On the security of two multi-use CCA-secure proxy re-encryption schemes

Jindan Zhang

Department of Electronic Information,
Xianyang Vocational Technical College,
Xianyang, Shaanxi Province, 712000, China
E-mail: 69957106@qq.com

Xu An Wang*, Yi Ding and Xiaoyuan Yang

Key Laboratory of Information and Network Security,
Engineering University of Chinese Armed Police Force,
Xi'an, Shaanxi Province, 710086,China
E-mail: wangxazjd@163.com
E-mail: 624337493@qq.com
E-mail: xyyangwj@163.com
*Corresponding author

Abstract: In proxy re-encryption (PRE), a semi-trusted proxy can convert a ciphertext originally intended for Alice into one which can be decrypted by Bob, while the proxy cannot know the corresponding plaintext. PRE can be classified as *single-use* PRE and *multi-use* PRE according to the times the ciphertext can be transformed. In multi-use PRE schemes, the ciphertext can be transformed from A to B and to C and so on. In CCS'09 (post session), Wang et al. proposed a multi-use unidirectional CCA-secure proxy re-encryption scheme. Unfortunately, we show their proposal is not CCA-secure in the corresponding security models by giving concrete attacks. In 2010, Ren et al. proposed a hierarchical identity-based proxy re-encryption scheme without random oracles, and claimed their scheme was also multi-use and CCA-secure, we also show their scheme is not secure.

Keywords: proxy re-encryption; PRE; multi-use; CCA security; attack.

Reference to this paper should be made as follows: Zhang, J., Wang, X.A., Ding, Y. and Yang, X. (2013) 'On the security of two multi-use CCA-secure proxy re-encryption schemes', *Int. J. Intelligent Information and Database Systems*, Vol. 7, No. 5, pp.422–440.

Biographical notes: Jindan Zhang obtained her Master degree from University of Shaanxi Science and Technology. Currently, she is a Lecturer in Xianyang Vocational Technical College. Her main research interests include cryptography, and information hiding.

Xu An Wang obtained his Master degree from Engineering University of Chinese Armed Police Force. Currently, he is an Associate Professor in the same university. His main research interests include public key cryptography and information security.

Yi Ding is currently an Associate Professor in the Engineering University of Chinese Armed Police Force. Her main research interest is information security.

Xiaoyuan Yang obtained his Master and Bachelor degree from Xidian University. Currently, he is a Professor in the Engineering University of Chinese Armed Police Force. His main research interest is information security.

This paper is a revised and expanded version of a paper entitled 'On the security of a multi-use CCA-secure proxy re-encryption scheme' presented at Incos2012 Conference, Bucharest, Romania, 19–21 September 2012.

1 Introduction

Proxy re-encryption (PRE) allows a semi-trusted proxy transform the ciphertext for the delegator Alice to the one for the delegatee Bob, without the proxy knowing the corresponding plaintext. According to the direction of transformation, PRE can be classified as *bidirectional* PRE and *unidirectional* PRE. In bidirectional PRE schemes, the proxy can transform from Alice to Bob and vice versa. While in uni-directional PRE schemes, the proxy can only transform in one direction. According to the times the ciphertext can be transformed, PRE can be classified as *single-use* PRE and *multi-use* PRE. In single-use PRE schemes, the ciphertext can only be transformed from Alice to Bob. While in multi-use PRE schemes, the ciphertext can be transformed from Alice to Bob and to Charlie and so on.

CCA-secure PRE

In Eurocrypt'98, Blaze et al. (1998) introduced the concept of PRE. It was recently investigated by Ateniese et al. (2005, 2006). They proposed the first construction of unidirectional PRE and demonstrated several applications of PRE. However, their schemes can only achieve chosen plaintext security. It is important to achieve chosen ciphertext security (CCA-security) for many practical applications. However, due to the transformation property, it is difficult to achieve CCA-security for PRE. The adversary can launch the chosen ciphertext attack on PRE in many ways: such as

1 query to the delegator's decryption oracle

2 query to the delegatee's decryption oracle

3 query to the re-encryption oracle

4 query to the re-encryption key generation oracle, etc.

In CCS'07, the first CCA-secure bidirectional PRE was proposed by Canetti and Hohenberger (2007). To achieve CCA security, Canetti and Hohenberger used the CHK paradigm proposed by Canetti et al. (2003). However, their schemes suffer from collusion attacks. They left how to construct a CCA-secure unidirectional PRE scheme in the standard model and a multi-use unidirectional PRE scheme as open problems. In PKC'08, Libert and Vergnaud (2008) proposed the first unidirectional PRE scheme which is replayable chosen ciphertext attack (RCCA) secure and collusion resistant in the standard model. They left an important open problem, that is, how to construct a CCA-secure unidirectional PRE scheme in the standard model. Recently, this open problem was solved by Weng et al. (2010a, 2010b). Shao et al. (2009) also give a generic construction for CCA-secure PRE based on the CCA-secure threshold encryption systems. The first multi-use unidirectional PRE (IBPRE) was proposed by Green and Ateniese (2007). However, their scheme cannot achieve CCA-security.

Concretely, there are about eight PRE schemes which can achieve CCA security (there are other constructions which can achieve CCA security for PRE, but they are either generic construction or conditional PRE, here we do not consider them), here we review the constructions and underlying ideas:

- *CH scheme:* In CCS'07, the first bidirectional PRE with CCA security was proposed by Canetti and Hohenberger. They relied on the CHK transformation. Assume the ciphertext needs to be re-encrypted is $c = (X, Y)$. If the encrypter signs (X, Y) in the CHK transformation, then the proxy cannot re-encrypt (X, Y) without invalidating the signature. But if the encrypter only signs, say Y, then the adversary can arbitrarily mutate X, thus changes the decryption value. To solve this problem, they smartly add an element Z to the ciphertext, such that (Y, Z) will be signed and Z allows anyone to check that the unsigned value X was not mutated in any meaningful way. Although the scheme can be constructed without random oracle, it is a bidirectional PRE.
- *DWLC scheme:* In CANS'08, Deng et al. (2008) proposed the first IND-CCA2 secure bidirectional PRE scheme without pairings. They smartly integrated an CCA secure hashed Elgamal encryption and a modified Schnnor's signature into a PRE scheme, while achieving IND-CCA2 secure, which no longer follows the CHK transformation. This scheme is constructed with random oracle, it is a bidirectional PRE.
- *LV scheme:* In PKC'08, Libert and Vergnaud (2008) proposed the first IND-RCCA secure PRE scheme without random oracle. They follow the paradigm of Canetti and Hohenberger (2007). But if directly apply CHK transformation, the validity of translated ciphertext cannot be publicly checked. Thus, they randomise the re-encryption algorithm to make the re-encrypted cipher-texts publicly verifiable. Although the scheme can be constructed without random oracle and is unidirectional, it can only achieve IND-RCCA secure.
- *SC scheme:* In PKC'09, Shao and Cao (2009) proposed the first unidirectional CCA-secure PRE scheme without pairing. By using signature of knowledge to provide $\log_g^A = \log_g^B$, hence obtaining public verifiability for original ciphertexts. Furthermore, they use Fujisaki-Okamoto conversion to provide the validity check of both original ciphertexts and re-encrypted ciphertexts for the decryptor (Alice or Bob). Unfortunately, their scheme is pointed out not secure for the first level ciphertext by Weng et al. (2010a, 2010b). This scheme is constructed with random oracle.
- *CWYD scheme:* In Africacrypt'10, Chow et al. (2010) proposed new efficient CCA-secure unidirectional PRE schemes. Their design is based on ElGamal encryption and Schnorr signature, which is (arguably) simple. Their construction extends the bidirectional scheme proposed by Weng et al. (2009), again by the

token-controlled encryption technique. Although this scheme is efficient and unidirectional, it is constructed with random oracle.

- *WCYDCB scheme:* Recently, Weng et al. proposed the first CCA-secure unidirectional PRE scheme in the adaptive corruption model without random oracles, they use a technique inspired by Hohenberger and Waters' (2009) recent signatures scheme. This technique enables the challenger to successfully generate the challenge ciphertext for the adversary, even if the adversary is allowed to adaptively corrupt users.
- *SCL scheme:* Independently, Shao et al. (2010) recently proposed a CCA-secure unidirectional PRE scheme without random oracle. Actually, the encryption algorithm of their proposal is almost the same as that in Canetti and Hohenberger (2007), except that they use the technique of Kurosawa and Desmedt (2004). However, this modification provides the validity check for re-encrypted ciphertexts. Note that the Canetti-Hohenberger technique can only be used to answer decryption queries for the original ciphertext, and the Kurosawa-Desmedt cannot be used to answer decryption queries. To simulate the decryption oracle correctly, the re-encrypted ciphertext should contain the plaintext ciphertext. Nevertheless, the re-encrypted ciphertext cannot explicitly contain the original ciphertext according to the CCA security. Thus, their scheme relies on a 'twice encryption' paradigm to hide the original ciphertext. Obliviously this is not an efficient solution.

Identity-based PRE

Green and Ateniese (2007) proposed the first IBPRE schemes in ACNS'07. They defined the algorithms and security models for IBPRE, by using a variant of the efficient Dodis and Ivan (2003) key splitting approach to the settings with a bilinear map, they constructed a concrete scheme. But it was found later that it cannot resist collusion attack by Koo et al. (2009). Based on Waters' IBE, Chu and Tzeng (2007) proposed the first CCA-secure IBPRE scheme in the standard model. However, due to the structure of Waters' IBE and Green's paradigm, this scheme is not efficient. Matsuo (2007) showed PRE schemes for identity-based system in Pairing'07. They constructed an IBPRE scheme and a hybrid PRE scheme. But recently it was shown this scheme has some flaws by Wang and Yang (2010). In Inscrypt'08, Tang (2008) proposed the inter-domain identity-based PRE scheme. In SDM'08, Ibraimi et al. (2008) constructed a type-and-identity-based PRE and discussed its application in healthcare. Recently based on identity-based mediated encryption, Lai et al. (2010) gave new generic constructions on IBPRE with master secret security.

1.1 Our contribution

In CCS'09 poster session, Wang and Cao (2009) proposed a multi-use fully CCA-secure PRE schemes, but we will show this scheme is not CCA-secure. We give the first formal security models for multi-use CCA-secure PRE, then we give a concrete attack on a PRE scheme proposed by Wang and Cao (2009) in this security model. Our result show that their scheme is not CCA-secure.

Recently, Ren et al. (2010) claimed to propose a fully secure hierarchical identity-based proxy re-encryption (HIBPRE) scheme without random oracle, but we shall show that this scheme is yet not secure either.

1.2 Organisation

We organise our paper as follows. In Section 2, we first give the definition and security models of multi-use CCA-secure PRE, then we review of a PRE scheme proposed by Wang et al. and give an attack. In Section 3, we review of a multi-use HIBPRE scheme proposed by Ren et al. and its security model. and then we give some security analysis on their scheme. In the last Section 4, we give our conclusion.

2 On the security of Wang et al.'s PRE scheme

2.1 Definition and security model

Definition 2.1: A multi-use unidirectional PRE scheme is a tuple of algorithms KeyGen, ReKeyGen, Encrypt, ReEncrypt, Decrypt:

1 KeyGen(1^k) → ($params$, pk, sk): on input a security parameter k, the algorithm outputs the system's public parameters ($params$), and a public key pk and a secret key sk

2 ReKeyGen($params$, sk_i, pk_j) → $rk_{i \to j}$: on input the delegator's secret key ski and the delegatee's public key pk_j, this algorithm outputs a re-encryption key $rk_{i \to j}$

3 Encrypt($params$, pk, m) → $C_{pk}^{(1)}$: on input a public key pk and a message $m \in \{0, 1\}^n$, this algorithm outputs a first-level ciphertext $C_{pk}^{(1)}$

4 ReEncrypt($params$, $rk_{i \to j}, C_{pk_i}^{(l)} \to C_{pk_j}^{(l+1)} (l \geq 1))$: on input a re-encryption key $rk_{i \to j}$ and a l-level ciphertext $C_{pk_i}^{(l)}$ under public key pk_i, this algorithm outputs a l + 1-level ciphertext $C_{pk_j}^{(l+1)}$ under the public key pk_j or the error symbol ⊥.

5 Decrypt($params$, sk_i, $C_{pk_i}^{(l)}$): on input a secret key sk_i and a l-level ciphertext $C_{pk_i}^{(l)}$ under public key pk_i, this algorithm outputs a message $m \in \{0, 1\}^n$ or ⊥.

Roughly speaking, the correctness requires that, for all (pk_i, sk_i) ← KeyGen(1^k) and (pk_j, sk_j) ← KeyGen(1^k), the corresponding re-encryption key $rk_{id_i \to id_j}$ ← ReKeyGen, and an l^{th}-level ($l \geq 1$) ciphertext $C_{pk_i}^{(l)}$ under public key pk_i output by Encrypt or Reencrypt, it holds that

$$\text{Decrypt}\left(params, sk_i, \text{Encrypt}\left(params, pk_i, m\right)\right) = m.$$

$$\text{Decrypt}\left(params, sk_j, \text{ReEncrypt}\left(params, rk_{i \to j}, C_{pk_i}^{(l)}\right)\right) = m.$$

Definition 2.2 (CCA-security): A multi-use unidirectional PRE scheme is CCA-secure if the advantage of any PPT adversary $\mathcal{A}$ in the following game played between a challenger $\mathcal{C}$ and $\mathcal{A}$ is negligible in the security parameter k. Note that we work in the static corruption model, where the adversary should decide the corrupted users before the game starts.

1 *Setup phase:* the challenger $\mathcal{C}$ sets up the system parameters.

2 *Find phase:* the adversary $\mathcal{A}$ adaptively issues queries $q_1, \ldots, q_{n_1}$ where query q_i is one of:

- on $\mathcal{A}$'s any query of the form (*keygen*, *i*), the challenger $\mathcal{C}$ and responds by running algorithm KeyGen(1^k) to generate a key pair (pk_i, sk_i), gives pk_i to $\mathcal{A}$ and records (pk_i, sk_i) in table T_K
- on $\mathcal{A}$'s any query of the form (*corrupt*, pk_i), the challenger $\mathcal{C}$ searches pk_i in table T_K and returns sk_i; otherwise, the challenger returns $\perp$
- on $\mathcal{A}$'s any query of the form (*rekeygen*, pk_i, pk_j), the challenger $\mathcal{C}$ returns the re-encryption key $rk_{pk_i \to pk_j} = \text{ReKeyGen}(params, sk_i, pk_j)$, where sk_i is the secret key corresponding to pk_i
- on $\mathcal{A}$'s any query of the form (*reencrypt*, pk_i, pk_j, $C_{pk_i}^{(l)}$), the challenger $\mathcal{C}$ returns the re-encrypted ciphertext

  $$C_{pk_j}^{(l+1)} = \text{ReEncrypt}\left(params, \text{ReKeyGen}\left(params, sk_i, pk_j\right), C_{pk_i}^{(l)}\right)$$

  where sk_i, sk_j is the secret keys corresponding to pk_i, pk_j.
- on $\mathcal{A}$'s any query of the form (*decrpt*, pk_i, $C_{pk_i}^{(l)}$), the challenger $\mathcal{C}$ returns Decrypt(*params*, *ski*, $C_{pk_i}^{(l)}$), where sk_i is the secret key corresponding to pk_i.

3 *Challenge phase:* Once the adversary $\mathcal{A}$ decides that Find Phase is over, it outputs two equal length plaintexts m_0, m_1 from the message space, and a public key pk^* on which it wishes to challenge. There are three constraints on the public key pk^*,

a it is in the table T_K

b it is uncorrupted

c if (pk^*, ★) did appear in any query of the form (*rekeygen*, pk^*, ★), then ★ is uncorrupted.

The challenger $\mathcal{C}$ picks a random bit $b \in \{0, 1\}$ and sets c^* = Encrypt(*params*, pk^*, m_b). It sends C^* as the challenge to $\mathcal{A}$.

4 *Guess phase:* $\mathcal{A}$ continues to make queries $q_{k+1}, \ldots, q_n$ as in the Find Phase, with the following restrictions. Let $\mathcal{C}$ be a set of ciphertext/public pairs, initially containing the single pair (c^*, pk^*). For all (c, *) $\in \mathcal{C}$, and for all rk given to $\mathcal{A}$ or can be computed by $\mathcal{A}$. Let C' be the set of all possible values derived via (one or more) consecutive calls to Reencrypt:

a $\mathcal{A}$ is not permitted to launch any query of the form (*decrypt*, *pk*, *c*), where $(c, pk) \in C \cup C'$.

b $\mathcal{A}$ is not permitted to launch any query of the form (*corrupt*, *i*) or (*rkextract*, pk_i, pk_j) that would permit trivial decryption of any ciphertext in $C \cup C'$.

c $\mathcal{A}$ is not permitted to launch any query of the form (*reencrypt*, pk_i, pk_j, *c*), where $\mathcal{A}$ possesses the keys to trivially decrypt ciphertexts under pk_j and $(c, pk) \in C \cup C'$. On successful execution of any re-encryption query, let c' be the result and add the pair (c', pk_j) to the set *C*.

These queries maybe asked adaptively as in the Find Phase. At the end of this phase, $\mathcal{A}$ outputs his guess b', where $b = b'$, then $\mathcal{A}$ wins the game.

We define adversary $\mathcal{A}$'s advantage in attacking PRE as

$$Adv_{PRE}^{CCA}(k) = \left| \Pr[b = b'] - \frac{1}{2} \right|.$$

We say that a PRE scheme is secure against adaptive chosen ciphertext if for all probabilistic polynomial time algorithms $\mathcal{A}$, $Adv_{PRE}^{CCA}(k)$ is negligible with respect to *k*.

2.2 Review of a multi-use CCA-secure PRE scheme

Here we review a PRE scheme proposed by Wang and Cao (2009) in CCS'09 poster session. Let 1^k be the security parameter, $(q, g, \mathbb{G}_1, \mathbb{G}_T, e)$ be generated by a bilinear group generator on input (1^k), and $Sig = (G, S, V)$ be a strongly unforgeable signature scheme. Let g_1, h_1, h_2, and h_3 be four random elements in $\mathbb{G}_1 / \{g\}$. Further, let $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$ and $H_2 : \mathbb{G}_T \to \mathbb{G}_1$ be two one-way, collision-resistant cryptographic hash functions. The public parameters are:

$$param = (q, g, g_1, h_1, h_2, h_3, \mathbb{G}_1, \mathbb{G}_T, e, Sig, H_1, H_2)$$

Our PRE scheme consists of the following five algorithms (KeyGen, Encrypt, ReKeyGen, ReEncrypt, Decrypt):

1 KeyGen(*param*) → (*pk*, *sk*): on input *param*, select $x \in_R \mathbb{Z}_q^*$. Set

$$pk = g^x, sk = x$$

2 Encrypt(*param*, *pk*, *m*) → $C^{(1)}$: to encrypt a message $m \in \mathbb{G}_T$ under *pk*, select $r \leftarrow_R \mathbb{Z}_q^*$, then compute $C^{(1)} = (c_{1,1}, c_{1,2}, c_{1,3})$, where

$$c_{1,1} = g^r, c_{1,2} = m \cdot e(g_1, pk)^r,$$

$$c_{1,3} = \left(h_1^{H_1(c_{1,1})} h_2^{H_1(c_{1,1} \| c_{1,2})} h_3 \right)^r$$

Finally, output the first level ciphertext $C^{(1)}$.

3 ReKeyGen(*param*, sk_i, pk_j) → $rk_{i \to j}$ ($i \neq j$): to generate a re-encryption key from pk_i to pk_j for pk_i's proxy P_i, do the following:

a Select $r_i \leftarrow_R \mathbb{Z}_q^*$, $K_i \leftarrow_R \mathbb{G}_T$.

b Compute

$$R_1^{(i)} = g^{r_i},$$
$$R_2^{(i)} = K_i \cdot e\left(g_1, pk_j\right)^{r_i},$$
$$R_3^{(i)} = svk_{P_i},$$
$$R_4^{(i)} = \left(h_1^{H_1\left(R_1^{(i)}\right)} h_2^{H_1\left(R_1^{(i)} \| R_2^{(i)} \| R_3^{(i)}\right)}\right)^{r_i},$$
$$R_5^{(i)} = H_2\left(K_i\right) \cdot g_1^{-x_i}$$

where svk_{P_i} is a publicly available verification key of pk_i's proxy P_i.

c Output

$$rk_{i \to j} = \left(R_1^{(i)}, R_2^{(i)}, R_3^{(i)}, R_4^{(i)}, R_5^{(i)}\right)$$

The re-encryption key is sent to P_i via a secure channel.

4 ReEncrypt(*param*, $rk_{i \to j}$, $C_i^{(l)}$) → $\{C_j^{(i+1)}, \perp\}$ ($i \neq j$, $l \geq 1$):

- To re-encrypt a first-level ciphertext $C_i^{(l)}$, denoted by $C_i^{(1)}$, do

  a Parse $C_i^{(1)}$ as ($c_{1,1}$, $c_{1,2}$, $c_{1,3}$), and $rk_{i \to j}$ as $(R_1^{(i)}, R_2^{(i)}, R_3^{(i)}, R_4^{(i)}, R_5^{(i)})$.

  b Check if

$$\begin{aligned} e\left(g, c_{1,3}\right) &= e\left(c_{1,1}, h_1^{H_1\left(c_{1,1}\right)} h_2^{v} h_3\right) e\left(g, R_4^{(i)}\right) \\ &= e\left(R_1^{(i)}, h_1^{H_1\left(R_1^{(i)}\right)} hH_{12}^{\left(R_1^{(i)} \| R_2^{(i)} \| R_3^{(i)}\right)}\right) \end{aligned}$$

  hold. If either of them fails, return ⊥, otherwise, do the following.

  c Compute

$$C = \left(c'_{1,1}, c'_{1,2}, c'_{1,3}, c'_{2,1}, c'_{2,2}, c'_{2,3}, c'_{2,4}\right)$$

  where

$$c'_{1,1} = c_{1,1}, c'_{1,2} = c_{1,2} \cdot e\left(c_{1,1}, R_5^{(i)}\right),$$
$$c_{1,3} = c_{1,3}, c'_{2,1} = R_1^{(i)}, c'_{2,2} = R_2^{(i)},$$
$$c_{2,3} = R_3^{(i)}, c'_{2,4} = R_4^{(i)}$$

  d Let P_i be pk_i's proxy, and ssk_{P_i} be the signing key of P_i corresponding to P_i's verification key $R_3^{(i)}$.

  e Run the signing algorithm $S(ssk_{P_i}, (c'_{1,1}, c'_{1,2}, c'_{1,3}, c'_{2,1}, c'_{2,3}, c'_{2,4}))$, and denote the signature as $S_i^{(1)}$.

f Output the ciphertext $C_j^{(2)} = \langle C, S_i^{(1)} \rangle$.

- To encrypt an l^{th} level ($l \geq 1$) ciphertext $C_i^{(l)}$,

a Parse $C_i^{(l)}$ as $(c_{1,1}, \cdots, c_{l,1}, c_{l,2}, c_{l,3}, c_{l,4}, c_{l,5})$, and $rk_{i \to j}$ as $(R_1^{(i)}, R_2^{(i)}, R_3^{(i)}, R_4^{(i)}, R_5^{(i)})$.

b Check if

$$\begin{aligned} e(g, c_{l,3}) &= e\left(c_{l,1}, h_1^{H_1(c_{l,1})} h_2^{H_1(c_{l,1} \| c_{l,2} \| c_{l,3})}\right) e\left(g, R_4^{(i)}\right) \\ &= e\left(R_1^{(i)}, h_1^{H_1\left(R_1^{(i)}\right)} h_2^{H_1\left(R_1^{(i)} \| R_2^{(i)} \| R_3^{(i)}\right)}\right) \end{aligned}$$

hold. If either of them fails, return ⊥, otherwise, do the following.

c For $\forall k \in [2, l]$, check

$$V\left(c_{k,3}, c_{k,5}, \left(c_{1,1}, \cdots, c_{k,1}, c_{k,3}, c_{k,4}\right)\right) = 1$$

whenever one of them fails, return ⊥. Otherwise, do the following:

d Compute

$$C = \left(c'_{1,1}, \cdots, c'_{l,1}, c'_{l,2}, \cdots, c'_{l,5}, c'_{l+1,1}, c'_{l+1,2}, c'_{l+1,3}, c'_{l+1,4}\right)$$

where

$$c'_{l,2} = c_{l,2} \cdot e\left(c_{l,1}, R_5^{(i)}\right), c'_{l+1,1} = R_1^{(i)},$$
$$c'_{l+1,2} = R_2^{(i)}, c'_{l+1,3} = R_3^{(i)}, c'_{l+1,4} = R_4^{(i)})$$

and all other elements remain unchanged.

e Let P_i be pk_i's proxy, and ssk_{P_i} be the signing key of P_i corresponding to P_i's verification key $R_3^{(i)}$.

f Run the signing algorithm $S(ssk_{P_i}, (c'_{1,1}, \ldots, c'_{l+1,1}, c'_{l+1,3}, c'_{l+1,4}))$ to generate a signature on the ciphertext tuple $(c'_{1,1}, \cdots, c'_{l+1,1}, c'_{l+1,3}, c'_{l+1,4})$, and denote the signature as $S_i^{(l)}$.

g Output the ciphertext $C_j^{(l+1)} = \langle C, S_i^{(l)} \rangle$.

5 Decrypt(*param*, *ski*, $C_i^{(l)}$) $\to \{m, \perp\}$ ($l \geq 1$): if $C_i^{(l)}$ cannot be parsed as ($c_{1,1}$, $c_{1,2}$, $c_{1,3}$) for a first-level ciphertext, or ($c_{1,1}$,⋯,$c_{l,1}$,⋯,$c_{l,5}$) for an l^{th}-level ciphertext ($l \geq 1$), then return ⊥. Otherwise, continue the following process:

- For a first level ciphertext,

a Verify that

$$e(g, c_{1,3}) = e\left(c_{1,1}, h_1^{H_1(c_{1,1})} h_2^{H_1(c_{1,1} \| c_{1,2})} h_3\right)$$

If not, return ⊥.

b Otherwise, compute

$$m \leftarrow \frac{c_{1,2}}{e\left(c_{1,1}, g_1^{sk_i}\right)}$$

c Output m.

- For an l^{th}-level ciphertext ($l \geq 1$),

a Check if

$$e\left(g, c_{l,4}\right) = e\left(c_{l,1}, h_1^{H_1\left(c_{l,1}\right)} h_2^{H_1\left(c_{l,1} \| c_{l,2} \| c_{l,3}\right)}\right)$$

If not, return $\perp$. Otherwise,

b For $\forall k \in [2, l]$, check

$$V\left(c_{k,3}, c_{k,5}, \left(c_{1,1}, \cdots, c_{k,1}, c_{k,3}, c_{k,4}\right)\right) = 1$$

whenever one of them fails, output $\perp$. Otherwise do the following:

c Compute

$$K_{l-1} \leftarrow \frac{c_{l,2}}{e\left(c_{l,1}, g_1^{sk_i}\right)}$$

d For i from $l - 2$ down to 1, compute

$$K_i \leftarrow \frac{c_{i+1,2}}{e\left(c_{i+1,1}, H_2\left(K_{i+1}\right)\right)}$$

e Compute

$$m \leftarrow \frac{c_{1,2}}{e\left(c_{1,1}, H_2\left(K_1\right)\right)}$$

f Output m.

2.3 *Our attack*

The authors claimed that their multi-use PRE scheme is CCA secure. However, in this section, we show that this is not true. Concretely, there exists a polynomial time adversary $\mathcal{A}$ who has non-negligible advantage against the CCA security of this multi-use PRE scheme. Adversary $\mathcal{A}$ works as follows:

1 In *Setup Phase*, adversary $\mathcal{A}$ obtains the public parameters *params* from the challenger $\mathcal{C}$.

2 In *Find Phase*, adversary $\mathcal{A}$ needs not issue any queries.

3 In *Challenge Phase*, adversary $\mathcal{A}$ returns a challenged public key $pk^* = gx^*$, and two equal-length plaintexts m_0, m_1. Then challenger $\mathcal{C}$ picks $b \in_R \{0, 1\}$, sets the challenge ciphertext to $c^* = \text{Encrypt}(pk^*, m_b))$, and gives c^* to $\mathcal{A}$. Recall that $\mathcal{A}$'s goal is to correctly guess the value b. Note here

$$c^* = \left(c_{1,1}^* = g^r, c_{1,2}^* = m_b \cdot e\left(g_1, pk^*\right)^r, c_{1,3} = \left(h_1^{H_1(c_{1,1})} h_2^{H_1(c_{1,1} \| c_{1,2})} h_3 \right)^r \right).$$

4 In *Guess Phase*, adversary $\mathcal{A}$ does as the follows:

a First he issues a query of the form (*rekeygen*, *pk**, *pk*$_j$), and he will get

$$R_1^{(*)} = g^{r_1}, R_2^{(*)} = K \cdot e\left(g_1, pk_j\right)^{r_1},$$

$$R_3^{(*)} = svk_P,$$

$$R_4^{(*)} = \left(h_1^{H_1\left(R_1^{(*)}\right)} h_2^{H_1\left(R_1^{(*)} \| R_2^{(*)} \| R_3^{(*)}\right)} \right)^{r_1},$$

$$R_5^{(*)} = H_2(K) \cdot g_1^{-x^*}$$

where r_1, K are randomly selected by *pk** from $\mathbb{Z}_q^*, \mathbb{G}_T$.

b Now the adversary $\mathcal{A}$ is actually the proxy *P* between *pk** and *pk*$_j$, then it compute a 'valid' ciphertext for *pk*$_j$ as follows:

$$c_j = \left(c_{1,1}', c_{1,2}', c_{1,3}', c_{2,1}', c_{2,2}', c_{2,3}', c_{2,4}' \right).$$

where

$$c_{1,1}' = c_{1,1} \cdot g^t = g^{r+t},$$

$$c_{1,3}' = c_{1,3} = \left(h_1^{H_1(c_{1,1})} h_2^{H_1\left(c_{1,1}^* \| c_{1,2}^*\right)} h_3 \right)^r,$$

$$\begin{aligned} c_{1,2}' = c_{1,2} &= e\left(g_1^t, p_k^*\right) \cdot e\left(c_{1,1}^* \cdot g^t, R_5^*\right) \\ &= \left(m_b \cdot e\left(g_1, p_k^*\right)^{r+t} \cdot e\left(g^{r+t}, H_2(K) g_1^{-x^*}\right)\right) \\ &= m_b e\left(g^{r+t}, H_2(K)\right) \end{aligned}$$

$$c_{2,1}' = R_1^{(*)} = g^{r_1}, c_{2,2}' = R_2^{(*)} = K \cdot e\left(g_1, pk_j\right)^{r_1},$$

$$c_{2,3} = R_3^{(*)} = svk_P,$$

$$c_{2,4}' = R_4^{(*)} = \left(h_1^{H_1\left(R_1^{(*)}\right)} h_2^{H_1\left(R_1^{(*)} \| R_2^{(*)} \| R_3^{(*)}\right)} \right)^{r_1}.$$

where t is randomly chosen from $\mathbb{Z}_q^*$.

c Run the signing algorithm $S(ssk_P, (c_{1,1}', c_{1,2}', c_{1,3}', c_{2,1}', c_{2,3}', c_{2,4}'))$, and denote the signature as $c_{2,5}' = S$ and the transformed ciphertext as

$$C_j = \left(c_j, S\right) = \left(c_{1,1}', c_{1,2}', c_{1,3}', c_{2,1}', c_{2,3}', c_{2,4}', c_{2,5}' \right)$$

Note this C_j is different from the C_j^*, which is the valid re-encryption result returning from the re-encryption oracle with input (*reencrypt*, *pk**, *pk*$_j$, *c**).

d Then the adversary $\mathcal{A}$ issues a query of the form (*decrypt*, *pk*$_j$, *C*$_j$), and he will get m_b as follows

1. The challenger $\mathcal{C}$ checks

   $$e\left(g, c'_{2,4}\right) = e\left(c'_{2,1}, h_1^{H_1\left(c'_{2,1}\right)} h_2^{H_1\left(c'_{2,1} \| c'_{2,2} \| c'_{2,3}\right)}\right)$$

   holding, then he goto the next step,
2. Check

   $$V\left(c'_{2,3}, c'_{2,5}, \left(c'_{1,1}, \cdots, c'_{2,1}, c'_{2,3}, c'_{2,4}\right)\right) = 1$$

   holding, then he goto the next step,
3. Compute

   $$K = \frac{c'_{2,2}}{e\left(c'_{2,1}, g_1^{sk_j}\right)}$$
4. Compute

   $$m = \frac{c'_{1,2}}{e\left(c'_{1,1}, H_2(K)\right)} = \frac{m_b e\left(g^{r+t}, H_2(K)\right)}{e\left(g^{r+t}, H_2(k)\right)} = m_b$$

Now, knowing the plaintext m_b, adversary $\mathcal{A}$ can certainly know the underlying bit b chosen by the challenger in *Challenge Phase*, and hence always wins the game.

Note here we just attack the single-hop variant of their scheme, it can be easily extended to the multi-hop variants. And this attack is allowed in the above security model. The adversary $\mathcal{A}$ actually is a corrupted proxy and it makes use of the decryption oracle of the delegatee.

3 On the security of Ren et al.'s HIBPRE scheme

3.1 Definition and security model

The definition refers to the notion of an encryption level as an implicit property of a ciphertext. A ciphertext generated directly using the Encrypt algorithm is termed as a level-1 ciphertext. Applying the Re-encrypt algorithm to a level-*n* ciphertext results in a level-(n + 1) ciphertext.

Definition 3.1: A non-interactive HIBPRE scheme is a tuple of algorithms as follows:

1. Setup(1^k): a probabilistic algorithm takes a security parameter k as input, output the public parameters (*params*) which are distributed to users, and the master secret key (*msk*) which is kept private.
2. KeyGen(*params*, *msk*, ID = (ID_1, ID_2,···,ID_i)): on input an identity $ID \in (Z_p^*)^i$ and the master secret key, output a decryption key d_{ID} corresponding to that identity. The private key can also be generated by its parent (ID_1, ID_2,···,ID^{i-1}).
3. Encrypt(*params*, *ID*, *m*): on input an identity *ID*, and a plaintext $m \in G_2$, output the ciphertext c_{ID}.

4 RKGen(*params*, d_{ID}, ID, ID′): on input a secret key d_{ID} and identities (ID, ID'), output a re-encryption key $rk_{ID\to ID'}$.

5 Re-encrypt(*params*, $rk_{ID\to ID'}$, c_{ID}): on input a ciphertext c_{ID} under identity ID, and a re-encryption key $rk_{ID\to ID'}$, output a re-encrypted ciphertext $c_{ID'}$.

6 Decrypt(*params*, d_{ID}, c_{ID}): on input a ciphertext c_{ID} and a secret key d_{ID}, output a plaintext or an error message.

Correctness: Let $c_{ID} \leftarrow Encrypt(params, ID, m)$ be a correct ciphertext, then the following propositions hold:

1 Decrypt(*params*, d_{ID}, c_{ID}) = m.

2 Decrypt(*params*, $d_{ID'}$, Re-encrypt(*params*, $rk_{ID\to ID'}$, c_{ID})) = m,
where $rk_{ID\to ID'} = RKGen(params, d_{ID}, ID, ID')$, $d_{ID} = KeyGen(params, msk, ID)$,
$d_{ID'} = KeyGen(params, msk, ID')$.

Now we give the security model for HIBPRE.

Definition 3.2: Semantic security of an HIBPRE scheme against an adaptive chosen ciphertext attack (IND-PrID-CCA2) is defined according to the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$.

1 *Setup:* $\mathcal{B}$ runs the Setup algorithm and gives $\mathcal{A}$ the resulting system parameters *params*, keeping the master key *msk* to itself

2 *Phase 1:* $\mathcal{A}$ adaptively issues a set of queries as below:
   - Extract query (ID): $\mathcal{B}$ returns KeyGen(*params*, *msk*, *ID*) to $\mathcal{A}$
   - Rkextract query (ID, ID'): $\mathcal{B}$ runs KeyGen(*params*, *msk*, *ID*) to obtain d_{ID}, and returns RKGen(*params*, d_{ID}, ID, ID') to $\mathcal{A}$.
   - Decrypt query (ID, c): $\mathcal{B}$ runs KeyGen(*params*, *msk*, *ID*) to obtain d_{ID}, and returns Decrypt(*params*, d_{ID}, c) to $\mathcal{A}$.

Challenge: Once $\mathcal{A}$ decides that Phase 1 is over, it outputs an identity ID^*, where $ID^* = (ID_1^*, \cdots, ID_i^*)$, and two equal length plaintext m_0, m_1 on which it wishes to be challenged. $\mathcal{B}$ selects $w \in \{0, 1\}$ and returns $c_{ID^*} = Encrypt(params, ID^*, m_w)$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ adaptively issues queries as Phase 1, and $\mathcal{B}$ answers these queries in the same way as Phase 1.

Guess: $\mathcal{A}$ submits a guess $w' \in \{0, 1\}$, and wins the game if $w = w'$ and the following queries are not allowed:

- Extract query (ID* or its prefix)
- Rkextract query (ID* or its prefix, ID') and Extract query (ID' or its prefix) for any identity ID'
- Rkextract query (ID* or its prefix, ID'), and Decrypt query (ID', $c_{ID'}$) for any identity ID' and any ciphertext $c_{ID'}$

- Re-encrypt query (ID*, ID', c_{ID*}) and Extract query (ID' or its prefix) for any identity ID'
- Decrypt query (ID*; c^*_{ID})
- Decrypt query (ID', $c_{ID'}$) for any identity ID', where c$_{ID'}$ = Reencrypt($params$, $rk_{ID^* \to ID'}$, c_{ID^*})

We call an adversary $\mathcal{A}$ in the above game an IND-PrID-CCA2 adversary. The advantage of $\mathcal{A}$ is defined as | $Pr[w = w'] - 1 / 2$.

Definition 3.3: An HIBPRE system is (t, ϵ, q) IND-PrID-CCA2 secure if all t-time IND-PrID-CCA2 adversaries making at most q queries have advantage of at most ϵ in the above game.

3.2 Review of a multi-use CCA-secure HIBPRE scheme

1 *Setup:* let p be a large prime number, G_1, G_2 are groups of order p. $e : G_1 \times G_1 \to G_2$ is a bilinear map, g is a generator of G_1, $g_1 = g^\alpha$, where $\alpha \in Z_p^*$. l is the maximum number of levels in the HIBPRE, $h : G_2 \to G_1$ and $H : G_1^2 \times G_2^l \to Z_p^*$ are collision resistent hash functions, where $I \in Z_p^*$. The PKG randomly choose g_2, g_3, $h_i \in G_1$, $i = 0, 1 \cdots, l + 1$ and $f(x) = ax + b$, where $a, b \in Z_p^*$. If $g_2 = g_3^{-a}$ or $h_0 = g_3^{-b}$, choose another $f(x)$ again. The public parameters are

$$params = \left(g, g_1, g_2, g_3, f(x), h, H, h_i \left(i = 0, 1, \cdots, l+1\right)\right)$$

α is the private key of PKG.

2 *KeyGen:* to a user U with identity $\text{ID} = (ID_1, \cdots, ID_i) \in Z_p^i$, the PKG randomly choose $r_{-1,i}, r_{0,i} \in Z_p^*$, and computes

$$d_{0,i} = \left(h_0 g_2^{r-1,i} g_3^{f(r-1,i)}\right)\alpha \cdot \left(h_{l+1} \prod_{k=1}^{i} h_k^{ID_k} \right)^{r_{0,i}}, d_{-1,i} = r_{-1,i}, d_{1,i} = g^{r_{0,i}},$$

$$d_{i+1,i} = h_{i+1}^{r_{0,i}}, \cdots, d_{l,i} = h_l^{r_{0,i}}$$

so the private key of U is $d_{ID} = (d_{0,i}, d_{-1,i}, d_{1,i}, d_{i+1,i}, \cdots, d_{l,i})$. If $h_0 g_2^{r_{-1,i}} g_3^{f(r_{-1,i})} = 1$, randomly choose $r_{-1,i}$ again.

The private key for (ID_1, ID_2,$\cdots$,ID_i) can also be generated by its parent (ID_1, ID_2,$\cdots$,ID_{i-1}) having the secret key $K(d_{0,i-1}, d_{-1,i-1}, d_{1,i-1}, d_{i,i-1}, \cdots, d_{l,i-1})$. It computes:

$$d_{0,i} = d_{0,i-1} \cdot d_{l,i-1} \cdot d_{i,i-1}^{ID_i} \cdot \left(h_{l+1} \prod_{k=1}^{i} h_k^{ID_k} \right)^{t}, d_{-1,i} = d_{-1,i-1}, d_{1,i} = d_{1,i-1} \cdot g^t,$$

$$d_{k,i} = d_{k,i-1} \cdot h_k^t (k = i+1, \cdots, l)$$

where $r_{0,i} = r_{0,i-1} + t$ and t is randomly chosen from Z_p^*.

3 *Encrypt:* to encrypt a message $m \in G_2$ for the user with identity ID = $(ID_1,\cdots,ID_i)$, randomly choose $s \in Z_p^*$ and compute

$$c_1 = \left(h_{l+1} \prod_{k=1}^{i} h_k^{ID_k} \right)^s, c_2 = g^s, c_3 = e(g_1, g_2)^s, c_4 = e(g_1, g_3)^s,$$

$$c_5 = m \cdot e(g_1, h_0)^{s+\gamma}, c_6 = H\left(c_1, c_2, c_3, c_4, c_5, m \cdot e(g_1, h_0)^s\right)$$

where $\gamma = h(c_1, c_2, c_3, c_4, e(g_1, h_0)^s)$. The ciphertext of message m is $c_{ID} = (c_1, c_2, c_3, c_4, c_5, c_6)$.

4 RKGen: let $d_{ID} = (d_{0,i}, d_{-1,i}, d_{1,i}, d_{i+1,i},\cdots,d_{l,i})$. To compute a re-encryption key from ID to ID′, randomly choose $X \in G_2$ and compute

$$N = (N_1, N_2, \cdots, N_6) = Encrypt(params, ID', X), d_0' = d_{0,i}h(X)$$

So $rk_{ID \to ID'} = (N, d_0', d_{-1,i}, d_{1,i})$.

5 Re-encrypt: To re-encrypt a level-n ciphertext from ID to ID′, the proxy first parse $c_{ID} = (c_1,\cdots,c_{7n-1})$, $rk_{\text{ID}\to\text{ID}'} = (N, d_0', d_{-1,i}, d_{1,i}) = (N_1, \cdots, N_9)$.

- If $n = 1$, encrypt $c_7 = \dfrac{e(c_2, N_7)}{c_3^{N_8} c_4^{f(N_8)} e(c_1, N_9)}$, and so the re-encrypted ciphertext is

  $cID' = (c_1, c_2,\cdots,c_7, N)$.

- If $n > 1$, treat the element $(c_{7n-6}, c_{7n-5},\cdots,\mathrm{c}_{7n-1})$ as a first-level ciphertext δ, compute

  $$(c_1', c_2', \cdots, c_{13}') = Re\text{-}encrypt(params, rk_{ID \to ID'}, \delta)$$

  and output the ciphertext $c_{ID'} = (c_1, c_2, \cdots, c_{7n-7}, c_1', c_2', \cdots, c_{13}')$. Each level-$n$ ciphertext contains $7n - 1$ elements. In principle, the scheme permits an arbitrary number of re-encryptions on a ciphertext, with a seven-element ciphertext expansion on each re-encryption.

6 Decrypt. Let $c_{ID} = (c_1,\cdots,c_{7n-1})$, $d_{ID} = (d_{0,i}, d_{-1,i}, d_{1,i}, d_{i+1,i},\cdots,d_{l,i})$.

- If $n = 1$, decrypt

  $$\frac{e(c_2, d_{0,i})}{c_4^{f(d_{-1,i})} c_3^{d-1,i} e(c_1, d_{1,i})} = e(g_1, h_0)^s$$

  and

  $$\gamma = h\left(c_1, c_2, c_3, c_4, e(g_1, h_0)^s\right), \frac{c_5}{e(g_1, h_0)^\gamma} = R, \frac{R}{e(g_1, h_0)^s} = m$$

  Then he computes

  $$c_6' = H(c_1, c_2, c_3, c_4, c_5, m, R)$$

and verifies whether $c_6 = c_6'$, if the equation holds, the ciphertext is valid. Otherwise, the recipient returns an error message.

- If $n > 1$, treat the elements (c_{7n-6}, $c_{7n-5}, \cdots, c_{7n-1}$ as a first-level ciphertext δ, and decrypt $X_n = Decrypt(params, d_{ID}, \delta)$. For $i = (n - 1)$ descending to 1, decrypt

$$e(g_1, h_0)^{s_i} = \frac{c_{7i}}{e\left(c_{7i-5}, h(X_n)\right)},$$

$$\gamma_i = H\left(c_{7i-6}, c_{7i-5}, c_{7i-4}, c_{7i-3}, e(g_1, h_0)^{s_i}\right), R_i = \frac{c_{7i-2}}{e(g_1, h_0)^{\gamma_i}},$$

$$c_{7i-1}' = H\left(c_{7i-6}, c_{7i-5}, c_{7i-4}, c_{7i-3}, c_{7i-2}, R_i\right)$$

Then the recipient verifies whether $c_{7i-1}' = c_{7i-1}$. If yes, he computes $R_i = e(g_1, h_0)^{s_i} = X_i$. Finally, output X_1 as the plaintext m.

3.3 Our attack

Considering this PRE system: The proxy between user A and user B is P_1, and the proxy between user B and user C is P_2, and so on. In our attack, the corrupted user C, corrupted proxy P_2 and corrupted proxy P_1 can decrypt user A's any ciphertext. Note this attack is not a trivial attack, for it does not lie in any restriction in the above security model. The above security model just does not allow the collusion attack between P_1 and B, but does not say anything about the collusion attack among C, P_1, P_2. This attack just like Koo et al.'s (2009) attack on Green-Ateniese's IBPRE scheme. Concretely, the attack as following:

1 In Phase 1, $\mathcal{A}$ does not issue any query.

2 In Challenge Phase, $\mathcal{A}$ outputs an identity (User A) $\mathrm{ID}^* = (ID_1^*, ID_2^*, \cdots, ID_i^*)$ and two equal length plaintexts m_0, m_1 on which it wishes to be challenged. $\mathcal{B}$ picks a random bit $w \in \{0, 1\}$ and computes C^* = Encrypt($params$, ID*, m_w), sends C^* to $\mathcal{A}$. Here

$$C = \left(c_1 = \left(\prod_{k=1}^{i} h_l h_k^{ID_k^*} \right)^s, c_2 = g^s, c_3 = e(g_1, g_2)^s, c_4 = e(g_1, g_3)^s, \right.$$
$$\left. c_5 = m \cdot e(g_1, h_0)^{s+\gamma}, c_6 = H\left(c_1, c_2, c_3, c_4, c_5, m, m \cdot e(g_1, h_0)^s\right) \right)$$

where $\gamma = h(c_1, c_2, c_3, c_4, \mathrm{e}(g_1, h_0)^s)$.

3 $\mathcal{A}$ queries the re-encryption key between A and B, the re-encryption key between B and C, also the security key for C (NOTE these queries are allowed in Ren et al.'s HIBPRE security model) and he will get

$$rk_{A\to B} = \left(N^A, \left(d_0'\right)^A, d_{-1,i}^A, d_{1,i}^A\right),$$
$$N^A = \left(N_1^A, N_2^A, \cdots, N_6^A\right) = Encrypt(params, \mathrm{ID}^*, X), \left(d_0'\right)^A = d_{0,i}^A h(X)$$
$$rk_{B\to C} = \left((N')^B, \left(d_0'\right)_B, d_{-1,i}^B, d_{1,i}^B\right),$$
$$N^B = \left(N_1^B, N_2^B, \cdots, N_6^B\right) = Encrypt\left(params, (\mathrm{ID})_B, X'\right), \left(d_0'\right)^B = d_{0,i}^B h(X')$$
$$sk_C = \left(d_{0,i}^C, d_{-1,i}^C, d_{1,i}^C\right)$$

4 Now he does as following:

- He first decrypts $(N')^B$ by using $\underline{sk_C}$ and he will get X'.
- Then he computes $d_{0,i}^B = \frac{(d_0')^B}{h(X')}$.
- Then he can decrypt N^A by using $sk_B = (d_{0,i}^B, d_{-1,i}^B, d_{1,i}^B)$ where the last two items coming from $rk_{B\to C}$, and he will get X.
- Then he computes $d_{0,i}^A = \frac{(d_0')^A}{h(X)}$.
- Now he get A secret key $sk_A = (d_{0,i}^A, d_{-1,i}^A, d_{1,i}^A)$ where the last two items coming from $rk_{A\to B}$. Certainly he can decrypt any challenge ciphertext by using the secret key.

The above attack shows the security model for multi-use CCA-secure PRE is much more complicated than single-use CCA-secure PRE, and it needs more careful consideration. To the best of our knowledge, there is even no agreed security model for multi-use CCA secure PRE.

4 Conclusions

In this paper, we indicate that the recent proposal on multi-use CCA-secure PRE proposed by Wang and Cao (2009) are not CCA-secure. Furthermore, we show a recently proposed fully secure HIBPRE scheme without random oracle by Ren et al. (2010) is not secure either. It is still open problems to find a compact multi-use CCA-secure PRE scheme.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Contract No. 61103230, 61272492.

References

Ateniese, G., Fu, K., Green, M. and Hohenberger, S. (2005) 'Improved proxy re-encryption schemes with applications to secure distributed storage', *ACM NDSS 2005*, pp.29–43.

Ateniese, G., Fu, K., Green, M. and Hohenberger, S. (2006) 'Improved proxy re-encryption schemes with applications to secure distributed storage', *ACM Transactions on Information and System Security*, pp.1–30.

Blaze, M., Bleumer, G. and Strauss, M. (1998) 'Divertible protocols and atomic proxy cryptography', *EUROCRYPT 1998*, LNCS 1403, pp.127–144.

Canetti, R. and Hohenberger, S. (2007) 'Chosen ciphertext secure proxy re-encryption', *ACM CCS 2007*, pp.185–194.

Canetti, R., Halevi, S. and Katz, J. (2003) 'A forward-secure public-key encryption scheme', *EUROCRYPT 2003*, LNCS 2656, pp.255–271.

Chow, S., Weng, J., Yang, Y. and Deng, R. (2010) 'Efficient unidirectional proxy re-encryption', *AFRICACRYPT 2010*, LNCS 6055, pp.316–332.

Chu, C. and Tzeng, W. (2007) 'Identity-based proxy re-encryption without random oracles', *ISC 2007*, LNCS 4779, pp.189–202.

Deng, R., Weng, J., Liu, S. and Chen, K. (2008) 'Chosen ciphertext secure proxy re-encryption without pairing', *CANS 2008*, LNCS 5339, pp.1–17.

Dodis, Y. and Ivan, A. (2003) 'Proxy cryptography revisited', *Internet Society (ISOC): NDSS 2003*.

Green, M. and Ateniese, G. (2007) 'Identity-based proxy re-encryption', *ACNS 2007*, LNCS 4521, pp.288–306.

Hohenberger, S. and Waters, B. (2009) 'Realizing hash-and-sign signatures under standard assumptions', *EUROCRYPT 2009*, LNCS 5479, pp.333–350.

Ibraimi, L., Tang, Q., Hartel, P. and Jonker, W. (2008) 'A type-and-identity-based proxy re-encryption scheme and its application in healthcare', *SDM 2008*, LNCS 5159, pp.185–198.

Koo, W., Hwang, J. and Lee, D. (2009) 'Security vulnerability in a non-interactive ID-based proxy re-encryption scheme', *Information Processing Letters*, Vol. 109, Nos. 23–24, pp.1260–1262.

Kurosawa, K. and Desmedt, Y. (2004) 'A new paradigm of hybrid encryption scheme', *CRYPTO 2004*, LNCS 3152, pp.426–442.

Lai, J., Zhu, W., Deng, R., Liu, S. and Kou, W. (2010) 'New constructions for identity-based unidirectional proxy re-encryption', *Journal of Computer Science and Technology*, Vol. 25, No. 4, pp.793–806.

Libert, B. and Vergnaud, D. (2008) 'Unidirectional chosen ciphertext secure proxy re-encryption', *PKC 2008*, LNCS 4939, pp.360–379.

Matsuo, T. (2007) 'Proxy re-encryption systems for identity-based encryption', *PAIRING 2007*, LNCS 4575, pp.247–267.

Ren, Y., Gu, D., Wang, S. and Zhang, X. (2010) 'Hierarchical identity based proxy re-encryption without random oracles', *International Journal of Foundations of Computer Science*, Vol. 21, No. 6, pp.1049–1063.

Shao, J. and Cao, Z. (2009) 'CCA-secure proxy re-encryption without pairing', *PKC 2009*, LNCS 5443, pp.357–376.

Shao, J., Cao, Z. and Lin, P. (2009) 'Generic construction for CCA-secure unidirectional proxy re-encryption', *Security and Communication Networks*, Vol. 4, No. 2, pp.1–16.

Shao, J., Cao, Z. and Liu, P. (2010) 'CCA-secure PRE scheme without random oracle', *Cryptology ePrint Archive*, Report 2010/112 [online] http://eprint.iacr.org.

Tang, Q. (2008) 'Inter-domain identity-based proxy re-encryption', *INSCRYPT 2008*, LNCS 5487, pp.332–347.

Wang, H. and Cao, Z. (2009) 'A fully secure unidirectional and multi-use proxy re-encryption scheme', *ACM CCS Poster Session*.

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

Weng, J., Chen, M., Yang, Y., Deng, R., Chen, K. and Bao, F. (2010a) 'CCA-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles', *Science China Information Sciences*, Vol. 53, No. 4, pp.593–606.

Weng, J., Chen, M., Yang, Y., Deng, R., Chen, K. and Bao, F. (2010b) 'CCA-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles', Report 2010/265, Cryptology ePrint Archive.

Weng, J., Deng, R.H., Chu, C., Ding, X. and Lai, J. (2009) 'Conditional proxy re-encryption secure against chosen-ciphertext attack', *ACM ASIACCS 2009*, pp.322–332.