International Journal of Security and Its Applications
Vol.9, No.12 (2015), pp.405-418
http://dx.doi.org/10.14257/ijsia.2015.9.12.36

Research on Non-interactive Construction based on Fuzzy Conditional Proxy Re-encryption

Chunpeng Ge*, Jiandong Wang and Liming Fang

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210000, China
** gecp@nuaa.edu.cn*

Abstract

In a conditional proxy re-encryption (C-PRE) scheme, a semi-trusted proxy can transform Alice's ciphertext into Bob's ciphertext without learning the underlying plaintext, if the ciphertext satisfies a certain condition. To achieve more fine-grained delegation on conditions, Fang, Wang and Ge introduced the notion of fuzzy conditional proxy re-encryption (FC-PRE), whereby the conditions is viewed a set of descriptive keywords. The proxy with a re-encryption key for a condition word set W can convert a ciphertext encrypted with a condition word set W', if and only if W and W' are close to each other as measured by the "set overlap" distance metric. Nonetheless, they left an open problem on how to construct a non-interactive FC-PRE scheme. Furthermore, their scheme is only proved secure in the random oracle model. In this paper, we answer the above problems affirmatively by presenting a non-interactive fuzzy conditional proxy re-encryption scheme. Moreover, our scheme is proved secure without random oracles.

Keywords: *Proxy re-encryption; Conditional proxy re-encryption; Non-interactive; without random oracles*

1. Introduction

Proxy re-encryption (PRE), introduced by Blaze, Bleumer and Strauss [1], enables a semi-trusted proxy to transform Alice's ciphertext into Bob's ciphertext of the same message without revealing the underlying message. PRE has found lots of applications such as distributed file system and outsourced filtering of encrypted spam [2]. Rather than allowing the proxy to convert all of Alice's ciphertext, Alice may only want the proxy to convert the ciphertexts that satisfies a certain condition. To overcome this problem, Weng, Deng and Ding [3] introduced the notion of conditional proxy re-encryption, whereby only ciphertext satisfying one certain condition set by Alice can be transformed by the proxy. Similarly to Weng's work [3], Libert and Vergnaud [4] proposed a PRE scheme that provides warrant-based and keyword-based delegations in PKC08. Later, Tang proposed a notion called typed-based proxy re-encryption [5]. To achieve anonymous on conditions, Fang, Susilo and Wang proposed an anonymous conditional proxy re-encryption scheme [6].

In the former C-PRE, the condition is set to be a specific keyword, thus the C-PRE cannot enable a more flexible combination of keywords. To overcome this problem, Fang [7] proposed the notion of fuzzy conditional proxy re-encryption. Whereby, ciphertext is encrypted with a condition set W, and proxy re-encryption key is generated with a condition set W'. The proxy is able to convert the ciphertext if and only if W and W' are close to each other as measured by the "set overlap" distance metric. Unfortunately, their scheme is interactive, which means Alice and Bob should interact to generate the re-encryption key. This is undesirable as Bob should be on line when doing the delegation.

*Corresponding Author

ISSN: 1738-9976 IJSIA

Moreover, their scheme is proved against chosen-ciphertext attack (CCA) secure in the random oracle model, unfortunately, a proof in the random oracle model can only servers as a heuristic argument and admittedly using quite contrived constructions, it has been shown in [8] to possibly lead to insecure schemes when the random oracles are implemented in the standard model.

1.1. Our Contributions

This paper aims to resolving the aforementioned open problems. In this paper, we present a non-interactive fuzzy conditional proxy re-encryption scheme. Furthermore, our scheme is proved CCA secure under the 3-weak decisional bilinear Diffie-Hellman inversion (3-wDBDHI) assumption without random oracles.

1.2. Paper Organization

The rest of the paper is organized as follows. Section 2 gives some preliminaries, the definitions and complexity assumption and the security model of FC-PRE schemes. In section 3, we present our non-interactive FC-PRE scheme and prove its security in the standard model. In section 4, we conclude the paper.

2. Preliminaries

2.1. Negligible Function

A function $f : N \to [0,1]$ is said to be negligible if for all $c \in N$, there exists a $k_c \in N$ such that $f(k) < k^{-c}$ for all $k > k_c$.

2.2. Bilinear Map

Let G and G_T be two multiplicative cyclic groups with the same prime order p, and g be a generator of G. A bilinear pairing is a map $e : G \times G \to G_T$ with the following properties [9,10]:

1. $e(g^a, g^b) = e(g,g)^{ab}$ for all $a, b \in Z_p$ and $g \in G$.
2. $e(g,g) \neq 1$.
3. There is an efficient algorithm to compute $e(g,g)$ for all $g \in G$.

2.3. The 3-wDBDHI Assumption

Let $e : G \times G \to G_T$ be a bilinear map. We define the advantage function $Adv_{G,\mathcal{B}}^{3-wDBDHI}(\lambda)$ of an adversary $\mathcal{B}$ as:

$$|Pr[\mathcal{B}(g, g^{\frac{1}{a}}, g^a, g^{(a^2)}, g^b, e(g,g)^{\frac{b}{a^2}}) = 1] - Pr[\mathcal{B}(g, g^{\frac{1}{a}}, g^a, g^{(a^2)}, g^b, e(g,g)^r) = 1]|$$

Where $a, b, r \in Z_p$ are randomly chosen. We say that the 3-wDBDHI assumption [11] relative to generator G holds if $Adv_{G,\mathcal{B}}^{3-wDBDHI}(\lambda)$ is negligible for all PPT $\mathcal{B}$.

2.4. Pseudorandom Function Family

We here review the definition of pseudorandom function family [11]. Let $F : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ be a function family, where $\mathcal{K}$ is the set of keys of F, D is the domain, and R is the range. Let $G : D \to R$ be a true random function family. Let $\mathcal{F}$ be a PPT adversary which outputs a bit. We consider the following two experiments:

Experiment $Exp_F^{PRF-1}(\mathcal{F})$: [$K \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \mathcal{F}^{\mathcal{F}(\mathcal{K},.)}$, Return b].

Experiment $Exp_F^{PRF-0}(\mathcal{F})$: [$g \xleftarrow{\$} G, b \xleftarrow{\$} \mathcal{F}^{\mathcal{G}(.)}$, Return b].

We define $\mathcal{F}$'s advantage $Adv_{F,\mathcal{F}}^{PRF}$ in attacking the pseudorandom of the function F as

$$|Pr[Exp_F^{PRF-1}(\mathcal{F}) = 1] - Pr[Exp_F^{PRF-0}(\mathcal{F}) = 1]|.$$

If for any PPT adversary $\mathcal{F}$, his advantage in attacking the pseudorandomness of the function family F is negligible, the we say that F is a pesudorandom function family.

2.5. Fuzzy Conditional Proxy Re-encryption

In this section, we review the definition and security notions for FC-PRE [7].

Definition 1 (Fuzzy conditional proxy re-encryption). A (single-hop) fuzzy conditional proxy re-encryption scheme consists the following eight algorithms:

- GlobalSetup(1^k): On input a security parameter 1^k, the global public parameters are outputted.
- KeyGen(i): The KeyGen algorithm outputs the public key pk_i and secret key sk_i for user i.
- RKeyGen(sk_i, W, pk_j): On input the delegator's secret key sk_i, the conditional keywords set $W = (\omega_1, \omega_2, \cdots, \omega_n)$, and the delegatee's public key sk_j, outputs the conditional re-encryption key $rk_{i,W,j}$.
- Enc(pk_i, m, W): On input the public key pk_i, a message $m \in M$ and a conditional keywords set $W = (\omega_1, \omega_2, \cdots, \omega_n)$, it outputs a second level ciphertext CT_i associated with W under the public key pk_i.
- REnc($CT_i, rk_{i,W,j}$): On input a second level ciphertext CT_i associated with conditions W, and a conditional re-encryption key $rk_{i,W',j}$. Outputs the first level ciphertext CT_j under pk_j if $|W \bigcap W'| \geq d$, or an error symbol $\perp$ otherwise.
- Dec1(CT_j, sk_j): On input a re-encrypted ciphertext (first level) CT_j under pk_j and the private key sk_j. Outputs the message $m \in M$ or an error symbol $\perp$.
- Dec2(CT_i, sk_i): On input the original ciphertext (second level) CT_i under pk_i and the private key sk_i. It output the message $m \in M$ or an error symbol $\perp$.

Note that we omit the global parameters PP as the other algorithms' input for simplicity. The correctness of FC-PRE means that, for any condition set W, any message $m \in M$, any $(pk_i, sk_i) \leftarrow KeyGen(i)$, $(pk_j, sk_j) \leftarrow KeyGen(j)$, any $CT_j \leftarrow REnc(CT_i, rk_{i,W',j})$, where $|W \bigcap W'| \geq d$ and $CT_i = Enc2(pk_i, m, W)$, we have:

$$Pr[Dec2(CT_i, sk_i) = m] = 1 \text{ and}$$
$$Pr[Dec1(sk_j, ReEnc(CT_i, RKeyGen(sk_i, W', pk_j))) = m] = 1.$$

2.6. FC-PRE-IND-CCA Game

Next we consider the following two games, which correspond to the security of the first level ciphertext and the second level ciphertext individually.

Game1: IND-level2-CCA game, which considers the security of level2 ciphertext.

1. Setup. The challenger $\mathcal{C}$ perform $GlobalSetup(\lambda)$ to get the public parameter PP. Give the public parameter PP to the adversary $\mathcal{A}$.

2. Query phase 1. The adversary $\mathcal{A}$. makes the following queries:

- Uncorrupted key generation query (i): The challenger $\mathcal{C}$ first runs algorithm $KeyGen(i)$ to obtain a public/secret key pair(pk_i, sk_i), and then sends pk_i to the adversary $\mathcal{A}$.
- Corrupted key generation query (j): The challenger $\mathcal{C}$ first runs algorithm $KeyGen(j)$ to obtain a public/secret key pair (pk_j, sk_j), and then sends (pk_j, sk_j) to the adversary $\mathcal{A}$.
- Re-encryption key query (pk_i, W, pk_j) : The challenger $\mathcal{C}$ runs algorithm $RKeyGen(sk_i, W, pk_j)$ to generate a conditional re-encryption key $rk_{i,W,j}$ and returns it to the adversary $\mathcal{A}$. It is required that pk_i and pk_j have been generated beforehand by algorithm $KeyGen$.
- Re-encryption query $(pk_i, pk_j, (W, CT_i))$: The challenger $\mathcal{C}$ runs algorithm $CT_j = ReEnc(CT_i, RKeyGen(sk_i, W, sk_j))$ and returns the resulting ciphertext CT_j to the adversary $\mathcal{A}$. It is required that pk_i and pk_j have been generated beforehand by algorithm $KeyGen$.
- Decryption query $(pk_i, (W, CT_i))$: Here $(pk_i, (W, CT_i))$ denotes the queries on original conditional ciphertext. Challenger $\mathcal{C}$ returns the result of $Dec2(CT_i,\ sk_i)$ to $\mathcal{A}$. It is required that pk_i has been generated beforehand by algorithm $KeyGen$
- Decryption query (pk_j, CT_j) : Here (pk_j, CT_j) denotes the queries on re-encrypted ciphertext. Challenger $\mathcal{C}$ returns the result of $Dec1(CT_j, sk_j)$ to $\mathcal{A}$. It is required that pk_j has been generated beforehand by algorithm $KeyGen$.

3. Challenge. Once $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal length message (m_0, m_1). Challenger $\mathcal{C}$ chooses a bit $b \in \{0, 1\}$ and sets the challenge ciphertext to be $CT^* = Enc2(pk_{i^*}, m_b, W^*)$, which is sent to $\mathcal{A}$.

4. Query phase 2. $\mathcal{A}$ continues making queries as in the query phase 1.

5. Guess. $\mathcal{A}$ outputs the guess b'. The adversary wins if $b' = b$.

During the above game, adversary $\mathcal{A}$ is subject to the following restrictions:

- $\mathcal{A}$ cannot issue corrupted key generation queries on (i^*) to obtain the target secret key sk_{i^*}.
- $\mathcal{A}$cannot issue decryption queries on neither $(pk_{i^*}, (W^*, CT^*))$ nor (pk_j, CT_j) where (pk_j, CT_j) is a re-encryption of the challenge pair $(pk_{i^*}, (W^*, CT^*))$.
- $\mathcal{A}$ cannot issue re-encryption queries on $(pk_{i^*}, pk_j, (W, CT^*))$, if $|W \bigcap W^*| \geq d$ and pk_j appears in a previous corrupted key generation query.
- $\mathcal{A}$ cannot obtain the conditional re-encryption key $rk_{i^*,W,j}$, if $|W \bigcap W^*| \geq d$ and $|W \bigcap W^*| \geq d$ appears in a previous corrupted key generation query.

We refer to the above adversary $\mathcal{A}$ as an IND-L2-CCA adversary. His advantage is defined as $Succ_{\mathcal{A}}^{Game_1}(\lambda) = |Pr[b' = b] - 1/2|$.

Game 2: IND-level1-CCA game, which considers the security of level1 ciphertext.

The above definition provides adversaries with a second level ciphertext in the challenge phase. A complementary definition of security captures their inability to distinguish first level ciphertext as well. The queries in IND-level1-CCA game is almost the same as IND-level2-CCA game unless in the following ways: (1) For single-use schemes, the adversary is provided with access to all re-encryption keys, thus the re-encryption oracle becomes useless; (2) The level 2 decryption oracle is unnecessary; (3) Since first level ciphertext cannot be re-encrypted, there is no need to keep attackers from obtaining all honest-to-corrupt re-encryption keys.

Master secret security. Atenises, Fu, Green and Hohenberger [2] defined another important security requirement, named master secret security. This suggests that even if the dishonest proxy collude with delegatees, it is still impossible for them to expose the private key of their common delegators in its entirety. As discussed in [4], the notion of CCA security of first level ciphertext implies the master secrete security.

3. Proposed non-interactive FC-PRE scheme

3.1. Construction

Let G be bilinear group of prime order p, and g be a generator of G. Additionally, let $e : G \times G \to G_T$ denote the bilinear map.

We also define the Lagrange coefficient $\Delta_{\omega,S}(x)$ for $\omega \in Z_p$ and a set S, of elements in Z_p:

$$\Delta_{\omega,S}(x) = \prod_{i \in S, i \neq \omega} \frac{x-i}{\omega-i}$$

Our proposed scheme consists of the following algorithms:

- GlobalSetup(λ): Let λ be the security parameter, and (p, g, G, G_T, e) be the bilinear map parameters. Let d be the "set overlap" distance metric, which means that the proxy is able to perform re-encryption if and only if at least d components of the second level ciphertext with their re-encryption key components. $M = \{0,1\}^k$ denotes the message space. Let $H_1 : G \times Z_p^* \to G^*$, $H_2 : G \times \{0,1\}^l \to Z_p^*$, $H_3 : \{0,1\}^k \to Z_p^*$ be collision-resistent hash functions. Choose a pesudorandom function (PRF) family $F : G_T \times G \to \{0,1\}^{l-k} \parallel \{0,1\}^k$. Randomly picks generators $g, g_1, g_2, g_3, g_4 \xleftarrow{\$} G$. Output the public parameters $PP = (p, g, g_1, g_2, g_3, g_4, G, G_T, e, d, H_1, H_2, H_3, F, l, k)$.
- KeyGen(i): User i picks a random $x_i \in Z_p^*$, and sets his public key as $pk_i = g^{x_i}$, and the private key as $sk_i = x_i$.
- Enc(pk_i, W, m): To encrypt a message $m \in \{0,1\}^k$, under a public key pk_i and a conditional keyword set W at the second level, the sender proceeds as follows:

  (a). Set $C_0 = W$;

  (b). Select a random $r \in Z_p^*$ and compute

  $$C_1 = g_1^r, \qquad C_2 = pk_i^r, \qquad C_4 = (C_\omega = H_1(pk_i, \omega)^r)_{\omega \in W};$$

  (c). Compute $K = e(g,g)^r$, and set $C_3 = [F(K, C_1)]_{l-k} \parallel ([F(K, C_1)]^k \oplus m)$;

(d). Select a random $t \in Z_p^*$ and compute $h = H_2(C_1, C_3)$ and $C_5 = (g_2^h g_3^t g_4)^r$;

(e). Output the ciphertext $CT_i = (t, C_0, C_1, C_2, C_3, C_4, C_5)$.

- RKeyGen(sk_i, W', pk_j): On input user i's private key $sk_i = x_i$, a conditional keyword set W', and user j's public key $pk_j = g^{x_j}$. The RKeyGen algorithm proceeds as follows. Fist choose a random value $\theta \in \{0,1\}^k$.

  (a). Randomly choose a $d-1$ degree polynomial $q(x)$, such that $q(0) = H_3(\theta)/x_i$. For each $\omega \in W'$, select a random $s_\omega \in Z_p^*$, and computes

  $$a = (a_\omega = g^{q(\omega)} H_1(pk_i, \omega)^{s_\omega})_{\omega \in W'}, \qquad b = (b_\omega = pk_i^{s_\omega})_{\omega \in W'},$$

  (b). Select a random $r' \in Z_p^*$ and compute

  $$rk_1 = g_1^{r'}, \qquad rk_2 = pk_j^{r'};$$

  (c). Compute $K' = e(g,g)^{r'}$, and set

  $$rk_3 = [F(K', rk_1)]_{l-k} \parallel ([F(K', rk_1)]^k \oplus \theta);$$

  (d). Select a random $t' \in Z_p^*$ and compute

  $$h' = H_2(rk_1, rk_3) \qquad rk_4 = (g_2^{h'} g_3^{t'} g_4)^{r'};$$

  (e). Set the conditional re-encryption key

  $$rk_{i,W',j} = (a, b, W', rk_1, rk_2, rk_3, t', rk_4).$$

- ReEnc $(rk_{i,W',j}, CT_i)$: On input a re-encryption key $rk_{i,W',j} = (a, b, W', rk_1, rk_2, rk_3, t', rk_4)$ and a second level ciphertext $CT_i = (t, C_0, C_1, C_2, C_3, C_4, C_5)$. Recovers $W = C_0$ and computers $h = H_2(C_1, C_3)$, the proxy checks whether the following equalities hold:

$$e(C_1, g_2^h g_3^t g_4) \stackrel{?}{=} e(g_1, C_5), \tag{1}$$

$$e(C_1, pk_i) \stackrel{?}{=} e(g_1, C_2), \tag{2}$$

$$e(C_1, H_1(pk_i, \omega))_{\omega \in W} \stackrel{?}{=} e(g_1, C_\omega)_{\omega \in W} \tag{3}$$

If not, output $\perp$. Otherwise, choose an arbitrary d element subset, $S \subseteq W \cap W'$. Computer

$$C_2' = \prod_{\omega \in S} \left(\frac{e(C_2, a_\omega)}{e(b_\omega, C_4)} \right)^{\Delta_{\omega,S}(0)}.$$

The re-encrypted ciphertext (level 1) is $CT_j = (t, C_1, C_2', C_3, C_5, rk_1, rk_2, rk_3, t', rk_4)$.

Note, for a valid ciphertext and conditional re-encryption keyword set, we have

$$\begin{aligned} C_2' &= \prod_{\omega \in S} \left(\frac{e(C_2, a_\omega)}{e(b_\omega, C_4)} \right)^{\Delta_{\omega,S}(0)} \\ &= \prod_{\omega \in S} \left(\frac{e(pk_i^r, g^{q(\omega)} H_1(pk_i, \omega)^{s_\omega})}{e(pk_i^{s_\omega}, H_1(pk_i, \omega)^r)} \right)^{\Delta_{\omega,S}(0)} \\ &= e(pk_i, g)^{rq(0)} = e(g,g)^{rH_3(\theta)}. \end{aligned}$$

- Dec2 (sk_i, CT_i): On input a private key sk_i and a second level ciphertext $CT_i = (t, C_0, C_1, C_2, C_3, C_4, C_5)$, he proceeds as follows:

  (a). First check the validity of the ciphertexts as in equations $(1) - (3)$. If the verification fails, output $\perp$ and abort.

  (b). Compute $K = e(C_2, g)^{1/sk_i}$. If $F[K, C_1]_{l-k} = [C_3]_{l-k}$ holds, output $m = F[K, C_1]^k \oplus [C_3]^k$; else output $\perp$ and abort.

- Dec1 (sk_j, CT_j): On input a private key sk_j and a first level ciphertext $CT_j = (t, C_1, C_2', C_3, C_5, rk_1, rk_2, rk_3, t', rk_4)$, he proceeds as follows:

  (a). computers $h' = H_2(rk_1, rk_3)$, the proxy checks whether the following equalities hold:

$$e(rk_1, g_2^{h'} g_3^{t'} g_4) \stackrel{?}{=} e(g_1, rk_4), \tag{4}$$

$$e(rk_1, pk_j) \stackrel{?}{=} e(g_1, rk_2) \tag{5}$$

  If not, output $\perp$. Otherwise, compute $K' = e(rk_2, g)^{1/sk_j}$.

  If $F[K', rk_1]_{l-k} = [rk_3]_{l-k}$ holds, output $\theta = F[K', rk_1]^k \oplus [rk_3]^k$; else output $\perp$ and abort.

  (b). Check the validity of the ciphertexts as in equations $(1) - (2)$. If the verification fails, output $\perp$ and abort.

  (c). Compute $K = C_2'^{1/H_3(\theta)}$. If $F[K, C_1]_{l-k} = [C_3]_{l-k}$ holds, output $m = F[K, C_1]^k \oplus [C_3]^k$; else output $\perp$ and abort.

Correctness. It is straightforward to verify that all the correctly generated original/re-encrypted ciphertexts can be correctly decrypted.

3.2. Security of our FC-PRE Scheme

In this subsection, we prove the FC-PRE-IND-CCA security for our scheme without random oracles. The analysis of Game 1 and Game 2 are as follows.

Theorem 1. If the 3-wDBDHI assumption holds, then the above scheme is FC-PRE-IND-CCA secure without random oracles.

Lemma 1. If there exists an IND-L2-CCA adversary $\mathcal{A}$ against our scheme, then there exists an algorithm $\mathcal{B}$ which can solve the 3-wDBDHI problem.

Proof. Our approach to proving lemma 1 closely follows the security proof for Weng's scheme [11] and the Fuzzy IBE scheme [12]. In the following, we call HU be the set of honest parties, including the target user i^*, and CU be the set of corrupt parties. $\mathcal{A}$first generates an keyword set $W^* = (\omega_1^*, \omega_2^*, \cdots, \omega_n^*)$ that it intends to attack.

We let the challenger set the group G and G_T with an efficient bilinear map e and a generator g of G. $\mathcal{B}$ inputs a 3-wDBDHI instance $(g, A_{-1} = g^{1/a}, A_1 = g^a, A_2 = g^{a^2}, B = g^b, T)$ and has to distinguish $T = e(g, g)^{b/a^2}$ from a random element in G_2.

Next, $\mathcal{B}$ sets up the global parameters as follows: $\mathcal{B}$ sets the generators $g_1 = A_2^{\alpha_0}$,

$g_2 = A_1^{\alpha_1} A_2^{\beta_1}, g_3 = A_1^{\alpha_2} A_2^{\beta_2}, g_4 = A_1^{\alpha_3} A_2^{\beta_3}$for a randomly chosen $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2,$

$\beta_3 \in Z_p^*$. Finally, $\mathcal{B}$ samples a pairwise independent hash functions H_1, such that $H_1(pk_i, \omega) = pk_i^{H_1(\omega)}$. Without loose of generality, assume that H_1, H_2, H_3 are target collision resistant and F is a pesudorandom function family. The system parameters are $(p, g, g_1, g_2, g_3, g_4, G, G_T, e, d, H_1, H_2, H_3, F, l, k)$.

- Uncorrupted key: On input i to $\mathcal{O}_{HU}$, public keys of honest users $i \in HU \setminus \{i^*\}$ are defined as follows: $\mathcal{B}$ select a random $x_i \xleftarrow{R} Z_p^*$, then $\mathcal{B}$ computes $pk_i = A_1^{x_i}$. The target user's public key is set as $pk_{i^*} = A_2^{x_{i^*}}$ for randomly chosen $x_{i^*} \xleftarrow{R} Z_p^*$. Sends public key pk_i, pk_{i^*} to $\mathcal{A}$.
- Corrupted key: On input j to $\mathcal{O}_{CU}$, public keys of corrupted user $j \in CU$ are defined as follows: $\mathcal{B}$ select a random $x_j \xleftarrow{R} Z_p^*$, then $\mathcal{B}$ computes $sk_j = x_j, pk_i = g^{x_j}$. Sends public and private key (sk_j, pk_j) to $\mathcal{A}$.
- Re-encryption key query(pk_i, W, pk_j): Consider a query for the re-encryption key corresponding to a conditional set W, the only restriction is that $|W \cap W^*| < d$ when $pk_i = pk_i^*$ and $j \in CU$, otherwise $\mathcal{B}$ abort and output $\perp$.

If the above condition is met, $\mathcal{B}$ randomly chooses $\theta \in \{0,1\}^k$, let $\Phi = H_3(\theta)/sk_i$, $g^{\Phi} = g^{H_3(\theta)/sk_i}$, to answer the Re-encryption key query (pk_i, W, pk_j), $\mathcal{B}$ has to distinguish several situations:

(a). If $i \in CU$, since $\mathcal{B}$ knows the private key x_i for user i, $\mathcal{B}$ can compute the conditional re-encryption $rk_{i,W,j}$ as the re-encryption key generation algorithm.

(b). If $i \in HU \setminus \{i^*\}, \Phi = H_3(\theta)/sk_i = H_3(\theta)/(ax_i), g^{\Phi} = A_{-1}^{H_3(\theta)/x_i}$, $\mathcal{B}$ first defines three sets Γ, Γ', S in the following manner: $\Gamma = W \bigcap W^*$, Γ' be any set such that $\Gamma \subseteq \Gamma' \subseteq W$ and $|\Gamma'| = d - 1$, and $S = \Gamma' \cup \{0\}$.

Next, we define the re-encryption key components a_ω and b_ω for $\omega \in W$ as:

If $\omega \in \Gamma'$:

$$a_\omega = g^{q(\omega)} \cdot H_1(pk_i, \omega)^{s_\omega}, \qquad b_\omega = pk_i^{s_\omega},$$

Where s_ω, $q(\omega)$ are chosen randomly in Z_p^*.

If $\omega \in W - \Gamma'$:

$$\begin{aligned} a_\omega &= \prod_{\omega \in \Gamma', i = index(\omega)} g^{q(i)\Delta_{i,S}(0)} \cdot (g^{\Phi})^{\Delta_{0,S}(0)} \cdot H_1(pk_i, \omega)^{s_\omega} \\ &= g^{q(\omega)} \cdot H_1(pk_i, \omega)^{s_\omega}, \\ b_\omega &= pk_i^{s_\omega}. \end{aligned}$$

Where s_ω are randomly chosen in Z_p^*.

(c). If $i = i^*$, $\mathcal{B}$ outputs a random bit in $\{0,1\}$ and abort.

Next, $\mathcal{B}$ computes $rk_1, rk_2, rk_3, t', rk_4$ as the Enc algorithm. Thus, $\mathcal{B}$ can derive a valid re-encryption key for W.

- First level decryption query (pk_j, CT_j): On given a first level ciphertext $CT_j = (t, C_1, C_2', C_3, C_5, rk_1, rk_2, rk_3, t', rk_4)$, if $j \in CU$, since $\mathcal{B}$ knows the private key $sk_j = x_j$, $\mathcal{B}$ returns $Dec1(sk_j, CT_j)$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$proceeds as follows:

(a). Compute $h' = H_2(rk_1, rk_3)$ and check whether the equation $(4) - (5)$ hold, if not, output "$\perp$" and abort. Else,

(b). Compute

$$A_1^{r'} = \left(\frac{rk_4}{rk_1^{\frac{\beta_1 h' + \beta_2 t' + \beta_3}{\alpha_0}}} \right)^{\frac{1}{\alpha_1 h' + \alpha_2 t' + \alpha_3}}. \quad (6)$$

Then compute $K' = e(A_{-1}, A_1^{r'}) = e(g,g)^{r'}$. Notice that, in equation (6), the quality $\alpha_1 h' + \alpha_2 t' + \alpha_3 = 0 \bmod p$ holds with probability at most $\frac{1}{p}$.

(c). If $F[K', rk_1]_{l-k} \neq [rk_3]_{l-k}$, output "$\perp$" indicating an invalid ciphertext. Otherwise, output $\theta = F[K', rk_1]^k \oplus [rk_3]^k$.

(d). Using θ, $\mathcal{B}$ can get m as the Dec1 algorithm.

- Second level decryption query (pk_i, CT_i): On given a second level ciphertext $CT_i = (t, C_0, C_1, C_2, C_3, C_4, C_5)$, $\mathcal{B}$ first checks whether the equations (1)-(3) hold, if not, $\mathcal{B}$ abort and output "$\perp$". Otherwise, if $i \in CU$, it means that$sk_i = x_i$, and $\mathcal{B}$ return $Dec2(sk_i, CT_i)$ to $\mathcal{A}$. Otherwise, if $i \in HU$, it means that $pk_i = g^{ax_i}$, $\mathcal{B}$ proceeds as follows:

(a). Compute $h = H_2(C_1, C_3)$, $A_1^r = \left(\frac{C_5}{C_1^{\frac{\beta_1 h + \beta_2 t + \beta_3}{\alpha_0}}} \right)^{\frac{1}{\alpha_1 h + \alpha_2 t + \alpha_3}}$. Then compute $K = e(A_{-1}, A_1^r) = e(g,g)^r$.

(b). If $F[K, C_1]_{l-k} \neq [C_3]_{l-k}$, output "$\perp$" indicating an invalid ciphertext. Otherwise, output$m = F[K, C_1]^k \oplus [C_3]^k$.

- Re-encryption query(pk_i, pk_j, CT_i): On input a second level ciphertext $CT_i = (t, C_0, C_1, C_2, C_3, C_4, C_5)$, $\mathcal{B}$ first make sure that $(pk_i, pk_j, W, CT_i) \neq (pk_{i^*}, pk_j, W^*, CT_{i^*})$ if $|W \cap W^*| \geqslant d$ and $j \in CU$, otherwise $\mathcal{B}$ abort and output "$\perp$".

Next $\mathcal{B}$ first check whether the equations (1)-(3) hold, if not, output "$\perp$". If the above condition is met, $\mathcal{B}$ has to distinguish the following several situations:

(a). If $i = i^*$, $\mathcal{B}$ selects a random $\theta \in \{0,1\}^k$ and computes $h = H_2(C_1, C_3)$, $A_1^r = \left(\frac{C_5}{C_1^{\frac{\beta_1 h + \beta_2 t + \beta_3}{\alpha_0}}} \right)^{\frac{1}{\alpha_1 h + \alpha_2 t + \alpha_3}}$.Then compute $C_2' = e(A_{-1}, A_1^r)^{H_3(\theta)} = e(g,g)^{rH_3(\theta)}$. Next $\mathcal{B}$ can compute $rk_1, rk_2, rk_3, t', rk_4$ as the Enc algorithm.

Output $CT_j = (t, C_1, C_2', C_3, C_5, rk_1, rk_2, rk_3, t', rk_4)$ as the re-encrypted ciphertext.

(b). Else, since $\mathcal{B}$ can compute the re-encryption key$rk_{i,W,j}$, so $\mathcal{B}$ can re-encrypted it correctly.

- Challenge. Once $\mathcal{A}$ decided that Phase 1 is over, it outputs two equal length plaintexts (m_0, m_1). Let $r^* = b/a^2$, $\mathcal{B}$ picks a random $\beta \in \{0,1\}$, and responds as follows:

$$C_0^* = W^*,$$
$$C_1^* = B^{\alpha_0} = A_2^{\alpha_0 \frac{b}{a^2}} = g_1^{r^*},$$
$$C_2^* = B^{x_{i^*}} = A_2^{x_{i^*} \frac{b}{a^2}} = pk_i^{r^*},$$
$$C_3^* = [F(T, C_1^*)]_{l-k} \parallel ([F(T, C_1^*)]^k \oplus m_\beta),$$
$$C_4^* = (C_{\omega^*} = B^{x_{i^*} H_1(\omega^*)} = A_2^{x_{i^*} H_1(\omega^*) \frac{b}{a^2}} = H_1(pk_{i^*}, \omega^*)^{r^*})_{\omega^* \in W^*},$$
$$h^* = H_2(C_1^*, C_3^*), \qquad t^* = -\frac{\alpha_1 h^* + \alpha_3}{\alpha_2},$$
$$C_5^* = B^{\beta_1 h^* + \beta_2 t^* + \beta_3} = (A_2^{\beta_1 h^* + \beta_2 t^* + \beta_3})^{r^*} = (g_2^{h^*} g_3^{t^*} g_4)^{r^*}.$$

- Query phase 2. $\mathcal{A}$ continues making queries as in the query phase 1 with the restrictions described in the IND-L2-CCA game.

- Guess. $\mathcal{A}$ outputs the guess β', if $\beta' = \beta$, then output 1 meaning $T = e(g,g)^{b/a^2}$; else output 0 meaning $T = e(g,g)^r$.

Observe that, if $T = e(g,g)^{b/a^2}$, CT_i^* is indeed a valid challenge ciphertext under public key pk_{i^*}. On the other hand, when T is uniform in G_T, the challenge ciphertext CT_i^* is independent of β in the adversary's view.

Lemma 2. If there exists an IND-L1-CCA adversary $\mathcal{A}$ against our scheme, then there exists an algorithm $\mathcal{B}$ which can solve the 3-wDBDHI problem.

Proof. Suppose there exists a polynomial-time adversary $\mathcal{A}$ that can attack our scheme in the standard model. In the following, we call HU be the set of honest parties, including the target user i^*, and CU be the set of corrupt parties. $\mathcal{A}$ first generates an keyword set $W^* = (\omega_1^*, \omega_2^*, \cdots, \omega_n^*)$ that it intends to attack.

We let the challenger set the group G and G_T with an efficient bilinear map e and a generator g of G. Simulator $\mathcal{B}$ inputs a 3-wDBDHI instance $(g, A_{-1} = g^{\frac{1}{a}}, A_1 = g^a, A_2 = g^{(a^2)}, B = g^b, T)$ and has to distinguish $T = e(g,g)^{\frac{b}{a^2}}$ from a random element in G_T.

Next, $\mathcal{B}$ sets up the global parameters as follows: $\mathcal{B}$ sets the generators $g_1 = A_2^{\alpha_0}$, $g_2 = A_1^{\alpha_1} A_2^{\beta_1}, g_3 = A_1^{\alpha_2} A_2^{\beta_2}, g_4 = A_1^{\alpha_3} A_2^{\beta_3}$ for a randomly chosen $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3 \in Z_p^*$. Without loose of generality, assume that H_1, H_2, H_3 are target collision resistant and F is a pesudorandom function family.

Output the system parameters $(p, g, g_1, g_2, g_3, g_4, G, G_T, e, d, H_1, H_2, H_3, F, l, k)$.

- Uncorrupted key generation query(i): Public keys of honest user $i \in HU$(including target public key) are defined as follows: $\mathcal{B}$ select a random $x_i \in Z_p^*$, then $\mathcal{B}$ computes $pk_i = A_1^{x_i}$. Sends public key pk_i to $\mathcal{A}$.

- Corrupted key generation query(j): Public keys of corrupted user $j \in CU$ are defined as follows: $\mathcal{B}$ select a random $x_j \in Z_p^*$, then $\mathcal{B}$ computes $sk_j = x_j, pk_i = g^{x_j}$. Sends public and private key (sk_j, pk_j) to $\mathcal{A}$.

- Re-encryption key query(pk_i, W, pk_j): Consider a query for the re-encryption key corresponding to a conditional set W, the only restriction is that $|W \cap W^*| < d$ when $pk_i = pk_i^*$ and $j \in CU$, otherwise $\mathcal{B}$ abort and output $\perp$. If the above condition is met, $\mathcal{B}$ selects a random value $\theta \in \{0,1\}^k$ and proceeds as follows:

(a). If $i \in CU$. since $\mathcal{B}$ can compute $q(0) = H_3(\theta)/x_i$, next $\mathcal{B}$ can compute the conditional re-encryption $rk_{i,W,j}$ as the re-encryption key generation algorithm.

(b). If $i \in HU$. let $\Phi = H_3(\theta)/sk_i$, $g^{\Phi} = g^{H_3(\theta)/(ax_i)}$, $\mathcal{B}$ first defines three sets Γ, Γ', S in the following manner: $\Gamma = W \bigcap W^*$, Γ' be any set such that $\Gamma \subseteq \Gamma' \subseteq W$ and $|\Gamma'| = d - 1$, and $S = \Gamma' \cup \{0\}$.

Next, we define the re-encryption key components a_ω and b_ω for $\omega \in W$ as:

If $\omega \in \Gamma'$:

$$a_\omega = g^{q(\omega)} \cdot H_1(pk_i, \omega)^{s_\omega}, \qquad b_\omega = pk_i^{s_\omega},$$

Where s_ω, $q(\omega)$ are chosen randomly in Z_p^*.

If $\omega \in W - \Gamma'$:

$$\begin{aligned} a_\omega &= \prod_{\omega \in \Gamma', i = index(\omega)} g^{q(i)\Delta_{i,S}(0)} \cdot (g^{\Phi})^{\Delta_{0,S}(0)} \cdot H_1(pk_i, \omega)^{s_\omega} \\ &= g^{q(\omega)} \cdot H_1(pk_i, \omega)^{s_\omega}, \\ b_\omega &= pk_i^{s_\omega}. \end{aligned}$$

Where s_ω are randomly chosen in Z_p^*.

Next, $\mathcal{B}$ computes $rk_1, rk_2, rk_3, t', rk_4$ as the Enc algorithm. Thus, B can derive a valid re-encryption key for W.

- First level decryption query(pk_j, CT_j): The same as in Lemma 1.
- Challenge. Once $\mathcal{A}$ decided that Phase 1 is over, it outputs two equal length plaintexts (m_0, m_1). $\mathcal{B}$ chooses a random value $\theta^* \in \{0,1\}^k$ and computes $rk_1^*, rk_2^*, rk_3^*, t'^*, rk_4^*$ as the Enc algorithm. Let $r^* = b/a^2$, $\mathcal{B}$ randomly choose $\beta \in \{0,1\}$, and responds as follows:

$$\begin{aligned} C_1^* &= B^{\alpha_0} = A_2^{\alpha_0 \frac{b}{a^2}} = g_1^{r^*}, \\ C_2'^* &= T^{H_3(\theta^*)} = e(g,g)^{H_3(\theta^*)\frac{b}{a^2}} = e(g,g)^{r^* H_3(\theta^*)}, \\ C_3^* &= [F(T, C_1^*)]_{l-k} \parallel ([F(T, C_1^*)]^k \oplus m_\beta), \\ h^* &= H_2(C_1^*, C_3^*), \qquad t^* = -\frac{\alpha_1 h^* + \alpha_3}{\alpha_2}, \\ C_5^* &= B^{\beta_1 h^* + \beta_2 t^* + \beta_3} = (A_2^{\beta_1 h^* + \beta_2 t^* + \beta_3})^{r^*} = (g_2^{h^*} g_3^{t^*} g_4)^{r^*}, \end{aligned}$$

Output the challenge ciphertext $CT_{i^*} = (t^*, C_1^*, C_2'^*, C_3^*, C_5^*, rk_1^*, rk_2^*, rk_3^*, t'^*, rk_4^*)$.

- Query phase 2. $\mathcal{A}$ continues making queries as in the query phase 1 with the restrictions described in the IND-L2-CCA game.
- Guess. $\mathcal{A}$ outputs the guess β', if $\beta' = \beta$, then output 1 meaning$T = e(g,g)^{b/a^2}$; else output 0 meaning $T = e(g,g)^r$.

Observe that, if $T = e(g,g)^{b/a^2}$, CT_i^* is indeed a valid challenge ciphertext under public key pk_{i^*}. On the other hand, when T is uniform in G_2, the challenge ciphertext CT_i^* is independent of β in the adversary's view.

4. Conclusions and Open Questions

In this paper, we solve the problem left by Fang, Wang, Ge and Ren by proposing a non-interactive fuzzy conditional proxy re-encryption. Furthermore, our scheme is proved CCA secure without random oracles. Many interesting questions still remain to be solved.

Anonymous conditions. Designing FC-PRE schemes with anonymous conditions is very necessary. The technique introduced in [13] might be possible approaches to achieve CCA-secure. We leave it as our future work.

Without pairings. Designing FC-PRE schemes without relying on pairings is also an interesting work, as pairing takes a more computation cost compared with modular exponentiation.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61272083, 61300236), the National Natural Science Foundation of Jiangsu (No. BK20130809), the National Science Foundation for Post-doctoral Scientists of China (No. 2013M530254), the National Science Foundation for Post-doctoral Scientists of Jiangsu (No. 1302137C), and the China Postdoctoral Science special Foundation (No. 2014T70518).

References

[1] M. Blaze, G. Bleumer and M. Strauss, "Divertible protocols and atomic proxy cryptography", Proceedings of EUROCRYPT 1998, Finland, **(1998)** May 31 – June 4.
[2] G. Ateniese, k. Fu, M. Green and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage", Proceedings of the 12th Annual Network and Distributed System Security Symposium 2005, San Diego, California, USA, **(2005)** February 3-4.
[3] J. Weng, R.H. Deng and C. Chu, "Conditional proxy re-encryption secure against chosen-ciphertext attack", Proceedings of the 4th International Symposium on ACM Symposium on Information, Computer and Communications Security 2009, Sydeny, Australia, **(2009)** March 10-12.
[4] B. Libert and D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy re-encryption", Proceedings of PKC 2008, Barcelona, Spain, **(2008)** March 9-12.
[5] Q. Tang, "Type-based proxy re-encryption and its construction", Proceedings of INDOCRYPT 2008, Kharagpur, India, **(2008)** December 14-17.
[6] L. Fang, W. Susilo and J. Wang, "Anonymous conditional proxy re-encryption without random oracle", Proceedings of ProvSec 2009, Guangzhou, China, **(2009)** November 11-13.
[7] L. Fang and J. Wang, "Fuzzy conditional proxy re-encryption", Journal of Sci China Inf Sci, Vol.56, no.11 **(2013)**, pp.1-13.
[8] R. Canetti, O. Goldreich and S. Halevi, "The random oracle methodology, revisited", Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, STOC 1998, Texas, USA, **(1998)** May 23-26.
[9] D. Boneh and X. Boyen, "Efficient selective-ID based encryption without random oracles", Proceedings of EUROCRYPT 2004 Interlaken, Switzerland, **(2004)** May 2-6.
[10] D. Boneh and M. Franklin, M, "Identity-based encryption from the weil pairing", Proceedings of CRYPTO 2001, Santa Barbara, California, USA, **(2001)** August 19-23.
[11] J. Weng, M. Chen, Y. Yang, R. DENG, K. Chen and F. Bao, "CCA-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles", Journal of Sci China Inf Sci, vol.53, no.11, **(2010)**, pp.593-606.
[12] A. Sahai and B. Waters, B, "Fuzzy identity-based encryption", Proceedings of Eurocrypt 2005, Aarhus, Denmark, **(2005)** May 22-26.
[13] C. Gentry, "Practical identity-based encryption without random oracles", Proceedings of EUROCRYPT 2006, Saint Petersburg, Russia, **(2006)** May 28- June 1.
[14] K. Liang, L. Fang, D.S. Wong and W. Susilo, "A Ciphertext-Policy Attribute-Based Proxy Re-Encryption with Chosen-Ciphertext Security", Proceedings of the 5th International Conference on Intelligent Networking and Collaborative Systems 2013, Xi'an, China, **(2013)** September 9-13.
[15] B. Waters, "A CCA-Secure Identity-Based Conditional Proxy Re-Encryption without Random Oracles", Proceedings of ICISC 2012, Seoul, Korea, **(2012)** November 28-30.
[16] T. Mizuno and H. Doi, "Hybrid proxy re-encryption scheme for attribute-based encryption". Proceedings of ICISC 2011, Seoul, Korea, **(2011)** November 30 – December 2.

[17] J. Shao, P. Liu and Y. zhou, "Achieving key privacy without losing CCA security in proxy re-encryption", Journal of systems and software, vol.85, no.8, **(2012)**, pp.655-665.
[18] G. Hanaoka, Y. Kawai, N. Kunihiro, T. Matsuda, J. Weng, R. Zhang and Y. Zhao, "Generic constructions of chosen ciphertext secure proxy re-encryption", Proceedings of CT-RSA 2012, San Francisco, USA, **(2012)** Feb 27-Mar 2.