International Journal of Security and Its Applications
Vol. 10, No. 5 (2016) pp.347-360
http://dx.doi.org/10.14257/ijsia.2016.10.5.33

A Chosen-Ciphertext Secure Fuzzy Identity-Based Proxy Re-Encryption Scheme

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

Green and Ateniese introduced the notion of identity-based proxy re-encryption (IB-PRE), whereby the proxy can covert a ciphertext encrypted under the delegator's identity to an encryption under the delegatee's identity of the same message. In some situations, biometric, such as dactylogram, was used as identities. However, these biometric identities will inherently have some noise when they are sampled each time. To make identity-based proxy re-encryption flexible on identities, we introduced a new primitive called fuzzy identity-based proxy re-encryption (FIB-PRE), in which an identity is viewed as a set of descriptive attributes. In a fuzzy identity-based proxy re-encryption scheme, an identity W' can decrypt a ciphertext re-encrypted under another identity W, if and only if W and W' are close to each other as measured by the "set overlap" distance metric. In this work, we first formulate the security model of a FIB-PRE scheme. Finally, we present a construction of FIB-PRE and prove its CCA security under the decisional bilinear Diffie-Hellman (DBDH) assumption in the random model.

***Keywords**: Proxy re-encryption; Identity-based proxy re-encryption; Fuzzy identity-based proxy re-encryption; Chosen-ciphertext security*

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]. If a re-encryption key can transform Alice's ciphertext to Bob's ciphertext and vice versa, then the scheme is *bidirectional*. Otherwise, it is *unidirectional*. PRE can be further divided into two categories: *multi-hop* and *single-hop* PRE. In a *multi-hop* PRE scheme, ciphertexts can be transformed from Alice to Bob and then to Carol, and so on. If ciphertexts can be transformed to Bob only, then the scheme is *single-hop*.

To simplifying the public key certificates management, Green and Ateniese [3] introduced the notion of identity-based proxy re-encryption. In an IB-PRE scheme, the proxy is allowed to convert an encryption under Alice's identity into the encryption under Bob's identity. Chun and Tzeng [4] introduced two IB-PRE schemes, which are both secure in the standard model based on Waters IBE scheme [5]. However, both Green and Ateniese's scheme [3] and Chun and Tzeng's scheme [4] are not collusion resistant. The collusion resistant property means that, the proxy colluding with a set of delegatees is only able to decrypt the delegator's ciphertext, but cannot reveal the delegator's private key. Shao and Cao [6] proposed a collusion resistant multi-use unidirectional IB-PRE from a CPA-secure non-anonymous hierarchical identity based encryption. To capture fine-grain delegations, Liang, Liu, Tang, Wong and Tang [7] proposed an identity-based

[*] Corresponding author: gecp@nuaa.edu.cn

conditional proxy re-encryption scheme, whereby ciphertext only satisfying a special condition can be convert from Alice's identity to Bob's identity.

As described in [8], in some situations a user's biometric, such as iris scan, may used as his identity. Since biometric measurements are noisy, an error-tolerance on identities may need to enable the delegatee to decrypt a ciphertext re-encrypted with a slightly distance. Unfortunately, none of the existed IB-PRE schemes achieve the fuzzy property. This work focuses filling such a gap. To our best knowledge, this is the first solution for fuzzy identity-based proxy re-encryption. Here, we compare our scheme with previous IB-PRE schemes in terms of security and properties.

Table 1. Comparison with [3, 5, 6, 7]

Schemes	Fuzzy on identity?	CPA or CCA	Collusion resistant?
IB-PRE [3]	No	CCA	No
IB-PRE [5]	No	CCA	No
IB-PRE [6]	No	CCA	Yes
IB-CPRE [7]	No	CCA	Yes
Our scheme	Yes	CCA	Yes

1.1. Our Contributions

In this work, we first formalize the definition and security models for FIB-PRE scheme. In our definition, a proxy re-encrypted a ciphertext from identity W_1 to identity W_2. Then, an identity W_2' can decrypt the ciphertext, if and only if W_2 and W_2' are within a certain distance of each other as judged by some metric. Furthermore, we present a unidirectional single-hop FIB-PRE scheme and prove its against chosen-ciphertext attack (CCA) security under the decisional bilinear Diffie-Hellman assumption.

Here we further describe the difficulty of constructing a CCA-secure unidirectional single-hop FIB-PRE scheme. One may think it is trivial to construct a CCA-secure FIB-PRE by combing a fuzzy IBE and a CCA secure proxy re-encryption. However it is not true for some reasons. Firstly, as in former proxy re-encryption schemes [2, 9], the secret and public key for a user i is set to be x_i and g^{x_i}, where $x_i \in Z_p^*$ and $g^{x_i} \in G$. The mast secret $sk_j - sk_i = x_j - x_i \in Z_p^*$ of a re-encryption key is set to be the exponent for element $g^{x_j - x_i}$. However, in the identity-based setting [4, 10], the secret key for an identity is set be an element in G and the public key is ID. The element $sk_j - sk_i$ cannot be used as the exponent anymore. Secondly, the former fuzzy IBE scheme [8] only achieves chosen-plaintext attack (CPA) secure, thus we cannot combine the fuzzy IBE scheme [8] with a CCA-secure PRE scheme straightly.

To solve the above problems, we first use the F-O conversion [11] to get a CCA-secure fuzzy IBE scheme. Secondly, we use the “token-controlled” method to achieve a FIBE-PRE scheme. Formally, in the re-encryption key generation algorithm, we select a random token θ as the secret for the polynomial. And also we let θ involved in the delegatee's identity. When performing the re-encryption, the original ciphertext is “semi-decrypted” by the proxy using the re-encryption key. At last, the re-encrypted ciphertext is only relevant to the token θ, which is encrypted under W_2. Using the secret key of W_2', he can recover θ and furthermore recover the message. The details will be described in section 3.

1.2. Paper Organization

The rest of the paper is organized as follows. We first provide the basic primitives and the security model for an FIB-PRE scheme in section 2. In Section 3, we present our FIB-PRE scheme and prove its CCA security. Finally, we conclude the paper in section 4.

2. Preliminaries

2.1. Negligible function

A function $\varepsilon(n) : N \longrightarrow R$ is said to be negligible if for all positive integer $c \in N$ there exists a $n_c \in N$ such that $\varepsilon(n) < n_c$ for all $n > n_c$.

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 $e : G \times G \longrightarrow G_T$ with the following properties [10, 12]:

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

2.3. Decisional bilinear Diffie-Hellman (DBDH) assumption

Let $e : G \times G \longrightarrow G_T$ be a bilinear map. The DBDH assumption [8] relative to G and G_T holds, if the following probability is negligible:

$$| \; Pr[a, b, c, r \xleftarrow{R} Z_p^*; \quad T_0 = e(g, g)^{abc}; \quad T_1 = e(g, g)^r;$$
$$z \in \{0, 1\}; \quad z' \leftarrow \mathcal{A}(g, g^a, g^b, g^c, T_z) : \quad z = z'] \; | - \frac{1}{2}.$$

2.4. Fuzzy identity-based proxy re-encryption

In this subsection, we will provide the definition of fuzzy identity-based proxy re-encryption. In our scheme, an identity is viewed as a set of descriptive attributes.

Definition 1 (FIB-PRE). A (single-use) fuzzy identity-based proxy re-encryption scheme [3] consists of the following algorithms:

- $Setup(\lambda)$: The $Setup$ algorithm is run by the private key generator (PKG), on input a security parameter λ, the global public parameters PP including the threshold d, and mast secret key msk are outputted.
- $KeyGen(msk, W)$: On input the mast secret key msk and an identity W, output the private key sk_W for identity W.
- $Enc(m, W')$: On input a message $m \in M$ and an identity W', output an original (level 2) ciphertext $C_{W'}^{(2)}$. Here M denotes the message space.
- $RKeyGen(sk_{W_1}, W_1, W_2)$: On input identities W_1, W_2 and the secret key sk_{W_1}, output the re-encryption key $rk_{W_1 \rightarrow W_2}$.

- $ReEncrypt(C^{(2)}_{W'_1}, W_1, rk_{W_1 \to W_2})$: On input a re-encryption key $rk_{W_1 \to W_2}$ and a level 2 ciphertext $C^{(2)}_{W'_1}$ under identity W'_1, output an re-encrypted ciphertext (level 1) $C^{(1)}_{W_2}$.
- $Dec2(W_1, sk_{W_1}, C^{(2)}_{W'_1})$: On input an identity W_1, a private key sk_{W_1} and a second level ciphertext $C^{(2)}_{W'_1}$, output the plaintext m or an error symbol $\perp$.
- $Dec1(W'_2, sk_{W'_2}, C^{(1)}_{W_2})$: On input an identity W'_2, a private key $sk_{W'_2}$ and a first level ciphertext $C^{(1)}_{W_2}$, output the plaintext m or an error symbol $\perp$.

Note that we omit the global parameters PP as the other algorithms' input for simplicity. The correctness of FIB-PRE means that, for any message $m \in M$, $sk_{W_1} \leftarrow KeyGen(msk, W_1)$, $sk_{W'_2} \leftarrow KeyGen(msk, W'_2)$, $C^{(2)}_{W'_1} = Enc(m, W'_1)$ and $rk_{W_1 \to W_2} \leftarrow RKeyGen(sk_{W_1}, W_1, W_2)$, where $|W_1 \cap W'_1| \geq d$ and $|W_2 \cap W'_2| \geq d$.

$$Pr[Decrypt(C^{(2)}_{W'_1}, sk_{W_1}) = m] = 1, \text{ and}$$

$$Pr[Decrypt(sk_{W'_2}, ReEncrypt(C^{(2)}_{W'_1}, rk_{W_1 \to W_2})) = m] = 1.$$

2.5. Security notion for FIB-PRE

In this subsection, we will provide the game-based security definition of FIB-PRE scheme. As in [3, 5], our security model considers the CCA-security of the second level ciphertext and the first level ciphertext in the selective-identity model. The two security games are denoted as IND-level2-CCA and IND-level1-CCA individually. As in [8], in a selective-identity model, the adversary $\mathcal{A}$ declares the target identity W^* that he wishes to be challenged at the beginning of the security game.

Definition 2 (IND-level2-CCA). An (single-use) FIB-PRE scheme is IND-level2-CCA secure if no probabilistic polynomial time (PPT) adversary $\mathcal{A}$ can win the game below with non-negligible advantage. Next in the game, let $\mathcal{B}$ be the challenger and λ be the security parameter.

1. Setup. The challenger $\mathcal{B}$ Runs the $Setup(\lambda)$ algorithm to get (PP, msk), and returns PP to the adversary $\mathcal{A}$.

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

(a)$Extract(W)$: run the $KeyGen(msk, W)$ algorithm to get sk_W, and returns sk_W to the adversary $\mathcal{A}$.

(b) $RKExtract(W_1, W_2)$: run the $RKeyGen(sk_{W_1}, W_1, W_2)$ algorithm to get $rk_{W_1 \to W_2}$, and returns the result to $\mathcal{A}$.

(c) $ReEnc(W_1, W_2, C^{(2)}_{W_1})$: run the $ReEncrypt(C^{(2)}_{W_1}, rk_{W_1 \to W_2})$ algorithm to get a first level ciphertext $C^{(1)}_{W_2}$, and returns $C^{(1)}_{W_2}$ to $\mathcal{A}$.

(d) $Dec2(W'_1, C^{(2)}_{W_1})$: run the $Dec2(ID_{W'_1}, sk_{W'_1}, C^{(2)}_{W_1})$ algorithm to get the underlying message m, and returns the result to $\mathcal{A}$.

(e)$Dec1(W_2', C_{W_2}^{(1)})$: run the $Dec1(W_2', sk_{W_2'}, C_{W_2}^{(1)})$ algorithm to get the underlying message m, and returns the result to $\mathcal{A}$.

3. Challenge. Once $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal length message (m_0, m_1). The challenger $\mathcal{B}$ chooses a random bit $b \in \{0, 1\}$ and sends the challenge ciphertext $C_{W^*}^{(2)*} = Enc(W^*, m_b)$ to $\mathcal{A}$, if the following queries are never made:

-- $Extract(W)$, where $|W \cap W^*| \geq d$, and

-- $RKExtract(W^*, W_2)$ and $Extract(W_2')$, where $|W_2' \cap W_2| \geq d$.

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

(a)$Extract(W)$, where $|W \cap W^*| \geq d$;

(b)$RKExtract(W^*, W_2)$ and $Extract(W_2')$, where $|W_2' \cap W_2| \geq d$;

(c) $ReEnc(W^*, W_2, C_{W^*}^{(2)})$ and $Extract(W_2')$, where $|W_2' \cap W_2| \geq d$;

(d)$Dec2(W, C_{W^*}^{(2)*})$, where $|W \cap W^*| \geq d$;

(e)$Dec1(W_2', C_{W_2}^{(1)})$, where $|W_2' \cap W_2| \geq d$, if $C_{W_2}^{(1)}$ is a derivative of $C_{W^*}^{(2)*}$. As defined in [18], derivative of $C_W^{(2)} = Enc(m, W)$ is defined as follows:

-- $C_W^{(2)}$ is a derivative of itself;

-- If the adversary $\mathcal{A}$ has issued $RKExtract(W, W_2)$ query to obtain a re-encryption key $rk_{W \to W_2}$, and compute $C_{W_2}^{(1)} \longleftarrow ReEnc(W, W_2, C_W^{(2)})$, then $C_{W_2}^{(1)}$ is a derivative of $C_W^{(2)}$.

-- If the adversary $\mathcal{A}$ has issued $ReEnc(W, W_2, C_W^{(2)})$ and obtained $C_{W_2}^{(1)}$. Then $C_{W_2}^{(1)}$ is a derivative of $C_W^{(2)}$.

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

We say that a FIB-CPRE scheme is IND-level2-CCA secure, if the following probability is negligible for all probabilistic polynomial time adversary $\mathcal{A}$:

$$Adv_{\mathcal{A}}^{IND-level2-CCA}(\lambda) = |Pr[b' = b] - 1/2|.$$

Definition 3 (IND-level1-CCA). The IND-level1-CCA considers the indistinguishability of the original ciphertext. The adversary is privided with an original ciphertext in the challenge phase. A complementary definition of security captures the inability to distinguish the re-encrypted 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 proxy re-encryption schemes, the adversary is provided with access to all re-encryption keys, thus the re-encryption oracle becomes useless; (2) The adversary can re-encrypted any second level ciphertext, and the make a level1 decryption oracle, thus the level 2 decryption oracle is unnecessary;

The FIB-CPRE scheme is said to be IND-CCA secure if both $Adv_{\mathcal{A}}^{IND-level2-CCA}(\lambda)$ and $Adv_{\mathcal{A}}^{IND-level1-CCA}(\lambda)$ are negligible.

3. Our proposed FIB-PRE Scheme

3.1. Construction

We first 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 FIB-PRE consists of the following algorithms:

- $Setup(\lambda)$: Let λ be the security parameter, and (p, g, G, G_T, e) be the bilinear map parameters. d is the threshold, which means an identity W' is able to decrypt a ciphertext re-encrypted under identity W, if and only if $|W \cap W'| \geq d$. Let $M \in \{0,1\}^k$ be the message space, $H_1 : \{0,1\}^k \times G_T \longrightarrow Z_p^*$, $H_2 : G_T \longrightarrow \{0,1\}^k$, $H_3 : Z_p^* \times G \times G \times G_T \times \{0,1\}^k \longrightarrow G$, $H_4 : Z_p^* \longrightarrow G$, $H_5 : \{0,1\}^k \longrightarrow G$ be hash functions. Randomly chooses $y \in Z_p^*, Z \in G$, and computes $Y = e(g, Z)^y$. An identity W will be a set of described attributes \$. Let the public key is $PP = (\lambda, p, g, G, G_T, e, Z, d, H_1, H_2, H_3, H_4, H_5)$ and the mast secret key is $msk = y$.

- $KeyGen(msk, W)$: Given an identity W, the PKG randomly chooses a $d-1$ degree polynomial $q(x)$, such that $q(0) = y$. For each $\omega \in W$, selects a random value $s_\omega \in Z_p^*$, and computes

$$d_1 = (d_{1,\omega} = Z^{q(\omega)} H_4(\omega)^{s_\omega})_{\omega \in W} \qquad d_2 = (d_{2,\omega} = g^{s_\omega})_{\omega \in W},$$

  and sets the secret key $d = (d_1, d_2)$.

- $Enc(m, W')$: To encrypt a message $m \in M$ under identity W', the sender selects a random $R \in G_T$, and computers $t = H_1(m, R)$, and outputs the second level ciphertext $C_{W'}^{(2)} = (C_1, C_2, C_3, C_4, C_5, C_6)$ as follows:

$$C_1 = W', \qquad C_2 = g^t, \qquad C_3 = (C_\omega = H_4(\omega)^t)_{\omega \in W'},$$
$$C_4 = R \cdot Y^t, \qquad C_5 = m \oplus H_2(R), \qquad C_6 = H_3(C_1, C_2, C_3, C_4, C_5)^t.$$

- $RKeyGen(sk_{W_1}, W_1, W_2)$: On input two identities W_1, W_2, and W_1's private key $sk_{W_1} = (d_1, d_2)$. The $ReKeyGen$ algorithms proceeds as follows:

  (a) Select random values $\rho \in Z_p^*, \theta \in \{0,1\}^k$, computes

$$rk_0 = H_5(\theta), \qquad rk_1 = (rk_{1,\omega} = d_{1,\omega} H_4(\omega)^\rho)_{\omega \in W_1},$$
$$rk_2 = g^\rho, \qquad rk_3 = (rk_{3,\omega} = d_{2,\omega})_{\omega \in W_1}.$$

  (b) Select random values $\theta \in \{0,1\}^k, R' \in G_T$, computes $t' = H_1(\theta, R')$ and sets

$$rk_4 = W_2, \qquad rk_5 = g^{t'}, \qquad rk_6 = (rk_{6,\omega} = H_4(\omega)^{t'})_{\omega \in W_2},$$
$$rk_7 = R' \cdot Y^{t'}, \qquad rk_8 = \theta \oplus H_2(R'), \qquad rk_9 = H_3(rk_4, rk_5, rk_6, rk_7, rk_8)^{t'}.$$

  (c) Return $rk_{W_1 \to W_2} = (rk_0, rk_1, rk_2, rk_3, rk_4, rk_5, rk_6, rk_7, rk_8, rk_9)$.

- $ReEnc(C_{W_1'}^{(2)}, W_1, rk_{W_1 \to W_2})$: On input a re-encryption key $rk_{W_1 \to W_2} = (rk_0, rk_1,$ $rk_2, rk_3, rk_4, rk_5, rk_6, rk_7, rk_8, rk_9)$ and a second level ciphertext $C_{W_1'}^{(2)} = (C_1,$

$C_2, C_3, C_4, C_5, C_6)$ under the keyword set W_1'. Recover $W_1' = C_1$, for all $\omega \in W'$, the proxy checks whether the following equalities hold:

$$e(H_4(\omega), C_2) \overset{?}{=} e(C_\omega, g), \qquad (1)$$

$$e(H_3(C_1, C_2, C_3, C_4, C_5), C_2) \overset{?}{=} e(C_6, g). \qquad (2)$$

If not, outputs $\perp$. Otherwise if $|W_1 \cap W_1'| < d$, outputs $\perp$, else chooses an arbitrary d elements subset, $S \subseteq W_1 \cap W_1'$, and computes

$$C_4' = C_4 \cdot e(rk_0, C_2) / \prod_{\omega \in S} \left(\frac{e(rk_{1,\omega}, C_2)}{e(C_\omega, rk_{3,\omega}) \cdot e(C_\omega, rk_2)} \right)^{\Delta_{\omega,S}(0)}.$$

Finally, output the re-encrypted ciphertext

$$C_{W_2}^{(1)} = (C_2, C_4', C_5, rk_4, rk_5, rk_6, rk_7, rk_8, rk_9).$$

- $Dec2(W_1, sk_{W_1}, C_{W_1'}^{(2)})$: First check the validity of the ciphertexts as in equations (1)-(2). If the verification fails, output $\perp$ and abort. Otherwise, proceeds

  If $|W_1 \cap W_1'| < d$, output $\perp$, else chooses an arbitrary d elements subset, $S \subseteq W_1 \cap W_1'$, and computes

$$R = C_4 / \prod_{\omega \in S} \left(\frac{e(d_{1,\omega}, C_2)}{e(C_\omega, d_{2,\omega})} \right)^{\Delta_{\omega,S}(0)}, \quad m = C_5 \oplus H_2(R), \quad t = H_1(m, R)$$

  and checks whether

$$C_2 \overset{?}{=} g^t, \quad (C_\omega \overset{?}{=} H_4(\omega)^t)_{\omega \in W'}, \quad C_6 \overset{?}{=} H_3(C_1, C_2, C_3, C_4, C_5)^t$$

  hold. If yes, returns m, else returns $\perp$.

- $Dec1(W_2', sk_{W_2'}, C_{W_2}^{(1)})$: On input a first level ciphertext $C_{W_2}^{(1)} = (C_2, C_4', C_5, rk_4, rk_5, rk_6, rk_7, rk_8, rk_9)$, an identity W_2' and the secret key $sk_{W_2'} = (d_{1,\omega}', d_{2,\omega}')$, proceeds

  (a) For each $\omega \in W_2$, Verifies

$$e(H_4(\omega), rk_5) \overset{?}{=} e(rk_{6,\omega}, g), \qquad (3)$$

$$e(H_3(rk_4, rk_5, rk_6, rk_7, rk_8), rk_5) \overset{?}{=} e(rk_9, g). \qquad (4)$$

  If the equations (3)-(4) do not hold, output $\perp$ and abort. Otherwise, proceeds.

  (b) If $|W_2 \cap W_2'| < d$, output $\perp$, else chooses an arbitrary d elements subset, $S' \subseteq W_2 \cap W_2'$, and computers

$$R' = rk_7 / \prod_{\omega \in S'} \left(\frac{e(d_{1,\omega}', rk_5)}{e(rk_{3,\omega}, d_{2,\omega}')} \right)^{\Delta_{\omega,S}(0)},$$

$$\theta = rk_7 \oplus H_2(R'), \quad t' = H_1(\theta, R').$$

  Then, checks whether

$$rk_5 \overset{?}{=} g^{t'}, \quad (rk_{6,\omega} \overset{?}{=} H_4(\omega)^{t'})_{\omega \in W_2'}, \quad rk_9 \overset{?}{=} H_3(C_1, C_2, C_3, C_4, C_5)^{t'}$$

  hold. If not returns $\perp$ and abort, else

(c) Compute $R = C_4'/e(C_2, H_5(\theta))$, $m = C_5 \oplus H_2(R)$, $t = H_1(m, R)$. Output m if $C_2 = g^t$, 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 FIB-PRE scheme

In this subsection, we prove the IND-CCA security for our scheme without random oracles.

Theorem 1. Our FIB-PRE scheme is IND-CCA secure assuming the DBDH assumption holds, and H_1, H_2, H_3, H_4, H_5, are collision-resistant hash functions.

Lemma 1. Our FIB-PRE scheme is IND-level2-CCA secure assuming the DBDH assumption holds, and H_1, H_2, H_3, H_4, H_5, are collision-resistant hash functions.

Proof. Suppose there is an adversary $\mathcal{A}$ who can break the IND-leval2-CCA security of our FIB-PRE scheme with non-negligible probability ϵ. We can construct a simulator $\mathcal{B}$ to solve DBDH problem with probability ϵ', such that $\epsilon' \geq \frac{\epsilon}{\grave{e}(1+q_e)}$. Simulator $\mathcal{B}$ inputs a DBDH instance (g, g^a, g^b, g^c, T) and has to distinguish $T = e(g,g)^{abc}$ from a random value in G_T. $\mathcal{A}$ first generates the target identity W^* that it intends to attack. $\mathcal{B}$ selects a random value $\gamma \in Z_p^*$, and sets $Z = g^{b+\gamma}$. $\mathcal{B}$ also set $Y = e(g^a, Z)$. The intuition behind this assignment is that the mast secret key is set to be $y = a$ for some unknown value a. The random oracles H_1, H_2, H_3, H_4, H_5 are controlled by $\mathcal{B}$ as follows.

If $\mathcal{A}$ queries (m, R) to the random oracle H_1, $\mathcal{B}$ searches H_1^{List} for an entry (m, R, t). If such an entry exists , returns t as the answer. Otherwise, selects a random $t \in Z_p^*$, and returns t as the answer. Finally, adds (m, R, t) to H_1^{List}.

If $\mathcal{A}$ queries (R) to the random oracle H_2, $\mathcal{B}$ searches H_2^{List} for an entry (R, κ). If such an entry exists , returns κ as the answer. Otherwise, selects a random $\kappa \in \{0,1\}^k$, and returns κ as the answer. Finally, adds (R, κ) to H_2^{List}.

If $\mathcal{A}$ queries $(C_1, C_2, C_3, C_4, C_5)$ to the random oracle H_3, $\mathcal{B}$ searches H_3^{List} for an entry $(C_1, C_2, C_3, C_4, C_5, \phi, \psi)$. If such an entry exists , returns ψ as the answer. Otherwise, selects a random $\phi \in Z_p^*$, and computers $\psi = g^\phi$. Returns ψ as the answer and adds $(C_1, C_2, C_3, C_4, C_5, \phi, \psi)$ to H_3^{List}.

If $\mathcal{A}$ queries (ω) to the random oracle H_4, $\mathcal{B}$ searches H_4^{List} for an entry (ω, ν, υ). If such an entry exists , returns υ as the answer. Otherwise, selects a random $\nu \in Z_p^*$. If $\omega \in W^*$, computes $\upsilon = g^\nu$, else computes $\upsilon = Zg^\nu$. Returns υ as the answer, and adds (ω, ν, υ) to H_4^{List}.

If $\mathcal{A}$ queries (θ) to the random oracle H_5, $\mathcal{B}$ searches H_5^{List} for an entry (θ, ϑ). If such an entry exists , returns ϑ as the answer. Otherwise, selects a random $\vartheta \in G$, and returns ϑ as the answer. Finally, adds (θ, ϑ) to H_5^{List}\$.

Next, $\mathcal{B}$ maintains the following tables which are initially empty.

- Key^{List}: records the tuples (β, W, sk_W), which are the information of secret keys;
- $ReKey^{List}$: records the tuples $(\sigma, W_1, W_2, rk_{W_1 \to W_2}, \theta, flag_1)$, which are the result of the queries to $RKExtract(W_1, W_2)$, where $flag_1 = 1$ denotes the re-encryption key is a valid one, and $flag_1 = 0$ denotes the re-encryption key is a random value.

- $ReEnc^{List}$: records the tuples$(W_1, W_2, C^{(2)}_{(W_1)}), C^{(1)}_{(W_2)}, flag_2)$, which are the result of the queries to $ReEnc(W_1, W_2, C^{(2)}_{(W_1)}))$, where $flag_2 = 1$ denotes the re-encrypted ciphertext is generated under a valid re-encryption key, and $flag_2 = 0$ denotes the re-encrypted ciphertext is generated under a random re-encryption key.

1. **Setup:** Let λ be the security parameter, d be the threshold, and (p, g, G, G_T, e) be bilinear map. The global public parameters are
$$PP = (\lambda, d, p, g, G, G_T, e, Y, Z, H_1, H_2, H_3, H_4, H_5).$$
2. **Query phase 1:** $\mathcal{A}$ issues a series of queries to which $\mathcal{B}$ responds as follows:

   (a) $Extract(W)$: $\mathcal{B}$ first checks whether $|W \cap W^*| < d$, if not output $\perp$ and abort. Otherwise $\mathcal{B}$ searches Key^{List}, if $(1, W, sk_W)$ exists in Key^{List}, returns sk_W as the result. else, $\mathcal{B}$ generates a biased coin β so that $Pr[\beta = 1] = \delta$ for some δ that will be determined later.

   -- If $\beta = 0$, $\mathcal{B}$ outputs a random bit and aborts.

   -- If $\beta = 1$, $\mathcal{B}$ first defines three sets Γ, Γ', S in the following manner
$$\Gamma = W \cap W^*.$$
   Γ' be any set such that $\Gamma \subseteq \Gamma' \subseteq W$ and $|\Gamma'| = d - 1$, and $S = \Gamma' \cup \{0\}$. For each $\omega \in \Gamma'$, sets
$$d_{1,\omega} = Z^{\lambda_\omega} H_4(\omega)^{s_\omega}, \qquad d_{2,\omega} = g^{s_\omega},$$
   where λ_ω, s_ω are randomly chosen in Z_p^*. The intuition behind these assignments is that we implicitly choosing a randomly $d-1$ degree polynomial $q(x)$ by selecting $d-1$ points $q(\omega) = \lambda_\omega$ in Γ' in addition to setting $q(0) = a$ for some unknown a.

   For each $\omega \in W - \Gamma'$, computes
$$d_{1,\omega} = \left(\prod_{j \in \Gamma'} Z^{\lambda_j \Delta_{j,S}(\omega)}\right) \cdot \left(Z^{s'_\omega} g^{\nu s'_\omega} (g^a)^{-\nu}\right)^{\Delta_{0,S}(\omega)},$$
$$d_{2,\omega} = g^{s'_\omega \Delta_{0,S}(\omega)} (g^a)^{-\Delta_{0,S}(\omega)},$$
   where s'_ω is randomly chosen in Z_p^*. Let $s_\omega = (s'_\omega - a)\Delta_{0,S}(\omega)$, and $q(x)$ defined as above. We then have
$$\begin{aligned} d_{1,\omega} &= \left(\prod_{j \in \Gamma'} Z^{\lambda_j \Delta_{j,S}(\omega)}\right) \cdot \left(Z^{s'_\omega} g^{\nu s'_\omega} (g^a)^{-\nu}\right)^{\Delta_{0,S}(\omega)} \\ &= \left(\prod_{j \in \Gamma'} Z^{\lambda_j \Delta_{j,S}(\omega)}\right) \cdot Z^{a\Delta_{0,S}(\omega)} \cdot (Zg^\nu)^{s_\omega} \\ &= Z^{q(\omega)} H_4(\omega)^{s_\omega}, \end{aligned}$$
   and

$$\begin{aligned} d_{2,\omega} &= g^{s'_\omega \Delta_{0,S(\omega)}} (g^a)^{-\Delta_{0,S(\omega)}} \\ &= g^{(s'_\omega - a)\Delta_{0,S(\omega)}} \\ &= g^{s_\omega}. \end{aligned}$$

Therefor, the simulator is able to construct a private key for an identity W. Furthermore, the distribution of the private key for identity W is identical to that of the real scheme. Finally $\mathcal{B}$ adds $(1, W, sk_W)$ to Key^{List}.

(b) $RKExtract(W_1, W_2)$: $\mathcal{B}$ first checks that $W_1 \neq W^*$ if $(1, W'_2, sk_{W'_2})$ exists in Key^{List} and $|W'_2 \cap W_2| \geq d$, if this checking do not hold, output $\perp$ and abort. Otherwise, $\mathcal{B}$ searches whether there is a tuple $(*, W_1, W_2, rk_{W_1 \to W_2}, \theta, *)$ in $ReKey^{List}$. If yes, $\mathcal{B}$ returns $rk_{W_1 \to W_2}$ as the result, where $*$ is the wildcard. Otherwise, $\mathcal{B}$ proceeds as follows:

-- If $(1, W_1, sk_{W_1})$ exists in Key^{List}, $\mathcal{B}$ uses sk_{W_1} to generate the re-encryption key $rk_{W_1 \to W_2}$ visa algorithm $RKeyGen$ as in the real scheme. Returns the re-encryption key to $\mathcal{A}$ and adds $(*, W_1, W_2, rk_{W_1 \to W_2}, \theta, 1)$ to $ReKey^{List}$, where $\theta \in \{0,1\}^k$ is randomly chosen in the $RKeyGen$ algorithm.

-- Otherwise, $\mathcal{B}$ flips a biased coin β. If $\beta = 1$, $\mathcal{B}$ queries the $Extract(W_1)$ oracle to get sk_{W_1}, and then generates $rk_{W_1 \to W_2}$ visa algorithm $RKeyGen$ as in the real scheme. Returns the re-encryption key to $\mathcal{A}$ and adds $(1, W_1, sk_{W_1})$

and $(*, W_1, W_2, rk_{W_1 \to W_2}, \theta, 1)$ to Key^{List} and $ReKey^{List}$ respectively, where $\theta \in \{0,1\}^k$ is randomly chosen in the $RKeyGen$ algorithm. If $\beta = 0$, $\mathcal{B}$ sets $rk_0 = \xi_0$, $rk_1 = (rk_{1,\omega} = \xi_{1,\omega})_{\omega \in W_1}$, $rk_2 = \xi$, $rk_3 = (rk_{3,\omega} = \xi_{3,\omega})_{\omega \in W_1}$ for randomly chosen $\xi_0, \xi_{1,\omega}, \xi, \xi_{3,\omega} \in G$. Then $\mathcal{B}$ constructs$rk_4, rk_5, rk_6, rk_7, rk_8, rk_9$ to encrypted a random $\theta \in \{0,1\}^k$ as in the real scheme. $\mathcal{B}$ forwards the re-encryption key to $\mathcal{A}$ and adds $(*, W_1, W_2, rk_{W_1 \to W_2}, \theta, 0)$ to $ReKey^{List}$.

(c) $ReEnc(W_1, W_2, C^{(2)}_{W_1})$: $\mathcal{B}$ first searches whether there is a tuple $(W_1, W_2, C^{(2)}_{(W_1)}, C^{(1)}_{(W_2)}, *)$ in $ReEnc^{List}$. If yes, $\mathcal{B}$ returns $C^{(1)}_{(W_2)}$ as the result, where $*$ is the wildcard. Otherwise, $\mathcal{B}$ proceeds as follows:

-- If $(*, W_1, W_2, rk_{W_1 \to W_2}, \theta, *)$ exists in $ReKey^{List}$, $\mathcal{B}$ uses $rk_{W_1 \to W_2}$ to generate $C^{(1)}_{W_2}$ visa algorithm $ReEnc$ as in the real scheme. Returns $C^{(1)}_{W_2}$ to $\mathcal{A}$ and adds $(W_1, W_2, C^{(2)}_{(W_1)}, C^{(1)}_{(W_2)}, *)$ to $ReEnc^{List}$.

-- Otherwise, $\mathcal{B}$ first issues $RKExtract(W_1, W_2)$, to obtain re-encryption key $rk_{W_1 \to W_2}$. Next, generates $C^{(1)}_{(W_2)}$ as the real scheme and adds $(*, W_1, W_2, rk_{W_1 \to W_2}, \theta, *)$ and $(W_1, W_2, C^{(2)}_{(W_1)}, C^{(1)}_{(W_2)}, *)$ to the $ReKey^{List}$ and $ReEnc^{List}$ respectively.

(d) $Dec2(W_1, C^{(2)}_{W_1})$: $\mathcal{B}$ first verifies equations (1)-(2). If the equations do not hold, output $\perp$ and abort. Otherwise, $\mathcal{B}$ proceeds:

-- If $(1, W'_1, sk_{W'_1})$ exists in Key^{List}, where $|W'_1 \cap W_1| \geq d$, using $sk_{W'_1}$to recover m.

-- Otherwise, $\mathcal{B}$ searches H_1^{List}, H_2^{List} to see whether there exists (m, R, t) and (R, κ) such that

$$C_1 = W_1, \qquad C_2 = g^t, \qquad C_3 = (C_\omega = H_4(\omega)^t)_{\omega \in W_1},$$
$$C_4 = R \cdot Y^t, \quad C_5 = m \oplus H_2(R), \quad C_6 = H_3(C_1, C_2, C_3, C_4, C_5)^t.$$

If yes, sends m to the adversary to $\mathcal{A}$, otherwise output $\perp$ and abort.

(e) $Dec1(W_2, C_{W_2}^{(1)})$: $\mathcal{B}$ first verifies equations (2)-(4). If the equations do not hold, output $\perp$ and abort. Otherwise, $\mathcal{B}$ proceeds:

-- If $(W_1, W_2, C_{(W_1)}^{(2)}), C_{(W_2)}^{(1)}, 1)$ exists in $ReEnc^{List}$, return $C_{(W_1)}^{(2)}$ as the result.

-- Else, $\mathcal{B}$ searches whether $(1, W_2', sk_{W_2'})$ exists in Key^{List}, where $|W_2' \cap W_2| \geq d$, if yes, $\mathcal{B}$ recovers m using $sk_{W_2'}$. Otherwise, $\mathcal{B}$ issues $(rk_3, rk_4, rk_5, rk_6, rk_7, rk_8, rk_9)$ to the Dec2 oracle to get θ. Then using θ, $\mathcal{B}$ can recover m as in the real scheme.

3. **Challenge:** Once $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal length message (m_0, m_1), $\mathcal{B}$ first checks whether the following two conditions hold:

-- $(1, W, sk_W)$ does not exists in Key^{List}, where $|W \cap W^*| \geq d$, and

--$(*, W^*, W_2, rk_{W^* \to W_2}, \theta, *)$, and $(1, W_2', sk_{W_2'})$ do not exists in $ReKey^{List}$ and Key^{List}, where $|W_2' \cap W_2| \geq d$.

If the above two conditions do not hold, $\mathcal{B}$ outputs a random bit and aborts. Else, $\mathcal{B}$ chooses a random $b \in \{0, 1\}$, $R^* \in G_T$ and computes

$$\begin{aligned}
C_1^* &= W^*, \\
C_2^* &= g^c, \\
C_3^* &= (C_\omega = (g^c)^{\nu_\omega})_{\omega \in W^*}, \\
C_4^* &= T \cdot e(g^a, g^c)^\gamma \cdot R^*, \\
C_5^* &= m_b \oplus H_2(R^*), \\
g^{\phi^*} &= H_3(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*), \\
C_6^* &= (g^c)^{\phi^*}
\end{aligned} \quad ,$$

where $\mathcal{B}$ makes a query $(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$ to the random oracle H_3 for an entry $(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, \phi^*, \psi^*)$ and query (ω) for $\omega \in W^*$ to the random oracle H_4 for an entry $(\omega, \nu_\omega, \upsilon_\omega)$. Finally $\mathcal{B}$ sends ${C_{(W^*)}^{(2)}}^* = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*)$ to $\mathcal{A}$. Notice, let $H_1(m_b, R^*) = t^* = c$, if $T = e(g,g)^{abc}$, then

$$\begin{aligned}
C_2^* &= g^{t^*}, \\
C_3^* &= (C_\omega = (g^c)^{\nu_\omega})_{\omega \in W^*} = (H_4(\omega)^{t^*})_{\omega \in W^*}, \\
C_4^* &= T \cdot e(g^a, g^c)^\gamma \cdot R^* = e(g,g)^{abc} \cdot e(g^a, g^c)^\gamma \cdot R^* \\
&= e(g^a, g^{b+\gamma})^c \cdot R^* = R^* \cdot Y^{t^*}, \\
C_6^* &= (g^c)^{\phi^*} = H_3(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)^{t^*}
\end{aligned} \quad ,$$

This is a valid second level ciphertext for message m_b under identity W^*.

4. **Query phase 2:** $\mathcal{A}$ continues making queries as in the query phase 1 with the restrictions described in the IND-level2-CCA game.

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

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

Probability analysis. If $\mathcal{B}$ does not abort, $\mathcal{A}$ view is identical to the real scheme. We define $Abort$ be the event of $\mathcal{B}'s$ aborting during the simulation of $Extract$ query. Let q_e denote the total number of $Extract$ queries, we have $Pr[\neg Abort] \geq \delta^{q_e}(1-\delta)$, which is maximized at $\delta_{opt} = \frac{q_e}{(1+q_e)}$. Using δ_{opt}, the probability $Pr[\neg Abort]$ is at least $\frac{1}{\dot{e}(1+q_e)}$, where $\dot{e}$ is the base of the nature logarithm. Therefor, we have $\varepsilon' \geq \frac{\varepsilon}{\dot{e}(1+q_e)}$.

This completes the proof of Lemma 1.

Lemma 2. Our FIB-PRE scheme is IND-level1-CCA secure assuming the DBDH assumption holds, and H_1, H_2, H_3, H_4, H_5, are collision-resistant hash functions.

Proof. Suppose there is an adversary $\mathcal{A}$ who can break the IND-leval1-CCA security of our FIB-PRE scheme with non-negligible probability ϵ. We can construct a simulator $\mathcal{B}$ to solve DBDH problem with probability ϵ', such that $\epsilon' \geq \frac{\epsilon}{\dot{e}(1+q_e)}$. Simulator $\mathcal{B}$ inputs a DBDH instance (g, g^a, g^b, g^c, T) and has to distinguish $T = e(g,g)^{abc}$ from a random value in G_T. $\mathcal{A}$ first generates the target identity W^* that it intends to attack. $\mathcal{B}$ selects a random value $\gamma \in Z_p^*$, and sets $Z = g^{b+\gamma}$. $\mathcal{B}$ also set $Y = e(g^a, Z)$. The random oracles H_1, H_2, H_3, H_4, H_5 are defined in the same way as above.

1. **Setup:** Let λ be the security parameter, d be the threshold, and (p, g, G, G_T, e) be bilinear map. The global public parameters are

$$PP = (\lambda, d, p, g, G, G_T, e, Y, Z, H_1, H_2, H_3, H_4, H_5).$$

2. **Query phase 1:**

   (a) $Extract(W)$: $\mathcal{B}$ responds the $Extract(W)$ query in the same way as above.

   (b) $RKExtract(W_1, W_2):$ $\mathcal{B}$ searches whether there is a tuple $(*, W_1, W_2, rk_{W_1 \to W_2}, \theta, *)$ in $ReKey^{List}$. If yes, $\mathcal{B}$ returns $rk_{W_1 \to W_2}$ as the result, where $*$ is the wildcard. Otherwise, $\mathcal{B}$ proceeds as follows:

   -- If $(1, W_1, sk_{W_1})$ exists in Key^{List}, $\mathcal{B}$ uses sk_{W_1} to generate the re-encryption key $rk_{W_1 \to W_2}$ visa algorithm $RKeyGen$ as in the real scheme. Returns the re-encryption key to $\mathcal{A}$ and adds $(*, W_1, W_2, rk_{W_1 \to W_2}, \theta, 1)$ to $ReKey^{List}$, where $\theta \in \{0,1\}^k$ is randomly chosen in the $RKeyGen$ algorithm.

   -- Otherwise, $\mathcal{B}$ flips a biased coin β. If $\beta = 1$, $\mathcal{B}$ queries the $Extract(W_1)$ oracle to get sk_{W_1}, and then generates $rk_{W_1 \to W_2}$ visa algorithm $RKeyGen$ as in the real scheme. Returns the re-encryption key to $\mathcal{A}$ and adds $(1, W_1, sk_{W_1})$

   and $(*, W_1, W_2, rk_{W_1 \to W_2}, \theta, 1)$ to Key^{List} and $ReKey^{List}$ respectively, where $\theta \in \{0,1\}^k$ is randomly chosen in the $RKeyGen$ algorithm. If $\beta = 0$, $\mathcal{B}$ sets $rk_0 = \xi_0$, $rk_1 = (rk_{1,\omega} = \xi_{1,\omega})_{\omega \in W_1}$, $rk_2 = \xi$, $rk_3 = (rk_{3,\omega} = \xi_{3,\omega})_{\omega \in W_1}$ for randomly chosen $\xi_0, \xi_{1,\omega}, \xi, \xi_{3,\omega} \in G$. Then $\mathcal{B}$ constructsrk_4, rk_5, rk_6, rk_7,

rk_8, rk_9 to encrypted a random $\theta \in \{0,1\}^k$ as in the real scheme. $\mathcal{B}$ forwards the re-encryption key to $\mathcal{A}$ and adds $(*, W_1, W_2, rk_{W_1 \to W_2}, \theta, 0)$ to $ReKey^{List}$

(c) $Dec1(W_2, C^{(1)}_{W_2})$: $\mathcal{B}$ responds the $Dec1(W_2, C^{(1)}_{W_2})$ query in the same way as above.

3. **Challenge:** Once $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal length message (m_0, m_1), $\mathcal{B}$ first checks that $(1, W, sk_W)$ does not exists in Key^{List}, where $|W \cap W^*| \geq d$. If not, $\mathcal{B}$ outputs a random bit and aborts. Else, $\mathcal{B}$ chooses a random $b \in \{0,1\}, R^* \in G_T, \theta^* \in \{0,1\}^k$ and computes

$$\begin{aligned} C_2^* &= g^c, \\ C_4'^* &= R^* \cdot e(g^c, H_5(\theta^*)), \\ C_5^* &= m_b \oplus H_2(R^*), \qquad , \end{aligned}$$

Next, $\mathcal{B}$ generates $(rk_4^*, rk_5^*, rk_6^*, rk_7^*, rk_8^*, rk_9^*)$ to encrypt θ^* under identity W^*. Finally, $\mathcal{B}$ sends ${C^{(1)}_{(W^*)}}^* = (C_2^*, C_4'^*, C_5^*, rk_4^*, rk_5^*, rk_6^*, rk_7^*, rk_8^*, rk_9^*)$ to $\mathcal{A}$.

4. **Query phase 2:** $\mathcal{A}$ continues making queries as in the query phase 1 with the restrictions described in the IND-level1-CCA game.

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

Probability analysis. The same as Lemma 1.

This completes the proof of Lemma 2.

4. Conclusions

In this work, we proposed a new primitive called fuzzy identity-based proxy re-encryption which allow an identity W' to decrypt an encryption a re-encrypted ciphertext under identity W, if and only if are close to each other measured by a certain metric. We formalize the FIB-PRE IND-CCA secure model and present an FIB-PRE scheme. Our scheme allows a certain error-tolerance on identities. Also, our scheme is CCA-secure and collusion-resistant. Many interesting questions are still remaining to be solved. One is how to construct a CCA-secure FIB-PRE scheme without random oracles. Additionally, as our scheme is proved in the select-ID model, constructions proved in the adaptive-ID model also remains to be solved.

Acknowledgement

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] M. Green and G. Ateniese, "Identity-based proxy re-encryption", Proceedings of ACNS 2007, Zhuhai, China, **(2007)** June 5-8.

[4] C. Chu and W. Tzeng, "Identity-based proxy re-encryption without random oracles". Proceedings of ISC 2007, Valparaiso, Chile, **(2007)** October 9-12.

[5] B. Waters, "Efficient identity-based encryption without random oracles". Proceedings of EUROCRYPT 2005, Aarhus, Denmark, **(2005)** May 22-26.

[6] B. Waters, "Multi-use unidirectional identity-based proxy re-encryption from hierarchical identity-based encryption", Journal of Information Science, 206, **(2012)** pp.83-95.

[7] B. Waters, "A CCA-Secure Identity-Based Conditional Proxy Re-Encryption without Random Oracles", Proceedings of ICISC 2012, Seoul, Korea, **(2012)** November 28-30.

[8] A. Sahai and B. Waters, "Fuzzy identity-based encryption", Proceedings of Eurocrypt 2005, Aarhus, Denmark, **(2005)** May 22-26.

[9] B. Libert and D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy re-encryption", Proceedings of PKC 2008, Barcelona, Spain, **(2008)** March 9-12.

[10] D. Boneh, and M. Franklin, "Identity-based encryption from the weil pairing". Proceedings of CRYPTO 2001, Santa Barbara, California, USA, **(2001)** August 19-23.

[11] E. Fujisaki and T. Okamota, "How to enhance the security of public-key encryption at minimum cost", Proceedings of Eurocrypt 2004, Interlaken, Switzerland, **(2004)** May 2-6.

[12] D. Boneh and X. Boyen, "Efficient selective-ID based encryption without random oracles", Proceedings of EUROCRYPT 2004 Interlaken, Switzerland, **(2004)** May 2-6.

[13] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-policy attribute-based encryption". Proceedings of IEEE Symposium on Security and Privacy 2007, Berkeley, California, **(2007)** May 20-23.

[14] X. Liang, Z. Cao, H. Lin and J. Shao, "Attribute-Based Proxy Re-Encryption with Delegating Capabilities" Proceedings of ASIACCS 2009, Sydney, Australia, **(2009)** March 10-12.

[15] S. Luo, J. Hu and Z. Chen, "Ciphertext policy attribute-based proxy re-encryption", Proceedings of ICICS 2010, Barcelona, Spain, **(2010)** December 15-17.

[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] R. Canetti, S. Halevi and J. Katz, "Chosen-ciphertext security from identity-based encryption", Proceedings of PKC 1999, Kamakura, Japan, **(1999)** May 21-23.

[18] R. Canetti and S. Hohenberger, "Chosen-ciphertext secure proxy re-encryption", Proceedings of the 14th ACM conference on Computer and Communication, Security 2007, **(2007)** October 29-31 - November 1.

[19] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data", Proceedings of ACM CCS 2006, Alexandria, Virginia, USA, **(2006)** October 30 - November 3.

[20] J. Lai, W. Zhu and R. Deng, "New constructions for identity-based unidirectional proxy re-encryption", Journal of Computer Sci Technology, 25, **(2010)** pp.793-806.