mathcal{A}_I$, $\mathcal{O}' = \{\mathcal{O}_{\text{hkg}}, \mathcal{O}_{\text{deleg}}, \mathcal{O}_{\text{1-dec}}, \mathcal{O}_{\text{2-dec}}, \mathcal{O}_{\text{pex}}, \mathcal{O}_{\text{pkr}}\}$ *and* $\delta = 1$. *Suppose the challenger ciphertext* C^* *is generated under* ID^* *and* $\widehat{PK^*}$, *where* $\widehat{PK^*}$ *denotes the current public key of* ID^*. *We require that* $ID^* \in HU$ *and* $|m_0| = |m_1|$. *If* $\widehat{PK^*} = PK^*$ *where* PK^* *denotes the original public key of* ID^*, $\mathcal{A}$ *is not allowed to make decryption query* $\mathcal{O}_{\text{1-dec}}(ID^*, C^*)$ *after seeing the challenge ciphertext* C^*. *We say* Π_s *is secure against (replayable) chosen ciphertext attacks at the 1st level if for any polynomial time adversary* $\mathcal{A}$, *the advantage function* $\text{Adv}^{\text{clpre,1-rcca}}_{\Pi_s, \mathcal{A}_I}(\lambda)$ *is negligible in* λ.

Remark 1. In our model, when an honest entity's public key has been replaced, the challenger will still use his original secret key to decrypt and generate re-encryption keys. Since the honest entity would not possess the secret key corresponding with the replaced public key, we cannot force the honest entity to run algorithms with an unknown value in reality. Therefore, it is a reasonable assumption that the challenger manages oracles $\mathcal{O}_{\text{deleg}}$, $\mathcal{O}_{\text{1-dec}}$ and $\mathcal{O}_{\text{2-dec}}$ with the secret key related to the original public key.

Remark 2. Adversary $\mathcal{A}_I$ could corrupt honest entity in a specific way: first $\mathcal{A}_I$ replaces the honest user's public key; then queries the oracle $\mathcal{O}_{\text{pex}}$ to gain the partial private key. Since we do not allow the adversary to directly query the secret key of an honest user, our model is partially adaptive. With the oracle $\mathcal{O}_{\text{pex}}$, the adversary can also generate secret keys of corrupted users.

Remark 3. In [22], Hanaoka et al. illustrated the adversary with both the 2nd level decryption oracle and the 1st level decryption oracle is strictly stronger than the adversary who can only access to the 1st level decryption oracle. Therefore, in our model we provide the adversary with the oracle $\mathcal{O}_{\text{2-dec}}$ as well as the oracle $\mathcal{O}_{\text{1-dec}}$ to achieve a higher level security.

Security Against the Type II Adversary. Let's consider the security definition against the Type II adversary. Since the Type II adversary stands for the curious KGC, we provide $\mathcal{A}_{II}$ with an oracle $\mathcal{O}_{\text{msk}}$ to obtain the master secret key. $\mathcal{A}_{II}$ could produce arbitrary partial private key with the master secret key, therefore the oracle $\mathcal{O}_{\text{pex}}$ is unnecessary in the security experiment. In Definition 3 and Definition 4, the oracles work as follows:

- Master secret key $\mathcal{O}_{\text{msk}}$: on input the security parameter λ, return the master secret key msk.
- Honest key generation $\mathcal{O}_{\text{hkg}}$: on input ID, compute $(sk, pk) \leftarrow \text{UserKeyGen}(ID)$, $(psk, ppk) \leftarrow \text{PKeyExt}(ID)$ and $(PK, SK) \leftarrow \text{KeyGen}(param, ID, ppk, psk)$. Return PK.
- Delegation $\mathcal{O}_{\text{deleg}}$: on input (ID_i, ID_j, PK_j), compute the re-encryption key $rk_{i\to j} = \text{ReKeyGen}(ID_i, SK_i, PK_i, ID_j, PK_j)$. Return $rk_{i\to j}$.
- Re-encryption $\mathcal{O}_{\text{renc}}$: on input $(ID_i, ID_j, PK_j; C)$, compute the re-encrypted ciphertext $C' = \text{ReEnc}(\text{ReKeyGen}(ID_i, ID_j, PK_j); C)$. Return C'.
- First level decryption $\mathcal{O}_{\text{1-dec}}$: on input $(ID; C)$, compute the plaintext $m = \text{Dec}_1(ID; C)$. Return m.
- Second level decryption $\mathcal{O}_{\text{2-dec}}$: on input $(ID; C)$, compute plaintext $m = \text{Dec}_2(ID; C)$. Return m.

We define the RCCA security against Type II adversary at the 2nd level ciphertext as follows:

Definition 3 (RCCA Security against the Type II Adversary at the 2nd Level Ciphertext). *For any single-hop unidirectional CL-PRE scheme Π_s, we instantiate the CL-PRE RCCA experiment with the Type II adversary A_{II}, the $\mathcal{O}' = \{\mathcal{O}_{\text{msk}}, \mathcal{O}_{\text{deleg}}, \mathcal{O}_{\text{renc}}, \mathcal{O}_{\text{1-dec}}, \mathcal{O}_{\text{2-dec}}\}$ and $\delta = 2$. We require that $ID^* \in HU$ and $|m_0| = |m_1|$. If C^* denotes the challenge ciphertext, A_{II} can never make following queries:*

- *Delegation query $\mathcal{O}_{\text{deleg}}(ID^*, ID_x)$, if $ID_x \notin HU$.*
- *Decryption query $\mathcal{O}_{\text{2-dec}}(ID^*, C^*)$.*
- *Re-encryption query $\mathcal{O}_{\text{renc}}(ID^*, ID_x, C^*)$, if $ID_x \notin HU$.*
- *Decryption query $\mathcal{O}_{\text{1-dec}}(ID', C')$, if $\text{Dec}_1(ID', C') \in \{m_0, m_1\}$.*

We say Π_s is secure against (replayable) chosen ciphertext attacks at the 2nd level if for any polynomial time adversary $\mathcal{A}$, the advantage function $\text{Adv}^{\text{clpre,2-rcca}}_{\Pi_s, A_{II}}(\lambda)$ is negligible in λ.

Then, we consider the security notion at the 1st level ciphertext. As Definition 2, the oracle $\mathcal{O}_{\text{renc}}$ is unnecessary. We define the RCCA security against the Type II adversary at the 1st level ciphertext as follows:

Definition 4 (RCCA Security against the Type II Adversary at the 1st Level Ciphertext). *For any single-hop unidirectional CL-PRE scheme Π_s, we instantiate the CL-PRE RCCA experiment with the Type II adversary A_{II}, $\mathcal{O}' = \{\mathcal{O}_{\text{msk}}, \mathcal{O}_{\text{deleg}}, \mathcal{O}_{\text{1-dec}}, \mathcal{O}_{\text{2-dec}}\}$ and $\delta = 1$. We require that $ID^* \in HU$, $|m_0| = |m_1|$ and A_{II} is not allowed to make decryption query $\mathcal{O}_{\text{1-dec}}(ID^*, C^*)$ after seeing the*

challenge ciphertext C^*. We say Π_s is secure against (replayable) chosen ciphertext attacks at the 1st level if for any polynomial time adversary A_{II}, the advantage function $\mathrm{Adv}^{\mathrm{clpre,1\text{-}rcca}}_{\Pi_s, A_{II}}(\lambda)$ is negligible in λ.

4 Discussion on CL-PRE Scheme

In this section, we first observe and give attack to Xu et al.'s [1] scheme. Then, we show an insecure generic construction of CL-PRE, to illustrate the key point to present a RCCA secure CL-PRE scheme.

4.1 Security Analysis of Xu et al.'s Scheme

In order to leverage cloud for encryption based access control and key management, Xu et al. [1] proposed a certificateless proxy re-encryption scheme in 2012. Their scheme was claimed to be chosen plaintext secure in the random oracle model. However, the scheme is vulnerable when facing the Type I adversary.

In their scheme, the public key of user ID is $pk = (H(ID), g^{s \cdot x})$, where $H(\cdot)$ is a hash function, s is the master secret key and x is chosen by the user. The encryption algorithm is $C = (g^r, m \cdot e(H(ID)^r, g^{s \cdot x}))$. If the Type I adversary replaces $pk = (H(ID), g^{s \cdot x})$ with $pk = (H(ID), g^t)$, where t is selected on the adversary's choice, the ciphertext would be $C = (C_1, C_2) = (g^r, m \cdot e(H(ID)^r, g^t))$. Consequently, the adversary can successfully decrypt the ciphertext with t by computing $m = m \cdot e(H(ID)^r, g^t)/e(H(ID), g^r)^t = C_2/e(H(ID), C_1)^t$. The Type I adversary breaks the CPA security of Xu et al.'s scheme.

4.2 An Extension of a Generic Construction Is Vulnerable

Libert et al. [14] proposed a generic construction from a CPA secure PKE scheme and a CPA secure IBE sheme to a CCA CL-PKE scheme. Intuitively, can we directly combine a CCA PRE scheme and a CCA IB-PRE scheme to obtain a RCCA secure CL-PRE scheme by using their technique? Unfortunately, we find the resulting scheme is vulnerable to the Type I adversary. We will present a Type I attack after the description of the generic scheme.

Let Π^{I} be a CCA secure IB-PRE scheme and Π^{P} denote a CCA secure PRE scheme. Using a CCA secure Π^{P} and a CCA secure Π^{I} as building blocks, we construct a CL-PRE scheme Π by Libert et al.'s generic construction technique [14] as follows:

- The key generation algorithm for Π is to run the key generation algorithms $\Pi^{\mathrm{P}}.\mathrm{KeyGen}$ of Π^{P} and $\Pi^{\mathrm{I}}.\mathrm{KeyExtract}$ of Π^{I}. Return $SK = (SK^{\mathrm{P}}, SK^{\mathrm{I}})$ and $PK = PK^{\mathrm{P}}$.
- The re-encryption key generation algorithm for Π is to run both the re-encryption key generation algorithms of Π^{P} and Π^{I}. Return $(rk_1, rk_2) = (rk^{\mathrm{P}}, rk^{\mathrm{I}})$.
- The second level encryption algorithm for Π first split a plaintext m into $m = m_1 \oplus m_2$. Run the second level encryption of Π^{I} and Π^{P} to generate ciphertexts $C_1 = \mathcal{E}^{\mathrm{P}}_{PK}(m_1||\sigma, H(m||\sigma||pk||ID))$ and $C_2 = \mathcal{E}^{\mathrm{I}}_{ID}(m_2||\sigma, H(m||\sigma||pk||ID))$. Return $C = (C_1, C_2)$.

- The second level decryption algorithm Π with input $C = (C_1, C_2)$ runs $\Pi^{\mathrm{P}}.\mathrm{Dec}_2$ with C_1 and runs $\Pi^{\mathrm{I}}.\mathrm{Dec}_2$ with C_2. If the result is m_1 and m_2, compute $m = m_1 \oplus m_2$ and return m.
- The re-encryption algorithm for Π with $C = (C_1, C_2)$ runs $\Pi^{\mathrm{P}}.\mathrm{ReEnc}$ with C_1 to obtain re-encrypted ciphertext C_1' and runs $\Pi^{\mathrm{I}}.\mathrm{ReEnc}$ with C_2 to obtain re-encrypted ciphertext C_2'. Return $C' = (C_1', C_2')$.
- The first level decryption algorithm for Π with $C' = (C_1', C_2')$ as input runs $\Pi^{\mathrm{P}}.\mathrm{Dec}_1$ with C_1' to obtain the plaintext m_1 and runs $\Pi^{\mathrm{I}}.\mathrm{Dec}_1$ with C_2' to obtain the plaintext m_2. Compute $m = m_1 \oplus m_2$ and return m.

If we just consider the key generation algorithm, the encryption algorithm and the decryption algorithm of Π, the resulting scheme $\Pi' = (\Pi.\mathrm{KeyGen}, \Pi.\ \mathrm{Enc}_2, \Pi.\mathrm{DEC}_2)$ is a CCA CL-PKE according to Libert et al.'s result [14]. However, Π is an insecure CL-PRE scheme against the Type I adversary. We show that the Type I adversary can break the 2nd level RCCA security of Π as follows:

1. After receiving the challenge ciphertext $C^* = (C_1^*, C_2^*)$, the adversary first queries the partial private key of ID^*, namely $SK_{ID^*}^{\mathrm{I}}$ of ID^*. $\mathcal{A}$ decrypts C_2^* with $SK_{ID^*}^{\mathrm{I}}$ and obtains m_2.
2. The adversary replaces an honest user ID's public key with $\widehat{PK}$ on his choice. Note that he knows the corresponding secret value, i.e. $\widehat{SK}^{\mathrm{P}}$.
3. The adversary queries the re-encryption of C^* from ID^* to ID, and obtains $C' = (C_1', C_2')$, where $C_1' = \Pi^{\mathrm{P}}.\mathrm{ReEnc}(C_1^*)$. With the secret value $\widehat{SK}^{\mathrm{P}}$, he can easily decrypt C_1' and obtain m_1.
4. The adversary computes $m = m_1 \oplus m_2$ and breaks the RCCA security of Π.

A CCA IB-PRE plus a CCA PRE can not trivially make a RCCA CL-PRE using Libert et al.'s generic construction technique [14]. Why not? Let us have a look at the re-encryption keys first. A delegator's private key has two parts, one part is his partial private key, and the other part is the secret value. When the delegator generates a re-encryption key, he should insert his private key into the re-encryption key. Unfortunately, we find that rk_2 is only relevant to his partial private key, and it has no relation with the secret value, while rk_1 is just the reverse. This kind of construction destroys the bindings of delegator's identity and public key. Such weakness in the re-encryption key generation directly results in vulnerability of the scheme.

Informally speaking, the re-encryption key of CL-PRE should integrate the receiver's public key and identity tightly to achieve the RCCA security notion. We will present an efficient solution to this problem in the next section.

5 Replayable CCA Secure CL-PRE Scheme

In this section, we extend Sun et al's scheme [13] to the pairing based setting and construct the first RCCA secure CL-PRE. In order to achieve the RCCA security notion, we derive the re-encryption key in a manner somewhat like that in [23].

5.1 Construction

$\mathrm{Setup}(\lambda)$: Let λ be the security parameter, $\mathbb{G}$ and $\mathbb{G}_T$ be groups of prime order p, and e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map . It then performs as follows:

1. Choose a group generator $g \in \mathbb{G}$.
2. Select $x, \alpha \in \mathbb{Z}_p$ at random and set $y = g^x$.
3. Choose target collision resistant hash functions $H_0 : \{0,1\}^* \to \mathbb{G}$, $H_1 : \{0,1\}^* \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_p$, $H_2 : \{0,1\}^{l_0} \times \{0,1\}^{l_1} \to \mathbb{Z}_p$, $H_3 : \mathbb{G} \times \mathbb{G} \times \mathbb{G} \to \mathbb{G}$, $H_4 : \mathbb{G}_T \to \{0,1\}^l$ and $H_5 : \{0,1\}^* \times \mathbb{G} \times \mathbb{G} \to \mathbb{G}$, where $l = l_0 + l_1 \in \mathbb{N}$. Here, l_0 and l_1 denote the bit-length of a plaintext and a random bit string.

The public parameters are $param = (p, g, y, e(g,g)^\alpha, H_0, H_1, H_2, H_3, H_4, H_5)$. The master secret key is (x, g^α). The plaintext space is $\{0,1\}^{l_0}$.

$\mathrm{UserKeyGen}(param, ID)$: Pick $z \in \mathbb{Z}_p$ at random and compute $\mu = g^z$. Return user's key $(sk, pk) = (z, \mu)$.

$\mathrm{PKeyExt}(param, msk, ID, pk)$: Pick $s, s' \in \mathbb{Z}_p$ at random and compute $\omega = g^s$, $t = s + xH_1(ID, pk, \omega)$, $K = g^\alpha H_5(ID, pk, \omega)^{s'}$ and $L = g^{s'}$. Return the partial public key $ppk = (\omega, K, L)$ and the partial private key $psk = t$.

$\mathrm{KeyGen}(param, ID, psk, ppk, sk, pk)$: Set public key $PK = (\mu, \omega, K, L)$ and private key $SK = sk + psk = z + t$. Return (PK, SK).

$\mathrm{ReKeyGen}(param, ID_i, SK_i, PK_i, ID_j, PK_j)$: On input ID_i, SK_i, PK_i and ID_j, PK_j, this algorithm generates the re-encryption key $rk_{i \to j}$ as follows:

1. Parse PK_j as $(\mu_j, \omega_j, K_j, L_j)$.
2. Check whether $e(K_j, g) = e(g,g)^\alpha e(H_5(ID_j, \mu_j, \omega_j), L_j)$. If not, return "$\perp$".
3. Select $\theta \in \mathbb{Z}_p$ at random.
4. Compute $A_{ij} = (\mu_j \omega_j y^{H_1(ID_j, \mu_j, \omega_j)})^{SK_i^{-1}} H_0(ID_i)^\theta$ and $B_{ij} = (\mu_i \omega_i y^{H_1(ID_i, \mu_i, \omega_i)})^\theta$.
5. Return $rk_{i \to j} = (A_{ij}, B_{ij}, PK_i)$.

Note that PK_i here is corresponding with the private key SK_i.

$\mathrm{Enc}_1(param, m, ID, PK)$: On input ID, PK and a message $m \in \{0,1\}^{l_0}$, this algorithm encrypts m to a 2nd level ciphertext as follows:

1. Parse PK as (μ, ω, K, L).
2. Check whether $e(K, g) = e(g,g)^\alpha e(H_5(ID, \mu, \omega), L)$. If not, return "$\perp$".
3. Pick $\sigma \in \{0,1\}^{l_1}$ at random and compute $r = H_2(m, \sigma)$.
4. Compute $c_0 = H_4(e(g,g)^r) \oplus (m || \sigma), c_1' = e(\mu w y^{H_1(ID, \mu, \omega)}, g)^r$.
5. Return the 2nd level ciphertext $C' = (c_0, c_1')$.

$\mathrm{Enc}_2(param, m, ID, PK)$: On input ID, PK and a message $m \in \{0,1\}^{l_0}$, this algorithm encrypts m to a 1st level ciphertext as follows:

1. Parse PK as (μ, ω, K, L).
2. Check whether $e(K, g) = e(g,g)^\alpha e(H_5(ID, \mu, \omega), L)$. If not, return "$\perp$".
3. Pick $\sigma \in \{0,1\}^{l_1}$ at random, and compute $r = H_2(m, \sigma)$.
4. Compute $c_0 = H_4(e(g,g)^r) \oplus (m || \sigma), c_1 = (\mu \omega y^{H_1(ID, \mu, \omega)})^r, c_2 = H_0(ID)^r, c_3 = H_3(c_0, c_1, c_2)^r$.
5. Return the 1st level ciphertext $C = (c_0, c_1, c_2, c_3)$.

$\mathrm{ReEnc}(param, C, ID_i, ID_j, rk_{i \to j})$: On input re-encryption key $rk_{i \to j}$ and the ciphertext C of ID_i, re-encrypt the ciphertext C to ID_j as following:

1. Parse C as (c_0, c_1, c_2, c_3), and $rk_{i\to j}$ as (A_{ij}, B_{ij}, PK_i) and PK_i as $(\mu_i, \omega_i, K_i, L_i)$.
2. Check whether $e(c_3, \mu_i\omega_i y^{H_1(ID_i,\mu_i,\omega_i)}) = e(c_1, H_3(c_0, c_1, c_2))$ and $e(H_0(ID_i), c_3) = e(c_2, H_3(c_0, c_1, c_2))$. If not, return "$\perp$".
3. Compute $c_1' = e(c_1, A_{ij})/e(c_2, B_{ij})$.
4. Return the re-encrypted ciphertext $C' = (c_0, c_1')$.

$\mathrm{Dec}_1(param, C', SK)$: On input the ciphertext C', user ID's private key SK and public key PK, recover the plaintext m as follows:

1. Parse C' as (c_1', c_2), and PK as (μ, ω, K, L).
2. Compute $m||\sigma = H_4({c_1'}^{SK^{-1}}) \oplus c_0$ and $r = H_2(m||\sigma)$.
3. Check whether $c_1' = e(\mu w y^{H_1(ID,\mu,\omega)}, g)^r$. If not, return "$\perp$".
4. Return plaintext m

$\mathrm{Dec}_2(param, C, SK)$: On input the ciphertext C ,user ID's private key SK and public key PK, recover the plaintext m as:

1. Parse C as (c_0, c_1, c_2, c_3), and PK as (μ, ω, K, L).
2. Compute $m||\sigma = c_0 \oplus H_4(e(g, c_1^{SK^{-1}}))$ and $r = H_2(m||\sigma)$.
3. Check whether $c_1 = (\mu\omega y^{H_1(ID,w,\mu)})^r$, $c_2 = H_0(ID)^r$, $c_3 = H_3(c_0, c_1, c_2)^r$. If not, return "$\perp$".
4. Return plaintext m.

Correctness. To simplify the computation, we denote $\mu\omega y^{H_1(ID,\mu,\omega)}$ as $\mathcal{Y}$. Then we have $\mathcal{Y} = \mu\omega y^{H_1(ID,\mu,\omega)} = g^{SK}$. The CL-PRE scheme satisfies the correctness property at each level:

– Decryption of a 2nd level ciphertext is correct. If $C = (c_0, c_1, c_2, c_3)$ is a 2nd level ciphertext, we obtain

$$c_0 \oplus H_4(e(g, C_1^{SK^{-1}})) = H_4(e(g, g)^r) \oplus (m||\sigma) \oplus H_4(e(g, \mathcal{Y}^{r\cdot SK^{-1}})) = m||\sigma.$$

– Decryption of a 1st level ciphertext is correct. If $C' = (c_0, c_1')$ is a 1st level ciphertext, we obtain

$$H_4({c_1'}^{(SK)^{-1}}) \oplus c_0 = H_4(e(\mathcal{Y}, g)^{r\cdot SK^{-1}}) \oplus H_4(e(g, g)^r) \oplus (m||\sigma) = m||\sigma.$$

– Decryption of a re-encrypted ciphertext is correct. If $C' = (c_0, c_1')$ is a re-encrypted ciphertext of $C = (c_0, c_1, c_2, c_3)$ and $rk_{i\to j} = (A_{ij}, B_{ij}, PK_i)$ is the re-encryption key, first we obtain

$$c_1' = e(c_1, A_{ij})/e(c_2, B_{ij}) = \frac{e(\mathcal{Y}_i^r, \mathcal{Y}_j^{SK_i^{-1}} \cdot H_0(ID_i)^\theta)}{e(H_0(ID_i)^r, \mathcal{Y}_i^\theta)} = e(g, \mathcal{Y}_j)^r$$

Then, as the decryption of original ciphertext at level 1, we have $H_4({c_1'}^{SK^{-1}}) \oplus c_0 = m||\sigma$.

Remark 4. The scheme is replayable CCA secure at the second level ciphertext which is arguably sufficient for most practical applications [24]. Since a re-encryption key $rk_{*\to *} = (H_0(ID^*)^\theta, PK^{*\theta}, PK^*)$ can always be generated by picking θ at random, the adversary can re-encrypt the challenge ciphertext to ID^* itself [25], resulting in the replayable CCA security.

5.2 Discussions

In our scheme, each user has to generate a secret key using $\mathrm{UserKeyGen}$ before querying the partial public key and partial private key. This method enables us to reach a security proof. Though readers might consider that the partial keys would be independently generated from the choice of users in a certificateless scheme, we note that it is not always the case. Actually, in a survey of certificateless encryption [26,27], the authors classified certificateless schemes into three different infrastructures, namely, the AP formulation [12], the BSS formulation [13] and the LK formulation [16]. In the AP formulation, the receiver can generate the public key at anytime. While in the BSS formulation, the receiver can only generate the public key after receiving the partial private key from the KGC. In this paper, we have adopted the LK formulation for the CL-PRE scheme, namely, when generating the public key the receiver should complete a protocol with the KGC.[1] The BSS or LK formulations are the minimum requirements to achieve denial of decryption security [28] in CL-PKE.

Interestingly, Dent [27] also instantiated the LK formulation of certificateless encryption by the traditional notion of PKI-based encryption as follows: first the receiver generates encryption key pair and send it to the KGC; then the KGC creates a digital signature to bind the encryption key to his identity. The receiver's full public key contains the public key and the digital certificate. If a sender wishes to encrypt a message, he should first checks the certificate. The difference between such a certificateless scheme and a traditional public-key scheme in the PKI system is the security consideration. Interested readers may refer to [29] for a discussion on self-generated-certificate encryption versus public-key encryption.

In this paper, we adopted the Dent's instantiation to the PRE setting. But there are two main differences in our scheme 1) the full private key of a user is generated from two resources to protect the users privacy: one part is generated by the user himself and the other part related to his identity is from the KGC, and 2) the KGC creates a proof on not only the public key (generated by the user) but also an additional group element (picked up by KGC itself). The differences let us achieve a strong security without harming the efficiency, which seems optimal for a PRE scheme to the best of our knowledge. However, we also left the problem of designing a PRE in other formulation (e.g., the AP formulation) in our future work.

5.3 Security and Efficiency Comparisons

Now, we give some intuitions for the security of the scheme.

1. Public verification. Since the 2nd level ciphertext includes two short signatures, i.e. (c_1, c_3) and (c_2, c_3), everyone in the system can verify its validity. Therefore, the re-encryption algorithm will not reveal sensitive information to the adversary.

[1] The LK formulation is a reasonable relaxed formulation, since *"The Lai-Kou formulation can be viewed as a generalisation of the BSS formulation. Instead of a single message (the partial private key) being passed between the receiver and the KGC prior to public key publication, the receiver and the KGC must undertake a protocol before the receiver can publish its public key"* [27].

Table 1. Comparisons between the IB-PRE scheme in [9] and our CL-PRE scheme. $n(\cdot)$ denotes a polynomial function of the security parameter λ. $|\mathbb{G}|, |\mathbb{G}_T|, |m|$ and $|ID|$ denote the bit-length of an element in $\mathbb{G}$, an element in $\mathbb{G}_T$, a plaintext and an identifier of the user. (*: The scheme is unfortunately vulnerable to a collusion attack [30].)

Shemes	IB-PRE scheme in [9]	Our CL-PRE scheme
ReKeyGen	$1t_p$	$3t_e$
Enc_2	$3t_e$	$4t_e$
Dec_1	$1t_e$+$2t_p$	$2t_e + 1t_p$
Dec_2	$4t_e + 2t_p$	$4t_e + 1t_p$
$\|C'\|$	$\|\mathbb{G}\| + \|\mathbb{G}_T\| + \|m\| + \|n(\lambda)\| + \|ID\|$	$\|\mathbb{G}_T\| + l$
$\|C\|$	$2\|\mathbb{G}\| + \|\mathbb{G}_T\| + \|m\|$	$3\|\mathbb{G}\| + l$
Secrutiy	CCA? *	RCCA
Assumption	DBDH	1-wDBDHI&CDH
Random oracle	Yes	Yes
Other property	$\text{Dec}_1(\cdot)$ requires the identifier of the delegator.	$\text{Dec}_1(\cdot)$ does not requires the identifier of the delegator.

2. RCCA security at level 2 & level 1. Fujisaki and Okamoto [31] transformation ensures its RCCA security.

Theorem 1. *Our CL-PRE scheme is RCCA secure in the random oracle model, assuming that the CDH problem and 1-wDBDHI problem are intractable.*

The above theorem is obtained by combining of Lemma 1-4. Due to the space limit, proofs of Lemma 1- 4 will appear in the full version of this paper.

Lemma 1. *Assume $H_0, H_1, H_2, H_3, H_4, H_5$ are random oracles and the CDH problem and 1-wDBDHI problem are intractable. The CL-PRE scheme is RCCA secure at the 2nd level ciphertext against the Type I adversary.*

Lemma 2. *Assume $H_0, H_1, H_2, H_3, H_4, H_5$ are random oracles and the CDH problem and 1-wDBDHI problem are intractable. The CL-PRE scheme is RCCA secure at the 1st level ciphertext against the Type I adversary.*

Lemma 3. *Assume $H_0, H_1, H_2, H_3, H_4, H_5$ are random oracles and the CDH problem and 1-wDBDHI problem are intractable. The CL-PRE scheme is RCCA secure at the 2nd level ciphertext against the Type II adversary.*

Lemma 4. *Assume $H_0, H_1, H_2, H_3, H_4, H_5$ are random oracles and the CDH problem and 1-wDBDHI problem are intractable. The CL-PRE scheme is RCCA secure at the 1st level ciphertext against the Type II adversary.*

Efficiency. In Table 1, we compare our CL-PRE scheme with the IB-PRE scheme in [9]. t_e and t_p denote the the computation time for an exponentiation and a bilinear pairing. In our scheme, we assume $\mu\omega y^{H_1(ID,\mu,\omega)}$ is pre-computed and $e(K, g) = e(g, g)^{\alpha} e(H_5(ID, \mu, \omega), L)$ is pre-checked. The comparison indicates that the efficiency of our scheme is comparable with the IB-PRE scheme.

6 Conclusion

We introduced the RCCA security model for CL-PRE. We showed a vulnerable generic construction to illustrate constructing a RCCA secure scheme is nontrivially and meaningful. Finally, we presented a CL-PRE scheme and proved it to be RCCA secure in the random oracle model.

References

1. Xu, L., Wu, X., Zhang, X.: Cl-pre: a certificateless proxy re-encryption scheme for secure data sharing with public cloud. In: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security. ASIACCS 2012, pp. 87–88. ACM (2012)
2. Blaze, M., Bleumer, G., Strauss, M.J.: Divertible protocols and atomic proxy cryptography. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 127–144. Springer, Heidelberg (1998)
3. Canetti, R., Hohenberger, S.: Chosen-ciphertext secure proxy re-encryption. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 185–194. ACM (2007)
4. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Transactions on Information and System Security (TISSEC) 9, 1–30 (2006)
5. Yu, S., Wang, C., Ren, K., Lou, W.: Attribute based data sharing with attribute revocation. In: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2010, pp. 261–270. ACM (2010)
6. Li, M., Yu, S., Zheng, Y., Ren, K., Lou, W.: Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Transactions on Parallel and Distributed Systems, 131–143 (2013)
7. Libert, B., Vergnaud, D.: Unidirectional chosen-ciphertext secure proxy re-encryption. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 360–379. Springer, Heidelberg (2008)
8. Chu, C.-K., Tzeng, W.-G.: Identity-based proxy re-encryption without random oracles. In: Garay, J.A., Lenstra, A.K., Mambo, M., Peralta, R. (eds.) ISC 2007. LNCS, vol. 4779, pp. 189–202. Springer, Heidelberg (2007)
9. Green, M., Ateniese, G.: Identity-based proxy re-encryption. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 288–306. Springer, Heidelberg (2007)
10. Liang, X., Cao, Z., Lin, H., Shao, J.: Attribute based proxy re-encryption with delegating capabilities. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, pp. 276–286. ACM (2009)
11. Luo, S., Hu, J., Chen, Z.: Ciphertext policy attribute-based proxy re-encryption. In: Soriano, M., Qing, S., López, J. (eds.) ICICS 2010. LNCS, vol. 6476, pp. 401–415. Springer, Heidelberg (2010)
12. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
13. Baek, J., Safavi-Naini, R., Susilo, W.: Certificateless public key encryption without pairing. In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 134–148. Springer, Heidelberg (2005)
14. Libert, B., Quisquater, J.-J.: On constructing certificateless cryptosystems from identity based encryption. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 474–490. Springer, Heidelberg (2006)

15. Chow, S.S.M., Boyd, C., González Nieto, J.M.: Security-mediated certificateless cryptography. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 508–524. Springer, Heidelberg (2006)
16. Lai, J., Kou, W.: Self-generated-certificate public key encryption without pairing. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 476–489. Springer, Heidelberg (2007)
17. Chow, S.S.M., Weng, J., Yang, Y., Deng, R.H.: Efficient unidirectional proxy re-encryption. In: Bernstein, D.J., Lange, T. (eds.) AFRICACRYPT 2010. LNCS, vol. 6055, pp. 316–332. Springer, Heidelberg (2010)
18. Sun, Y., Zhang, F.T., Baek, J.: Strongly secure certificateless public key encryption without pairing. In: Bao, F., Ling, S., Okamoto, T., Wang, H., Xing, C. (eds.) CANS 2007. LNCS, vol. 4856, pp. 194–208. Springer, Heidelberg (2007)
19. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
20. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
21. Libert, B., Vergnaud, D.: Unidirectional chosen-ciphertext secure proxy re-encryption. IEEE Transactions on Information Theory 57, 1786–1802 (2011)
22. Hanaoka, G., Kawai, Y., Kunihiro, N., Matsuda, T., Weng, J., Zhang, R., Zhao, Y.: Generic construction of chosen ciphertext secure proxy re-encryption. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 349–364. Springer, Heidelberg (2012)
23. Chu, C.-K., Weng, J., Chow, S.S.M., Zhou, J., Deng, R.H.: Conditional proxy broadcast re-encryption. In: Boyd, C., González Nieto, J. (eds.) ACISP 2009. LNCS, vol. 5594, pp. 327–342. Springer, Heidelberg (2009)
24. Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 565–582. Springer, Heidelberg (2003)
25. Isshiki, T., Nguyen, M.H., Tanaka, K.: Proxy re-encryption in a stronger security model extended from CT-RSA2012. In: Dawson, E. (ed.) CT-RSA 2013. LNCS, vol. 7779, pp. 277–292. Springer, Heidelberg (2013)
26. Dent, A.W.: A survey of certificateless encryption schemes and security models. International Journal of Information Security 7, 349–377 (2008)
27. Dent, A.W.: A brief introduction to certificateless encryption schemes and their infrastructures. In: Martinelli, F., Preneel, B. (eds.) EuroPKI 2009. LNCS, vol. 6391, pp. 1–16. Springer, Heidelberg (2010)
28. Liu, J.K., Au, M.H., Susilo, W.: Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model. In: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, pp. 273–283. ACM (2007)
29. Chow, S.S.M.: Certificateless Encryption. In: Identity-Based Cryptography. IOS, pp. 135–155 (2008)
30. Koo, W.K., Hwang, J.Y., Lee, D.H.: Security vulnerability in a non-interactive id-based proxy re-encryption scheme. Information Processing Letters 109, 1260–1262 (2009)
31. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)