On the Security of two Identity-Based Conditional Proxy Re-Encryption Schemes

Kai He\textsuperscript{a,c}, Jian Weng\textsuperscript{a,}\textsuperscript{*}, Robert H. Deng\textsuperscript{b}, Joseph K. Liu\textsuperscript{c}

\textsuperscript{a}Department of Computer Science, Jinan University, Guangzhou 510632, China
\textsuperscript{b}School of Information Systems, Singapore Management University, Singapore 178902
\textsuperscript{c}Faculty of Information Technology, Monash University, Australia

Abstract

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 exists 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. \cite{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

\*Corresponding author
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 ciphertexts 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 ciphertexts can continually 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, 3, 4, 5, 6, 7, 8, 9, 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 unidirectional PRE schemes from bilinear maps and both of the two schemes are chosen-ciphertext 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, 16, 17, 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 \[21, 10, 22, 23, 24, 25, 26, 27, 28, 29\].

In order to facilitate the 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 \(CT_{ID_i}^* = \text{Enc}(ID_i^*, m_\beta) = (\cdots, C^*, \cdots)\) under the target identity \(ID_i^*\), where the ciphertext component \(C^*\) is not verified. First, the adversary modifies \(C^*\) to \(C'\) and obtains another ciphertext \(CT_{ID_i}^{'ID_i} = (\cdots, C', \cdots)\). Then, it issues a re-encryption query on \(CT_{ID_i}^{'ID_i}\) to achieve another ciphertext \(CT_{ID_j}^{'ID_i}\) under a corrupted user \(ID_j\). Note that it is legal for the adversary to issue the re-encryption query. Since \((ID_i^*, CT_{ID_i}^{'ID_i})\) is not a derivative of \((ID_i^*, CT_{ID_i}^{'ID_i})\).

Next, the adversary uses the corrupted user \(ID_j\)’s private key \(sk_{ID_j}\) to derive the underlying plaintext from the ciphertext \(CT_{ID_j}^{'ID_i}\).

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

$G$ and $G_T$ are cyclic multiplicative groups of order $p$, $g$ is a generator of $G$. A bilinear map is a map $e: G \times G \rightarrow G_T$ with the following properties:

- **Bilinearity:** $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $g_1, g_2 \in G$ and $a, b \in \mathbb{Z}_p^*$.  

- **Non-degeneracy:** There exists $g_1, g_2 \in G$ such that $e(g_1, g_2) \neq 1_G$.

- **Computability:** There exists an efficient algorithm to compute $e(g_1, g_2)$ for $g_1, g_2 \in G$.

2.2. Decisional Bilinear Diffie-Hellman (DBDH) Assumption

The DBDH problem in a bilinear group $(p, G, G_T, e)$ 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 $A$ in solving the DBDH problem is defined as $Adv^\text{DBDH}_A = |\Pr[A(g, g^a, g^b, g^c, e(g, g)^{abc}) = 1] - \Pr[A(g, g^a, g^b, g^c, T) = 1]|$. where $g \in G, a, b, c \leftarrow \mathbb{Z}_p^*$, $T$ is chosen randomly from $G_T$. We say that the DBDH assumption holds in the bilinear group $(p, G, G_T, e)$ 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 \cite{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 $\prod = (\text{Setup}, \text{KeyGen}, \text{ReKeyGen}, \text{Enc}, \text{ReEnc}, \text{Dec}_2, \text{Dec}_1)$ consists of the following seven algorithms.

- **Setup$(\lambda)$:** On input a security parameter $\lambda$, 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$(mpk, sk_{ID_1}, ID_2, w)$:** On input $mpk$, the private key $sk_{ID_1}$ of an identity $ID_1$, an identity $ID_2$ and a condition $w \in \{0,1\}^*$, output a re-encryption key $rk_{w|ID_1\rightarrow ID_2}$ from $ID_1$ to $ID_2$ under $w$.
- **Enc$(mpk, ID_1, w, m)$:** On input $mpk$, an identity $ID_1$, a condition $w$ and a plaintext $m \in \{0,1\}^\lambda$, output a original ciphertext $C^{(2)}_{(ID_1, w)}$.
- **ReEnc$(mpk, rk_{w|ID_1\rightarrow ID_2}, ID_1, w, C^{(2)}_{(ID_1, w)})$:** On input $mpk$, a re-encryption key $rk_{w|ID_1\rightarrow ID_2}$, an identity $ID_1$, a condition $w$ and a original ciphertext $C^{(2)}_{(ID_1, w)}$, output a transformed ciphertext $C^{(1)}_{(ID_2, w)}$.
- **Dec$_2$(mpk, ID$_1$, sk$_{ID_1}$, w, $C^{(2)}_{(ID_1, w)}$):** On input $mpk$, an identity $ID_1$ and the corresponding private key $sk_{ID_1}$, a condition $w$ and a original ciphertext $C^{(2)}_{(ID_1, w)}$, output a plaintext $m$ or $\perp$ for failure.
- **Dec$_1$(mpk, ID$_1$, ID$_2$, sk$_{ID_2}$, w, $C^{(1)}_{(ID_2, w)}$):** On input $mpk$, an identity $ID_1$, an identity $ID_2$ and the corresponding private key $sk_{ID_2}$, a condition $w$ and a transformed ciphertext $C^{(1)}_{(ID_2, w)}$, 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.
In their model, $C$ is a challenger who plays the below game with an adversary $A$.

**Setup:** Challenger $C$ runs $\text{Setup}(1^\lambda)$ and sends $mpk$ to $A$.

**Query Phase I:** Adversary $A$ is given access to the following oracles:

- $\text{Extract}(ID)$: Given an identity $ID$, return $sk_{ID} \leftarrow \text{KeyGen}(msk, ID)$ and $ID$ is considered as corrupted.
- $\text{ReKeyExtract}(ID_i, ID_j, w)$: Given two distinct identities $ID_i$ and $ID_j$, and a condition $w$, return $rk_{w|ID_i \rightarrow ID_j} \leftarrow \text{ReKeyGen}(sk_{ID_i}, ID_j, w)$, where $sk_{ID_i} \leftarrow \text{KeyGen}(msk, ID_i)$.
- $\text{ReEnc}(ID_i, ID_j, w, C^{(2)}_{(ID_i, w)})$: 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 $C^{(1)}_{(ID_j, w)} \leftarrow \text{ReEnc}(rk_{w|ID_i \rightarrow ID_j}, ID_j, w, C^{(2)}_{(ID_i, w)})$, where $sk_{ID_j} \leftarrow \text{KeyGen}(msk, ID_j)$ and $rk_{w|ID_i \rightarrow ID_j} \leftarrow \text{ReKeyGen}(sk_{ID_i}, ID_j, w)$.
- $\text{Dec}_2(ID_i, w, C^{(2)}_{(ID_i, w)})$: Given an identity $ID_i$, a condition $w$ and a original ciphertext $C^{(2)}_{(ID_i, w)}$, return $m \leftarrow \text{Dec}_2(ID_i, sk_{ID_i}, w, C^{(2)}_{(ID_i, w)})$, where $sk_{ID_i} \leftarrow \text{KeyGen}(msk, ID_i)$.
- $\text{Dec}_1(ID_i, ID_j, w, C^{(1)}_{(ID_j, w)})$: Given two identity $ID_i$, $ID_j$, a condition $w$ and a transformed ciphertext $C^{(1)}_{(ID_j, w)}$, return $m \leftarrow \text{Dec}_1(ID_i, ID_j, sk_{ID_j}, w, C^{(1)}_{(ID_j, w)})$, where $sk_{ID_j} \leftarrow \text{KeyGen}(msk, ID_j)$.

**Challenge:** Adversary $A$ outputs two equal-length plaintexts $m_0, m_1$, a target identity $ID^*$ and a target condition $w^*$ to $C$. If the following queries:

- $\text{Extract}(ID^*)$: $ID^*$ is uncorrupt identity.
- $\text{ReKeyExtract}(ID^*, ID_j, w^*)$: Extracts $ID_j$ for any identity $ID_j$ are never made, $C$ outputs $C^{(2)*}_{(ID^*, w^*)} = \text{Enc}(ID^*, w^*, m_b)$, where $b \in R\{0, 1\}$.

**Query Phase II:** Adversary $A$ makes further queries as in Query Phase I except the following: $\text{Extract}(ID)$ if $ID = ID^*$; $\text{ReKeyExtract}(ID^*, ID_j, w^*)$ and $\text{Extract}(ID_j)$ for any identity $ID_j$; $\text{ReEnc}(ID^*, ID_j, w^*, C^{(2)*}_{(ID^*, w^*)})$ and Extracts $ID_j$ for any identity $ID_j$; $\text{Dec}_2(ID^*, w^*, C^{(2)*}_{(ID^*, w^*)})$ and $\text{Dec}_1(ID^*, ID_j, w^*, C^{(1)}_{(ID_j, w^*)})$ 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 [III], the derivative of $(ID^*, w^*, C^{(2)*}_{(ID^*, w^*)})$ is defined as follows.
1. If adversary \( A \) has issued a re-encryption key query on \((ID^*, ID_j, w^*)\) to obtain the re-encryption key \( r_{k_{w^*|ID_j|1ID_j}} \), computed \( C_{(ID_j, w^*)}^{(1)} \leftarrow \text{ReEnc}(r_{k_{w^*|ID_j|1ID_j}, ID^*, w^*, C_{(ID_j, w^*)}^{(2)*})) \), then \((ID_j, w^*, C_{(ID_j, w^*)}^{(1)})\) is a derivative of \((ID^*, w^*, C_{(ID^*, w^*)}^{(2)*})\).

2. If adversary \( A \) has issued a re-encryption query on \((ID^*, w^*, C_{(ID^*, w^*)}^{(2)*})\) and obtained \( C_{(ID^*, w^*)}^{(1)} \), then \((ID_j, w^*, C_{(ID_j, w^*)}^{(1)})\) is a derivative of \((ID^*, w^*, C_{(ID^*, w^*)}^{(2)*})\).

**Guess:** Adversary \( A \) outputs a guess bit \( b' \in \{0, 1\} \). If \( b' = b \), \( A \) wins.

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

### 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^\lambda):** Run \((q, g, G_1, G_2, e) \leftarrow G\). Let \( w \in \{0, 1\}^n \) be an \( n \)-bit condition string. Choose \( \alpha \in_R Z_p^\ast, g_2, u_1', u_2', u_3', u_{3,0} \in_R G_1 \), three random \( n \)-length sets \( U_1 = \{u_{1,i} | 1 \leq i \leq n\}, U_2 = \{u_{2,i} | 1 \leq i \leq n\}, U_3 = \{u_{3,i} | 1 \leq i \leq n\}, u_{1,i}, u_{2,i}, u_{3,i} \in_R G_1 \), a pseudorandom function \( \text{PRF}: G_2 \times G_1 \rightarrow \{0, 1\}^{\lambda_1} \), and a TCR hash function \( H_1 : G_2 \rightarrow G_1 \), where \( \lambda_1 \) is a security parameter. The master secret key is \( msk = g_2^\alpha \), the master public key is \( mpk = (\lambda, \lambda_1, g, g_1, g_2, u_1', u_2', u_3', u_{3,0}, U_1, U_2, U_3, \text{PRF}, H_1, (\text{Sign.KeyGen, Sign, Verify})) \), where \( g_1 = g^\alpha \).

**KeyGen**(msk, ID): Output \( sk_{ID} = (sk_{ID_1}, sk_{ID_2}) = (g_2^{\alpha (u_1' \prod_{i \in V_{ID}} u_{1,i})^r}, g^r) \), where \( r \in_R Z_q^\ast \), \( ID \in \{0, 1\}^n \), and let \( V_{ID} \) be the set of all \( i \) for which the \( i \)th bit of \( ID \) is set to 1.

**Enc**(ID, w, m): Run \((K_S, K_V) \leftarrow \text{Sign.KeyGen}(1^\lambda)\), choose \( t \in_R Z_q^\ast, \sigma \in_R G_2 \),

\[ \text{Enc}(ID, w, m) = (K_S, K_V, t, \sigma) \]
generate the ciphertext: $C_0 = [\text{PRF}(\sigma, C_2)]^{(\lambda_1 - \lambda)}[\text{PRF}(\sigma, C_2)]_{\lambda} \oplus m$, $C_1 = e(g_1, g_2)^t \cdot \sigma$, $C_2 = g_1^t$, $C_3 = (u_1' \prod_{i \in V_{ID_1}} u_{1,i})^t$, $C_4 = (u_2' \prod_{i \in \xi_w} u_{2,i})^t$, $C_5 = (u_3' u_{3,0} \prod_{i \in K_{K_v}} u_{3,i})^t$, $C_6 = \text{Sign}(K_S, (C_0, C_2, C_3, C_4, C_5))$, and output $C_{(ID, w)}^{(2)} = (K_V, C_0, C_1, C_2, C_3, C_4, C_5, C_6)$, where $ID_i \in \{0,1\}^n, m \in \{0,1\}^\lambda$, let $\xi_w, X_{K_v}, V_{ID_1}$ be the sets of all $i$ for which the $i$th bit of $w, K_V, ID_1$ is set to 1, respectively.

ReKeyGen$(sk_{ID_1}, ID_j, w)$: Choose $\rho, t \in R Z_p^*, \theta \in R \mathcal{G}_2$, compute $rk_0 = sk_{ID_1, t}$, $(u_2' \prod_{i \in \xi_w} u_{2,i})^\rho$, $rk_1 = g_\rho$, $rk_2 = sk_{ID_2} \cdot H_1(\theta)$, $rk_3 = e(g_1, g_2)^{t \cdot \theta}$, $rk_4 = g_\rho^t$, $rk_5 = (u_1' \prod_{i \in V_{ID_j}} u_{1,i})^{t \cdot \theta}$, $rk_6 = (u_3' u_{3,0} \prod_{i \in K_{K_v}} u_{3,i})^{t \cdot \theta}$, $rk_7 = \text{Sign}(K_S, rk_3, rk_4, rk_5, rk_6)$, and output $rk_w | ID_j \rightarrow ID_1 = (K_V', rk_0, rk_1, rk_2, rk_3, rk_4, rk_5, rk_6, rk_7)$, where $ID_j \in \{0,1\}^n$ and $(K_S', K_V') \leftarrow \text{Sign.KeyGen}(\sigma_1^\lambda)$.

ReEnc$(rk_w | ID_1 \rightarrow ID_j, ID_1, w, C_{(ID, w)}^{(2)})$: Verify the following equations $[1]$ hold or not:

$$e(g, C_3) = e(C_2, u_1' \prod_{i \in V_{ID_j}} u_{1,i}), \quad e(g, C_4) = e(C_2, u_2' \prod_{i \in \xi_w} u_{2,i})$$

$$e(g, C_5) = e(C_2, u_3' u_{3,0} \prod_{i \in K_{K_v}} u_{3,i})$$

Verify$(K_V, C_6, (C_0, C_2, C_3, C_4, C_5)) \overset{?}{=} 1$

If equations $[1]$ don’t hold, output $\bot$; else compute: $C_1' = \frac{C_1 \cdot e(rk_2, C_3)}{e(rk_0, C_2) / e(rk_1, C_3)}$, and output the transformed ciphertext $C_{(ID, w)}^{(1)} = (K_V, C_0, C_1', C_2, C_3, C_4, C_5, C_6, K_V', rk_0, rk_1, rk_2, rk_3, rk_4, rk_5, rk_6, rk_7)$.

Dec$(2)(ID_1, sk_{ID_1}, w, C_{(ID, w)}^{(2)})$: Verify equations $[1]$. If equations $[1]$ don’t hold, output $\bot$; else compute $\sigma = C_1' / e(s_{ID_1}, C_2)$, and output $m = \lfloor \text{PRF}(\sigma, C_2) \rfloor$, if $\text{PRF}(\sigma, C_2)^{\lambda_1 - \lambda} = C_0^{\lambda_1 - \lambda}$ holds; else output $\bot$.

Dec$(1)(ID_1, ID_j, sk_{ID_j}, w, C_{(ID_1, w)}^{(1)})$: Verify the following equations $[2]$ hold or not:

$$e(g, rk_5) = e(rk_4, u_1' \prod_{i \in V_{ID_j}} u_{1,i}), \quad e(g, rk_6) = e(rk_4, u_3' u_{3,0} \prod_{i \in K_{K_v}} u_{3,i})$$

Verify$(K_V', rk_7, (rk_3, rk_4, rk_5, rk_6)) \overset{?}{=} 1$

If equations $[2]$ don’t hold, output $\bot$; else compute $\theta = rk_3 \cdot \frac{e(s_{ID_2, \lambda}, C_3)}{e(s_{ID_1}, C_3)}$. If equations $[1]$ don’t hold, output $\bot$, 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 \( \bot \).

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 we find that some original ciphertext component in their construction is not verified, there might exist an adversary 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 \( A_1 \) to break the security of Liang et al.’s UniSH-IBCPRE construction. The outside adversary \( A_1 \) does not collude with the semi-trusted proxy. Second, we present an inside adversary \( A_2 \) (semi-trusted proxy) who colluded with a delegatee before and recovers a part of the delegator’s private key. Although the semi-trusted proxy \( A_2 \) 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 \( A_1 \) modifies the challenge ciphertext component \( C^{(2)_s}(ID_{*,w^*}) \) to obtain a new (ill-formed) ciphertext \( \overline{C^{(2)_s}(ID_{*,w^*})} \). Then, in the phase 2, adversary \( A_1 \) asks the re-encryption oracle to re-encrypt the new ciphertext \( \overline{C^{(2)_s}(ID_{*,w^*})} \) and gets a transformed ciphertext \( \overline{C^{(1)_s'}(ID_{j,w^*})} \), where \( ID_j \) is a corrupted user (note that according to the security model, it is legal for adversary \( A_1 \)). Next, adversary \( A_1 \) modifies the transformed ciphertext \( \overline{C^{(1)_s'}(ID_{j,w^*})} \) to obtain the right re-encrypted ciphertext \( C^{(1)_s}(ID_{j,w^*}) \) corresponding to the challenge ciphertext \( C^{(2)_s}(ID_{*,w^*}) \). Thus, adversary \( A_1 \) derives the underlying plaintext by decrypting \( C^{(1)_s}(ID_{j,w^*}) \) 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.

**Setup:** Adversary $A_1$ first obtains the public parameters from challenger $C$.

**Query Phase I:** Adversary $A_1$ issues the $\text{Extract}(ID_j)$ oracle to obtain the private key of the $ID_j$ and adds $ID_j$ to a corrupt list.

**Challenge:** Adversary $A_1$ submits $(ID_{i^*}, m_0, m_1, w^*)$ to challenger $C$, and then given the challenge ciphertext $C^{(2)*}_{(ID_{i^*}, w^*)} = (K_V, C_0^*, C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*)$:

\[ C_6^* = \left[ \text{PRF}(\sigma^*, C_2^*) \right]^{\lambda_1 - \lambda} \cdot \left[ \text{PRF}(\sigma^*, C_5^*) \right] = e(g_1, g_2)^{t^*} \cdot \sigma^*, \]

\[ C_2^* = g^{t^*}, C_3^* = (u_1^t \prod_{i \in V \cap D_1^*} u_{i, i})^t, C_4^* = (u_2^t \prod_{i \in \xi} u_{2, i})^t, C_5^* = (u_3^t u_{3, 0} \prod_{i \in \chi \cap \xi} u_{3, i})^t, \]

\[ C_0^* = \text{Sign}(K_{V}^*, (C_0^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*)). \]

**Query Phase II:** Adversary $A_1$ issues the re-encryption oracle as follows:

First, adversary $A_1$ picks $C_1^* \in_R G_2$, and lets $C_1^* = C_1^* \cdot C_1^*$. Then adversary $A_1$ modifies the challenge ciphertext $C^{(2)*}_{(ID_{i^*}, w^*)}$ to obtain a new (ill-formed) ciphertext $C^{(2)*}_{(ID_{i^*}, w^*)} = (K_V^*, C_0^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*)$. Now adversary $A_1$ submits $(ID_{i^*}, ID_j, w^*, C^{(2)*}_{(ID_{i^*}, w^*)})$ to the re-encryption oracle. (Note that, although $ID_j$ is in the corrupt list, it is legal for adversary $A_1$ to issue this query, since $(ID_{i^*}, ID_j, w^*, C^{(2)*}_{(ID_{i^*}, w^*)})$ is not a derivate of $(ID_{i^*}, ID_j, w^*, C^{(2)*}_{(ID_{i^*}, w^*)})$).

The main reason is that the re-encryption algorithm $\text{ReEnc}$ cannot check the validity of the ciphertext component $C_1^*$. So the re-encryption oracle still responds the re-encryption ciphertext $\tilde{C}^{(1)}_{(ID_{i^*}, w^*)} = \text{ReEnc}(\text{params}, \text{ReKeyGen}(\text{params}, sk_{ID_{i^*}}, ID_j, w^*), ID_{i^*}, ID_j, w^*, C^{(2)*}_{(ID_{i^*}, w^*)})$ to adversary $A_1$, where $\tilde{C}^{(1)}_{(ID_{i^*}, w^*)} = (K_V^*, C_0^*, C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*, K_V, r_k^3, r_k^4, r_k^5, r_k^6, r_k^7)$. In fact, we have

\[ \tilde{C}_1^* = \frac{C_1^* \cdot e(rk_2, C_3)}{e(rk_0, C_2)/e(rk_1, C_4)} = \frac{C_1^* \cdot e(rk_2, C_3)}{e(rk_0, C_2)/e(rk_1, C_4)}. \]

Next, adversary $A_1$ uses $\tilde{C}_1^*$ to recover the real transformed ciphertext $\tilde{C}_1^*$:

\[ \tilde{C}_1^* = \frac{\tilde{C}_1^*}{C_1^*} = \frac{C_1^* \cdot e(rk_2, C_3)}{C_1^* \cdot e(rk_0, C_2)/e(rk_1, C_4)} = \frac{C_1^* \cdot e(rk_2, C_3)}{e(rk_0, C_2)/e(rk_1, C_4)}. \]

Observe that, $\tilde{C}_1^*$ is indeed transformed by the challenge ciphertext component $C_1^*$. Thus, the ciphertext $\tilde{C}^{(1)}_{(ID_{i^*}, w^*)} = (K_V^*, C_0^*, \tilde{C}_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*, K_V, r_k^3, r_k^4, r_k^5, r_k^6, r_k^7)$ is indeed transformed by the challenge ciphertext $C^{(2)*}_{(ID_{i^*}, w^*)}$.
Now, adversary $A_1$ owns the colluded private key $sk_{ID_j}$ to obtain the underlying plaintext $m_\beta$ by decrypting the re-encryption ciphertext $C_{(ID_j, w^*)}^{(1)'_j}$.

**Guess:** Adversary $A_1$ outputs a bit $\beta'$.

Obviously, adversary $A_1$ has non-negligible advantage to output $\beta' = \beta$. It implies that Liang et al.'s UniSH-IBCPRE scheme cannot obtain chosen-ciphertext security.

**Inside Attack**

For a semi-trusted proxy $A_2$, given the re-encryption key $rk_{w[ID_i \rightarrow ID_j]} = (K'_V, rk_0, rk_1, rk_2, rk_3, rk_4, rk_5, rk_6, rk_7)$, where $rk_2 = sk_{ID_{1_2}} \cdot H_1(\theta)$. Then the semi-trusted proxy $A_2$ can transform the delegator’s ciphertext to the delegatee’s ciphertext. Assuming that the semi-trusted proxy $A_2$ corrupts with a corresponding delegatee or it acts as a delegatee before, then it can obtain a additional value $H_1(\theta)$. At last $A_2$ can compute the second part private key $sk_{ID_{1_2}}$ of the delegator. After that, every time, once the delegator gives the re-encryption key to the semi-trusted proxy $A_2$, then the semi-trusted proxy $A_2$ can compute $H_1(\theta')$ from the corresponding re-encryption key. Thus, the semi-trusted proxy $A_2$ 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 $A_2$, 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 $A_2$ be a semi-trust proxy, $A_2$ interacts with challenger $\mathcal{C}$ in the following game.

**Setup**: Adversary $A_2$ obtains the public parameters from challenger $\mathcal{C}$.

**Query phase I**: Adversary $A_2$ issues the following queries:

- Adversary $A_2$ issues the corrupted extract $\text{Extract}(ID_j)$ oracle to obtain the private key of $ID_j$ and adds $ID_j$ to a corrupt list.
- Adversary $A_2$ issues the re-encryption key generation oracle $\text{ReKeyExtract}(ID_i^*, ID_j, w)$, where $ID_j$ is corrupted, and gets $rk_{w[ID_i^* \rightarrow ID_j]} = (K'_V, rk_0, rk_1,$
\(rk_2, rk_3, rk_4, rk_5, rk_6, rk_7\), where \(rk_0 = sk_{1D_i} \cdot (u_1^3 \prod_{i \in \xi_w} u_{2,i})^{u}, \quad rk_1 = g^u, \quad rk_2 = sk_{1D_i} \cdot H_1(\theta), \quad rk_3 = e(g_1, g_2)^{\theta}, \quad rk_4 = g^{\theta}, \quad rk_5 = (u_1^3 \prod_{i \in \xi_w} u_{2,i})^{\theta}, \quad rk_6 = (u_3^3 \cdot 3 \prod_{i \in \xi_w} u_{3,i})^{\theta}, \quad rk_7 = \text{Sign}(K_{S_i}^{\ast}, (rk_3, rk_4, rk_5, rk_6))\). Then, semi-trusted proxy \(A_2\) colludes the user \(ID_j\) and uses the corrupted private key \(sk_{1D_j} = (sk_{1D_j}, sk_{1D_j})\) to compute \(\theta = rk_3 \cdot e(sk_{1D_j}^{\ast}, sk_{1D_j})^{\ast}, \) and then compute the second part of the delegator’s private key \(sk_{1D_j}^{\ast} = rk_2 / H_1(\theta)\). (Note that although semi-trusted proxy \(A_2\) cannot recover the first private key \(sk_{1D_i}\) of the delegator \(ID_i\), but it is enough for \(A_2\) to recover all the message encrypted under \(ID_i\).

**Challenge:** Adversary \(A_2\) submits \((ID_i^{\ast}, m_0, m_1, w^\ast)\) to challenger \(C\), and then given the challenge ciphertext \(C_{ID_i^{\ast}, w^\ast} = (K_{V_i}^{\ast}, C_0^{\ast}, C_1^{\ast}, C_2^{\ast}, C_3^{\ast}, C_4^{\ast}, C_5^{\ast}, C_6^{\ast})\) as follows: \(C_0^{\ast} = [\text{PRF}(\sigma^\ast, C_2^{\ast}) \mid \lambda - \lambda = | [\text{PRF}(\sigma^\ast, C_2^{\ast}) \mid \lambda \oplus m_\beta, \quad C_1^{\ast} = e(g_1, g_2)^{\ast \cdot \sigma^\ast}, \quad C_2^{\ast} = g^{\ast}, \quad C_3^{\ast} = (u_1^3 \prod_{i \in \xi_{ID_j}} u_{1,i})^{\ast}, \quad C_4^{\ast} = (u_2^3 \prod_{i \in \xi_w} u_{2,i})^{\ast}, \quad C_5^{\ast} = (u_3^3 \cdot 3 \prod_{i \in \xi_w} u_{3,i})^{\ast}, \quad C_6^{\ast} = \text{Sign}(K_{S_i}^{\ast}, (C_0^{\ast}, C_2^{\ast}, C_3^{\ast}, C_4^{\ast}, C_5^{\ast})).

**Query Phase II:** In this phase, \(A_2\) issues the following queries:

Semi-trusted proxy \(A_2\) continues to issue the re-encryption key generation oracle \(\text{ReKeyExtract}(ID_i^{\ast}, ID_k, w^\ast)\), where \(ID_k\) is uncorrupt, challenger \(C\) gives the re-encryption key \(rk_{w^\ast, ID_i^{\ast}} \rightarrow ID_k = (K_{V_i}^{\ast}, rk_0, rk_1, rk_2, rk_3, rk_4, rk_5, rk_6, rk_7)\) to \(A_2\), where \(rk_2 = sk_{1D_j} \cdot H_1(\theta)\). Thus, the semi-trusted proxy \(A_2\) can compute \(H_1(\theta) = rk_2 / sk_{1D_j}^{\ast}\) using the second private key \(sk_{1D_j}^{\ast}\). (Semi-trusted proxy \(A_2\) does not need to corrupt any delegatee now).

Next, \(A_2\) runs the re-encryption algorithm \(\text{ReEnc}(rk_{w^\ast, ID_i^{\ast}} \rightarrow ID_k, ID_i^{\ast}, w^\ast, \ C_{ID_i^{\ast}, w^\ast}^{\ast})\) to obtain re-encryption ciphertext \(C_{ID_i^{\ast}, w^\ast}^{\ast} = (K_{V_i}^{\ast}, C_0^{\ast}, C_1^{\ast}, C_2^{\ast}, C_3^{\ast}, \ C_4^{\ast}, C_5^{\ast}, C_6^{\ast}, K_{V_i}^{\ast}, rk_3, rk_4, rk_5, rk_6, rk_7)\), where \(C_1^{\ast} = e(rk_0, C_2^{\ast}) / e(rk_1, C_4^{\ast}) = \sigma^\ast \cdot e(H_1(\theta), C_3^{\ast})\). Thus, semi-trusted proxy \(A_2\) can compute \(\sigma^\ast = C_1^{\ast} / e(H_1(\theta), C_3^{\ast})\).

Last, semi-trusted proxy \(A_2\) can easily compute \(m_\beta = [\text{PRF}(\sigma^\ast, C_2^{\ast}) \mid \lambda \oplus [C_0^{\ast}] \lambda \cdot \lambda\). (Note that \(A_2\) does not use the colluded private key of user \(ID_j\) to obtain the plaintext \(m_\beta\).

**Guess:** Semi-trusted proxy \(A_2\) outputs \(\beta^\ast\).

Obviously, semi-trusted proxy \(A_2\) has non-negligible advantage to output
It implies that semi-trusted proxy $A_2$ can recover all the message of the delegator, as long as $A_2$ colludes with one delegatee one time. So 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 $\Pi = (\text{Setup}, \text{KeyGen}, \text{Re-Encryption Key Generation Protocol}, \text{Enc}, \text{ReEnc}, \text{Dec})$ consists of the following algorithms and protocols.

\textbf{Setup}(1^\lambda, n): On input a security parameter $1^\lambda$, and $n \in \mathcal{N}$ the allowable maximum number of condition in the system, output a master public key $\text{mpk}$ and a master secret key $\text{msk}$ for Private Key Generator (PKG).

\textbf{KeyGen}(mpk, msk, ID): On input $mpk, msk$, and an identity $ID \in \{0, 1\}^*$, output a private key $sk_{ID}$ for the identity $ID$.

\textbf{Re-Encryption Key Generation Protocol}: For simplicity, $i$ and $j$ denotes $ID_i$ and $ID_j$ in the re-encryption key.

\textbf{PReKeyGen}(mpk, sk_{ID_j}, W): On input $mpk$, a private key $sk_{ID_j}$ for identity $ID_j$, and a condition set $W = \{w_z|1 \leq z \leq n, w_z \in \{0, 1\}^*\}$, the partial re-encryption key generation algorithm PReKeyGen outputs a partial re-encryption key $prk_{(W,j)}$ under $W$ and $ID_j$.

\textbf{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 $rk_{i \rightarrow j|W}$ from $ID_i$ to $ID_j$ under $W$. Note $ID_i$ and $ID_j$ are two distinct identities.

\textbf{ReKeyBiGen}(rk_{i \rightarrow j|W}): On input a re-encryption key $rk_{i \rightarrow j|W}$ 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 \( rk_{j \rightarrow i|W} \) from ID_j to ID_i under W. Note that this algorithm also allows the re-encryption key \( rk_{j \rightarrow i|W} \) holder to generate a new re-encryption key \( rk_{i \rightarrow j|W} \).

**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, \( rk_{i \rightarrow j|W} \), C_{(ID_i,W)}): On input mpk, a re-encryption key \( rk_{i \rightarrow j|W} \), and a ciphertext \( C_{(ID_i,W)} \), the re-encryption algorithm ReEnc outputs a re-encrypted ciphertext \( C_{(ID_j,W)} \) or a symbol ⊥ 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 ⊥ 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 al.’s BiMH-IBCPRE scheme [34]. 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 \( sk_{ID} \leftarrow KeyGen(msk, ID) \) 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 \( rk_{i \rightarrow j|W} \leftarrow ReKeyGen(sk_{ID_i}, PReKeyGen(sk_{ID_j}, W), W) \), where \( sk_{ID_i} \leftarrow KeyGen(msk, ID_i) \), \( sk_{ID_j} \leftarrow KeyGen(msk, ID_j) \), and \( ID_i, ID_j \in \{0, 1\}^* \). Note that \( \mathcal{A} \) can derive \( rk_{j \rightarrow i|W} \) from \( rk_{i \rightarrow j|W} \) with algorithm ReKeyBiGen.

*Re-encryption oracle* \( \mathcal{O}_{re}(ID_i, ID_j, W, C_{(ID_i,W)}) \): On input two distinct identi-
ties $ID_i$ and $ID_j$, a condition set $W$, and a ciphertext $C_{(ID_i,W)}$ under $ID_i$ and $W$, $C$ returns a re-encrypted ciphertext $C_{(ID_j,W)} \leftarrow \text{ReEnc}(rk_{i\rightarrow j|W}, C_{(ID_i,W)})$, where $rk_{i\rightarrow j|W} \leftarrow \text{ReKeyBiGen}(ri_{j\rightarrow i|W})$, and further re-encrypts $C_{(ID_i,W)}$ to $O_{re}$. If so, $C$ will first generate $rk_{j\rightarrow i|W}$ and get $rk_{i\rightarrow j|W} \leftarrow \text{ReKeyBiGen}(rk_{j\rightarrow i|W})$, and further re-encrypt $C_{(ID_i,W)}$ using $rk_{i\rightarrow j|W}$.

**Decryption oracle $O_{dec}(ID_i, C_{(ID_i,W)})$:** On input an identity $ID_i$, and a ciphertext $C_{(ID_i,W)}$, $C$ returns $m \leftarrow \text{Dec}(sk_{ID_i}, C_{(ID_i,W)})$, where $sk_{ID_i} \leftarrow \text{KeyGen}(msk, ID_i)$, $ID_i \in \{0,1\}^*$. Note that if $A$ issues invalid ciphertext to $O_{re}$ or $O_{dec}$, $C$ simply outputs $\bot$. Moreover, the following the queries cannot be issued:

1. $O_{sk}(ID)$, if $ID^* = ID$ or for any $ID$ in an uncorrupted delegation chain under $W^*$ which includes $ID^*$;
2. $O_{rk}(ID_i, ID_j, W^*)$ 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:** $A$ outputs two distinct equal length messages $(m_0, m_1)$ to $C$. $C$ returns the challenge ciphertext $C^*_{(ID^*,W^*)} = \text{Enc}(ID^*, W^*, m_b)$ to $A$, where $b \in_R \{0,1\}$.

**Phase 2:** $A$ continues making queries except the followings:

1. $O_{sk}(ID)$ if $ID^* = ID$ or for any $ID$ in an uncorrupted delegation chain under $W^*$ which includes $ID^*$;
2. $O_{rk}(ID_i, ID_j, W^*)$ 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. $O_{re}(ID_i, ID_j, W^*, C_{(ID_i,W^*)})$ 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.

15
If \( A \) has issued a re-encryption key query on \( (ID_i, ID_j, W) \) to obtain \( rk_{i \rightarrow j|W} \), and achieved \( C(ID_j, W, C(ID_j, W)) \) is a derivative of \( (ID_i, W, C(ID_i, W)) \).

If \( A \) can run \( C(ID_j, W) \leftarrow \text{ReEnc}(\text{ReKeyGen}(sk_{ID_i, prk(W,j)}, W), C(ID_i, W)) \), then \( (ID_j, W, C(ID_j, W)) \) is a derivative of \( (ID_i, W, C(ID_i, W)) \), where \( sk_{ID_i} \leftarrow \text{KeyGen}(msk, ID_i) \), \( prk(W,j) \leftarrow \text{PreKeyGen}(sk_{ID_j}, W) \) and \( sk_{ID_j} \leftarrow \text{mathcalGen}(msk, ID_j) \).

If \( A \) 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) \( O_{dec}(ID_i, C(ID_i, W^*)) \) if \( (ID_i, W^*, C(ID_i, W^*)) \) is a derivative of \( (ID^*, W^*, C^*(ID^*, W^*)) \).

**Guess:** \( A \) 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 \( A \) wins the above game with negligible advantage
\[
\epsilon_1 = \text{Adv}_{\text{BiMH-IBCPRE}, A}^{\text{IND-sCon-sID-CCA}}(1^\lambda, n) = |\Pr[b' = b] - 1/2|.
\]

### 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 on 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 \( \lambda \) and \( n \) the allowable maximum number of conditions in the system (here \( n = 1 \)), run \((g, g, G_1, G_T, e) \leftarrow G(1^\lambda)\), choose \( \alpha \in R \mathbb{Z}_p^*, f_1, f_2, g_2, g_3, h_1, h_2, h_3 \in R G_1 \), and prepare a pseudorandom function \( \text{PRF}: G_T \times G_1 \rightarrow \{0, 1\}^{\lambda_1} \) (which takes an element in \( G_T \) as the function key and an element in \( G_1 \) as input, and outputs a \( \lambda_1 \)-bit pseudorandom string), where \( \lambda_1 \) is a security parameter as well. Let \( w \in \mathbb{Z}_q^* \) be a condition, \((\text{Sign.KeyGen}, \text{Sign}, \text{Verify})\) be an OTS scheme, and assume the verification key output by \( \text{Sign.KeyGen}(1^\lambda) \) is in \( \mathbb{Z}_q^* \). The master secret key is \( msk = g_2^a \),
and the master public key is $mpk = (q, g, G_1, G_T, e, f_1, f_2, g_1, g_2, g_3, h_1, h_2, h_3, PRF,(\text{Sign.KyGen, Sign.Verify}))$, where $g_1 = g^a$.

**KeyGen**(m$sk$, ID): Given the master secret key m$sk$ and an identity ID (i.e. an identity $I = \langle ID \rangle$), choose $r \in_R Z_1^*$ and output the key $sk_{ID} = (g_2^a \cdot (h_1^{ID} \cdot g_3)^r, g^r, h_2^r, h_3^r) \in G_1^4$. A system user with knowledge of $sk_{ID}$ can generate the following private key due to the key derivation of HIBE. Given $sk_{ID} = (a_0, a_1, b_2, b_3), I = \langle ID, w \rangle$, choose a $t \in_R Z_1^*$ and output $sk_{<ID,w>} = (a_0 \cdot b_2^a \cdot (h_1^{ID} \cdot h_2^w \cdot g_3)^t, a_1 \cdot g^t, b_3 \cdot h_3^t) = (g_2^a \cdot (h_1^{ID} \cdot h_2^w \cdot g_3)^t \cdot g^r, h_2^r, h_3^r) \in G_1^4$, where $r' = r + t$.

**Enc**(ID, $w$, m): Given an identity ID, a condition $w$ and a message $m$, choose an OTS key pair $(K_s, K_v)$ ← Sign.KyGen(1$^\lambda$) and $s \in_R G_T$, set $C_0 = K_v$, $C_1 = [PRF_s(C_3)]^{\lambda = 1}$, $[PRF_s(C_3)]^m \cdot m$, $C_2 = \sigma \cdot e(g_1, g_2)^s$, $C_3 = g^s$, $C_4 = (h_1^{ID} \cdot h_2^w \cdot h_3^r \cdot g_3)^s$, $C_5 = (f_1^w \cdot f_2)^s$, $C_6 = \text{Sign}(K_s, (C_1, C_3, C_4, C_5))$, and output the transformed ciphertext $C_{<ID,w>} = \langle \langle ID, w \rangle, C_0, C_1, C_2, C_3, C_4, C_5, C_6 \rangle$, where $ID \in Z_1^*$, $m \in \{0, 1\}^\lambda$.

**Re-Encryption Key Generation Protocol:**

$PReKeyGen$(sk$_{<ID,w>}$). ID$_j$ first deduces $sk_{<ID,w>} = (a_0, a_1, b_3)$ (under $I = \langle ID, w \rangle$) from $sk_{ID}$, next chooses $\rho_1, \rho_2 \in_R Z_1^*$ and sets $\beta_1 = a_{01}^{-1} \cdot (f_1^w \cdot f_2)^{\rho_1}, \beta_2 = g^{\rho_1}, \beta_3 = a_{1j}^{-1} \cdot g^{\rho_2}, \beta_4 = b_{3j} \cdot h_3^{\rho_2}, \beta_5 = (h_2^w \cdot g_3)^{\rho_2}, \beta_6 = h_2^{\rho_2}$. 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$_{ID}$, prk$_{<w,j>}$). ID$_i$ first generates $sk_{<ID,w>} = (a_0, a_{ij}, b_{3j})$ (under $I = \langle ID, w \rangle$), chooses $\rho_3, \rho_4 \in_R Z_1^*$, computes $rk_1 = a_0 \cdot \beta_1 \cdot (f_1^w \cdot f_2)^{\rho_3}$, $rk_2 = g^{\rho_3} \cdot \beta_2$, $rk_3 = a_{1j} \cdot \beta_3 \cdot g^{\rho_4}$, $rk_4 = b_{3j} \cdot \beta_4 \cdot h_3^{\rho_4}$, $rk_5 = (h_2^w \cdot g_3)^{\rho_4} \cdot \beta_5$, $rk_6 = h_1^{\rho_4} \cdot \beta_6$ and outputs $rk_{i \rightarrow j|w} = (rk_1, rk_2, rk_3, rk_4, rk_5, rk_6)$.

Derive a new re-encryption key $rk_{j \rightarrow i|w}$ from $rk_{i \rightarrow j|w}$ by running $rk_{j \rightarrow i|w}$ ← $ReKeyBiGen$(rk$_{i \rightarrow j|w}$). The proxy sets $rk_{j \rightarrow i|w} = rk_{i \rightarrow j|w}^{-1}$, i.e. $rk_{j \rightarrow i|w} = (rk_1^{-1}, rk_2^{-1}, rk_3^{-1}, rk_4^{-1}, rk_5^{-1}, rk_6^{-1})$.

**ReEnc**(rk$_{i \rightarrow j|w}$, C$_{<ID,w>}$): Given a re-encryption key $rk_{i \rightarrow j|w}$ and a ciphertext C$_{<ID,w>}$, the re-encryption algorithm works as follows.
The same as in the above Setup:

Outside Attack

A PPT attacker, outside attack for Liang et al.’s UniSH-IBC PRE construction [33]. Let et al.’s BiMH-IBC PRE construction [34]. The attack is the same as the above

4.4. Security Analysis for Liang et al.’s BiMH-IBC PRE Construction

In the following, we shall present one concrete outside attack against Liang et al.’s BiMH-IBC PRE construction [34]. The attack is the same as the above outside attack for Liang et al.’s UniSH-IBC PRE construction [33]. Let $A_1$ be a PPT attacker, $A_1$ interacts with challenger $C$ 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 $A_1$ submits $(ID_{i^*}, m_0, m_1, w^*)$ to challenger $C$, then given the challenge ciphertext $C^*_i(ID_{i^*}, w^*) = ((ID_{i^*}, w^*), C^*_0, C^*_1, C^*_2, C^*_3, C^*_4, C^*_5, C^*_6)$ as follows: 

$$C^*_0 = K^*_s, \quad C^*_1 = \text{PRF}_{\sigma^*}(C^*_3)^{\lambda_1-\lambda}(\text{PRF}_{\sigma^*}(C^*_3))_\lambda \oplus m_\beta, \quad C^*_2 = \sigma^* \cdot e(g_1, g_2)^*, \quad C^*_3 = g^*, \quad C^*_4 = (h^*_1 \cdot h^*_2 \cdot h^*_3 \cdot g_3)^*, \quad C^*_5 = (f^*_1 \cdot f^*_2)^*, \quad C^*_6 = \text{Sign}(K^*_s, (C^*_1, C^*_3, C^*_4, C^*_5))$$.
Query Phase II: Adversary $A_1$ issues the re-encryption oracle as follows:

Adversary $A_1$ first randomly picks $C_2 \in \mathcal{G}_2$ and lets $C_2^* = C_2 \cdot C_2$. Then adversary $A_1$ modifies the challenge ciphertext to obtain a new (ill-formed) ciphertext $C'_{(ID_i , w^*)} = ( (ID_i, w^*), C_0^*, C_1', C_2^*, C_3, C_4^*, C_5, C_6^* )$. Now adversary $A_1$ submits $(ID_{i'}, ID_j, w^*, C'_{(ID_{i'}, w^*)})$ to the re-encryption oracle. (Note that, although $ID_j$ is in the corrupt list, it is legal for $A_1$ to issue this query, since $(ID_{i'}, ID_j, w^*, C'_{(ID_{i'}, w^*)})$ 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 $C_2^*$. So the re-encryption oracle can return the re-encryption ciphertext $C'_2 = ReEnc(params, ReKeyGen(params, sk_{ID_i}, ID_j, w^*))$ to adversary $A$, where $C'_{(ID_j, w^*)} = ( (ID_j, w^*), C_0^*, C_1', C_2^*, C_3, C_4^*, C_5, C_6^* )$ and $C_2' = C_{2'} = C_2' = C_2 \cdot C_2 = C_2 \cdot C_2$. Then, adversary $A_1$ uses $C_2'$ to compute $C_2$ as follows:

$$C_2 = \frac{C_2'}{C_2} = \frac{C_{2'} \cdot C_{2'} \cdot e(rk_3, C_4) \cdot e(rk_2, C_5)}{e(rk_1, r_k^4 \quad r_k^5 \quad r_k^6 \quad r_k^7 \quad r_k^8 \quad r_k^9 \quad r_k^{10} \quad r_k^{11} \quad r_k^{12} \quad r_k^{13} \quad r_k^{14})} = \frac{C_{2'} \cdot e(rk_3, C_4) \cdot e(rk_2, C_5)}{e(rk_1, r_k^4 \quad r_k^5 \quad r_k^6 \quad r_k^7 \quad r_k^8 \quad r_k^9 \quad r_k^{10} \quad r_k^{11} \quad r_k^{12} \quad r_k^{13} \quad r_k^{14})}.$$

Observe that, we find $C_2'$ is indeed transformed by the challenge ciphertext component $C_2^*$. Thus, the modified re-encryption ciphertext $C'_{(ID_j, w^*)} = ( (ID_j, w^*), C_0^*, C_1', C_2^*, C_3, C_4^*, C_5, C_6^* )$ is indeed the result of $ReEnc(rk_{w^*}, ID_{i'}, ID_{i'}, w^*)$, which is an encryption of $m_{\beta}$. Now, adversary $A_1$ can obtain the underlying plaintext $m_{\beta}$ by decrypting the re-encryption ciphertext $C'_{(ID_j, w^*)}$ using user $ID_j$’s private key $sk_{ID_j}$.

**Guess:** Adversary $A_1$ outputs $\beta'$. Obviously, adversary $A_1$ 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 \cite{33} and the component $C_2$ in the BiMH-IBCPRE scheme \cite{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.

Acknowledgment

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.

References


[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 with-


[34] K. Liang, C. Chu, X. Tan, D. S. Wong, C. Tang, J. Zhou, Chosen-


