11-2016

On the security of two identity-based conditional proxy re-encryption schemes

Kai HE
Jinan University - China

Jian WENG
Jinan University - China

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Joseph K. LIU
Monash University

Citation

HE, Kai; WENG, Jian; DENG, Robert H.; and LIU, Joseph K.. On the security of two identity-based conditional proxy re-encryption schemes. (2016). *Theoretical Computer Science*. 652, 18-27.
Available at: https://ink.library.smu.edu.sg/sis_research/3881

Published in Theoretical Computer Science, November 2016, Volume 652, Pages 18-27.
http://doi.org/10.1016/j.tcs.2016.08.023

On the security of two identity-based conditional proxy re-encryption schemes

Kai He [a,c], Jian Weng [a,*], Robert H. Deng [b], Joseph K. Liu [c]

[a] Department of Computer Science, Jinan University, Guangzhou 510632, China
[b] School of Information Systems, Singapore Management University, Singapore 178902, Singapore
[c] Faculty of Information Technology, Monash University, Australia

A B S T R A C T

Proxy re-encryption allows a semi-trusted proxy with a re-encryption key to convert a delegator's ciphertext into a delegatee's ciphertext, and the semi-trusted proxy cannot learn anything about the underlying plaintext. If a proxy re-encryption scheme is indistinguishable against chosen-ciphertext attacks, its initialized ciphertext should be non-malleable. Otherwise, there might exist an adversary who can break the chosen-ciphertext security of the scheme. Recently, Liang et al. proposed two proxy re-encryption schemes. They claimed that their schemes were chosen-ciphertext secure in the standard model. However, we find that the original ciphertext in their schemes are malleable. Thus, we present some concrete attacks and indicate their schemes fail to achieve chosen-ciphertext security in the standard model.

Keywords:
Conditional proxy re-encryption
Identity-based
Single hop
Multi-hop
Chosen-ciphertext security

1. Introduction

The notion of proxy re-encryption (PRE) was initially introduced by Blaze et al. [1]. In a PRE system, Alice can transform the ciphertext which is encrypted under her public key to another ciphertext which is encrypted under Bob's public key, so that Alice can securely share her information to Bob. According to the direction of transformation, PRE can be categorized into an undirectional PRE and a bidirectional PRE. In the unidirectional PRE, the ciphertext can be transformed from Alice to Bob. But in the bidirectional PRE, the ciphertext can be transformed not only from Alice to Bob, but it also can be transformed from Bob to Alice. According to another function, PRE can be categorized into a single-hop PRE and a multi-hop PRE. In the single-hop PRE, the ciphertext can only be transformed one time. But in the multi-hop PRE, the transformed ciphertext can continuously be transformed to the another user. PRE is a very useful primitive, it has many applications, such as encrypted e-mail forwarding, key distribution, access control and distributed file systems [2–10].

Chosen-ciphertext security is one of the most important goals to construct a PRE scheme. In 1998, Blaze et al. [1] proposed a bidirectional PRE scheme with chosen-plaintext security. In 2007, Canetti and Hohenberger [11] defined a chosen-ciphertext security model for the PRE scheme and proposed two bidirectional multi-hop PRE schemes with chosen-ciphertext security. One is proved in the random oracle model. The other one is proved in the standard model. After that, many bidirectional secure PRE schemes (e.g. [12,13]) have been proposed. Any unidirectional PRE scheme can be easily transformed to a bidirectional one by running the former in both directions, while whether the reverse holds is unknown. In 2005, Ateniese et al. [8,9] first presented two practical unidirectional PRE schemes from bilinear map and both of the

* Corresponding author.
E-mail address: cryptjweng@gmail.com (J. Weng).

two schemes are chosen-plaintext secure. In 2008, Libert and Vergnaud [14] proposed the first unidirectional PRE scheme against replayable chosen-ciphertext attacks in the standard model. Since then, many unidirectional PRE schemes with chosen-ciphertext security have been proposed (e.g., [15–18]) and all these schemes are single-hop PRE schemes.

If a PRE scheme is in the identity-based setting [19], each user's public key is the user's identity, (e.g. email address). In 2007, Green and Ateniese [20] proposed the first unidirectional identity-based proxy re-encryption (IBPRE) scheme, which is chosen-ciphertext secure in the random oracle model. Then, many IBPRE schemes have been proposed, such as [21,10, 22–29].

In order to facilitate fine-grained access control in the PRE or IBPRE system, the type-based PRE scheme [30] and the conditional PRE scheme [31] were proposed. In both cases, the proxy can re-encrypt the ciphertext if and only if the condition in the ciphertext is the same as in the re-encryption key. In 2009, Weng et al. [32] proposed a new conditional PRE scheme with chosen-ciphertext security and re-formed the definition and security notion for a conditional PRE scheme. Additionally, they pointed out the secure risk in the scheme [31].

Recently, Liang et al. proposed two identity-based conditional PRE schemes. One is a unidirectional single-hop conditional PRE (UniSH-IBCPRE) scheme [33], the other one is a bidirectional multi-hop conditional PRE (BiMH-IBCPRE) scheme [34]. They claimed that their schemes can achieve chosen-ciphertext security in the standard model. However, we find the original ciphertext in their schemes cannot ensure the non-malleability. There may exist an adversary who can break the security of their schemes. For example, given a challenge ciphertext $CT^*_{ID^*_i} = \mathrm{Enc}(ID^*_i, m_\beta) = (\cdots, C^*, \cdots)$ under the target identity ID^*_i, where the ciphertext component C^* is not verified. First, the adversary modifies C^* to C', so it obtains another ciphertext $CT'_{ID^*_i} = (\cdots, C', \cdots)$. Then, it issues a re-encryption query on $CT'_{ID^*_i}$ to achieve another ciphertext CT'_{ID_j} under a corrupted user ID_j. Note that it is legal for the adversary to issue the re-encryption query. Since $(ID^*_i, CT'_{ID^*_i})$ is not a derivative of $(ID^*_i, CT^*_{ID^*_i})$. Next, the adversity uses the corrupted user ID_j's private key sk_{ID_j} to derive the underlying plaintext from the ciphertext CT'_{ID_j}.

Based on the above analysis, in this paper, we present an outside adversary to break the chosen-ciphertext security of Liang et al.'s schemes [33,34] and an inside adversity to break the chosen-ciphertext security of [33]. The outside adversary does not collude with the semi-trusted proxy. The inside adversity is a semi-trusted proxy, who can collude with a delegatee. Thus, we indicate that their schemes fail to achieve chosen-ciphertext security.

1.1. Organization

The rest of the paper is organized as follows. In Section 2, we review the bilinear map and the decisional bilinear Diffie–Hellman assumption. In section 3, we first review the definition, the security model and the construction of Liang et al.'s UniSH-IBCPRE scheme [33], and then we present the security analysis for the UniSH-IBCPRE scheme. In section 4, we first review the definition, the security model and the construction of Liang et al.'s BiMH-IBCPRE scheme [34], and then we present the security analysis for the BiMH-IBCPRE scheme. Finally, we draw conclusions in Section 5.

2. Preliminaries

2.1. Bilinear map

$\mathcal{G}$ and $\mathcal{G}_T$ are cyclic multiplicative groups of order p, g is a generator of $\mathcal{G}$. A bilinear map is a map $e : \mathcal{G} \times \mathcal{G} \to \mathcal{G}_T$ with the following properties:

- **Bilinearity:** $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $g_1, g_2 \in \mathcal{G}$ and $a, b \in \mathcal{Z}_p^*$.
- **Non-degeneracy:** There exists $g_1, g_2 \in \mathcal{G}$ such that $e(g_1, g_2) \neq 1_{\mathcal{G}}$.
- **Computability:** There exists an efficient algorithm to compute $e(g_1, g_2)$ for $g_1, g_2 \in \mathcal{G}$.

2.2. Decisional Bilinear Diffie–Hellman (DBDH) assumption

The DBDH problem in a bilinear group $(p, \mathcal{G}, \mathcal{G}_T, e)$ is defined as follows: Given a tuple (g, g^a, g^b, g^c, T) as input, output 1 if $T = e(g, g)^{abc}$ and 0 otherwise. The advantage of an algorithm $\mathcal{A}$ in solving the DBDH problem is defined as $Adv^{\mathrm{DBDH}}_{\mathcal{A}} = |\Pr[\mathcal{A}(g, g^a, g^b, g^c, e(g, g)^{abc}) = 1] - \Pr[\mathcal{A}(g, g^a, g^b, g^c, T) = 1]|$, where $g \in \mathcal{G}, a, b, c \leftarrow \mathcal{Z}_p^*$, T is chosen randomly from $\mathcal{G}_{\mathcal{T}}$. We say that the DBDH assumption holds in the bilinear group $(p, \mathcal{G}, \mathcal{G}_{\mathcal{T}}, e)$ if all probabilistic polynomial-time (PPT) algorithms have negligible advantage in solving the DBDH problem.

3. Cryptanalysis of Liang et al.'s UniSH-IBCPRE scheme

In this section, first, we shall review the definition, the security model and the construction of Liang et al.'s UniSH-IBCPRE scheme [33]. Then, we give the security analysis for their construction.

3.1. Review the definition of Liang et al.'s UniSH-IBCPRE scheme

Definition 1. A UniSH-IBCPRE scheme $\prod = (\textbf{Setup}, \textbf{KeyGen}, \textbf{ReKeyGen}, \textbf{Enc}, \textbf{ReEnc}, \textbf{Dec}_2, \textbf{Dec}_1)$ consists of the following seven algorithms.

- **Setup**(1^λ)**:** On input a security parameter 1^λ, output a master public key mpk and a master secret key msk.
- **KeyGen**(msk, ID)**:** On input mpk, msk and an identity $ID \in \{0,1\}^n$, output a private key sk_{ID}.
- **ReKeyGen**$(mpk, sk_{ID_i}, ID_j, w)$**:** On input mpk, the private key sk_{ID_i} of an identity ID_i, an identity ID_j and a condition $w \in \{0,1\}^*$, output a re-encryption key $rk_{w|ID_i \to ID_j}$ from ID_i to ID_j under w.
- **Enc**(mpk, ID_i, w, m)**:** On input mpk, an identity ID_i, a condition w and a plaintext $m \in \{0,1\}^\lambda$, output a original ciphertext $C^{(2)}_{(ID_i,w)}$.
- **ReEnc**$(mpk, rk_{w|ID_i \to ID_j}, ID_i, w, C^{(2)}_{(ID_i,w)})$**:** On input mpk, a re-encryption key $rk_{w|ID_i \to ID_j}$, an identity ID_i, a condition w and a original ciphertext $C^{(2)}_{(ID_i,w)}$, output a transformed ciphertext $C^{(1)}_{(ID_j,w)}$.
- **Dec**$_2(mpk, ID_i, sk_{ID_i}, w, C^{(2)}_{(ID_i,w)})$**:** On input mpk, an identity ID_i and the corresponding private key sk_{ID_i}, a condition w and a original ciphertext $C^{(2)}_{(ID_i,w)}$, output a plaintext m or $\perp$ for failure.
- **Dec**$_1(mpk, ID_i, ID_j, sk_{ID_j}, w, C^{(1)}_{(ID_j,w)})$**:** On input mpk, an identity ID_i, an identity ID_j and the corresponding private key a sk_{ID_j}, a condition w and a transformed ciphertext $C^{(1)}_{(ID_j,w)}$, output a plaintext m or $\perp$ for failure.

3.2. Review the security model of Liang et al.'s UniSH-IBCPRE scheme

We review the adaptive condition and adaptive identity chosen ciphertext security (IND-aCon-aID-CCA) model of Liang et al.'s UniSH-IBCPRE scheme [33]. In their model, $\mathcal{C}$ is a challenger who plays the below game with a adversary $\mathcal{A}$.

- **Setup:** Challenger $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda)$ and sends mpk to $\mathcal{A}$.
- **Query Phase** I**:** Adversary $\mathcal{A}$ is given access to the following oracles:
  - *Extract(ID):* Given an identity ID, return $sk_{ID} \leftarrow \mathsf{KeyGen}(msk, ID)$ and ID is considered as corrupted.
  - *ReKeyExtract*(ID_i, ID_j, w): Given two distinct identities ID_i and ID_j, and a condition w, return $rk_{w|ID_i \to ID_j} \leftarrow \mathsf{ReKeyGen}(sk_{ID_i}, ID_j, w)$, where $sk_{ID_i} \leftarrow \mathsf{KeyGen}(msk, ID_i)$.
  - *ReEnc*$(ID_i, ID_j, w, C^{(2)}_{(ID_i,w)})$: Given two distinct identities ID_i and ID_j, a condition w and a original ciphertext $C^{(2)}_{(ID_i,w)}$, return a transformed ciphertext $C^{(1)}_{(ID_j,w)} \leftarrow \mathsf{ReEnc}(rk_{w|ID_i \to ID_j}, ID_i, w, C^{(2)}_{(ID_i,w)})$, where $sk_{ID_i} \leftarrow \mathsf{KeyGen}(msk, ID_i)$ and $rk_{w|ID_i \to ID_j} \leftarrow \mathsf{ReKeyGen}(sk_{ID_i}, ID_j, w)$.
  - *Dec*$_2(ID_i, w, C^{(2)}_{(ID_i,w)})$: Given an identity ID_i, a condition w and a original ciphertext $C^{(2)}_{(ID_i,w)}$, return $m \leftarrow \mathsf{Dec}_2(ID_i, sk_{ID_i}, w, C^{(2)}_{(ID_i,w)})$, where $sk_{ID_i} \leftarrow \mathsf{KeyGen}(msk, ID_i)$.
  - *Dec*$_1(ID_i, ID_j, w, C^{(1)}_{(ID_j,w)})$: Given two identity ID_i, ID_j, a condition w and a transformed ciphertext $C^{(1)}_{(ID_j,w)}$, return $m \leftarrow \mathsf{Dec}_1(ID_i, ID_j, sk_{ID_j}, w, C^{(1)}_{(ID_j,w)})$, where $sk_{ID_j} \leftarrow \mathsf{KeyGen}(msk, ID_j)$.
- **Challenge:** Adversary $\mathcal{A}$ outputs two equal-length plaintexts m_0, m_1, a target identity ID^* and a target condition w^* to $\mathcal{C}$. If the following queries: *Extract*(ID^*)*:* ID^* is uncorrupt identity. *ReKeyExtract*(ID^*, ID_j, w^*)*:* $\mathsf{Extract}(ID_j)$ for any identity ID_j are never queried, $\mathcal{C}$ outputs $C^{(2)*}_{(ID^*,w^*)} = \mathsf{Enc}(ID^*, w^*, m_b)$, where $b \in_R \{0,1\}$.
- **Query Phase** II**:** Adversary $\mathcal{A}$ makes further queries as in Query Phase I except the following: *Extract*(ID) if $ID = ID^*$; *ReKeyExtract*(ID^*, ID_j, w^*) and *Extract*(ID_j) for any identity ID_j; *ReEnc*$(ID^*, ID_j, w^*, C^{(2)*}_{(ID^*,w^*)})$ and *Extract*(ID_j) for any identity ID_j; *Dec*$_2(ID^*, w^*, C^{(2^*)}_{(ID^*,w^*)})$ and *Dec*$_1(ID^*, ID_j, w^*, C^{(1)}_{(ID_j,w^*)})$ for any $(ID_j, C^{(1)}_{(ID_j,w^*)})$, if $(ID_j, w^*, C^{(1)}_{(ID_j,w^*)})$ is a derivative of $(ID^*, w^*, C^{(2)*}_{(ID^*,w^*)})$. As of [11], the derivative of $(ID^*, w^*, C^{(2)*}_{(ID^*,w^*)})$ is defined as follows.
  1. If adversary $\mathcal{A}$ has issued a re-encryption key query on (ID^*, ID_j, w^*) to obtain the re-encryption key $rk_{w^*|ID_i^* \to ID_j}$, computed $C^{(1)}_{(ID_j,w^*)} \leftarrow \mathsf{ReEnc}(rk_{w^*|ID^* \to ID_j}, ID^*, w^*, C^{(2)*}_{(ID^*,w^*)})$, then $(ID_j, w^*, C^{(1)}_{(ID_j,w^*)})$ is a derivative of $(ID^*, w^*, C^{(2)*}_{(ID^*,w^*)})$.
  2. If adversary $\mathcal{A}$ has issued a re-encryption query on $(ID^*, w^*, C^{(2)*}_{(ID^*,w^*)})$ and obtained $C^{(1)}_{(ID_j,w^*)}$, then $(ID_j, w^*, C^{(1)}_{(ID_j,w^*)})$ is a derivative of $(ID^*, w^*, C^{(2)*}_{(ID^*,w^*)})$.
- **Guess**: Adversary $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$. if $b' = b$, $\mathcal{A}$ wins.

Definition 2. An IBCPRE scheme is IND-aCon-aID-CCA-secure at original ciphertext if for any probabilistic polynomial time (PPT) adversary $\mathcal{A}$, his advantage is negligible, where $\mathcal{A}$'s advantage is defined as $\epsilon = Adv^{\text{IBCPRE-2nd}}_{\mathcal{A}}(1^\lambda) = \left|\Pr[b' = b] - \frac{1}{2}\right|$.

3.3. Review Liang et al.'s UniSH-IBCPRE construction

We review Liang et al.'s UniSH-IBCPRE construction [33]. Their construction is based on Waters's IBE scheme [35], a strongly existential unforgeable one-time signature scheme [36], a pseudo-random function family [37] and a target collision resistant (TCR) hash function, it is specified by the following algorithms:

- **Setup**(1^λ)**:** Run $(q, g, \mathcal{G}_1, \mathcal{G}_2, e) \leftarrow \mathcal{G}$. Let $w \in \{0,1\}^n$ be an n-bit condition string. Choose $\alpha \in_R \mathcal{Z}_p^*$, $g_2, u_1', u_2', u_3', u_{3,0} \in_R \mathcal{G}_1$, three random n-length sets $U_1 = \{u_{1,i} | 1 \le i \le n\}, U_2 = \{u_{2,i} | 1 \le i \le n\}, U_3 = \{u_{3,i} | 1 \le i \le n\}$, $u_{1,i}, u_{2,i}, u_{3,i} \in_R \mathcal{G}_1$, a pseudorandom function PRF: $\mathcal{G}_2 \times \mathcal{G}_1 \to \{0,1\}^{\lambda_1}$, and a TCR hash function $H_1 : \mathcal{G}_2 \to \mathcal{G}_1$, where λ_1 is a security parameter. The master secret key is $msk = g_2^\alpha$, the master public key is $mpk = (\lambda, \lambda_1, g, g_1, g_2, u_1', u_2', u_3', u_{3,0}, U_1, U_2, U_3,$ PRF, H_1, (Sign.KeyGen, Sign, Verify)), where $g_1 = g^\alpha$.
- **KeyGen**(msk, ID)**:** Output $sk_{ID} = (sk_{ID_1}, sk_{ID_2}) = (g_2^\alpha \cdot (u_1' \prod_{i \in \mathcal{V}_{ID}} u_{1,i})^r, g^r)$, where $r \in_R \mathcal{Z}_q^*$, $ID \in \{0,1\}^n$, and let $\mathcal{V}_{ID}$ be the set of all i for which the i'th bit of ID is set to 1.
- **Enc**(ID_i, w, m)**:** Run $(K_S, K_V) \leftarrow$ Sign.KeyGen(1^λ), choose $t \in_R \mathcal{Z}_q^*, \sigma \in_R \mathcal{G}_2$, generate the ciphertext: $C_0 = [\text{PRF}(\sigma, C_2)]^{\lambda_1 - \lambda} || [\text{PRF}(\sigma, C_2)]_\lambda \oplus m$, $C_1 = e(g_1, g_2)^t \cdot \sigma$, $C_2 = g^t$, $C_3 = (u_1' \prod_{i \in \mathcal{V}_{ID_i}} u_{1,i})^t$, $C_4 = (u_2' \prod_{i \in \xi_w} u_{2,i})^t$, $C_5 = (u_3' u_{3,0} \prod_{i \in \mathcal{X}_{K_V}} u_{3,i})^t$, $C_6 = \text{Sign}(K_S, (C_0, C_2, C_3, C_4, C_5))$, and output $C^{(2)}_{(ID_i, w)} = (K_V, C_0, C_1, C_2, C_3, C_4, C_5, C_6)$, where $ID_i \in \{0,1\}^n, m \in \{0,1\}^\lambda$, let ξ_w, $\mathcal{X}_{K_V}$, $\mathcal{V}_{ID_i}$ be the sets of all i for which the i'th bit of w, K_V, ID_i is set to 1, respectively.
- **ReKeyGen**(sk_{ID_i}, ID_j, w)**:** Choose $\rho, t' \in_R \mathcal{Z}_p^*$, $\theta \in_R \mathcal{G}_2$, compute $rk_0 = sk_{ID_{i_1}} \cdot (u_2' \prod_{i \in \xi_w} u_{2,i})^\rho$, $rk_1 = g^\rho$, $rk_2 = sk_{ID_{i_2}} \cdot H_1(\theta)$, $rk_3 = e(g_1, g_2)^{t'} \cdot \theta$, $rk_4 = g^{t'}$, $rk_5 = (u_1' \prod_{i \in \mathcal{V}_{ID_j}} u_{1,i})^{t'}$, $rk_6 = (u_3' u_{3,0} \prod_{i \in \mathcal{X}_{K_V'}} u_{3,i})^{t'}$, $rk_7 = \text{Sign}(K_S', rk_3, rk_4, rk_5, rk_6))$, and output $rk_{w|ID_i \to ID_j} = (K_V', rk_0, rk_1, rk_2, rk_3, rk_4, rk_5, rk_6, rk_7)$, where $ID_j \in \{0,1\}^n$ and $(K_S', K_V') \leftarrow$ Sign.KeyGen(1^λ).
- **ReEnc**$(rk_{w|ID_i \to ID_j}, ID_i, w, C^{(2)}_{(ID_i, w)})$**:** Verify the following equations (1) hold or not:

$$\begin{aligned} &e(g, C_3) \stackrel{?}{=} e(C_2, u_1' \prod_{i \in \mathcal{V}_{ID}} u_{1,i}) \qquad e(g, C_4) \stackrel{?}{=} e(C_2, u_2' \prod_{i \in \xi_w} u_{2,i}) \\ &e(g, C_5) \stackrel{?}{=} e(C_2, u_3' u_{3,0} \prod_{i \in \mathcal{X}_{K_V}} u_{3,i}) \qquad Verify(K_V, C_6, (C_0, C_2, C_3, C_4, C_5)) \stackrel{?}{=} 1 \end{aligned} \tag{1}$$

  If equations (1) don't hold, output $\perp$; else compute: $C_1' = \frac{C_1 \cdot e(rk_2, C_3)}{e(rk_0, C_2)/e(rk_1, C_4)}$, and output the transformed ciphertext $C^{(1)}_{(ID_j, w)} = (K_V, C_0, C_1', C_2, C_3, C_4, C_5, C_6, K_V', rk_3, rk_4, rk_5, rk_6, rk_7)$.
- **Dec**$_2(ID_i, sk_{ID_i}, w, C^{(2)}_{(ID_i, w)})$**:** Verify equations (1). If equations (1) don't hold, output $\perp$; else compute $\sigma = C_1 \cdot \frac{e(sk_{ID_{i_2}}, C_3)}{e(sk_{ID_{i_1}}, C_2)}$, and output $m = [C_0]_\lambda \oplus [\text{PRF}(\sigma, C_2)]_\lambda$, if $[\text{PRF}(\sigma, C_2)]^{\lambda_1 - \lambda} = [C_0]^{\lambda_1 - \lambda}$ holds; else output $\perp$.
- **Dec**$_1(ID_i, ID_j, sk_{ID_j}, w, C^{(1)}_{(ID_j, w)})$**:** Verify the following equations (2) hold or not:

$$\begin{aligned} &e(g, rk_5) \stackrel{?}{=} e(rk_4, u_1' \prod_{i \in \mathcal{V}_{ID}} u_{1,i}), \qquad e(g, rk_6) \stackrel{?}{=} e(rk_4, u_3' u_{3,0} \prod_{i \in \mathcal{X}_{K_V}} u_{3,i}) \\ &Verify(K_V', rk_7, (rk_3, rk_4, rk_5, rk_6)) \stackrel{?}{=} 1 \end{aligned} \tag{2}$$

  If equations (2) don't hold, output $\perp$; else compute $\theta = rk_3 \cdot \frac{e(sk_{ID_{j_2}}, rk_5)}{e(sk_{ID_{j_1}}, rk_4)}$. If equations (1) don't hold, output $\perp$, else compute $\sigma = C_1' / e(H_1(\theta), C_3)$, and output $m = [\text{PRF}(\sigma, C_2)]_\lambda \oplus [C_0]_\lambda$, if $[\text{PRF}(\sigma, C_2)]^{\lambda_1 - \lambda} = [C_0]^{\lambda_1 - \lambda}$ holds. Otherwise, output $\perp$.

3.4. Security analysis for Liang et al.'s UniSH-IBCPRE construction

Liang et al.'s UniSH-IBCPRE construction [33] is based on Waters's identity-based encryption (IBE) [35] scheme. In order to capture the chosen-ciphertext security, Liang et al. extended Waters's IBE scheme by employing the technique introduced in [38]. Indeed, the extended Waters's IBE scheme can achieve the chosen-ciphertext security in the traditional public key encryption setting. However, it cannot achieve the chosen-ciphertext security in the proxy re-encrypted setting. As an orginal ciphertext component in their construction is not verified, there might exist an adversity who issues the re-encryption oracle to break the security of their construction. Thus, we present two concrete attacks against their UniSH-IBCPRE construction [33] in the following.

First, we present an outside adversary $\mathcal{A}_1$ to break the security of Liang et al.'s UniSH-IBCPRE construction. The outside adversary $\mathcal{A}_1$ does not collude with the semi-trusted proxy. Second, we present an inside adversary $\mathcal{A}_2$ (semi-trusted proxy) who colluded with a delegatee before and recovers a part of the delegator's private key. Although the semi-trusted proxy $\mathcal{A}_2$ cannot compromise the entire private key of the delegator, but it is enough for him to recover all the message of the delegator.

Outside Attack

First, in the challenge phase, adversary $\mathcal{A}_1$ modifies the challenge ciphertext component $C^{(2)*}_{(ID_{i^*},w^*)}$ to obtain a new (ill-formed) ciphertext $\overline{C}^{(2)*}_{(ID_{i^*},w^*)}$. Then, in the query phase II, adversary $\mathcal{A}_1$ asks the re-encryption oracle to re-encrypt the new ciphertext $\overline{C}^{(2)*}_{(ID_{i^*},w^*)}$ and gets a transformed ciphertext $\overline{C}^{(1)*'}_{(ID_j,w^*)}$, where ID_j is a corrupted user (note that according to the security model, it is legal for adversary $\mathcal{A}_1$ to query the re-encryption oracle). Next, adversary $\mathcal{A}_1$ modifies the transformed ciphertext $\overline{C}^{(1)*'}_{(ID_j,w^*)}$ to obtain the right re-encrypted ciphertext $C^{(1)*}_{(ID_j,w^*)}$ corresponding to the challenge ciphertext $C^{(2)*}_{(ID_{i^*},w^*)}$. Thus, adversary $\mathcal{A}_1$ derives the underlying plaintext by decrypting $C^{(1)*}_{(ID_j,w^*)}$ using the corrupted private key sk_{ID_j}.

To explain more clearly, we present the concrete outside attack against Liang et al.'s UniSH-IBCPRE construction in the following. Let $\mathcal{A}_1$ be an outside adversary, $\mathcal{A}_1$ interacts with challenger $\mathcal{C}$ in the following game.

- **Setup:** Adversary $\mathcal{A}_1$ first obtains the public parameters from challenger $\mathcal{C}$.
- **Query Phase I:** Adversary $\mathcal{A}_1$ issues the $\mathsf{Extract}(ID_j)$ oracle to obtain the private key of the ID_j and adds ID_j to a corrupted list.
- **Challenge:** Adversary $\mathcal{A}_1$ submits $(ID_{i^*}, m_0, m_1, w^*)$ to challenger $\mathcal{C}$, and then given the challenge ciphertext $C^{(2)*}_{(ID_{i^*},w^*)} = (K^*_V, C^*_0, C^*_1, C^*_2, C^*_3, C^*_4, C^*_5, C^*_6)$: $C^*_0 = [PRF(\sigma^*, C^*_2)]^{\lambda_1-\lambda}||([PRF(\sigma^*, C^*_2)]_\lambda \oplus m_\beta)$, $C^*_1 = e(g_1, g_2)^{t^*} \cdot \sigma^*$, $C^*_2 = g^{t^*}$, $C^*_3 = (u'_1 \prod_{i\in\mathcal{V}_{ID_{i^*}}} u_{1,i})^{t^*}$, $C^*_4 = (u'_2 \prod_{i\in\xi_w} u_{2,i})^{t^*}$, $C^*_5 = (u'_3 u_{3,0} \prod_{i\in\chi_{K^*_V}} u_{3,i})^{t^*}$, $C^*_6 = Sign(K^*_S, (C^*_0, C^*_2, C^*_3, C^*_4, C^*_5))$.
- **Query Phase II:** Adversary $\mathcal{A}_1$ issues the re-encryption oracle as follows: First, adversary $\mathcal{A}_1$ picks $\overline{C}_1 \in_R \mathcal{G}_2$, and lets $\overline{C^*_1} = C^*_1 \cdot \overline{C_1}$. Then adversary $\mathcal{A}_1$ modifies the challenge ciphertext $C^{(2)*}_{(ID_{i^*},w^*)}$ to obtain a new (ill-formed) ciphertext $\overline{C}^{(2)*}_{(ID_{i^*},w^*)} = (K^*_V, C^*_0, \overline{C^*_1}, C^*_2, C^*_3, C^*_4, C^*_5, C^*_6)$. Now adversary $\mathcal{A}_1$ submits $(ID_{i^*}, ID_j, w^*, \overline{C}^{(2)*}_{(ID_{i^*},w^*)})$ to the re-encryption oracle. (Note that, although ID_j is in the corrupt list, it is legal for adversary $\mathcal{A}_1$ to issue this query, since $(ID_{i^*}, ID_j, w^*, \overline{C}^{(2)*}_{(ID_{i^*},w^*)})$ is not a derivate of $(ID_{i^*}, ID_j, w^*, C^{(2)*}_{(ID_{i^*},w^*)})$). The main reason is that the re-encryption algorithm ReEnc cannot check the validity of the ciphertext component $\overline{C^*_1}$, so the re-encryption oracle still responds the re-encryption ciphertext $\widehat{C}^{(1)'}_{(ID_j,w^*)} = \mathsf{ReEnc}(params, \mathsf{ReKeyGen}(params, sk_{ID_{i^*}}, ID_j, w^*), ID_{i^*}, ID_j, w^*, \overline{C}^{(2)*}_{(ID_{i^*},w^*)})$ to adversary $\mathcal{A}_1$, where $\widehat{C}^{(1)'}_{(ID_j,w^*)} = (K^*_V, C^*_0, \widehat{C}'_1, C^*_2, C^*_3, C^*_4, C^*_5, C^*_6, K'_V, rk_3, rk_4, rk_5, rk_6, rk_7)$. In fact, we have

  $$\widehat{C}'_1 = \frac{\overline{C^*_1} \cdot e(rk_2, C_3)}{e(rk_0, C_2)/e(rk_1, C_4)} = \frac{C^*_1 \cdot \overline{C_1} \cdot e(rk_2, C_3)}{e(rk_0, C_2)/e(rk_1, C_4)}$$

  Next, adversary $\mathcal{A}_1$ uses $\overline{C}_1$ to recover the real transformed ciphertext $\overline{C}'_1$:

  $$\overline{C}'_1 = \frac{\widehat{C}'_1}{\overline{C}_1} = \frac{C^*_1 \cdot \overline{C_1} \cdot e(rk_2, C_3)}{\overline{C_1} \cdot e(rk_0, C_2)/e(rk_1, C_4)} = \frac{C^*_1 \cdot e(rk_2, C_3)}{e(rk_0, C_2)/e(rk_1, C_4)}.$$

  Observe that, $\overline{C}'_1$ is transformed by the challenge ciphertext component C^*_1. Thus, the ciphertext $\overline{C}^{(1)'}_{(ID_j,w^*)} = (K^*_V, C^*_0, \overline{C}'_1, C^*_2, C^*_3, C^*_4, C^*_5, C^*_6, K'_V, rk_3, rk_4, rk_5, rk_6, rk_7)$ is indeed transformed by the challenge ciphertext $C^{(2)*}_{(ID_{i^*},w^*)}$. Now, adversary $\mathcal{A}_1$ uses the colluded private key sk_{ID_j} to obtain the underlying plaintext m_β by decrypting the re-encryption ciphertext $\overline{C}^{(1)'}_{(ID_j,w^*)}$.
- **Guess:** Adversary $\mathcal{A}_1$ outputs a bit β'.

Obviously, adversary $\mathcal{A}_1$ has non-negligible advantage to output $\beta' = \beta$. It implies that Liang et al.'s UniSH-IBCPRE scheme cannot obtain chosen-ciphertext security in the standard model.

Inside Attack

For a semi-trusted proxy $\mathcal{A}_2$, given the re-encryption key $rk_{w|ID_i\to ID_j} = (K'_V, rk_0, rk_1, rk_2, rk_3, rk_4, rk_5, rk_6, rk_7)$, where $rk_2 = sk_{ID_{i_2}} \cdot H_1(\theta)$. Then the semi-trusted proxy $\mathcal{A}_2$ can transform the delegator's ciphertext to the delegatee's ciphertext. Assuming that the semi-trusted proxy $\mathcal{A}_2$ corrupts with a corresponding delegatee or it acts as a delegatee before, then it can obtain the value $H_1(\theta)$. At last $\mathcal{A}_2$ can compute the second part private key $sk_{ID_{i_2}}$ of the delegator. After that, every time, once the delegator gives the re-encryption key to the semi-trusted proxy $\mathcal{A}_2$, then the semi-trusted proxy $\mathcal{A}_2$ can compute $H_1(\theta')$ from the corresponding re-encryption key. Thus, the semi-trusted proxy $\mathcal{A}_2$ can use $H_1(\theta')$ to decrypt the re-encryption ciphertext, and then it derives the corresponding plaintext every time. Obviously, this is very dangerous for the delegator and the delegatee, because all plaintext information are transparent for the semi-trusted proxy $\mathcal{A}_2$, although it cannot obtain the entire delegator's private key.

To explain more clearly, we present a concrete inside attack against Liang et al.'s UniSH-IBCPRE scheme. Let $\mathcal{A}_2$ be a semi-trust proxy, $\mathcal{A}_2$ interacts with challenger $\mathcal{C}$ in the following game.

- **Setup:** Adversary $\mathcal{A}_2$ obtains the public parameters from challenger $\mathcal{C}$.

- **Query phase** I: Adversary $\mathcal{A}_2$ issues the following queries:
  - Adversary $\mathcal{A}_2$ issues the corrupted extract $\mathsf{Extract}(ID_j)$ oracle to obtain the private key of ID_j and adds ID_j to a corrupted list.
  - Adversary $\mathcal{A}_2$ issues the re-encryption key generation oracle $\mathsf{ReKeyExtract}(ID_{i^*}, ID_j, w)$, where ID_j is corrupted, and gets $rk_{w|ID_{i^*}\to ID_j} = (K'_V, rk_0, rk_1, rk_2, rk_3, rk_4, rk_5, rk_6, rk_7)$, where $rk_0 = sk_{ID_{i_1^*}} \cdot (u'_2 \prod_{i\in\xi_w} u_{2,i})^{\rho}$, $rk_1 = g^{\rho}$, $rk_2 = sk_{ID_{i_2^*}} \cdot H_1(\theta)$, $rk_3 = e(g_1, g_2)^{t'} \cdot \theta$, $rk_4 = g^{t'}$, $rk_5 = (u'_1 \prod_{i\in\mathcal{V}_{ID_j}} u_{1,i})^{t'}$, $rk_6 = (u'_3 u_{3,0} \prod_{i\in\mathcal{X}_{K'_V}} u_{3,i})^{t'}$, $rk_7 = \mathsf{Sign}(K'_S, (rk_3, rk_4, rk_5, rk_6))$. Then, semi-trusted proxy $\mathcal{A}_2$ colludes the user ID_j and uses the corrupted private key $sk_{ID_j} = (sk_{ID_{j1}}, sk_{ID_{j2}})$ to compute $\theta = rk_3 \cdot \frac{e(sk_{ID_{j2}}, rk_5)}{e(sk_{ID_{j1}}, rk_4)}$, and then computes the second part of the delegator's private key $sk_{ID_{i_2^*}} = rk_2 / H_1(\theta)$. (Note that although semi-trusted proxy $\mathcal{A}_2$ cannot recover the first private key $sk_{ID_{i_1^*}}$ of the delegator ID_{i^*}, but it is enough for $\mathcal{A}_2$ to recover all the message encrypted under ID_{i^*}).
- **Challenge:** Adversary $\mathcal{A}_2$ submits $(ID_{i^*}, m_0, m_1, w^*)$ to challenger $\mathcal{C}$, and then given the challenge ciphertext $C^{(2)*}_{(ID_{i^*}, w^*)} = (K^*_V, C^*_0, C^*_1, C^*_2, C^*_3, C^*_4, C^*_5, C^*_6)$ as follows: $C^*_0 = [\mathrm{PRF}(\sigma^*, C^*_2)]^{\lambda_1-\lambda} || [\mathrm{PRF}(\sigma^*, C^*_2)]_{\lambda} \oplus m_{\beta}$, $C^*_1 = e(g_1, g_2)^{t^*} \cdot \sigma^*$, $C^*_2 = g^{t^*}$, $C^*_3 = (u'_1 \prod_{i\in\mathcal{V}_{ID}} u_{1,i})^{t^*}$, $C^*_4 = (u'_2 \prod_{i\in\xi_w} u_{2,i})^{t^*}$, $C^*_5 = (u'_3 u_{3,0} \prod_{i\in\chi_{K^*_V}} u_{3,i})^{t^*}$, $C^*_6 = \mathsf{Sign}(K^*_S, (C^*_0, C^*_2, C^*_3, C^*_4, C^*_5))$.
- **Query Phase** II**:** In this phase, adversary $\mathcal{A}_2$ issues the following queries:
  Adversary $\mathcal{A}_2$ continues to issue the re-encryption key generation oracle $\mathsf{ReKeyExtract}(ID_{i^*}, ID_k, w^*)$, where ID_k is uncorrupt, challenger $\mathcal{C}$ gives the re-encryption key $rk_{w^*|ID_{i^*}\to ID_k} = (K'_V, rk_0, rk_1, rk_2, rk_3, rk_4, rk_5, rk_6, rk_7)$ to $\mathcal{A}_2$, where $rk_2 = sk_{ID_{i_2^*}} \cdot H_1(\vartheta)$. Thus, adversary $\mathcal{A}_2$ can compute $H_1(\vartheta) = rk_2 / sk_{ID_{i_2^*}}$ using the obtained private key $sk_{ID_{i_2^*}}$. (Adversary $\mathcal{A}_2$ does not need to corrupt any delegatee now.)
  Next, adversary $\mathcal{A}_2$ runs the re-encryption algorithm $\mathsf{ReEnc}(rk_{w^*|ID_{i^*}\to ID_k}, ID_{i^*}, w^*, C^{(2)}_{(ID_{i^*}, w^*)})$ to obtain re-encryption ciphertext $C^{(1)'}_{(ID_k, w^*)} = (K^*_V, C^*_0, C^{*'}_1, C^*_2, C^*_3, C^*_4, C^*_5, C^*_6, K^{*'}_V, rk_3, rk_4, rk_5, rk_6, rk_7)$, where $C^{*'}_1 = \frac{C^*_1 \cdot e(rk_2, C_3)}{e(rk_0, C_2)/e(rk_1, C_4)} = \sigma^* \cdot e(H_1(\vartheta), C^*_3)$. Thus, adversary $\mathcal{A}_2$ can compute $\sigma^* = C^{*'}_1 / e(H_1(\vartheta), C^*_3)$. At last, adversary $\mathcal{A}_2$ can easily compute $m_{\beta} = [\mathrm{PRF}(\sigma^*, C^*_2)]_{\lambda} \oplus [C^*_0]_{\lambda}$. (Note that $\mathcal{A}_2$ does not use the colluded private key of user ID_j to obtain the plaintext m_{β}.)
- **Guess:** Adversary $\mathcal{A}_2$ outputs β'.

Obviously, semi-trusted proxy $\mathcal{A}_2$ has non-negligible advantage to output $\beta' = \beta$. It implies that semi-trusted proxy $\mathcal{A}_2$ can recover all the message of the delegator as long as $\mathcal{A}_2$ colludes with one delegatee one time. Hence, Liang et al.'s UniSH-IBCPRE scheme cannot obtain chosen-ciphertext security.

4. Cryptanalysis of Liang et al.'s BiMH-IBCPRE scheme

In this section, first, we shall review the definition, security model and the construction of Liang et al.'s BiMH-IBCPRE scheme [34]. Then, we give the security analysis for their construction [34].

4.1. Review the definition of Liang et al.'s BiMH-IBCPRE scheme

Definition 3. A BiMH-IBVPRE scheme $\prod = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Re\text{-}Encryption\ Key\ Generation\ Protocol}, \mathsf{Enc}, \mathsf{ReEnc}, \mathsf{Dec})$ consists of the following algorithms and protocols.

- **Setup**$(1^{\lambda}, n)$**:** On input a security parameter 1^{λ}, and $n \in \mathcal{N}$ the allowable maximum number of condition in the system, output a master public key mpk and a master secret key msk for Private Key Generator (PKG).
- **KeyGen**(mpk, msk, ID)**:** On input mpk, msk, and an identity $ID \in \{0,1\}^*$, output a private key sk_{ID} for the identity ID.
- **Re-Encryption Key Generation Protocol:** For simplicity, i and j denotes ID_i and ID_j in the re-encryption key.
  - *PReKeyGen*(mpk, sk_{ID_j}, W): On input mpk, a private key sk_{ID_j} for identity ID_j, and a condition set $W = \{w_z | 1 \le z \le n, w_z \in \{0,1\}^*\}$, the partial re-encryption key generation algorithm PReKeyGen outputs a partial re-encryption key $prk_{(W,j)}$ under W and ID_j.
  - *ReKeyGen*$(mpk, sk_{ID_i}, prk_{(W,j)}, W)$: On input mpk, a private key sk_{ID_i} for identity ID_i, a partial re-encryption key $prk_{(W,j)}$ and a set W of conditions, the re-encryption key algorithm ReKeyGen outputs a re-encryption key $rk_{i\to j|W}$ from ID_i to ID_j under W. Note ID_i and ID_j are two distinct identities.
  - *ReKeyBiGen*$(rk_{i\to j|W})$: On input a re-encryption key $rk_{i\to j|W}$ from an identity ID_i to another identity ID_j under a set W of conditions, the re-encryption key derivation algorithm outputs a new re-encryption key $rk_{j\to i|W}$ from ID_j to ID_i under W. Note that this algorithm also allows the re-encryption key $rk_{j\to i|W}$ holder to generate a new re-encryption key $rk_{i\to j|W}$.
- **Enc**(mpk, ID_i, W, m)**:** On input mpk, an identity ID_i, a set W of conditions and a message $m \in \{0,1\}^{\lambda}$, the encryption algorithm Enc outputs a ciphertext (i.e. original ciphertext) $C_{(ID_i, W)}$ under ID_i and W. Note that ID_i and W are implicitly included in the ciphertext.

- **ReEnc**$(mpk, rk_{i\to j|W}, C_{(ID_i,W)})$**:** On input mpk, a re-encryption key $rk_{i\to j|W}$, and a ciphertext $C_{(ID_i,W)}$, the re-encryption algorithm ReEnc outputs a re-encrypted ciphertext $C_{(ID_j,W)}$ or a symbol $\perp$ indicating that the ciphertext $C_{(ID_i,W)}$ is invalid.
- **Dec**$(mpk, sk_{ID_i}, C_{(ID_i,W)})$**:** On input mpk, a private key sk_{ID_i} for identity ID_i, and a ciphertext $C_{(ID_i,W)}$, the decryption algorithm Dec outputs a message m or a symbol $\perp$ indicating that the ciphertext $C_{(ID_i,W)}$ is invalid.

4.2. Review the security model of Liang et al.'s BiMH-IBCPRE scheme

In this section, we review the IND-sCon-sID-CCA security model for Liang et at.'s BiMH-IBCPRE scheme [34]. $\mathcal{C}$ is the challenger who plays the game with adversary $\mathcal{A}$.

- **Init:** $\mathcal{A}$ outputs a challenge identity ID^* and a conditions set W^* to $\mathcal{C}$.
- **Setup:** $\mathcal{C}$ runs Setup$(1^\lambda, n)$ and sends mpk to $\mathcal{A}$.
- **Phase 1:** $\mathcal{A}$ is given access to the following oracles.
  - *Private key extraction oracle* $\mathcal{O}_{sk}(ID)$: On input an identity ID, $\mathcal{C}$ returns $sk_{ID} \leftarrow \mathsf{KeyGen}(msk, ID)$ to $\mathcal{A}$.
  - *Re-encryption key extraction oracle* $\mathcal{O}_{rk}(ID_i, ID_j, W)$: On input two distinct identities ID_i and ID_j, and a condition set W, $\mathcal{C}$ returns a re-encryption key $rk_{i\to j|W} \leftarrow \mathsf{ReKeyGen}(sk_{ID_i}, \mathsf{PReKeyGen}(sk_{ID_j}, W), W)$, where $sk_{ID_i} \leftarrow \mathsf{KeyGen}(msk, ID_i)$, $sk_{ID_j} \leftarrow \mathsf{KeyGen}(msk, ID_j)$, and $ID_i, ID_j \in \{0,1\}^*$. Note that $\mathcal{A}$ can derive $rk_{j\to i|W}$ from $rk_{i\to j|W}$ with algorithm ReKeyBiGen.
  - *Re-encryption oracle* $\mathcal{O}_{re}(ID_i, ID_j, W, C_{(ID_i,W)})$: On input two distinct identities ID_i and ID_j, a condition set W, and a ciphertext $C_{(ID_i,W)}$ under ID_i and W, $\mathcal{C}$ returns a re-encrypted ciphertext $C_{(ID_j,W)} \leftarrow \mathsf{ReEnc}(rk_{i\to j|W}, C_{(ID_i,W)})$, where $rk_{i\to j|W} \leftarrow \mathsf{ReKeyBiGen}(ri_{j\to i|W})$, and further re-encrypts $C_{(ID_i,W)}$ to $\mathcal{O}_{re}$. If so, $\mathcal{C}$ will first generate $rk_{j\to i|W}$ and get $rk_{i\to j|W} \leftarrow \mathsf{ReKeyBiGen}(rk_{j\to i|W})$, and further re-encrypt $C_{(ID_i,W)}$ using $rk_{i\to j|W}$.
  - *Decryption oracle* $\mathcal{O}_{dec}(ID_i, C_{(ID_i,W)})$: On input an identity ID_i, and a ciphertext $C_{(ID_i,W)}$, $\mathcal{C}$ returns $m \leftarrow Dec(sk_{ID_i}, C_{(ID_i,W)})$, where $sk_{ID_i} \leftarrow \mathsf{KeyGen}(msk, ID_i)$, $ID_i \in \{0,1\}^*$. Note that if $\mathcal{A}$ issues invalid ciphertext to $\mathcal{O}_{re}$ or $\mathcal{O}_{dec}$, $\mathcal{C}$ simply outputs $\perp$. Moreover, the following queries cannot be issued:
    (1) $\mathcal{O}_{sk}(ID)$, if $ID^* = ID$ or for any ID in an uncorrupted delegation chain under W^* which includes ID^*;
    (2) $\mathcal{O}_{rk}(ID_i, ID_j, W^*)$ for any distinct ID_i and ID_j, if ID^* will be in a corrupted delegation chain under W^* after issuing the corresponding re-encryption key.
- **Challenge:** $\mathcal{A}$ outputs two distinct equal length messages (m_0, m_1) to $\mathcal{C}$. $\mathcal{C}$ returns the challenge ciphertext $C^*_{(ID^*,W^*)} = \mathsf{Enc}(ID^*, W^*, m_b)$ to $\mathcal{A}$, where $b \in_R (0,1)$.
- **Phase 2:** $\mathcal{A}$ continues making queries except the followings:
  (1) $\mathcal{O}_{sk}(ID)$ if $ID^* = ID$ or for any ID in an uncorrupted delegation chain under W^* which includes ID^*.
  (2) $\mathcal{O}_{rk}(ID_i, ID_j, W^*)$ for any distinct ID_i and ID_j, if ID^* will be in a corrupted delegation chain under W^* after issuing the corresponding re-encryption key.
  (3) $\mathcal{O}_{re}(ID_i, ID_j, W^*, C_{(ID_i,W^*)})$ if $(ID_i, W^*, C_{(ID_i,W^*)})$ is a derivative of $(ID^*, W^*, C^*_{(ID^*,W^*)})$, but ID_j is a corrupted identity or ID_j is in a corrupted delegation chain. As of [11], a derivative of $(ID^*, W^*, C^*_{(ID^*,W^*)})$ is defined as follows.
  - $(ID^*, W^*, C^*_{(ID^*,W^*)})$ is a derivative of itself.
  - If $(ID_i, W^*, C^*_{(ID_i,W^*)})$ is a derivative of $(ID^*, W^*, C^*_{(ID^*,W^*)})$, and $(ID_{i'}, W^*, C_{(ID_{i'},W^*)})$ is a derivative of $(ID_i, W^*, C_{(ID_i,W^*)})$, then $(ID_{i'}, W^*, C_{(ID_{i'},W^*)})$ is a derivative of $(ID^*, W^*, C^*_{(ID^*,W^*)})$.
  - If $\mathcal{A}$ has issued a re-encryption key query on (ID_i, ID_j, W) to obtain $rk_{i\to j|W}$, and achieved $C_{(ID_j,W)} \leftarrow \mathsf{ReEnc}(rk_{i\to j|W}, C_{(ID_i,W)})$ then $(ID_j, W, C_{(ID_j,W)})$ is a derivative of $(ID_i, W, C_{(ID_i,W)})$.
  - If $\mathcal{A}$ can run $C_{(ID_j,W)} \leftarrow \mathsf{ReEnc}(\mathsf{ReKeyGen}(sk_{ID_i}, prk_{(W,j)}, W), C_{(ID_i,W)})$, then $(ID_j, W, C_{(ID_j,W)})$ is a derivative of $(ID_i, W, C_{(ID_i,W)})$, where $sk_{ID_i} \leftarrow \mathsf{KeyGen}(msk, ID_i)$, $prk_{(W,j)} \leftarrow \mathsf{PreKeyGen}(sk_{ID_j}, W)$ and $sk_{ID_j} \leftarrow \mathsf{KeyGen}(msk, ID_j)$.
  - If $\mathcal{A}$ has issued a re-encryption query on $(ID_i, ID_j, W, C_{(ID_i,W)})$ and obtained $C_{(ID_j,W)}$, then $(ID_j, W, C_{(ID_j,W)})$ is a derivative of $(ID_i, W, C_{(ID_i,W)})$.
  (4) $\mathcal{O}_{dec}(ID_i, C_{(ID_i,W^*)})$ if $(ID_i, W^*, C_{(ID_i,W^*)})$ is a derivative of $(ID^*, W^*, C^*_{(ID^*,W^*)})$.
- **Guess:** $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$.

Definition 4. A BiMH-IBCPRE scheme is IND-sCon-sID-CCA secure if for any PPT adversary $\mathcal{A}$ wins the above game with negligible advantage $\epsilon_1 = Adv^{\text{IND-sCon-sID-CCA}}_{\text{BiMH-IBCPRE},\mathcal{A}}(1^\lambda, n) = |\Pr[b' = b] - 1/2|$.

4.3. Review Liang et al.'s BiMH-IBCPRE construction

Let's review Liang et al.'s BiMH-IBCPRE construction [34]. Their construction is based a hierarchical identity-based encryption (HIBE) [39], a pseudorandom function [37], and a one-time signature scheme [36]. It is specified by the following algorithms:

- **Setup**$(1^\lambda, n)$**:** Given the security parameter λ and n the allowable maximum number of conditions in the system (here $n = 1$), run $(q, g, \mathcal{G}_1, \mathcal{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$, choose $\alpha \in_R \mathcal{Z}^*_p$, $f_1, f_2, g_2, g_3, h_1, h_2, h_3 \in_R \mathcal{G}_1$, and prepare a pseu-

dorandom function PRF:$\mathcal{G}_T \times \mathcal{G}_1 \to \{0,1\}^{\lambda_1}$ (which takes an element in $\mathcal{G}_T$ as the function key and an element in $\mathcal{G}_1$ as input, and outputs a λ_1-bit pseudorandom string), where λ_1 is a security parameter as well. Let $w \in \mathcal{Z}_q^*$ be a condition, (Sign.KeyGen,Sign,Verify) be an OTS scheme, and assume the verification key output by Sign.KeyGen(1^λ) is in $\mathcal{Z}_q^*$. The master secret key is $msk = g_2^\alpha$, and the master public key is $mpk = (q, g, \mathcal{G}_1, \mathcal{G}_T, e, f_1, f_2, g_1, g_2, g_3, h_1, h_2, h_3$, PRF, (Sign.KeyGen,Sign,Verify)), where $g_1 = g^\alpha$.

- **KeyGen**(msk, ID)**:** Given the master secret key msk and an identity ID (i.e. an identity $I = (ID)$), choose $r \in_R \mathcal{Z}_q^*$, and output the key $sk_{ID} = (g_2^\alpha \cdot (h_1^{ID} \cdot g_3)^r, g^r, h_2^r, h_3^r) \in \mathcal{G}_1^4$. A system user with knowledge of sk_{ID} can generate the following private key due to the key derivation of HIBE. Given $sk_{ID} = (a_0, a_1, b_2, b_3)$, $I = (ID, w)$, choose a $t \in_R \mathcal{Z}_q^*$ and output $sk_{(ID,w)} = (a_0 \cdot b_2^w \cdot (h_1^{ID} \cdot h_2^w \cdot g_3)^t, a_1 \cdot g^t, b_3 \cdot h_3^t) = (g_2^\alpha \cdot (h_1^{ID} \cdot h_2^w \cdot g_3)^{r'}, g^{r'}, h_2^{r'}, h_3^{r'}) \in \mathcal{G}_1^4$, where $r' = r + t$.
- **Enc**(ID_i, w, m)**:** Given an identity ID_i, a condition w and a message m, choose an OTS key pair $(K_s, K_v) \leftarrow$ Sign.KeyGen(1^λ) and $\sigma \in_R \mathcal{G}_T$, $s \in_R \mathcal{Z}_q^*$, set $C_0 = K_v$, $C_1 = [\mathrm{PRF}_\sigma(C_3)]^{\lambda_1-\lambda} || [\mathrm{PRF}_\sigma(C_3)]_\lambda \oplus m$, $C_2 = \sigma \cdot e(g_1, g_2)^s$, $C_3 = g^s$, $C_4 = (h_1^{ID_i} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^s$, $C_5 = (f_1^w \cdot f_2)^s$, $C_6 = \mathrm{Sign}(K_s, (C_1, C_3, C_4, C_5))$, and output the transformed ciphertext $C_{(ID_i,w)}$=$((ID_i, w), C_0, C_1, C_2, C_3, C_4, C_5, C_6)$, where $ID_i \in \mathcal{Z}_q^*$, $m \in \{0,1\}^\lambda$.
- **Re-Encryption Key Generation Protocol:**
  - *PReKeyGen*$(sk_{(ID_j,w)})$. ID_j first deduces $sk_{(ID_j,w)} = (a_{0j}, a_{1j}, b_{3j})$ (under $I = (ID_j, w)$) from sk_{ID_j}, next chooses $\rho_1, \rho_2 \in_R \mathcal{Z}_q^*$ and sets $\beta_1 = a_{0j}^{-1} \cdot (f_1^w \cdot f_2)^{\rho_1}$, $\beta_2 = g^{\rho_1}$, $\beta_3 = a_{1j}^{-1} \cdot g^{\rho_2}$, $\beta_4 = b_{3j}^{-1} \cdot h_3^{\rho_2}$, $\beta_5 = (h_2^w \cdot g_3)^{\rho_2}$, $\beta_6 = h_1^{\rho_2}$. ID_j then sends the partial re-encryption key $prk_{(w,j)} = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6)$ to ID_i.
  - *ReKeyGen*$(sk_{ID_i}, prk_{(w,j)}, w)$. ID_i first generates $sk_{(ID_i,w)} = (a_{0i}, a_{1i}, b_{3i})$ (under $I = (ID_i, w)$), chooses $\rho_3, \rho_4 \in_R \mathcal{Z}_q^*$, computes $rk_1 = a_{0i} \cdot \beta_1 \cdot (f_1^w \cdot f_2)^{\rho_3}$, $rk_2 = g^{\rho_3} \cdot \beta_2$, $rk_3 = a_{1i} \cdot \beta_3 \cdot g^{\rho_4}$, $rk_4 = b_{3i} \cdot \beta_4 \cdot h_3^{\beta_4}$, $rk_5 = (h_2^w \cdot g_3)^{\rho_4} \cdot \beta_5$, $rk_6 = h_1^{\rho_4} \cdot \beta_6$ and outputs $rk_{i \to j|w} = (rk_1, rk_2, rk_3, rk_4, rk_5, rk_6)$.
  - *Derive a new re-encryption key* $rk_{j \to i|w}$ from $rk_{i \to j|w}$ by running $rk_{j \to i|w} \leftarrow$ *ReKeyBiGen*$(rk_{i \to j|w})$. The proxy sets $rk_{j \to i|w} = rk_{i \to j|w}^{-1}$, i.e. $rk_{j \to i|w} = (rk_1^{-1}, rk_2^{-1}, rk_3^{-1}, rk_4^{-1}, rk_5^{-1}, rk_6^{-1})$.
- **ReEnc**$(rk_{i \to j|w}, C_{(ID_i,w)})$**:** Given a re-encryption key $rk_{i \to j|w}$ and a ciphertext $C_{(ID_i,w)}$, the re-encryption algorithm works as follows.
  (a). Verify the validity of the ciphertext:

$$\begin{aligned} &e(C_3, f_1^w \cdot f_2) \stackrel{?}{=} e(g, C_5) \\ &e(C_3, h_1^{ID_i} \cdot h_2^w \cdot h_3^{C_0} \cdot g_3) \stackrel{?}{=} e(g, C_4) \\ &Verify(C_0, C_6, (C_1, C_3, C_4, C_5)) \stackrel{?}{=} 1 \end{aligned} \tag{3}$$

  If Eq. (3) does not hold, output $\perp$.
  (b). Compute $C_2^{(\ell)} = \frac{C_2^{(\ell-1)} \cdot e(rk_3, C_4) \cdot e(rk_2, C_5)}{e(rk_1 \cdot rk_4^{C_0} \cdot rk_6^{ID_i}, rk_{5,C_3})}$, output the re-encrypted ciphertext $C_{(ID_j,w)} = ((ID_i, w), C_0, C_1, C_2^{(\ell)}, C_3, C_4, C_5, C_6)$, where $\ell \geq 2$ denotes the level of the ciphertext. If $\ell = 1$, $C_2^{(1)}$ is from the transformed ciphertext.
- **Dec**$(sk_{ID_i}, C_{(ID_i,w)})$**:** Given a private key sk_{ID_i} for ID_i and a ciphertext $C_{(ID_i,w)}$, the decryption algorithm works as follows. ID_i first deduces the private key sk_{ID_i} for $sk_{(ID_i,w)} = (a_{0i}, a_{1i}, b_{3i})$ (under $I = (ID_i, w)$) from sk_{ID_i} (under $I = (ID_i)$), and next does the followings. (a) Verify the validity of the ciphertext by checking Eq. (3). If the equation does not hold, output $\perp$. Otherwise, proceed. (b) Compute $\rho = C_2^{(l)} \cdot \frac{e(a_{1i}, C_4)}{e(a_{0i} \cdot b_{3i}^{C_0}, C_3)}$, and then verify $[\mathrm{PRF}_\delta(C_3)]^{\lambda_1-\lambda} = [C_1]^{\lambda_1-\lambda}$. If the equation holds, output $m = [C_1]_\lambda \oplus [\mathrm{PRF}_\delta(C_3)]_\lambda$. Otherwise, output $\perp$.

4.4. Security analysis for Liang et al.'s BiMH-IBCPRE construction

In the following, we shall present one concrete outside attack against Liang et al.'s BiMH-IBCPRE construction [34]. The attack is the same as the above outside attack for Liang et al.'s UniSH-IBCPRE construction [33]. Let $\mathcal{A}_1$ be a PPT attacker, $\mathcal{A}_1$ interacts with challenger $\mathcal{C}$ in the following.

Outside Attack

- **Setup:** The same as in the above Outside Attack.
- **Query phase I:** The same as in the above Outside Attack.
- **Challenge:** Adversary $\mathcal{A}_1$ submits $(ID_{i^*}, m_0, m_1, w^*)$ to challenger $\mathcal{C}$, then given the challenge ciphertext $C^*_{(ID_{i^*},w)} = ((ID_{i^*}, w^*), C_0^*, C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*)$ as follows: $C_0^* = K_v^*$, $C_1^* = [\mathrm{PRF}_{\sigma^*}(C_3^*)]^{\lambda_1-\lambda} || [\mathrm{PRF}_{\sigma^*}(C_3^*)]_\lambda \oplus m_\beta$, $C_2^* = \sigma^* \cdot e(g_1, g_2)^s$, $C_3^* = g^s$, $C_4^* = (h_1^{ID_{i^*}} \cdot h_2^{w^*} \cdot h_3^{K_v^*} \cdot g_3)^s$, $C_5^* = (f_1^{w^*} \cdot f_2)^s$, $C_6^* = \mathrm{Sign}(K_s^*, (C_1^*, C_3^*, C_4^*, C_5^*))$.
- **Query Phase II:** Adversary $\mathcal{A}_1$ issues the re-encryption oracle as follows:
  Adversary $\mathcal{A}_1$ first randomly picks $\overline{C}_2 \in \mathcal{G}_2$ and lets $\overline{C_2^*} = C_2^* \cdot \overline{C_2}$. Then adversary $\mathcal{A}_1$ modifies the challenge ciphertext to obtain a new (ill-formed) ciphertext $\overline{C}^*_{(ID_{i^*},w^*)} = ((ID_{i^*}, w^*), C_0^*, C_1^*, \overline{C_2^*}, C_3^*, C_4^*, C_5^*, C_6^*)$. Now adversary $\mathcal{A}_1$ submits $(ID_{i^*}, ID_j, w^*, \overline{C}^*_{(ID_{i^*},w^*)})$ to the re-encryption oracle. (Note that, although ID_j is in the corrupt list, it is legal for $\mathcal{A}_1$

to issue this query, since $(ID_{i^*}, ID_j, w^*, \overline{C}^*_{(ID_{i^*},w^*)})$ is not a derivate of $(ID_{i^*}, ID_j, w^*, C^*_{(ID_{i^*},w^*)})$.) As the re-encryption algorithm ReEnc cannot check the validity of the ciphertext component $\overline{C_2^*}$. So the re-encryption oracle can return the re-encryption ciphertext $\widehat{C}^*_{(ID_j,w^*)} = \mathsf{ReEnc}(params, \mathsf{ReKeyGen}(params, sk_{ID_{i^*}}, ID_j, w^*), ID_{i^*}, ID_j, w^*, \overline{C}^*_{(ID_{i^*},w^*)})$ to adversity $\mathcal{A}$, where $\widehat{C}^*_{(ID_j,w^*)} = ((ID_{i^*}, w^*), C_0^*, C_1^*, \widehat{C}_2^*, C_3^*, C_4^*, C_5^*, C_6^*)$ and $\widehat{C}_2^* = \frac{\overline{C_2^*}\cdot e(rk_3, C_4)\cdot e(rk_2, C_5)}{e(rk_1\cdot rk_4^{C_0}\cdot rk_6^{ID_i}\cdot rk_5, C_3)}$. Then, adversary $\mathcal{A}_1$ uses $\overline{C}_2$ to compute $\overline{C}_2'$ as follows:

$$\overline{C}_2' = \frac{\widehat{C}_2^*}{\overline{C}_2} = \frac{C_2^*\cdot\overline{C_2}\cdot e(rk_3, C_4)\cdot e(rk_2, C_5)}{\overline{C_2}\cdot e(rk_1\cdot rk_4^{C_0}\cdot rk_6^{ID_i}\cdot rk_5, C_3)} = \frac{C_2^*\cdot e(rk_3, C_4)\cdot e(rk_2, C_5)}{e(rk_1\cdot rk_4^{C_0}\cdot rk_6^{ID_i}\cdot rk_5, C_3)}.$$

Observe that, we find $\overline{C}_2'$ is transformed by the challenge ciphertext component C_2^*. Thus, the modified re-encryption ciphertext $\overline{C}'_{(ID_j,w^*)} = ((ID_{i^*}, w^*), C_0^*, C_1^*, \overline{C}_2', C_3^*, C_4^*, C_5^*, C_6^*)$ is indeed the result of $\mathsf{ReEnc}(rk_{w^*|ID_{i^*}\to ID_j}, ID_{i^*}, w^*, C^*_{(ID_{i^*},w^*)})$, which is an encryption of m_β. Now, adversary $\mathcal{A}_1$ can obtain the underlying plaintext m_β by decrypting the re-encryption ciphertext $\overline{C}^{(1)'}_{(ID_j,w^*)}$ using user ID_j's private key sk_{ID_j}.

- **Guess:** Adversary $\mathcal{A}_1$ outputs β'.

Obviously, adversary $\mathcal{A}_1$ has non-negligible advantage to output $\beta' = \beta$. It implies that Liang et al.'s BiMH-IBCPRE Scheme is not chosen-ciphertext secure.

5. Conclusion

We present some concrete attacks to Liang et al.'s UniSH-IBCPRE scheme [33] and BiMH-IBCPRE scheme [34]. In their schemes, the non-malleable cannot be ensured for their original ciphertexts. Although all components of the original ciphertext are signed by the one-time signature scheme, except one component (e.g. the component C_1 in the UniSH-IBCPRE scheme [33] and the component C_2 in the BiMH-IBCPRE scheme [34]). Of course, signing all components can prevent our concrete attack, but the resultant is that their schemes are not PRE schemes any longer. Hence, the problems of how to construct a UniSH-IBCPRE scheme and a BiMH-IBCPR scheme with chosen-ciphertext security in the standard model are still open.

Acknowledgement

This work was supported by National Science Foundation of China (Grant Nos. 61272413, 61133014, 61272415 and 61472165), Program for New Century Excellent Talents in University (Grant No. NCET-12-0680), Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20134401110011), Foundation for Distinguished Young Talents in Higher Education of Guangdong (Grant No. 2012LYM 0027), the Fundamental Research Funds for the Central Universities (Grant No. 11613106), and this work is also supported by China Scholarship Council (Grant No. [2014]3012).

References

[1] M. Blaze, G. Bleumer, M. Strauss, Divertible protocols and atomic proxy cryptography, in: Advances in Cryptology – EUROCRYPT '98, Proceeding of International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31–June 4, 1998, 1998, pp. 127–144.

[2] Y. Chiu, C. Lei, C. Huang, Secure multicast using proxy encryption, in: Information and Communications Security, Proceedings of 7th International Conference, ICICS 2005, Beijing, China, December 10–13, 2005, 2005, pp. 280–290.

[3] A. Ivan, Y. Dodis, Proxy cryptography revisited, in: Proceedings of the Network and Distributed System Security Symposium, NDSS 2003, San Diego, California, USA, 2003.

[4] H. Khurana, H. Hahm, Certified mailing lists, in: Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2006, Taipei, Taiwan, March 21–24, 2006, 2006, pp. 46–58.

[5] H. Khurana, A.J. Slagell, R. Bonilla, SELS: a secure e-mail list service, in: Proceedings of the 2005 ACM Symposium on Applied Computing, SAC, Santa Fe, New Mexico, USA, March 13–17, 2005, 2005, pp. 306–313.

[6] G. Taban, A.A. Cárdenas, V.D. Gligor, Towards a secure and interoperable DRM architecture, in: Proceedings of the Sixth ACM Workshop on Digital Rights Management, Alexandria, VA, USA, October 30, 2006, 2006, pp. 69–78.

[7] A. Talmy, O. Dobzinski, Abuse freedom in access control schemes, in: 20th International Conference on Advanced Information Networking and Applications, AINA 2006, 18–20 April 2006, Vienna, Austria, 2006, pp. 77–86.

[8] G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage, in: Proceedings of the Network and Distributed System Security Symposium, NDSS 2005, San Diego, California, USA, 2005.

[9] G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage, ACM Trans. Inf. Syst. Secur. 9 (1) (2006) 1–30.

[10] T. Matsuo, Proxy re-encryption systems for identity-based encryption, in: Pairing-Based Cryptography – Pairing 2007, Proceedings of First International Conference, Tokyo, Japan, July 2–4, 2007, 2007, pp. 247–267.

[11] R. Canetti, S. Hohenberger, Chosen-ciphertext secure proxy re-encryption, in: Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28–31, 2007, 2007, pp. 185–194.

[12] R.H. Deng, J. Weng, S. Liu, K. Chen, Chosen-ciphertext secure proxy re-encryption without pairings, in: Cryptology and Network Security, Proceedings of 7th International Conference, CANS 2008, Hong-Kong, China, December 2–4, 2008, 2008, pp. 1–17.

[13] T. Matsuda, R. Nishimaki, K. Tanaka, CCA proxy re-encryption without bilinear maps in the standard model, in: Public Key Cryptography – PKC 2010, Proceedings of 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26–28, 2010, 2010, pp. 261–278.
[14] B. Libert, D. Vergnaud, Unidirectional chosen-ciphertext secure proxy re-encryption, in: Public Key Cryptography – PKC 2008, Proceedings of 11th International Workshop on Practice and Theory in Public-Key Cryptography, Barcelona, Spain, March 9–12, 2008, 2008, pp. 360–379.
[15] G. Hanaoka, Y. Kawai, N. Kunihiro, T. Matsuda, J. Weng, R. Zhang, Y. Zhao, Generic construction of chosen ciphertext secure proxy re-encryption, in: Topics in Cryptology – CT-RSA 2012 – The Cryptographers' Track at the RSA Conference 2012, Proceedings, San Francisco, CA, USA, February 27–March 2, 2012, 2012.
[16] T. Isshiki, M.H. Nguyen, K. Tanaka, Proxy re-encryption in a stronger security model extended from CT-RSA2012, in: Topics in Cryptology – CT-RSA 2013 – The Cryptographers' Track at the RSA Conference 2013, Proceedings, San Francisco, CA, USA, February 25–March 1, 2013, 2013.
[17] T. Isshiki, M.H. Nguyen, K. Tanaka, Factoring-based proxy re-encryption schemes, in: Provable Security – Proceedings of 7th International Conference, ProvSec 2013, Melaka, Malaysia, October 23–25, 2013, 2013, pp. 309–329.
[18] J. Weng, M. Chen, Y. Yang, R.H. Deng, K. Chen, F. Bao, Cca-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles, Sci. China Inf. Sci. 53 (3) (2010) 593–606.
[19] A. Shamir, Identity-based cryptosystems and signature schemes, in: Advances in Cryptology, Proceedings, CRYPTO '84, Santa Barbara, California, USA, August 19–22, 1984, 1984, pp. 47–53.
[20] M. Green, G. Ateniese, Identity-based proxy re-encryption, in: Applied Cryptography and Network Security, Proceedings of 5th International Conference, ACNS 2007, Zhuhai, China, June 5–8, 2007, 2007, pp. 288–306.
[21] S. Luo, Q. Shen, Z. Chen, Fully secure unidirectional identity-based proxy re-encryption, in: Information Security and Cryptology – ICISC 2011 – 14th International Conference, Seoul, Korea, November 30–December 2, 2011, 2011, pp. 109–126. Revised Selected Papers.
[22] Q. Tang, P.H. Hartel, W. Jonker, Inter-domain identity-based proxy re-encryption, in: Information Security and Cryptology, 4th International Conference, Inscrypt 2008, Beijing, China, December 14–17, 2008, 2008, pp. 332–347. Revised Selected Papers.
[23] L. Wang, L. Wang, M. Mambo, E. Okamoto, Identity-based proxy cryptosystems with revocability and hierarchical confidentialities, IEICE Trans. 95-A (1) (2012) 70–88.
[24] L. Wang, L. Wang, M. Mambo, E. Okamoto, New identity-based proxy re-encryption schemes to prevent collusion attacks, in: Pairing-Based Cryptography – Pairing 2010 – Proceedings of 4th International Conference, 2010, pp. 327–346.
[25] T. Mizuno, H. Doi, Secure and efficient IBE-PKE proxy re-encryption, IEICE Trans. 94-A (1) (2011) 36–44.
[26] M. Green, G. Ateniese, Identity-based proxy re-encryption, in: Applied Cryptography and Network Security, Proceedings of 5th International Conference, ACNS 2007, Zhuhai, China, June 5–8, 2007, 2007, pp. 288–306.
[27] C. Chu, W. Tzeng, Identity-based proxy re-encryption without random oracles, in: Information Security, Proceedings of 10th International Conference, ISC 2007, Valparaíso, Chile, October 9–12, 2007, 2007, pp. 189–202.
[28] H. Wang, Z. Cao, L. Wang, Multi-use and unidirectional identity-based proxy re-encryption schemes, Inform. Sci. 180 (20) (2010) 4042–4059.
[29] J. Shao, Z. Cao, Multi-use unidirectional identity-based proxy re-encryption from hierarchical identity-based encryption, Inform. Sci. 206 (2012) 83–95.
[30] Q. Tang, Type-based proxy re-encryption and its construction, in: Progress in Cryptology – INDOCRYPT 2008, Proceedings of 9th International Conference on Cryptology in India, Kharagpur, India, December 14–17, 2008, 2008, pp. 130–144.
[31] J. Weng, Y. Yang, Q. Tang, R.H. Deng, F. Bao, Efficient conditional proxy re-encryption with chosen-ciphertext security, in: Information Security, Proceedings of 12th International Conference, ISC 2009, Pisa, Italy, September 7–9, 2009, 2009, pp. 151–166.
[32] J. Weng, R.H. Deng, X. Ding, C. Chu, J. Lai, Conditional proxy re-encryption secure against chosen-ciphertext attack, in: Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2009, Sydney, Australia, March 10–12, 2009, 2009, pp. 322–332.
[33] K. Liang, Z. Liu, X. Tan, D.S. Wong, C. Tang, A CCA-secure identity-based conditional proxy re-encryption without random oracles, in: Information Security and Cryptology – ICISC 2012 – 15th International Conference, Seoul, Korea, November 28–30, 2012, 2012, pp. 231–246. Revised Selected Papers.
[34] K. Liang, C. Chu, X. Tan, D.S. Wong, C. Tang, J. Zhou, Chosen-ciphertext secure multi-hop identity-based conditional proxy re-encryption with constant-size ciphertexts, Theoret. Comput. Sci. 539 (2014) 87–105.
[35] B. Waters, Efficient identity-based encryption without random oracles, in: Advances in Cryptology – EUROCRYPT 2005, Proceedings of 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22–26, 2005, 2005, pp. 114–127.
[36] M. Bellare, S. Shoup, Two-tier signatures, strongly unforgeable signatures, and Fiat–Shamir without random oracles, IACR Cryptology ePrint Archive 2007, 2007, 273.
[37] O. Goldreich, S. Goldwasser, S. Micali, How to construct random functions, J. ACM 33 (4) (1986) 792–807.
[38] C. Chu, W. Tzeng, Identity-based proxy re-encryption without random oracles, in: Information Security, Proceedings of 10th International Conference, ISC 2007, Valparaíso, Chile, October 9–12, 2007, 2007, pp. 189–202.
[39] D. Boneh, X. Boyen, E.-J. Goh, Hierarchical identity based encryption with constant size ciphertext, Cryptology ePrint Archive, Report 2005/015, 2005.