Computer Science and Information Systems 15(3):585–614 https://doi.org/10.2298/CSIS171218024W

Improved Functional Proxy Re-encryption Schemes for Secure Cloud Data Sharing

Xu An Wang[1,2], Xiaoyuan Yang[1], Cong Li[1], Yudong Liu[1], and Yong Ding[2]

[1] Key Laboratory of Information and Network Security, Engineering University of Chinese Armed Police Force, P. R. China

[2] Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, P. R. China

wangxazjd@163.com, xyyangwj@126.com, wugongcong@163.com, 1269124170@qq.com, 284722748@qq.com

Abstract. Recently Liang et al. propose an interesting privacy-preserving ciphertext multi-sharing control for big data storage mechanism, which is based on the cryptographic primitive of anonymous multi-hop identity based conditional proxy re-encryption scheme AMH-IBCPRE. They propose a concrete AMH-IBCPRE scheme and conclude their scheme can achieve IND-sCon-sID-CCA secure (indistinguishable secure under selectively conditional selectively identity chosen ciphertext attack). However, our research show their scheme can not be IND-sCon-sID-CCA secure for single-hop and multi-hop data sharing. Also in 2014, Liang et al. propose an interesting deterministic finite automata-based functional proxy re-encryption scheme DFA-based FPRE for secure public cloud data sharing, they also conclude their scheme can achieve IND-CCA secure (indistinguishable secure under chosen ciphertext attack), we also show their scheme can not be IND-CCA secure either. For these two proposals, the main reason of insecurity is that part of the re-encryption key has the same structure as the valid ciphertext, thus the adversary can query on the decryption oracle with this part of the re-encryption key to get secret keys, which will break the CCA-security of their scheme. We give an improved AMH-IBCPRE scheme and an improved DFA-based FPRE scheme for cloud data sharing and show the new schemes can resist our attack and be CCA-secure. We also demonstrate our improved AMH-IBCPRE scheme's efficiency compared with other related identity based proxy re-encryption schemes, the results show our scheme is almost the most efficient one.

Keywords: Attack, multi-control for big data storage, secure cloud data sharing, proxy re-encryption, chosen ciphertext security.

1. Introduction

Nowadays cloud computation and big data are very hot research topics. Cloud computation can be seen as a paradigm of aggregating many various kinds of computation resources and storage resources into very powerful computation grids. There are many advanced techniques to support the smooth running of cloud computation like mapreduce, hadoop, vmvare etc [8, 35–39]. Big data mainly refers to the burst generation of massive data sets everytime/everywhere/everyspace, which is a very common scenario these days, it mainly concerns on how to organize and process data efficiently. If we use traditional

data management or knowledge discover techniques, many useful results can not be obtained, thus it is a challenge on how to organize/store/process big data.

Let us focus on one of the most basic challenge for big data, that is, how to secure and scalably store these large amount of data. Cloud computation is such a promising technique. Data owners first outsource datum to the cloud servers to reduce the local management overhead. They often worry about the security and privacy if directly outsourcing their data sets to the cloud, thus it is often a practice that the data owners first encrypt datum and then upload them to the cloud [18–21]. But this time scalability is a new problem, can these encrypted data be shared efficiently and scalably? Aiming at solving this issue, Blaze et al. [5] and Atenesis et al. [3,4] proposed a new notion called proxy re-encryption (PRE), which allows the proxy with re-encryption key can transform the original ciphertext associated with a public key to the re-encrypted ciphertext associated with another public key.

In proxy re-encryption(PRE), a proxy can transform ciphertexts for Alice to ciphertexts for Bob without the proxy knowing either Alice or Bob's private key and the underlying plaintexts. PRE has very important applications for secure cloud storage and its sharing [32–34]. CCA-security (chosen ciphertext secure) is an important security notion for proxy re-encryption, which means that the schemes can resist the attacking even with adaptive decryption oracle [6,7,14,15,24,26–31]. Multi-hop PRE refers the scheme can support ciphertexts being able to be re-encrypted again, while single-hop PRE has no this ability. For the users' secret keys or the underlying plaintexts can not be induced by the proxy, this primitive can be used for efficiently and scalably sharing encrypted contents for multi-users in cloud. The cloud servers can be acted as proxy and transform the ciphertexts for the users, otherwise the data owner has to first download his data sets and then re-encrypt them, which is very inefficient. Conditional proxy re-encryption (CPRE) aims at more flexible re-encryption for PRE, which can only re-encrypt the ciphertexts which satisfying some condition. Multi-hop identity based conditional proxy re-encryption (MH-IBPRE) [9, 16]combines the feature of identity based encryption (IBE) with CPRE, which simultaneously has the advantage of no relying on public key infrastructure, supporting fine-grained re-encryption and multi-hop ciphertext transformation. However, directly using MH-IBPRE in multi-sharing control for big data storage may leak the identities of the users, and thus break the user's privacy. Therefore, Liang et al. exploited to use anonymous MH-IBPRE (AMH-IBCPRE) scheme to achieve the privacy persevering property when sharing data among multi-users. Fairly to say, their work give an interesting solution on the issue of how to secure share data storage efficiently and scalably in big data era. Unfortunately, the authors claimed their scheme can achieve IND-sCon-sID-CCA security, but we show their scheme can not be CCA-secure.

The traditional proxy re-encryption schemes or conditional proxy re-encryption can not easily achieve fine-grained delegation, although attribute based proxy re-encryption schemes can achieve more flexible delegation than PRE etc, but most of them can only be proved to secure under chosen plaintext secure in the selective model. Furthermore, almost all existing ABPRE schemes only support access policy assembling with AND gates and fixed size inputs, which is less expressive than the access policy assemble with AND, OR gates and NOT, and access policy expressed by regular languages with arbitrary size which supporting unlimited input size. Therefore, Liang et al. [17] first proposed the notion of deterministic finite automata-based functional proxy re-encryption, which

combines the feature of DFA-based functional encryption with proxy re-encryption. The re-encrypted ciphertext can only be decrypted if and only if the delegatee's deterministic finite automata accepts the string associated with the re-encrypted ciphertext. Really this is a novel notion and generalize the concept of proxy re-encryption to a very widely extension. They also proposed a concrete DFA-based PRE scheme and claimed it is fully CCA-secure in the standard model. Fairly to say, this is an interesting and foundational work, but unfortunately, we show their scheme can also not achieve CCA-security by exploiting some shortcoming of the form of their re-encryption keys.

1.1. Cloud Storage and Chosen Ciphertext Security

Chosen ciphertext security is a very important security notion for encryption in cryptographic society, and now it is considered as the gold standard for evaluation of the encryption schemes' security. Also it is very important for cloud storage, Figure 1 demonstrates several chosen ciphertext attacks on the cloud storage, which shows that secure cloud storage also should achieve this gold security notion.

1. Assume the attacker is the malicious cloud, it can launch the chosen ciphertext attack on the data owners. First the data owner outsources his datum to the cloud by first encrypting and then uploading the ciphertexts, after that some day when the data owner want to retrieve his data, the cloud maliciously return the modified ciphertexts to him, and he will decrypt the modified ciphertexts and find the decrypted results are not correct. The data owner will notify the cloud on this error, and thus the cloud can utilize this information to launch the chosen ciphertext attack.
2. The malicious cloud can also launch the chosen ciphertext attack on the data users. First the data owner shares his datum with other data users via proxy re-encryption. The cloud, which is the proxy for implementing the proxy re-encryption, can easily launch the chosen ciphertext attack. When the proxy re-encrypts the original ciphertext to be an encrypted ciphertext, it sends the incorrect re-encrypted ciphertexts to the data user, the data user decrypts the incorrect re-encrypted ciphertexts and will find they are incorrect. He will return the error information to the cloud, and thus the cloud can utilize this information to launch the chosen ciphertext attack.

1.2. Our Contribution

In this paper, we concentrate on how to achieve secure functional proxy re-encryption scheme for cloud data sharing. We first analysis the multi-hop identity based conditional proxy re-encryption scheme and the deterministic finite automata-based functional proxy re-encryption scheme proposed by Liang et al. The main advantage of these schemes is its very flexible delegation and thus more fine-grained delegation on the decryption of ciphertexts for other users. Actually we think these concepts are very promising and deserved further exploiting. Unfortunately, we find flaws in their protocols, which result their proposals can not be CCA-secure. We also give an improved AMH-IBCPRE scheme and an improved DFA-based PRE scheme which can resist the attack and be CCA-secure.

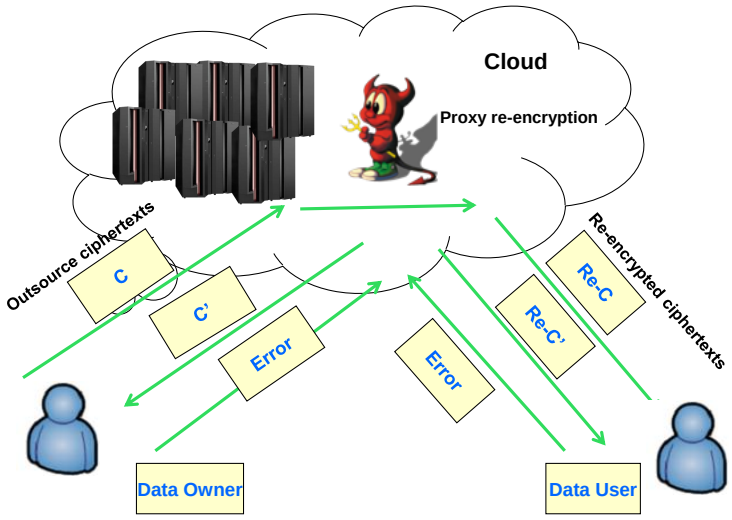

Fig. 1. Cloud storage and chosen ciphertext security

1.3. Organization

We organize our paper as the following. In section 2, we review Liang et al.'s AMH-IBCPRE scheme, then we demonstrate the attacks to show it can not achieve IND-CCA security in section 3. In section 4, we give an improved AMH-IBCPRE scheme and roughly evaluate its security and performance. In section 5, we review Liang et al.'s DFA-based PRE scheme, then we show the attacks to demonstrate their scheme can not achieve IND-CCA security in section 6. In section 7, we give an improved DFA-based PREscheme and roughly evaluate its security. We conclude our paper in the last section.

2. Review the AMH-IBCPRE Scheme

2.1. The Concrete Scheme

The definition and security model of AMH-IBCPRE can be found in [16], here we just review their concrete construction. The scheme consists of the Setup, Extract, Enc, ReKeyGen, ReEnc, Dec algorithms as the following:

1. Setup(1^k): A Target Collision Resistant (TCR) hash function $H_0 : \{0,1\}^* \rightarrow Z_q^*$ first be used to hash the conditions and identities which could be arbitrary length. Run $BSetup(1^k)$ to get $(q, g, \hat{g}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ with input parameter k. Denote $\omega \in \mathbb{Z}_q^*$ as a condition. $\alpha, \beta, \gamma, \delta_1, \delta_2, \delta_3, \eta \in_R \mathbb{Z}_q^*$ are randomly chosen, computes $g_1 = g^\alpha, g_2 = g^\beta, h = g^\gamma, f_1 = g^{\delta_1}, f_2 = g^{\delta_2}, f_3 = g^{\delta_3}, t = g^\eta, \hat{g}_1 = \hat{g}^\alpha, \hat{g}_2 = \hat{g}^\beta, \hat{h} = \hat{g}^\gamma, \hat{f}_1 = \hat{g}^{\delta_1}, \hat{f}_2 = \hat{g}^{\delta_2}, \hat{f}_3 = \hat{g}^{\delta_3}, \hat{t} = \hat{g}^\eta$. Denote $H_1 : \{0,1\}^k \rightarrow \mathbb{Z}_q^*$, $H_2 : \mathbb{G}_T \rightarrow \{0,1\}^{poly(1^k)}$ as two TCR hash functions, and $SYM = (SYM.Enc, SYM.Dec)$

as a CCA-secure one-time symmetric key encryption. Choose a sUF one-time signature scheme $(Sig.KG, Sign, Ver)$ and let its verification key K_v in $\mathbb{Z}_q^*$. The master public key is

$$mpk = (q, k, g, \hat{g}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, h, f_1, f_2, f_3, t, \hat{g_2}, \hat{f_2}, \hat{f_3}, \hat{h}, H_1, H_2, SYM, (Sig.KG, Sign, Ver))$$

and the master secret key is

$$msk = (\hat{g_0} = \hat{g}^{\alpha\beta}, \hat{f_1}, \hat{t})$$

.

2. Extract(msk, ID): Taken input an identity $ID \in \mathbb{Z}_q^*$, msk, $r, R \in_R \mathbb{Z}_q^*$, output

$$sk_{ID} = (sk_{ID_0}, sk_{ID_1}, sk_{ID_2}) = (\hat{g_0}(\hat{h}^{ID}\hat{f_1})^r \hat{t}^R, \hat{g}^r, \hat{g}^R)$$

The validity of the secret key can be checked as:

$$e(g, sk_{ID_0}) \stackrel{?}{=} e(g_1, \hat{g_2})e(h^{ID}f_1, sk_{ID_1})e(t, sk_{ID_2})$$

3. Enc(ID_i, ω, m): Chooses a one-time signature key pair

$$(K_s, K_v) \leftarrow Sig.KG(1^k)$$

and $s_0 \in_R \mathbb{Z}_q^*$ compute

$$C_0 = K_v, C_1 = e(g_1, \hat{g_2})^{s_0} \cdot m, C_2 = g^{s_0}, C_3 = (h^{ID_i}f_1)^{s_0}, C_4 = t^{s_0},$$
$$C_5 = (h^w f_2)^{s_0}, C_6 = (h^{K_v}f_3)^{s_0},$$
$$C_7 = Sign(K_s, (C_1, C_2, C_3, C_4, C_5, C_6))$$

output $C_{1,ID_i,w} = (C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7)$ as the 1-st level ciphertext, where the ciphertext implicitly includes w and $ID_i \in \mathbb{Z}_q^*, m \in \mathbb{G}_T$.

4. ReKeyGen$(ID_i, sk_{ID_i}, ID_{i'}, \omega)$: Chooses $(K_s^{(l)}, K_v^{(l)}) \leftarrow Sig.KG(1^k)$ as a one-time signature key pair and $\theta_1^{(l)} \in_R \mathbb{G}_T, \rho^{(l)}, s_1^{(l)}, \bar{r_1}^{(l)} \in \mathbb{Z}_q^*$, $rk_{\omega,ID_i \to ID_{i'}}$ can be computed as:

$$rk_0^{(l)} = (sk_{ID_{i_0}}(\hat{h}^w \hat{f_2})^{\rho^{(l)}})^{H_1(\theta_1^{(l)})}, rk_1^{(l)} = (\hat{g}^{\rho^{(l)}})^{H_1(\theta_1^{(l)})},$$
$$rk_2^{(l)} = (sk_{ID_{i_1}})^{H_1(\theta_1^{(l)})}, rk_3^{(l)} = (sk_{ID_{i_2}})^{H_1(\theta_1^{(l)})}, rk_4^{(l)} = e(g_1, \hat{g_2})^{s_1^{(l)}} \cdot \theta_1^{(l)},$$
$$rk_5^{(l)} = g^{s_1^{(l)}},$$
$$rk_6^{(l)} = (h^{ID_{i'}}f_1)^{s_1^{(l)}}, rk_7^{(l)} = t^{s_1^{(l)}}, rk_8^{(l)} = (h^w f_2)^{s_1^{(l)}}, rk_9^{(l)} = (h^{K_v^{(l)}}f_3)^{s_1^{(l)}},$$
$$rk_{10}^{(l)} = K_v^{(l)}$$
$$rk_{11}^{(l)} = Sign(K_s^{(l)}, (rk_4^{(l)}, rk_5^{(l)}, rk_6^{(l)}, rk_7^{(l)}, rk_8^{(l)}, rk_9^{(l)})),$$
$$rk_{12}^{(l)} = (h^{ID_{i'}}f_1)^{\bar{r_1}^{(l)}}, rk_{13}^{(l)} = g^{\bar{r_1}^{(l)}}, rk_{14}^{(l)} = t^{\bar{r_1}^{(l)}}, rk_{15}^{(l)} = h^{\bar{r_1}^{(l)}},$$
$$rk_{16}^{(l)} = e(g_1, \hat{g_2})^{\bar{r_1}^{(l)}},$$
$$rk_{17}^{(l)} = f_2^{\bar{r_1}^{(l)}}, rk_{18}^{(l)} = f_3^{\bar{r_1}^{(l)}},$$

where $ID_i, ID_{i'} \in \mathbb{Z}_q^*$ and $l \in \{1, \cdots, poly(1^k)\}$.

5. $\mathsf{ReEnc}(rk_{\omega,ID_i\to ID_{i'}}, C_{l,ID_i,w})$. If $l=1$,
   (a) Verify

$$e(\hat{h}^{K_v}\hat{f}_3, C_2) \stackrel{?}{=} e(\hat{g}, C_6), e(\hat{h}^{\omega}\hat{f}_2, C_2) \stackrel{?}{=} e(\hat{g}, C_5),$$
$$Ver(K_v, C_7, (C_1, C_2, C_3, C_4, C_5, C_6)) \stackrel{?}{=} 1 \quad (1)$$

   (b) Choose a one time signature key pair

$$(\bar{K}_s^{(1)}, \bar{K}_v^{(1)}) \leftarrow Sig.KG(1^k)$$

   and

$$\theta_2^{(1)} \in_R \mathbb{G}_T, s_2^{(1)} \in_R \mathbb{Z}_q^*$$

   compute

$$C_7^{(1)} = \frac{e(C_2, rk_0^{(l)})/e(C_5, rk_1^{(1)})}{e(C_3, rk_2^{(l)})e(C_4, rk_3^{(l)})},$$
$$\delta^{(1)} = SYM.Enc(C_0||C_1||\cdots||C_7||C_7^{(1)}, H_2(\theta_2^{(1)})),$$
$$C_8^{(1)} = rk_{16}^{s_2^{(1)}} \cdot \theta_2^{(1)}, C_9^{(1)} = rk_{13}^{(1)s_2^{(1)}}, C_{10}^{(1)} = rk_{12}^{(1)s_2^{(1)}},$$
$$C_{11}^{(1)} = rk_{14}^{(1)s_2^{(1)}}, C_{12}^{(1)} = (rk_{15}^{(1)\omega} rk_{17}^{(1)})^{s_2^1},$$
$$C_{13}^{(1)} = (rk_{15}^{(1)\bar{K}_v^{(1)}} rk_{17}^{(1)})^{s_2^1}, C_{14}^{(1)} = \bar{K}_v^{(1)},$$
$$C_{15}^{(1)} = Sign(\bar{K}_s^{(1)}, (C_8^{(1)}, C_9^{(1)}, C_{10}^{(1)}, C_{11}^{(1)}, C_{12}^{(1)}, C_{13}^{(1)}))$$

   and

$$C_{2,ID_{i'},w} = (\delta^{(1)}, C_8^{(1)}, C_9^{(1)}, C_{10}^{(1)}, C_{11}^{(1)}, C_{12}^{(1)}, C_{13}^{(1)}, C_{14}^{(1)}, C_{15}^{(1)}, rk_4^{(1)}, rk_5^{(1)}, rk_6^{(1)}, rk_7^{(1)}, rk_8^{(1)}, rk_9^{(1)}, rk_{10}^{(1)}, rk_{11}^{(1)})$$

   is outputted.

If $l \geq 2$,

(a) Verify

$$e(rk_5^{(l-1)}, \hat{h}^{\omega}\hat{f}_2) \stackrel{?}{=} e(rk_8^{(l-1)}, \hat{g}),$$
$$e(rk_5^{(l-1)}, \hat{h}^{K_v^{(l-1)}}\hat{f}_3) \stackrel{?}{=} e(rk_9^{(l-1)}, \hat{g}),$$
$$Ver(rk_{10}^{(l-1)}, rk_{11}^{(l-1)}, (rk_4^{(l-1)}, rk_5^{(l-1)}, rk_6^{(l-1)}, rk_7^{(l-1)}, rk_8^{(l-1)}, rk_9^{(l-1)})) \stackrel{?}{=} 1 \quad (2)$$
$$e(C_9^{(l-1)}, \hat{h}^{\omega}\hat{f}_2) \stackrel{?}{=} e(C_{12}^{(l-1)}, \hat{g}),$$
$$e(C_9^{(l-1)}, \hat{h}^{\bar{K}_v^{(l-1)}}\hat{f}_3) \stackrel{?}{=} e(C_{13}^{(l-1)}, \hat{g}),$$
$$Ver(C_{14}^{(l-1)}, C_{15}^{(l-1)}, (C_8^{(l-1)}, C_9^{(l-1)}, C_{10}^{(l-1)}, C_{11}^{(l-1)}, C_{12}^{(l-1)}, C_{13}^{(l-1)})) \stackrel{?}{=} 1 \quad (3)$$

   output $\perp$, if Eq. (2) and (3) do not hold. proceed otherwise.

(b) Choose a one time signature key pair

$$(\bar{K}_s^{(l)}, \bar{K}_v^{(l)}) \leftarrow Sig.KG(1^k)$$

and

$$\theta_2^{(l)} \in_R \mathbb{G}_T, s_2^{(l)} \in_R \mathbb{Z}_q^*$$

and then compute

$$C_{7,0}^{(l)} = \frac{e(rk_5^{(l-1)}, rk_0^{(l)})/e(rk_8^{(l-1)}, rk_1^{(l)})}{e(rk_6^{(l-1)}, rk_2^{(l)})e(rk_7^{(l-1)}, rk_3^{(l)})},$$

$$C_{7,1}^{(l)} = \frac{e(C_9^{(l-1)}, rk_0^{(l)})/e(C_{12}^{(l-1)}, rk_1^{(l)})}{e(C_{10}^{(l-1)}, rk_2^{(l)})e(C_{11}^{(l-1)}, rk_3^{(l)})}$$

$$\delta^{(l)} = SYM.Enc(\delta^{(l-1)}||C_8^{(l-1)}|| \cdots ||C_{15}^{(l-1)}||rk_4^{(l-1)}|| \cdots$$
$$||rk_{11}^{(l-1)}||C_{7,0}^{(l-1)}||C_{7,1}^{(l-1)}, H_2(\theta_2^{(l)}))$$

$$C_8^{(l)} = rk_{16}^{s_2^{(l)}} \cdot \theta_2^{(l)}, C_9^{(l)} = rk_{13}^{(l)s_2^{(l)}}, C_{10}^{(l)} = rk_{12}^{(l)s_2^{(l)}}, C_{11}^{(l)} = rk_{14}^{(l)s_2^{(l)}},$$

$$C_{12}^{(l)} = (rk_{15}^{(l)\omega} rk_{17}^{(l)})^{s_2^l},$$

$$C_{13}^{(l)} = (rk_{15}^{(l)\bar{K}_v^{(l)}} rk_{17}^{(l)})^{s_2^l}, C_{14}^{(l)} = \bar{K}_v^{(l)},$$

$$C_{15}^{(l)} = Sign(\bar{K}_s^{(l)}, (C_8^{(l)}, C_9^{(l)}, C_{10}^{(l)}, C_{11}^{(l)}, C_{12}^{(l)}, C_{13}^{(l)}))$$

and

$$C_{l,ID_{i'},\omega} = (\delta^{(l)}, C_8^{(l)}, C_9^{(l)}, C_{10}^{(l)}, C_{11}^{(l)}, C_{12}^{(l)}, C_{13}^{(l)}, C_{14}^{(l)}, C_{15}^{(l)}, rk_4^{(l)}, rk_5^{(l)},$$
$$rk_6^{(l)}, rk_7^{(l)}, rk_8^{(l)}, rk_9^{(l)}, rk_{10}^{(l)}, rk_{11}^{(l)})$$

is outputted.

6. $\mathsf{Dec}(sk_{ID_i}, C_{l,ID_i,w})$.
   - If $l = 1$,
     (a) Check Eq. (1) whether holds or not. Output $\perp$ if Eq. (1) does not hold. Proceed otherwise.
     (b) Compute

$$C_1 / \frac{e(C_2, sk_{ID_0})}{e(C_3, sk_{ID_1})e(C_4, sk_{ID_2})}$$
$$= e(g_1, \hat{g}_2)^{s_0} \cdot m / \frac{e(g_{s_0}, \hat{g}_0(\hat{h}^{ID_i}\hat{f})^r \hat{t}^R)}{e((h^{ID_i}f)^{s_0}, \hat{g}^r)e(t^{s_0}, \hat{g}^R)}$$
$$= e(g_1, \hat{g}_2)^{s_0} \cdot m / e(g_1, \hat{g}_2)^{s_0} = m$$

   - If $l \geq 2$,
     (a) Verify

$$e(rk_5^{(l)}, \hat{h}^\omega \hat{f}_2) \stackrel{?}{=} e(rk_8^{(l)}, \hat{g}), e(rk_5^{(l)}, \hat{h}^{K_v^{(l)}} \hat{f}_3) \stackrel{?}{=} e(rk_9^{(l)}, \hat{g}),$$
$$Ver(rk_{10}^{(l)}, rk_{11}^{(l)}, (rk_4^{(l)}, rk_5^{(l)}, rk_6^{(l)}, rk_7^{(l)}, rk_8^{(l)}, rk_9^{(l)})) \stackrel{?}{=} 1 \quad (4)$$
$$e(C_9^{(l)}, \hat{h}^\omega \hat{f}_2) \stackrel{?}{=} e(C_{12}^{(l)}, \hat{g}), e(C_9^{(l)}, \hat{h}^{\bar{K}_v^{(l)}} \hat{f}_3) \stackrel{?}{=} e(C_{13}^{(l)}, \hat{g}),$$
$$Ver(C_{14}^{(l)}, C_{15}^{(l)}, (C_8^{(l)}, C_9^{(l)}, C_{10}^{(l)}, C_{11}^{(l)}, C_{12}^{(l)}, C_{13}^{(l)})) \stackrel{?}{=} 1 \quad (5)$$

Output $\perp$ if Eq. (4) and (5) do not hold. Proceed otherwise.

(b) Compute

$$\begin{aligned}&\frac{e(rk_5^{(l)}, sk_{ID_{i'_0}})}{e(rk_6^{(l)}, sk_{ID_{i'_1}})e(rk_7^{(l)}, sk_{ID_{i'_2}})}\\&= \frac{e(g^{s_1^{(l)}}, \hat{g}_0(\hat{h}^{ID_{i'}}\hat{f}_1)^r\hat{t}^R}{e((h^{ID_{i'}}f_1)^{s_1^{(l)}}, \hat{g}^{(r)})e(t^{s_1^{(l)}}, \hat{g}^R)}\\&= e(g_1, \hat{g}_2)^{s_1^{(l)}}\end{aligned}$$

and

$$\begin{aligned}&\frac{e(C_9^{(l)}, sk_{ID_{i'_0}})}{e(C_{10}^{(l)}, sk_{ID_{i'_1}})e(C_{11}^{(l)}, sk_{ID_{i'_2}})}\\&= \frac{e(g^{\bar{s}_2^{(l)}}, \hat{g}_0(\hat{h}^{ID_{i'}}\hat{f}_1)^r\hat{t}^R}{e((h^{ID_{i'}}f_1)^{\bar{s}_2^{(l)}}, \hat{g}^{(r)})e(t^{\bar{s}_2^{(l)}}, \hat{g}^R)}\\&= e(g_1, \hat{g}_2)^{\bar{s}_2^{(l)}}\end{aligned}$$

where $\bar{s}_2^{(l)} = s_2^{(l)} \cdot \bar{r}_1^{(l)}$

(c) Compute $\theta_1^{(l)} = rk_4^{(l)}/e(g_1, \hat{g}_2)^{s_1^{(l)}}$, and $\theta_2^{(l)} = C_8^{(l)}/e(g_1, \hat{g}_2)^{\bar{s}_2^{(l)}}$. Recover

$$\begin{aligned}&(\delta^{(l-1)}||C_8^{(l-1)}||\cdots||C_{15}^{(l-1)}||rk_4^{(l-1)}||\cdots||rk_{11}^{(l-1)}||C_{7,0}^{(l-1)}||C_{7,1}^{(l-1)})\\&= SYM.Dec(\delta^{(l)}, H_2(\theta_2^{(l)})\end{aligned}$$

(d) Compute

$$\begin{aligned}&C_{7,0}^{(l-1)(H_1(\theta_1^{(l)}))^{-1}}\\&= (e(g_1, \hat{g}_2)^{s_1^{(l-1)}})^{(H_1(\theta_1^{(l)}))(H_1(\theta_1^{(l)}))^{-1}}\\&= e(g_1, \hat{g}_2)^{s_1^{(l-1)}}\end{aligned}$$

if Eq. (2) holds, compute $\theta_1^{(l-1)} = rk_4^{(l-1)}/e(g_1, \hat{g}_2)^{s_1^{(l-1)}}$.
Compute

$$\begin{aligned}&C_{7,1}^{(l-1)(H_1(\theta_1^{(l)}))^{-1}}\\&= (e(g_1, \hat{g}_2)^{\bar{s}_2^{(l-1)}})^{(H_1(\theta_1^{(l)}))(H_1(\theta_1^{(l)}))^{-1}}\\&= e(g_1, \hat{g}_2)^{\bar{s}_2^{(l-1)}}\end{aligned}$$

if Eq. (3) holds, compute $\theta_2^{(l-1)} = C_8^{(l-1)}/e(g_1, \hat{g}_2)^{\bar{s}_2^{(l-1)}}$.

(e) As in the previous steps, compute $\theta_1^{(j)}$ and $\theta_2^{(j)}$ for $1 \leq j \leq l-2$, from $l-2$ to 1.

(f) Recover $C_0||C_1||\cdots||C_7||C_7^{(1)} = SYM.Dec(\delta^{(1)}, H_2(\theta_{(2)}^{(1)}))$
If Eq. (1) holds, compute

$$\begin{aligned} & C_1/C_7^{(1)(H_1(\theta_1^{(1)}))^{-1}} \\ &= e(g_1, \hat{g_2})^{s_0} \cdot m/(e(g_1, \hat{g_2})^{s_0 H_1(\theta_1^{(1)})(H_1(\theta_1^{(1)}))^{-1}} \\ &= m \end{aligned}$$

3. Cryptanalysis of the AMH-IBCPRE Scheme

3.1. On the IND-sCon-sID-CCA Property

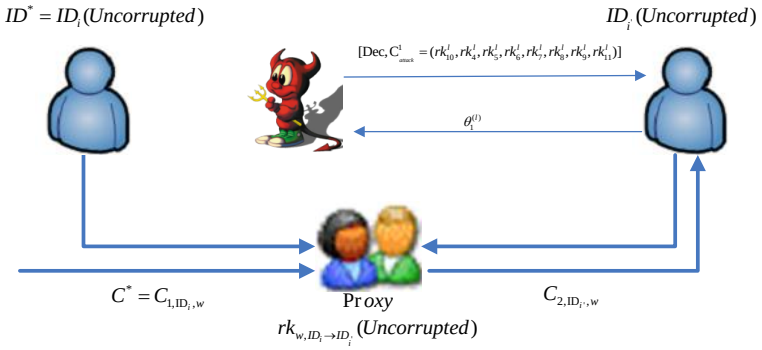

Fig. 2. Attack by querying the decryption oracle

Attack I Here we give the first attack on the IND-sCon-sID-CCA property, which can be seen in Figure 2. Note in this attack the delegator, delegatee and proxy are all uncorrupted. The main strategy is the following: the adversary just modifies the re-encryption key to be a valid ciphertext C^1_{attack} (which can be assumed at l level), then he queries this ciphertext to the decryption oracle (which can be assumed at 1 level) to get secret value $\theta_1^{(l)}$. Note here C^1_{attack} is not a derivative of C^*. By using this value, he can derive the delegator's secret key and thus can easily find the plaintext corresponding to C^*, which breaks the IND-sCon-sID-CCA property. Concretely, the attack can be described as following:

1. Assume the challenge identity and condition are (ID_i^*, w^*) and the challenge first level ciphertext is $C_{1,ID_i^*,w^*} = (C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7)$, where $ID_i^* \in \mathbb{Z}_q^*, m_b \in \mathbb{G}_T$ and w^* is implicitly included in the ciphertext, concretely

$$\begin{aligned} & C_0 = K_v, C_1 = e(g_1, \hat{g_2})^{s_0} \cdot m_b, C_2 = g^{s_0}, C_3 = (h^{ID_i^*} f_1)^{s_0}, C_4 = t^{s_0}, \\ & C_5 = (h^{w^*} f_2)^{s_0}, C_6 = (h^{K_v} f_3)^{s_0}, \\ & C_7 = Sign(K_s, (C_1, C_2, C_3, C_4, C_5, C_6)) \end{aligned}$$

2. The adversary first queries the re-encryption key generation oracle with input

$$(ID_i^*, ID_{i'}, w^*)$$

for the l hop, and he will get the re-encryption key as following:

$$\begin{aligned}
&rk_0^{(l)} = (sk_{ID_{i_0}^*}(\hat{h}^{w^*}\hat{f}_2)^{\rho^{(l)}})^{H_1(\theta_1^{(l)})}, rk_1^{(l)} = (\hat{g}^{\rho^{(l)}})^{H_1(\theta_1^{(l)})},\\
&rk_2^{(l)} = (sk_{ID_{i_1}^*})^{H_1(\theta_1^{(l)})}, rk_3^{(l)} = (sk^*_{ID_{i_2}})^{H_1(\theta_1^{(l)})},\\
&rk_4^{(l)} = e(g_1, \hat{g_2})^{s_1^{(l)}} \cdot \theta_1^{(l)}, rk_5^{(l)} = g^{s_1^{(l)}},\\
&rk_6^{(l)} = (h^{ID_{i'}} f_1)^{s_1^{(l)}}, rk_7^{(l)} = t^{s_1^{(l)}}, rk_8^{(l)} = (h^{w^*} f_2)^{s_1^{(l)}},\\
&rk_9^{(l)} = (h^{K_v^{(l)}} f_3)^{s_1^{(l)}}, rk_{10}^{(l)} = K_v^{(l)}\\
&rk_{11}^{(l)} = Sign(K_s^{(l)}, (rk_4^{(l)}, rk_5^{(l)}, rk_6^{(l)}, rk_7^{(l)}, rk_8^{(l)}, rk_9^{(l)})),\\
&rk_{12}^{(l)} = (h^{ID_{i'}} f_1)^{\bar{r_1}^{(l)}}, rk_{13}^{(l)} = g^{\bar{r_1}^{(l)}}, rk_{14}^{(l)} = t^{\bar{r_1}^{(l)}},\\
&rk_{15}^{(l)} = h^{\bar{r_1}^{(l)}}, rk_{16}^{(l)} = e(g_1, \hat{g_2})^{\bar{r_1}^{(l)}},\\
&rk_{17}^{(l)} = f_2^{\bar{r_1}^{(l)}}, rk_{18}^{(l)} = f_3^{\bar{r_1}^{(l)}},
\end{aligned}$$

3. Now the adversary can extract a valid ciphertext from this re-encryption key, which is

$$\begin{aligned}
&rk_4^{(l)} = e(g_1, \hat{g_2})^{s_1^{(l)}} \cdot \theta_1^{(l)}, rk_5^{(l)} = g^{s_1^{(l)}},\\
&rk_6^{(l)} = (h^{ID_{i'}} f_1)^{s_1^{(l)}}, rk_7^{(l)} = t^{s_1^{(l)}}, rk_8^{(l)} = (h^{w*} f_2)^{s_1^{(l)}},\\
&rk_9^{(l)} = (h^{K_v^{(l)}} f_3)^{s_1^{(l)}}, rk_{10}^{(l)} = K_v^{(l)}\\
&rk_{11}^{(l)} = Sign(K_s^{(l)}, (rk_4^{(l)}, rk_5^{(l)}, rk_6^{(l)}, rk_7^{(l)}, rk_8^{(l)}, rk_9^{(l)}))
\end{aligned}$$

This ciphertext has exactly the same structure as the first level ciphertext!

4. The adversary query this ciphertext to the first level ciphertext decryption oracle of $ID_{i'}$, which will return $\theta_1^{(l)}$. Note here the adversary does not need to corrupt the delegatee $ID_{i'}$, he just use his decryption oracle!
5. After obtaining $\theta_1^{(l)}$, the adversary can easily compute the partial private key of ID_i^*, which is enough to decrypt the challenge first level ciphertext:

$$\begin{aligned}
&\clubsuit = (sk_{ID_{i_0}^*}(\hat{h}^{w^*}\hat{f}_2)^{\rho^{(l)}}), \diamondsuit = (\hat{g}^{\rho^{(l)}}),\\
&\heartsuit = (sk_{ID_{i_1}^*}), \spadesuit = (sk^*_{ID_{i_2}})
\end{aligned}$$

6. He decrypt the challenge ciphertext as following:

$$\begin{aligned}
&C_1 / \frac{e(C_2, \clubsuit)}{e(C_3, \heartsuit)e(C_4, \spadesuit)e(C_5, \diamondsuit)}\\
&= e(g_1, \hat{g_2})^{s_0} \cdot m / \frac{e(g_{s_0}, \hat{g_0}(\hat{h}^{ID_i}\hat{f})^r \hat{t}^R (\hat{h}^{w^*}\hat{f}_2)^{\rho^{(l)}})}{e((h^{ID_i} f)^{s_0}, \hat{g}^r)e(t^{s_0}, \hat{g}^R)e((\hat{g}^{\rho^{(l)}}), (h^{w^*} f_2)^{s_0})}\\
&= e(g_1, \hat{g_2})^{s_0} \cdot m / e(g_1, \hat{g_2})^{s_0}\\
&= m
\end{aligned}$$

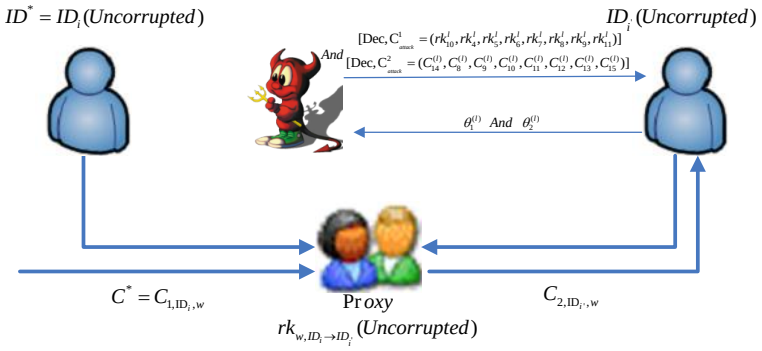

Fig. 3. Another attack by querying the decryption oracle

Attack II Here we give the second attack on the IND-sCon-sID-CCA property, which can be seen in Figure 2. Note in this attack the delegator, delegatee and proxy are all uncorrupted. The main strategy is similar as the first attack except the adversary modifies the challenge ciphertext to two valid ciphertexts. Concretely, the adversary first just modifies the re-encrypted challenge ciphertext C^* (which can be assumed at $l+1$ level) to be another ciphertext C^1_{attack} and C^2_{attack} which are valid original ciphertexts (which can be assumed at l level), then he queries this ciphertext to the decryption oracle (which can be assumed at l level) to get secret value $\theta_1^{(l)}$ and $\theta_2^{(l)}$. By using these value, he can derive the delegator's secret key and thus can easy find the plaintext corresponding to C^*, which breaks the IND-sCon-sID-CCA property. For this attack is similar with Attack I, here we omit the details.

On the IND-sCon-sID-CCA Security Model Here we give some comments on Liang et al.'s IND-sCon-sID-CCA security model. Here we first review the definition of uncorrupted delegation chain and corrupted delegation chain.

> *"Uncorrupted Delegation Chain. Suppose there is a delegation chain under w from ID_i to ID_j (i.e. $w|ID_i \to \cdots \to ID_j$). If there is no corrupted identity in the chain, it is an uncorrupted delegation chain, else it is corrupted."*

Then we review the part about uncorrupted delegation chain in the IND-sCon-sID-CCA security model

> *"In this phase, the followings are forbidden to issue:*
> 1. *$\mathcal{O}_{sk}(ID)$ for any ID, if there is an uncorrupted delegation chain under w^* from ID^* to ID, or $ID^* = ID$.*
> 2. *$\mathcal{O}_{rk}(ID_i, ID_{i'}, w^*)$ for any $ID_i, ID_{i'}$, if there is uncorrupted delegation chain under w^* from ID^* to ID_i, or $ID^* = ID_i$, but $ID_{i'}$ is in a corrupted delegation chain."*

We think this security model is too strong. In this model, the challenge delegator can only lie in the uncorrupted chain. In the multi-hop setting, this means there can be no corrupted

delegatees after the challenge delegator even these delegatees are far away from the delegator, which is obviously contradict our intuition in the multi-hop proxy re-encryption settings.

3.2. On the ANO-OC Property and the ANO-RK Property

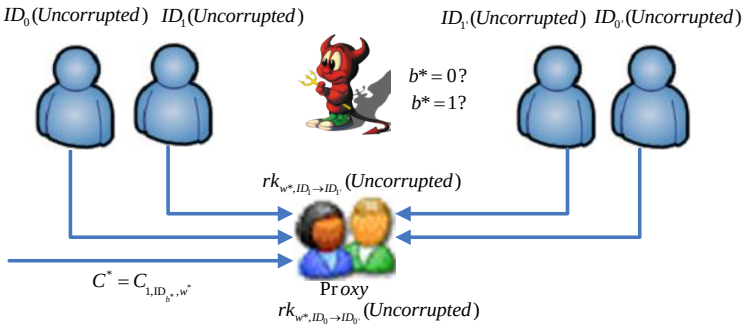

Fig. 4. On the ANO-OC property

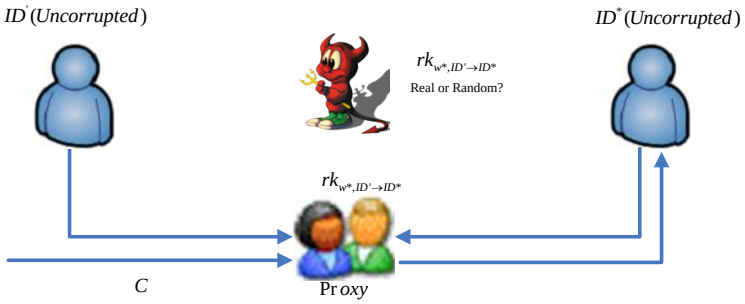

Fig. 5. On the ANO-RK property

The above two figures describe the ANO-OC property and ANO-RK property. It is easy to see, if the IND-sCon-sID-CCA security does not hold for the scheme, the ANO-OC property and ANO-RK property can not hold for this scheme either.

4. Our improved AMH-IBCPRE scheme

4.1. Concrete Construction

In this section, we try to give a new improved scheme which can resist our attack. The idea is the following: let the original ciphertext, part of the re-encrypted ciphertext and

part of the re-encryption key have different forms, then our attack can no longer work again. In particular, we just modify the signature structure in the original ciphertext, the re-encrypted ciphertext and part of the re-encryption key based on Liang et al.'s scheme [16]. Concretely, we let $C_7 = Sign(K_s, (C_0, C_1, C_2, C_3, C_4, C_5, C_6))$ instead of $C_7 = Sign(K_s, (C_1, C_2, C_3, C_4, C_5, C_6))$, $C_{15}^{(1)} = Sign(\bar{K}_s^{(1)}, (C_8^{(1)}, C_9^{(1)}, C_{10}^{(1)}, C_{11}^{(1)}, C_{12}^{(1)}, C_{13}^{(1)}, C_{14}^{(1)} = \bar{K}_v^{(l)}))$ instead of $C_{15}^{(1)} = Sign(\bar{K}_s^{(1)}, (C_8^{(1)}, C_9^{(1)}, C_{10}^{(1)}, C_{11}^{(1)}, C_{12}^{(1)}, C_{13}^{(1)}))$. In this way, the original ciphertext, part of the re-encrypted ciphertext and part of the re-encryption key shall have different forms and can not be used by the adversary improperly. Concretely, our improved scheme is the following:

1. Setup(1^k): The same as the original scheme [16].
2. Extract(msk, ID): The same as the original scheme [16].
3. Enc(ID_i, ω, m): Chooses $s_0 \in_R \mathbb{Z}_q^*$ and a one-time signature key pair

$$(K_s, K_v) \leftarrow Sig.KG(1^k)$$

compute

$$C_0 = K_v, C_1 = e(g_1, \hat{g}_2)^{s_0} \cdot m, C_2 = g^{s_0}, C_3 = (h^{ID_i} f_1)^{s_0}, C_4 = t^{s_0},$$
$$C_5 = (h^w f_2)^{s_0}, C_6 = (h^{K_v} f_3)^{s_0},$$
$$\boxed{C_7 = Sign(K_s, (C_0, C_1, C_2, C_3, C_4, C_5, C_6))}$$

and output the 1-st level ciphertext

$$C_{1,ID_i,w} = (C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7)$$

where $ID_i \in \mathbb{Z}_q^*, m \in \mathbb{G}_T$ and w is implicitly included in the ciphertext.
4. ReKeyGen($ID_i, sk_{ID_i}, ID_{i'}, \omega$): The same as the original scheme.
5. ReEnc($rk_{\omega, ID_i \to ID_{i'}}, C_{l,ID_i,w}$).
   – If $l = 1$,
   (a) Verify

$$e(\hat{h}^{K_v} \hat{f}_3, C_2) \stackrel{?}{=} e(\hat{g}, C_6),$$
$$e(\hat{h}^{\omega} \hat{f}_2, C_2) \stackrel{?}{=} e(\hat{g}, C_5), \tag{6}$$
$$\boxed{Ver(K_v, C_7, (C_0, C_1, C_2, C_3, C_4, C_5, C_6)) \stackrel{?}{=} 1}$$

   (b) Choose $\theta_2^{(1)} \in_R \mathbb{G}_T, s_2^{(1)} \in_R \mathbb{Z}_q^*$ and a one time signature key pair

$$(\bar{K}_s^{(1)}, \bar{K}_v^{(1)}) \leftarrow Sig.KG(1^k)$$

compute

$$C_7^{(1)} = \frac{e(C_2, rk_0^{(l)})/e(C_5, rk_1^{(1)})}{e(C_3, rk_2^{(l)})e(C_4, rk_3^{(l)})},$$
$$\delta^{(1)} = SYM.Enc(C_0||C_1||\cdots||C_7||C_7^{(1)}, H_2(\theta_2^{(1)})),$$
$$C_8^{(1)} = rk_{16}^{s_2^{(1)}} \cdot \theta_2^{(1)}, C_9^{(1)} = rk_{13}^{(1)s_2^{(1)}}, C_{10}^{(1)} = rk_{12}^{(1)s_2^{(1)}}, C_{11}^{(1)} = rk_{14}^{(1)s_2^{(1)}},$$
$$C_{12}^{(1)} = (rk_{15}^{(1)\omega} rk_{17}^{(1)})^{s_2^1},$$
$$C_{13}^{(1)} = (rk_{15}^{(1)\bar{K}_v^{(1)}} rk_{17}^{(1)})^{s_2^1}, C_{14}^{(1)} = \bar{K}_v^{(1)},$$
$$\boxed{C_{15}^{(1)} = Sign(\bar{K}_s^{(1)}, (C_8^{(1)}, C_9^{(1)}, C_{10}^{(1)}, C_{11}^{(1)}, C_{12}^{(1)}, C_{13}^{(1)}, C_{14}^{(1)}))}$$

Output $C_{2,ID_{i'},w} = (\delta^{(1)}, C_8^{(1)}, C_9^{(1)}, C_{10}^{(1)}, C_{11}^{(1)}, C_{12}^{(1)}, C_{13}^{(1)}, C_{14}^{(1)}, C_{15}^{(1)}, rk_4^{(1)}, rk_5^{(1)}, rk_6^{(1)}, rk_7^{(1)}, rk_8^{(1)}, rk_9^{(1)}, rk_{10}^{(1)}, rk_{11}^{(1)})$.

– If $l \geq 2$,

(a) Verify

$$\begin{aligned}
&e(rk_5^{(l-1)}, \hat{h}^{\omega} \hat{f}_2) \stackrel{?}{=} e(rk_8^{(l-1)}, \hat{g}),\\
&e(rk_5^{(l-1)}, \hat{h}^{K_v^{(l-1)}} \hat{f}_3) \stackrel{?}{=} e(rk_9^{(l-1)}, \hat{g})\\
&Ver(rk_{10}^{(l-1)}, rk_{11}^{(l-1)}, (rk_4^{(l-1)}, rk_5^{(l-1)}, rk_6^{(l-1)},\\
&rk_7^{(l-1)}, rk_8^{(l-1)}, rk_9^{(l-1)})) \stackrel{?}{=} 1\\
&e(C_9^{(l-1)}, \hat{h}^{\omega} \hat{f}_2) \stackrel{?}{=} e(C_{12}^{(l-1)}, \hat{g}),\\
&e(C_9^{(l-1)}, \hat{h}^{\bar{K}_v^{(l-1)}} \hat{f}_3) \stackrel{?}{=} e(C_{13}^{(l-1)}, \hat{g})\\
&\boxed{Ver(C_{14}^{(l-1)}, C_{15}^{(l-1)}, (C_8^{(l-1)}, C_9^{(l-1)}, \cdots, C_{13}^{(l-1)}, C_{14}^{(l-1)})) \stackrel{?}{=} 1}
\end{aligned} \tag{7}$$

If Eq. (7) do not hold, output $\perp$. Otherwise, proceed.

(b) Choose $\theta_2^{(l)} \in_R \mathbb{G}_T, s_2^{(l)} \in_R \mathbb{Z}_q^*$ and a one time signature key pair

$$(\bar{K}_s^{(l)}, \bar{K}_v^{(l)}) \leftarrow Sig.KG(1^k)$$

and then compute

$$C_{7,0}^{(l)} = \frac{e(rk_5^{(l-1)}, rk_0^{(l)})/e(rk_8^{(l-1)}, rk_1^{(l)})}{e(rk_6^{(l-1)}, rk_2^{(l)})e(rk_7^{(l-1)}, rk_3^{(l)})},$$

$$C_{7,1}^{(l)} = \frac{e(C_9^{(l-1)}, rk_0^{(l)})/e(C_{12}^{(l-1)}, rk_1^{(l)})}{e(C_{10}^{(l-1)}, rk_2^{(l)})e(C_{11}^{(l-1)}, rk_3^{(l)})}$$

$$\delta^{(l)} = SYM.Enc(\delta^{(l-1)}||C_8^{(l-1)}||\cdots||C_{15}^{(l-1)}||rk_4^{(l-1)}||\cdots||rk_{11}^{(l-1)}||$$
$$C_{7,0}^{(l-1)}||C_{7,1}^{(l-1)}, H_2(\theta_2^{(l)}))$$

$$C_8^{(l)} = rk_{16}^{s_2^{(l)}} \cdot \theta_2^{(l)}, C_9^{(l)} = rk_{13}^{(l)s_2^{(l)}}, C_{10}^{(l)} = rk_{12}^{(l)s_2^{(l)}}, C_{11}^{(l)} = rk_{14}^{(l)s_2^{(l)}},$$

$$C_{12}^{(l)} = (rk_{15}^{(l)\omega} rk_{17}^{(l)})^{s_2^l},$$

$$C_{13}^{(l)} = (rk_{15}^{(l)\bar{K}_v^{(l)}} rk_{17}^{(l)})^{s_2^l}, C_{14}^{(l)} = \bar{K}_v^{(l)},$$

$$\boxed{C_{15}^{(l)} = Sign(\bar{K}_s^{(l)}, (C_8^{(l)}, C_9^{(l)}, C_{10}^{(l)}, C_{11}^{(l)}, C_{12}^{(l)}, C_{13}^{(l)}, C_{14}^{(l)}))}$$

Output

$$C_{l,ID_{i'},\omega} = (\delta^{(l)}, C_8^{(l)}, C_9^{(l)}, C_{10}^{(l)}, C_{11}^{(l)}, C_{12}^{(l)}, C_{13}^{(l)}, C_{14}^{(l)}, C_{15}^{(l)}, rk_4^{(l)}, rk_5^{(l)}, rk_6^{(l)}, rk_7^{(l)}, rk_8^{(l)}, rk_9^{(l)}, rk_{10}^{(l)}, rk_{11}^{(l)})$$

6. $\mathsf{Dec}(sk_{ID_i}, C_{l,ID_i,w})$.
   - If $l = 1$,
     (a) Verify Eq. (6). If Eq. (6) does not hold, output $\perp$, otherwise, proceed.
     (b) Compute

$$\begin{aligned} & C_1 / \frac{e(C_2, sk_{ID_0})}{e(C_3, sk_{ID_1})e(C_4, sk_{ID_2})} \\ &= e(g_1, \hat{g_2})^{s_0} \cdot m / \frac{e(g_{s_0}, \hat{g_0}(\hat{h}^{ID_i}\hat{f})^r \hat{t}^R)}{e((h^{ID_i} f)^{s_0}, \hat{g}^r)e(t^{s_0}, \hat{g}^R)} \\ &= e(g_1, \hat{g_2})^{s_0} \cdot m / e(g_1, \hat{g_2})^{s_0} \\ &= m \end{aligned}$$

   - If $l \geq 2$,
     (a) Verify

$$e(rk_5^{(l)}, \hat{h}^\omega \hat{f}_2) \stackrel{?}{=} e(rk_8^{(l)}, \hat{g}), e(rk_5^{(l)}, \hat{h}^{K_v^{(l)}} \hat{f}_3) \stackrel{?}{=} e(rk_9^{(l)}, \hat{g}),$$
$$Ver(rk_{10}^{(l)}, rk_{11}^{(l)}, (rk_4^{(l)}, rk_5^{(l)}, rk_6^{(l)}, rk_7^{(l)}, rk_8^{(l)}, rk_9^{(l)})) \stackrel{?}{=} 1 \quad (8)$$
$$e(C_9^{(l)}, \hat{h}^\omega \hat{f}_2) \stackrel{?}{=} e(C_{12}^{(l)}, \hat{g}), e(C_9^{(l)}, \hat{h}^{\bar{K}_v^{(l)}} \hat{f}_3) \stackrel{?}{=} e(C_{13}^{(l)}, \hat{g}),$$
$$\boxed{Ver(C_{14}^{(l)}, C_{15}^{(l)}, (C_8^{(l)}, C_9^{(l)}, C_{10}^{(l)}, C_{11}^{(l)}, C_{12}^{(l)}, C_{13}^{(l)}, C_{14}^{(l)})) \stackrel{?}{=} 1} \quad (9)$$

If Eq. (8) and (9) do not hold, output $\perp$. Otherwise, proceed.

(b) The other steps are the same as the original scheme [16] except the verification equations changes to be the above corresponding equations .

For our scheme is almost the same as Liang et al.'s scheme, except this scheme can resist the attack we have proposed, we omit the security analysis of it which can be followed Liang et al's security proof, thus we get the following theorem:

Theorem 1. *Our improved AMH-IBCPRE scheme is IND-sCon-sID-CCA secure assuming the decisional* $P-BDH$ *assumption*[3] *holds,* $(Sig.KG, Sign, Ver)$ *is a sUF one-time signature scheme, SYM is a CCA-secure one-time symmetric key encryption, and* H_1, H_2 *are TCR hash functions.*

Proof. Here we just give the roughly security proof. For we just modify Liang et al.'s scheme on the Encrypt algorithm used in the re-encryption key generation process, and the corresponding Decrypt algorithm used in the decryption process for the first level ciphertexts. And other algorithms are the same as the original scheme, thus the security proof is almost the same as Liang et al's proof.

But there is a crucial difference, in our scheme

$$\boxed{C_7 = Sign(K_s, (C_0, C_1, C_2, C_3, C_4, C_5, C_6))}$$

which is not the same structure as part of the re-encryption key

$$\boxed{rk_{11}^{(l)} = Sign(K_s^{(l)}, (rk_4^{(l)}, rk_5^{(l)}, rk_6^{(l)}, rk_7^{(l)}, rk_8^{(l)}, rk_9^{(l)}))}$$

Note here $C_0 = K_v$, and the part of the re-encryption key can be successfully decrypted by querying to the decryption oracle only if the adversary successfully forge a signature

$$ForgedSignature = Sign(K_s^{(l)}, (rk_1 0^{(l)}, rk_4^{(l)}, rk_5^{(l)}, rk_6^{(l)}, rk_7^{(l)}, rk_8^{(l)}, rk_9^{(l)}))$$

which is not possible, due to the security of one-time signature and the tight connection between $rk_9^{(l)}$ and $K_v^{(l)}$ as following:

$$rk_9^{(l)} = (h^{K_v^{(l)}} f_3)^{s_1^{(l)}}$$

Table 1. Feature Comparison

Scheme	CCA-security	W.R.O.	M.U.	C.R.	Conditional Share	Anonymity
[16]	No	Yes	Yes	Yes	Yes	Yes
Our	Yes	Yes	Yes	Yes	Yes	Yes

[3] Please refer [16] for this assumption.

Table 2. Efficiency Comparison

Scheme	Enc	Check	Reenc	Dec		Ciph-Len	
				1stCiph	2ndCiph	1stCiph	2ndCiph
GA07B [9]	$1t_p + 1t_e$	$2t_p$	$2t_e + 2t_p$	$1t_e + 2t_p$	$2t_e + 2t_p$	$1\|G\| + 1\|G_T\|$ $+2\|m\| + \|id\|$	$2\|G\| + 1\|G_T\|$ $+\|m\|$
LZD$^+$10 [13]	$5t_e$	$6t_p$	$6t_e$	$24t_p$	$8t_p$	$13\|G\| + 1\|G_T\|$	$4\|G\| + 1\|G_T\|$
WCW10 [25]	$5t_e$	$4t_p$	$2t_p$	$1t_e + 2t_p$	$1t_e + 4t_p$	$2\|G\| + 1\|G_T\|$ $+\|m\| + \|id\|$	$4\|G\| + 1\|G_T\|$ $\|id\| + \|m\|$
Ours 4.1	$t_e^{''} + 2t_e$ $+3t'_{me} + 1t_s$	$4t'_p + 1t_v$ $+2t'_e$	$4t'_p + 3t_e$ $+3t'_{me} + 1t_s$	$3t'_p$	$6t'_p + 4t_e^{''}$ $+1t_{sd}$	$6\|G_1\| + 1\|s\|$ $+1\|SE\| + \|vk\|$	$5\|G_1\| + 1\|G_T\|$ $+1\|SE\|$

4.2. Comparison

First we compare our scheme's feature with Liang et al.'s in Table 1. In this table multiple ciphertext receiver update denotes as M.U., con- ditional (data) share denotes as C.S., collusion resistance denotes as C.R., anonymity, and without random oracle denotes as W.R.O.. From this table, we can see our scheme and Liang et al's scheme simultaneously have many good properties, but Liang et al.'s scheme is not CCA-secure, which is important for cloud storage.

Then we compare our scheme's efficiency with other related scheme's, the analysis results can be seen in Table 2. Note here we do not compare our improved scheme with Liang et al.'s scheme for the two schemes almost share the same efficiency, but their scheme can not be CCA-secure while our scheme can. To be fair, we just compare one-single hop of our scheme with the GA07B scheme (the second scheme in GA07) [9], the LZD$^+$10 scheme [13] and the WCW10 [25] scheme, which are all CCA-secure identity based proxy re-encryption schemes. Note in the performance evaluation analysis below, we just compare the length of the ciphertext overhead, not including the plaintext length in the total ciphertext length.

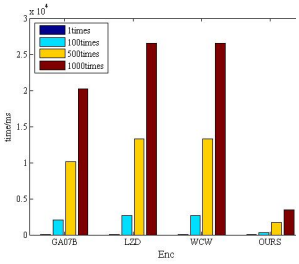

Fig. 6. Encryption cost

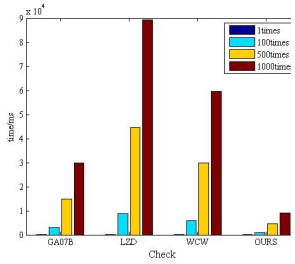

Fig. 7. Checking cost

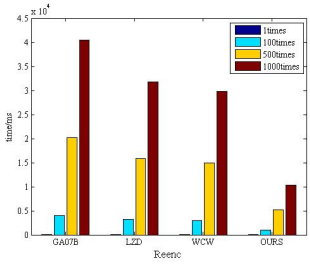

Fig. 8. Re-encryption cost

Notations: In Table 2, encryption is denoted as Enc, re-encryption is denoted as Reenc, decryption is denoted as Dec, ciphertext is denoted as Ciph and ciphertext length is denoted as Ciph-Len, t_p, t_e and t_{me} represent the computational cost of a pairing, a modular exponentiation and a modular multi-exponentiation in $\mathbb{G}_1$, respectively in type-I bilinear group($\mathbb{G}_1 = \mathbb{G}_2$). t'_p, t'_e, $t_e^{''}$ and t'_{me} represent the computational cost of a pairing, a mod-

ular exponentiation in $\mathbb{G}_1$, a modular exponentiation in $\mathbb{G}_2$ and a multi-exponentiation in $\mathbb{G}_1$ respectively in type-III bilinear group($\mathbb{G}_1 \neq \mathbb{G}_2$). t_{se}, t_{sd} and t_{sv} represent the computational cost of once symmetric encryption, once symmetric decryption and once symmetric checking decryption results' validity. t_{s} and t_{v} represent the computational cost of a one-time signature signing and verification respectively. $|\mathbb{G}|$ and $\mathbb{G}_T|$ denote the bit-length of an element in groups $\mathbb{G}$ and $\mathbb{G}_T$ respectively for type-I pairing, $|\mathbb{G}_1|$, $|\mathbb{G}_2|$ and $\mathbb{G}_T|$ denotes the bit-length of an element in groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ respectively for type-III pairing. $|SE|$ denotes the bit length of once symmetric encryption. Finally, $|vk|$ and $|s|$ denote the bit length of the one-time signature's public key and a one-time signature respectively.

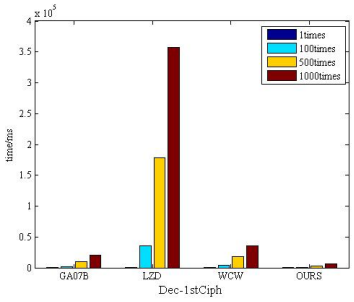

Fig. 9. First level decryption cost

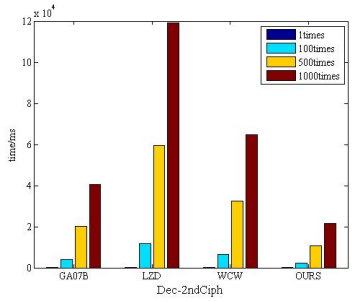

Fig. 10. Second level decryption cost

To further demonstrate our schemes efficiency, we roughly evaluated our scheme's practical performance according to implementation in [1]. We give the performance comparison results for GA07B, LZD+, WCW and Our schemes in Fig. 6,7,8,9,10, 11, 12, according to the efficiency benchmark of SS156 and BN256 groups [1] implemented in the highly efficient RELIC cryptographic toolkit version 0.4 [2](using the GMP library [12] for big number operations and the default configuration options for prime field arithmetic), measured on a standard workstation, which is 2.4GHz Intel Core i5 processor and 8GB of RAM (1067MHz DDR3) running Mac OS X Lion version 10.7.5. We have neglected some operations such as the computation cost of one-time symmetric encryption, one-time symmetric decryption and one-time checking for the decryption results' validity, one-time signature and verification. From Fig. 6,7,8,9,10, we can see our scheme is the most efficient scheme for encryption, checking, re-encryption, second-level ciphertext decryption and first-level ciphertext decryption. From Fig. 11, 12, we can see our scheme is a little inefficiency on the ciphertext length compared with some other schemes. Considering our scheme have many interesting properties while others do not have, we think this is not an essential shortcoming.

5. Review of the DFA-based FPRE Scheme

The definition and security model of DFA-based FPRE can be found in [17], here we just review their scheme.

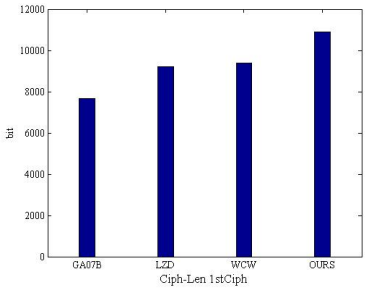

Fig. 11. First level ciphertext length

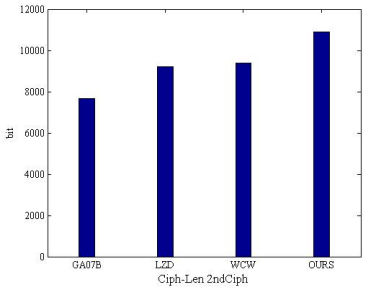

Fig. 12. Second level ciphertext length

1. Setup($1^n, \Sigma$): Choose $g, g_0, z, h_0 \in_R \mathbb{G}_{p_1}$, and $\alpha, k, a, b, \alpha_{end}, \alpha_{end} \in_R \mathbb{Z}_N^*$. Set $h_{start} = g^{\alpha_{start}}$, $h_{end} = g^{\alpha_{end}}$ and $h_k = g^k$. For each symbol $\delta \in \Sigma$, choose a $\alpha_\delta \in_R Z_N^*$, and set $h_\delta = g^{\alpha_\delta}$. Choose a one-time signature scheme OTS, a one-time symmetric encryption scheme $SYM = (SYM.Enc, SYM.Dec)$, and two hash functions: $H_1 : G_T \to \mathbb{Z}_N^*$ and $H_2 : \mathbb{G}_T \to \{0,1\}^{poly(n)}$. The PP is $(e(g,g)^\alpha, g, g^{ab}, g_0, z, h_0, h_{start}, h_{end}, h_k, \forall_{\delta\in\Sigma} h_\delta, OTS, SYM, H_1, H_2)$ along with the descriptions of $\mathbb{G}$ and the alphabet Σ. The MSK is $(g^{-\alpha}, X_3)$, where X_3 is a generator of $\mathbb{G}_{p_3}$.
2. KeyGen($MSK, M = (\mathcal{Q}, \mathcal{T}, q_0, F)$:The description of M includes a set $\mathcal{Q}$ of states $q_0, \cdots, q_{|\mathcal{Q}|-1}$ and a set of transitions $\mathcal{T}$ where each transition $t \in \mathcal{T}$ is a triple $(x, y, \delta) \in \mathcal{Q} \times \mathcal{Q} \times \Sigma$. q_0 is designated as a unique start state and $F \subseteq \mathcal{Q}$ is the set of accept states. The algorithm chooses $D_0, D_1, \cdots, D_{|\mathcal{Q}|-1} \in_R \mathbb{G}_{p_1}$ (associating D_i with q_i), for each $t \in \mathcal{T}$, it chooses $r_t \in_R \mathbb{Z}_N^*$, $\forall q_x \in F$ it chooses $r_{end_x} \in_R \mathbb{Z}_N^*$, and chooses a $u \in_R \mathbb{Z}_N^*$. It also chooses $R_{start1}, R_{start2}, R_{start3}, R_{t,1}, R_{t,2}, R_{t,3}$, $R_{end_{x,1}}, R_{end_{x,2}} \in_R \mathbb{G}_{p_3}$ and a $r_{start} \in_R \mathbb{Z}_N^*$. The algorithm constructs the key as the follows. First it sets:

$$K_{start1} = D_0 \cdot (h_{start})^{r_{start}} \cdot R_{start1}, K_{start2} = g^{r_{start}} \cdot R_{start2},$$
$$K_{start3} = g^u \cdot R_{start3}$$

For each $t = (x, y, \delta) \in \mathcal{T}$ the algorithm sets:

$$K_{t,1} = D_x^{-1} \cdot z^{r_t} \cdot R_{t,1}, K_{t,2} = g^{r_t} \cdot R_{t,2}, K_{t,3} = D_y \cdot (h_\delta)^{r_t} \cdot R_{t,3}$$

For each $q_x \in F$ it computes:

$$K_{end_{x,1}} = g^{-\alpha} \cdot D_x \cdot (h_{end} \cdot g^{ab})^{r_{end_x}} \cdot g^{ku} \cdot R_{end_{x,1}},$$
$$K_{end_{x,2}} = g^{r_{end_x}} \cdot R_{end_{x,2}}$$

Finally the key is

$$SK = (M, K_{start1}, K_{start2}, K_{start3}, \forall t \in \mathcal{T}(K_{t,1}, K_{t,2}, K_{t,3}),$$
$$\forall q_x \in F(K_{end_{x,1}}, K_{end_{x,2}}))$$

3. ReKeyGen(SK_M, ω):

(a) Choose a $y \in_R \mathbb{G}_T$, and $v_x \in_R \mathbb{Z}_N^*$ (for $\forall q_x \in F$) and set $rk_1 = K_{start1}^{H_1(y)}, rk_2 = K_{start2}^{H_1(y)}, rk_1 = K_{start3}^{H_1(y)}, \forall t \in \mathcal{T}(rk_{t,1} = K_{t,1}^{H_1(y)}, rk_{t,2} = K_{t,2}^{H_1(y)}, rk_{t,3} = K_{t,3}^{H_1(y)}), \forall q_x \in F(rk_{end_{x,1}} = K_{end_{x,1}}^{H_1(y)} \cdot h_{end}^{v_x}, rk_{end_{x,2}} = K_{end_{x,2}}^{H_1(y)} \cdot g^{v_x})$

(b) Run $rk_4 \leftarrow Encrypt(PP, \omega, y)$, and finally output

$$rk_{M\rightarrow w} = (M, rk_1, rk_2, rk_3, rk_4, \forall t \in \mathcal{T}(rk_{t,1}, rk_{t,2}, rk_{t,3}), \forall q_x \in F(rk_{end_{x,1}}, rk_{end_{x,2}}))$$

4. Encrypt(PP, ω, m): Choose $s_0, s_1, \cdots, s_l \in_R \mathbb{Z}_N^*$, run $(ssk, svk) \leftarrow KeyGen(1^n)$ and constructs CT as following. First set $C_m = m \cdot e(g,g)^{\alpha s_l}, C_{start1} = C_{0,1} = g^{s_0}, C_{start2} = (h_{start})^{s_0}, C_{start3} = (g_0^{svk} h_0)^{s_0}$, for $i = 1$ to l set $C_{i,1} = g^{s_i}, C_{i,2} = (h_{w_i})^{s_i} z^{s_{i-1}}$, finally set

$$C_{end1} = C_{l,1} = g^{s_l}, C_{end2} = (h_{end} g^{ab})^{s_l}, C_{end3} = (h_k)^{s_l},$$
$$C_{end4} = Sign(ssk, (\omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{end1}, C_{l,2}), C_{end2}, C_{end3})).$$

The original ciphertext is

$$CT = (svk, \omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{l,1}, C_{l,2}), C_{end2}, C_{end3}, C_{end4})$$

5. ReEnc$(rk_{M\rightarrow\omega'}, CT)$:

(a) If

$$Verify(ssk, (C_{end4}, (\omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{end1}, C_{l,2}), C_{end2}, C_{end3}))) = 1$$

and $e(C_{start1}, g_0^{svk} h_0) = e(g, C_{Start3})$
proceeds; otherwise, output $\perp$.

(b) CT is associated with a string $\omega = (\omega_1, \cdots, \omega_l)$ and the re-encryption key $rk_{M\rightarrow\omega'}$ is associated with a DFA $M = (\mathcal{Q}, \mathcal{T}, q_0, F)$ where $ACCEPT(M, \omega)$. There must exist a sequence of $l+1$ states $u_0, u_1, \cdots, u_l$ and l transitions $t_1, \cdots, t_l$ where $u_0 = q_0$ and $u_l \in F$ and for $i = 1, \cdots, l$, we have $t_i = (u_{i-1}, u_i, w_i) \in \mathcal{T}$, the proxy re-encrypts CT as follows.

i. If first computes $A_0 = e(C_{start1}, rk_1)\, e(C_{start2}, rk_2)^{-1} = e(g, D_0)^{s_0 H_1(y)}$

ii. For $i = 1$ to l, it computes:

$$A_i = A_{i-1} \cdot e(C_{i-1,1}, rk_{t_i,1}) \cdot e(C_{i,2}, rk_{t_i,2}) \cdot e(C_{i,1}, rk_{t_i,3}) = e(g, D_{u_i})^{s_i H_1(y)}$$

Since M accepts w, we have that $u_l = q_x$ for some $q_x \in F$ and $A_l = e(g, D_x)^{s_l H_1(y)}$.

iii. It sets

$$A_{end} = A_l \cdot e(C_{end_{x,1}}, rk_{end_{x,1}})^{-1} \cdot e(C_{end_{x,2}}, rk_{end_{x,2}}) \cdot e(C_{end_{x,3}}, rk_3)$$
$$= e(g,g)^{\alpha s_l H_1(y)}$$

iv. The proxy sets $C_1 = SYM.Enc(H_2(\delta), \xi), C_2 = Encrypt(PP, \omega', \delta)$, where $\delta \in_R \mathbb{G}_T$ and $\xi = (CT||A_{end}||rk_4)$. It finally outputs the re-encrypted ciphertext $C^R = (C_1, C_2)$.

6. Dec(SK_M, CT): If

$$Verify(svk, (C_{end4}, (\omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{end1}, C_{l,2}), C_{end2}, C_{end3}))) = 1$$

and $e(C_{start1}, g_0^{svk} h_0) = e(g, C_{start3})$ proceed; otherwise, output $\perp$. First compute

$$B_0 = e(C_{start1}, K_{start1}) e(C_{start2}, K_{start2})^{-1} = e(g, D_0)^{s_0}$$

For $i = 1$ to l compute

$$B_i = B_{i-1} e(C_{i-1,1}, K_{t_i,1}) \cdot e(C_{i,2}, K_{t_i,2})^{-1} \cdot e(C_{i,1}, K_{t_i,3}) = e(g, D_{u_i})^{s_i}$$

Since M accepts w, we have that $u_l = q_x$ for some $q_x \in F$ and $B_l = e(g, D_x)^{s_l}$. Finally compute

$$\begin{aligned} B_{end} &= B_l \cdot e(C_{end_{x,1}}, K_{end_{x,1}})^{-1} e(C_{end_{x,2}}, K_{end_{x,2}}) \cdot e(C_{end_{x,3}}, K_{start3}) \\ &= e(g, g)^{\alpha s_l} \end{aligned}$$

and output the message $m = \frac{C_m}{B_{end}}$.

7. Dec(SK_M, C^R):
   (a) Run $\delta \leftarrow Decrypt(SK_M, C_2)$, compute $\xi \leftarrow SYM.Dec(H_2(\delta), C_1)$, where $\xi = (CT||A_{end}||rk_4)$.
   (b) Run $y \leftarrow Decrypt(SK_M, rk_4)$, then compute $Key = A_{end}^{H_1(y)^{-1}}$.
   (c) Verify $e(C_{start1}, g_0^{svk} h_0) = e(g, C_{start3})$, Verify $(svk, (C_{end4}, (\omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{l,1}, C_{l,2}), C_{end2}, C_{end3}))) \stackrel{?}{=} 1$ If the equations hold, proceed; otherwise, output $\perp$.
   (d) Output the message $m = C_m / Key$.

6. Cryptanalysis of the CCA-secure DFA based FPRE Scheme

6.1. On the IND-CCA Property

Attack I Here we give the first attack on the IND-CCA property, which can be seen in Figure 13. Note in this attack the delegator, delegatee and proxy are all uncorrupted. The main strategy is the following: the adversary just modifies the re-encryption key to be a valid ciphertext C_{attack}, then he queries this ciphertext to the decryption oracle to get secret value y. Note here C_{attack} is not a derivative of C^*. By using this value, he can derive the delegator's secret key and thus can easily find the plaintext corresponding to C^*, which breaks the IND-CCA property. Concretely, the attack can be described as following:

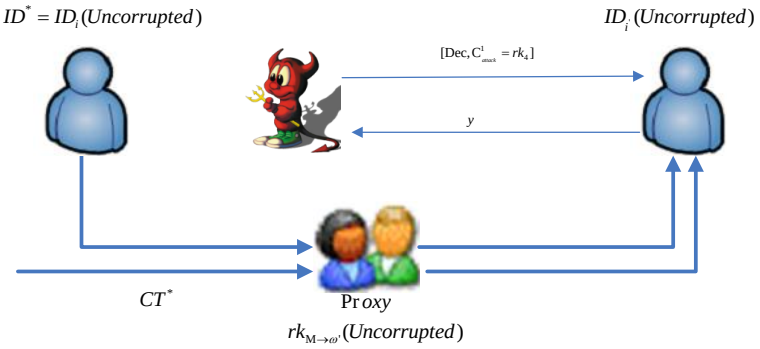

Fig. 13. Attack by querying the decryption oracle

1. Assume the challenge string is ω^* and the challenge original ciphertext is

$$CT = (svk, \omega^*, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{l,1}, C_{l,2}), C_{end2}, C_{end3}, C_{end4})$$

where $C_m = m_b \cdot e(g,g)^{\alpha s_l}, C_{start1} = C_{0,1} = g^{s_0}, C_{start2} = (h_{start})^{s_0}, C_{start3} = (g_0^{svk} h_0)^{s_0}$, for $i = 1$ to l set $C_{i,1} = g^{s_i}, C_{i,2} = (h_{\omega_i^*})^{s_i} z^{s_{i-1}}$, set

$$C_{end1} = C_{l,1} = g^{s_l}, C_{end2} = (h_{end} g^{ab})^{s_l}, C_{end3} = (h_k)^{s_l},$$
$$C_{end4} = Sign(ssk, (\omega^*, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}),$$
$$\cdots, (C_{end1}, C_{l,2}), C_{end2}, C_{end3})).$$

2. The adversary first queries the re-encryption key generation oracle with input ω^*, and the re-encryption key oracle runs as following:
   (a) Choose a $y \in_R \mathbb{G}_T$, and $v_x \in_R \mathbb{Z}_N^*$ (for $\forall q_x \in F$) and set $rk_1 = K_{start1}^{H_1(y)}, rk_2 = K_{start2}^{H_1(y)}, rk_1 = K_{start3}^{H_1(y)}, \forall t \in \mathcal{T}(rk_{t,1} = K_{t,1}^{H_1(y)}, rk_{t,2} = K_{t,2}^{H_1(y)}, rk_{t,3} = K_{t,3}^{H_1(y)}), \forall q_x \in F(rk_{end_{x,1}} = K_{end_{x,1}}^{H_1(y)} \cdot h_{end}^{v_x}, rk_{end_{x,2}} = K_{end_{x,2}}^{H_1(y)} \cdot g^{v_x})$
   (b) Run $rk_4 \leftarrow Encrypt(PP, \omega^*, y)$, and finally output

$$rk_{M \to w} = (M, rk_1, rk_2, rk_3, rk_4, \forall t \in \mathcal{T}(rk_{t,1}, rk_{t,2}, rk_{t,3}), \forall q_x \in F(rk_{end_{x,1}}, rk_{end_{x,2}})).$$

3. Now the adversary can extract a valid ciphertext from this re-encryption key, which is $rk_4 \leftarrow Encrypt(PP, \omega^*, y)$ This ciphertext has exactly the same structure as the original level ciphertext!
4. The adversary query this ciphertext to the original level ciphertext decryption oracle with $SK_{M'}$, where M' accept ω, and he will get y. Note here the adversary does not need to corrupt the delegatee, he just uses his decryption oracle!

5. After obtaining y, the adversary can easily compute the partial private key, which is enough to decrypt the challenge original level ciphertext:

$$K_{start1} = rk_1^{\frac{1}{H_1(y)}}, K_{start2} = rk_2^{\frac{1}{H_1(y)}}, K_{start3} = rk_1^{\frac{1}{H_1(y)}},$$
$$\forall t \in \mathcal{T}(K_{t,1} = rk_{t,1}^{\frac{1}{H_1(y)}}, K_{t,2} = rk_{t,2}^{\frac{1}{H_1(y)}}, K_{t,3} = rk_{t,3}^{\frac{1}{H_1(y)}}),$$
$$\forall q_x \in F(FK_{end_{x,1}} = (rk_{end_{x,1}}/h_{end}^{v_x})^{\frac{1}{H_1(y)}}, FK_{end_{x,2}} = (rk_{end_{x,2}}/g^{v_x})^{\frac{1}{H_1(y)}})$$

6. He decrypts the challenge ciphertext as following: First compute

$$B_0 = e(C_{start1}, K_{start1})e(C_{start2}, K_{start2})^{-1} = e(g, D_0)^{s_0}$$

For $i = 1$ to l compute

$$B_i = B_{i-1}e(C_{i-1,1}, K_{t_i,1}) \cdot e(C_{i,2}, K_{t_i,2})^{-1} \cdot e(C_{i,1}, K_{t_i,3}) = e(g, D_{u_i})^{s_i}$$

Since M accepts ω^*, we have that $u_l = q_x$ for some $q_x \in F$ and $B_l = e(g, D_x)^{s_l}$. Finally compute

$$\begin{aligned} B_{end} = \; & B_l \cdot e(C_{end_{x,1}}, (FK_{end_{x,1}}))^{-1} e(C_{end_{x,2}}, \\ & FK_{end_{x,2}}) \cdot e(C_{end_{x,3}}, K_{start3}) \\ = \; & e(g, g)^{\alpha s_l} \end{aligned}$$

and output the message $m_b = \frac{C_m}{B_{end}}$.

In this way, the adversary can easily break the IND-CCA security.

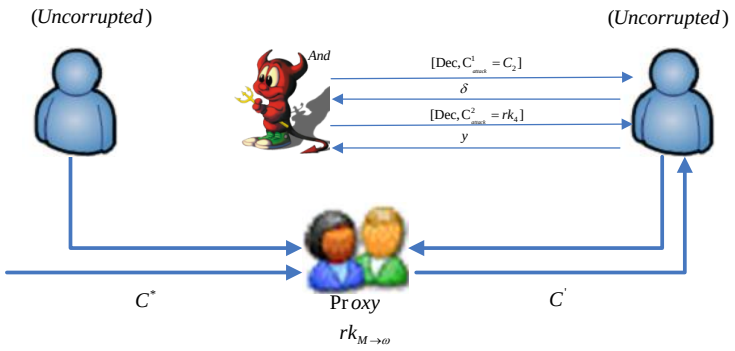

Fig. 14. Another attack by querying the decryption oracle

Attack II Here we give the second attack on the IND-CCA property, which can be seen in Figure 14. Note in this attack the delegator, delegatee and proxy are all uncorrupted. The main strategy is similar as the first attack except the adversary modifies the challenge

ciphertext to two valid ciphertexts. Concretely, the adversary first just modifies the re-encrypted challenge ciphertext C^* to be another ciphertext C^1_{attack} and C^2_{attack} which are valid original ciphertexts , then he queries this ciphertext to the decryption oracle to get secret value y. By using this secret value, he can derive the delegator's secret key and thus can easy find the plaintext corresponding to C^*, which breaks the IND-CCA property. For this attack is similar with Attack I, here we omit the details.

7. Our improved DFA-FPRE scheme

Our main idea is to modify the encryption algorithm used in the re-encryption process to be another one, which is not the same as the encryption algorithm used for second level ciphertext generation.

1. Setup($1^n, \Sigma$): Choose $g, g_0, z, h_0 \in_R \mathbb{G}_{p_1}$, and $\alpha, k, a, b, \alpha_{end}, \alpha_{end} \in_R \mathbb{Z}^*_N$. Set $h_{start} = g^{\alpha_{start}}$, $h_{end} = g^{\alpha_{end}}$ and $h_k = g^k$, $\boxed{h'_k = g^{k'}}$. For each symbol $\delta \in \Sigma$, choose a $\alpha_\delta \in_R Z^*_N$, and set $h_\delta = g^{\alpha_\delta}$. Choose a one-time signature scheme OTS, a one-time symmetric encryption scheme $SYM = (SYM.Enc, SYM.Dec)$, and two hash functions: $H_1 : G_T \to \mathbb{Z}^*_N$ and $H_2 : \mathbb{G}_T \to \{0,1\}^{poly(n)}$. The PP is $\{e(g,g)^\alpha, g, g^{ab}, g_0, z, h_0, h_{start}, h_{end}, h_k, \forall_{\delta\in\Sigma} h_\delta, OTS, SYM, H_1, H_2\}$ along with the descriptions of $\mathbb{G}$ and the alphabet Σ. The MSK is $(g^{-\alpha}, X_3)$, where X_3 is a generator of $\mathbb{G}_{p_3}$.
2. KeyGen($MSK, M = (\mathcal{Q}, \mathcal{T}, q_0, F)$):The description of M includes a set $\mathcal{Q}$ of states $q_0, \cdots, q_{|\mathcal{Q}|-1}$ and a set of transitions $\mathcal{T}$ where each transition $t \in \mathcal{T}$ is a triple $(x, y, \delta) \in \mathcal{Q} \times \mathcal{Q} \times \Sigma$. q_0 is designated as a unique start state and $F \subseteq \mathcal{Q}$ is the set of accept states. The algorithm chooses $D_0, D_1, \cdots, D_{|\mathcal{Q}|-1} \in_R \mathbb{G}_{p_1}$ (associating D_i with q_i), for each $t \in \mathcal{T}$, it chooses $r_t \in_R \mathbb{Z}^*_N$, $\forall q_x \in F$ it chooses $r_{end_x} \in_R \mathbb{Z}^*_N$, and chooses a $u \in_R \mathbb{Z}^*_N$. It also chooses $R_{start1}, R_{start2}, R_{start3}, R_{t,1}, R_{t,2}, R_{t,3}, R_{end_{x,1}}$, $R_{end_{x,2}} \in_R \mathbb{G}_{p_3}$ and a $r_{start} \in_R \mathbb{Z}^*_N$. The algorithm constructs the key as the follows. First it sets:

   $$K_{start1} = D_0 \cdot (h_{start})^{r_{start}} \cdot R_{start1}, K_{start2} = g^{r_{start}} \cdot R_{start2}, K_{start3} = g^u \cdot R_{start3}.$$

   For each $t = (x, y, \delta) \in \mathcal{T}$ the algorithm sets:

   $$K_{t,1} = D_x^{-1} \cdot z^{r_t} \cdot R_{t,1}, K_{t,2} = g^{r_t} \cdot R_{t,2}, K_{t,3} = D_y \cdot (h_\delta)^{r_t} \cdot R_{t,3},$$

   For each $q_x \in F$ it computes:

   $$k_{end_{x,1}} = g^{-\alpha} \cdot D_x \cdot (h_{end} \cdot g^{ab})^{r_{end_x}} \cdot g^{ku} \cdot R_{end_{x,1}}, K_{end_{x,2}} = g^{r_{end_x}} \cdot R_{end_{x,2}}$$

   Finally the key is

   $$SK = (M, K_{start1}, K_{start2}, K_{start3}, \forall t \in \mathcal{T}(K_{t,1}, K_{t,2}, K_{t,3}),$$
   $$\forall q_x \in F(K_{end_{x,1}}, K_{end_{x,2}})$$

3. ReKeyGen(SK_M, ω):

(a) Choose a $y \in_R \mathbb{G}_T$, and $v_x \in_R \mathbb{Z}_N^*$ (for $\forall q_x \in F$) and set $rk_1 = K_{start1}^{H_1(y)}, rk_2 = K_{start2}^{H_1(y)}, rk_1 = K_{start3}^{H_1(y)}, \forall t \in \mathcal{T}(rk_{t,1} = K_{t,1}^{H_1(y)}, rk_{t,2} = K_{t,2}^{H_1(y)}, rk_{t,3} = K_{t,3}^{H_1(y)}), \forall q_x \in F(rk_{end_{x,1}} = K_{end_{x,1}}^{H_1(y)} \cdot h_{end}^{v_x}, rk_{end_{x,2}} = K_{end_{x,2}}^{H_1(y)} \cdot g^{v_x})$

(b) Run $\boxed{rk_4 \leftarrow Encrypt'(PP, \omega, y)}$, and finally output $rk_{M\to w} = (M, rk_1, rk_2, rk_3, rk_4, \forall t \in \mathcal{T}(rk_{t,1}, rk_{t,2}, rk_{t,3}), \forall q_x \in F(rk_{end_{x,1}}, rk_{end_{x,2}}))$. Here the $Encrypt'(PP, \omega, y)$ algorithm runs as following: Choose $s_0, s_1, \cdots, s_l \in_R \mathbb{Z}_N^*$, run

$$(ssk, svk) \leftarrow KeyGen(1^n)$$

and constructs CT as following. First set $C_m = m \cdot e(g,g)^{\alpha s_l}, C_{start1} = C_{0,1} = g^{s_0}, C_{start2} = (h_{start})^{s_0}, C_{start3} = (g_0^{svk} h_0)^{s_0}$, for $i = 1$ to l set $C_{i,1} = g^{s_i}, C_{i,2} = (h_{w_i})^{s_i} z^{s_{i-1}}$, finally set

$$\boxed{C_{end1} = C_{l,1} = g^{s_l}, C_{end2} = (h_{end} g^{ab})^{s_l}, C_{end3} = (h_k)^{s_l}, C_{end4} = (h_k')^{s_l}}$$

$$\boxed{C_{end5} = Sign\,(ssk, (\omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{end1}, C_{l,2}), C_{end2}, C_{end3}, C_{end4}))}$$

The ciphertext is

$$CT = (svk, \omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{l,1}, C_{l,2}), C_{end2}, C_{end3}, C_{end4}, C_{end5})$$

4. Encrypt(PP, ω, m): Choose $s_0, s_1, \cdots, s_l \in_R \mathbb{Z}_N^*$, run $(ssk, svk) \leftarrow KeyGen(1^n)$ and constructs CT as following. First set $C_m = m \cdot e(g,g)^{\alpha s_l}, C_{start1} = C_{0,1} = g^{s_0}, C_{start2} = (h_{start})^{s_0}, C_{start3} = (g_0^{svk} h_0)^{s_0}$, for $i = 1$ to l set $C_{i,1} = g^{s_i}, C_{i,2} = (h_{w_i})^{s_i} z^{s_{i-1}}$, finally set

$$C_{end1} = C_{l,1} = g^{s_l}, C_{end2} = (h_{end} g^{ab})^{s_l}, C_{end3} = (h_k)^{s_l}$$
$$C_{end4} = Sign(ssk, (\omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{end1}, C_{l,2}), C_{end2}, C_{end3}))$$

The original ciphertext is

$$CT = (svk, \omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{l,1}, C_{l,2}), C_{end2}, C_{end3}, C_{end4})$$

5. ReEnc$(rk_{M\to\omega'}, CT)$:
(a) If

$$Verify(ssk, (C_{end4}, (\omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{end1}, C_{l,2}), C_{end2}, C_{end3}))) = 1$$

and $e(C_{start1}, g_0^{svk} h_0) = e(g, C_{Start3})$ proceeds; otherwise, output $\perp$.

(b) CT is associated with a string $\omega = (\omega_1, \cdots, \omega_l)$ and the re-encryption key $rk_{M\to\omega'}$ is associated with a DFA $M = (\mathcal{Q}, \mathcal{T}, q_0, F)$ where $ACCEPT(M, \omega)$. There must exist a sequence of $l+1$ states $u_0, u_1, \cdots, u_l$ and l transitions $t_1, \cdots, t_l$ where $u_0 = q_0$ and $u_l \in F$ and for $i = 1, \cdots, l$, we have $t_i = (u_{i-1}, u_i, w_i) \in \mathcal{T}$, the proxy re-encrypts CT as follows.

i. If first computes $A_0 = e(C_{start1}, rk_1)\, e(C_{start2}, rk_2)^{-1} = e(g, D_0)^{s_0 H_1(y)}$

ii. For $i = 1$ to l, it computes:

$$\begin{aligned} A_i &= A_{i-1} \cdot e(C_{i-1,1}, rk_{t_i,1}) \cdot e(C_{i,2}, rk_{t_i,2}) \cdot e(C_{i,1}, rk_{t_i,3}) \\ &= e(g, D_{u_i})^{s_i H_1(y)} \end{aligned}$$

Since M accepts w, we have that $u_l = q_x$ for some $q_x \in F$ and $A_l = e(g, D_x)^{s_l H_1(y)}$.

iii. It sets

$$\begin{aligned} &A_{end} = A_l \cdot e(C_{end_{x,1}}, rk_{end_{x,1}})^{-1} \cdot e(C_{end_{x,2}}, rk_{end_{x,2}}) \cdot e(C_{end_{x,3}}, rk_3) \\ &= e(g, g)^{\alpha s_l H_1(y)} \end{aligned}$$

iv. The proxy sets $C_1 = SYM.Enc(H_2(\delta), \xi), C_2 = Encrypt(PP, \omega', \delta)$, where $\delta \in_R \mathbb{G}_T$ and $\xi = (CT||A_{end}||rk_4)$. It finally outputs the re-encrypted ciphertext $C^R = (C_1, C_2)$.

6. Dec(SK_M, CT): If

$$\begin{aligned} &Verify(svk, (C_{end4}, (\omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, \\ &(C_{end1}, C_{l,2}), C_{end2}, C_{end3}))) = 1 \end{aligned}$$

and $e(C_{start1}, g_0^{svk} h_0) = e(g, C_{start3})$ proceed; otherwise, output $\perp$. First compute $B_0 = e(C_{start1}, K_{start1}) e(C_{start2}, K_{start2})^{-1} = e(g, D_0)^{s_0}$. For $i = 1$ to l compute $B_i = B_{i-1} e(C_{i-1,1}, K_{t_i,1}) \cdot e(C_{i,2}, K_{t_i,2})^{-1} \cdot e(C_{i,1}, K_{t_i,3}) = e(g, D_{u_i})^{s_i}$. Since M accepts w, we have that $u_l = q_x$ for some $q_x \in F$ and $B_l = e(g, D_x)^{s_l}$. Finally compute

$$\begin{aligned} B_{end} &= B_l \cdot e(C_{end_{x,1}}, K_{end_{x,1}})^{-1} e(C_{end_{x,2}}, K_{end_{x,2}}) \cdot e(C_{end_{x,3}}, K_{start3}) \\ &= e(g, g)^{\alpha s_l} \end{aligned}$$

and output the message $m = \frac{C_m}{B_{end}}$.

7. Dec(SK_M, C^R):

(a) Run $\delta \leftarrow Decrypt(SK_M, C_2)$, compute $\xi \leftarrow SYM.Dec(H_2(\delta), C_1)$, where $\xi = (CT||A_{end}||rk_4)$.

(b) Run $\boxed{y \leftarrow Decrypt'(SK_M, rk_4)}$, then compute $Key = A_{end}^{H_1(y)^{-1}}$, while

$$Decrypt'(SK_M, rk_4)$$

runs as following If

$$\begin{aligned} &Verify(svk, (C_{end5}, (\omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, \\ &(C_{end1}, C_{l,2}), C_{end2}, C_{end3}, C_{end4}))) = 1 \end{aligned}$$

and $e(C_{start1}, g_0^{svk} h_0) = e(g, C_{start3})$ proceed; otherwise, output $\perp$. First compute $B_0 = e(C_{start1}, K_{start1}) e(C_{start2}, K_{start2})^{-1} = e(g, D_0)^{s_0}$. For $i = 1$ to l compute $B_i = B_{i-1} e(C_{i-1,1}, K_{t_i,1}) \cdot e(C_{i,2}, K_{t_i,2})^{-1} \cdot e(C_{i,1}, K_{t_i,3}) = e(g, D_{u_i})^{s_i}$. Since M accepts w, we have that $u_l = q_x$ for some $q_x \in F$ and $B_l = e(g, D_x)^{s_l}$. Finally compute $B_{end} = B_l \cdot e(C_{end_{x,1}}, K_{end_{x,1}})^{-1} e(C_{end_{x,2}}, K_{end_{x,2}}) \cdot e(C_{end_{x,3}}, K_{start3}) = e(g, g)^{\alpha s_l}$ and output the message $m = \frac{C_m}{B_{end}}$.

(c) Verify $e(C_{start1}, g_0^{svk} h_0) = e(g, C_{start3})$, Verify

$$(svk, (C_{end4}, (\omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{l,1}, C_{l,2}), C_{end2}, C_{end3}))) \stackrel{?}{=} 1$$

If the equations hold, proceed; otherwise, output $\perp$.

(d) Output the message $m = C_m / Key$.

Theorem 2. *Suppose Liang et al's original scheme is IND-CPA secure, SYM is a CCA-secure symmetric encryption, OTS is a strongly existential unforgeable one-time signature and H_1, H_2 are TCR hash functions, our DFA-based FPRE system is IND-CCA secure in the standard model.*

Proof. Here we just give the roughly security proof. For we just modify Liang et al.'s scheme on the RekeyGen algorithm used in the re-encryption key generation process, and the corresponding Decrypt algorithm. And other algorithms are the same as the original scheme, thus the security proof is almost the same as Liang et al's proof.

But there is a crucial difference, in our scheme

$$rk_4 \leftarrow Encrypt'(PP, \omega, y)$$

and

$$\boxed{C_{end1} = C_{l,1} = g^{s_l}, C_{end2} = (h_{end} g^{ab})^{s_l}, C_{end3} = (h_k)^{s_l}, C_{end4} = (h'_k)^{s_l}}$$

$$\boxed{C_{end5} = Sign\,(ssk, (\omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{end1}, C_{l,2}), C_{end2}, C_{end3}, C_{end4}))}$$

which has not the same structure as the ciphertexts outputted by the Encrypt algorithm

$$C_{end1} = C_{l,1} = g^{s_l}, C_{end2} = (h_{end} g^{ab})^{s_l}, C_{end3} = (h_k)^{s_l}$$
$$C_{end4} = Sign(ssk, (\omega, C_m, C_{start1}, C_{start2}, C_{start3}, (C_{1,1}, C_{1,2}), \cdots, (C_{end1}, C_{l,2}), C_{end2}, C_{end3}))$$

that means, part of the re-encryption key can not be used to recover y by querying to the decryption oracle. Thus the above described attack can not be launched.

8. Conclusion

In this paper, we show two recently proposed proxy re-encryption schemes are not chosen ciphertext secure, the reason why their schemes can not be CCA secure is that the adversary can easily extract valid original level ciphertext from the re-encryption keys or the re-encrypted ciphertexts, while these extracted ciphertexts are not the challenge original level ciphertexts, neither the derivatives of challenge ciphertexts. We also propose a new CCA-secure AMH-IBCPRE and a CCA-secure DFA-based FPRE system and roughly analysis their security.

Acknowledgment This work is supported by National Natural Science Foundation of China (Grant Nos. 61772550, 61572521, U1636114, 61402531), National Cryptography Development Fund of China Under Grants No. MMJJ20170112, National Key Research and Development Program of China Under Grants No. 2017YFB0802000, Natural Science Basic Research Plan in Shaanxi Province of china (Grant Nos. 2018JM6028) and Guangxi Key Laboratory of Cryptography and Information Security (No. GCIS201610).

References

1. Akinyele, A., Garman, C., Hohenberger, S.: Authomating fast and secure translations from Type-I to Type-III paring schemes. ACM CCS 20015, pp. 1370–1381 (2015)
2. Aranha, D., Gouvea, C.: RELLC is an Efficient Library for Cryptography. http://code.google.com/p/relic-toolkit/.
3. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage. NDSS 2005, pp. 29–43 (2005)
4. Ateniese, G., Fu, K., Green M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Transation Information System Security 9(1), pp. 1–30 (2016)
5. Blaze, M., Bleumer G., Strauss, M.: Divertible Protocols and Atomic Proxy Cryptography. Advances in Cryptology - Eurocrypt'98, LNCS 1403, pp. 127–144 (1998)
6. Canetti, R., Hohenberger S.: Chosen Ciphertext Secure Proxy Re-encryption. ACM CCS 2007, pp. 185–194 (2007)
7. Deng, R., Weng, J., Liu, S., Chen, K.: Chosen Ciphertext Secure Proxy Re-encryption without Pairing. CANS'08, LNCS 5339, pp.1–17 (2008).
8. Li, F., Liu, B., Hong, J: An efficient signcryption for data access control incloud computing. Computing, 99(5), pp. 465-479 (2017).
9. Green, M., Ateniese, G.: Identity-based proxy re-encryption. ACNS 2007, volume 4521 of LNCS, pp. 288–306 (2007)
10. Guo, S., Xu, H.: A secure delegation scheme of large polynomial computation in multi-party cloud. International Journal of Grid and Utility Computing, 6(2), pp.1-7 (2015)
11. Hegarty, R., Haggerty, J.: Extrusion detection of illegal files in cloud-based systems. International Journal of Space-Based and Situated Computing, 5(3), pp. 150-158 (2015)
12. Torbjorn, G.(and the GMP development team): GNU MP: The GNU Multiple Precision Arithmetic Library, 5.0.5 edition, http://gmplib.org (2012)
13. Lai, J., Zhu, W., Deng, R., Liu, S., Kou W.: New constructions for identity-based unidirectional proxy re-encryption. Journal of Computer Science and Technology, 25(4), pp. 793–806 (2010)
14. Libert, B., Vergnaud, D.: Unidirectional Chosen-Ciphertext Secure Proxy Re-encryption. PKC 2008, LNCS 4939, pp. 360–379, Springer–Verlag (2008)
15. Libert, B., Vergnaud, D.: Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption. IEEE Transactions on Information Theory, 57(3), pp. 1786–1802 (2011)
16. Liang, K., Susilo, W., Liu J. K.: Privacy-Preserving Ciphertext Multi-Sharing Control for Big Data Storage. IEEE Transactions on Information Forensics and Security, 10(8), pp. 1578–1589 (2015)
17. Liang, K., Au, M., Liu, J., Susilo, W., Wong, D., Yang, G., Phuong, T., Xie, Q.: A DFA-Based Functional Proxy Re-Encryption Scheme for Secure Public Cloud Data Sharing. IEEE Transactions on Information Forensics and Security, 9(10), pp. 1667–1680 (2014)
18. Miao, Y., Ma, J., Liu, X., Li, X., Jiang, Q., Zhang, J.: Attribute-Based Keyword Search over Hierarchical Data in Cloud Computing. IEEE Transactions on Service Computing, DOI:10.1109/TSC.2017.2757467,(2017)

19. Miao, Y., Ma, J., Liu, X., Li, X., Liu, Z., Li. H.: Practical Attribute-Based Multi-keyword Search Scheme in Mobile Crowdsourcing. IEEE Internet of Things Journal, DOI:10.1109/JIOT.2017.2779124, (2017)
20. Miao, Y., Ma, J., Liu, X., Li, X., Jiang, Q., Zhang, J., Shen L., Liu, Z.: VCKSM:Verifiable conjunctive keyword search over mobile e-health cloud in shared multi-owner settings. Pervasive and Mobile Computing, 40, pp. 205-219 (2017)
21. Miao, Y., Ma, J., Liu, X., Zhang J., Liu Z.: VKSE-MO: verifiable keyword search over encrypted data in multi-owner settings. SCIENCE CHINA Information Sciences, 60(12), 122105:1-122105:15 (2017)
22. Liang, K., Susilo, W.: Searchable Attribute-Based Mechanism With Efficient Data Sharing for Secure Cloud Storage. IEEE Transactions on Information Forensics and Security, 10(9), pp. 1981–1992 (2015)
23. Petrlic, R., Sekula, S., Sorge, C.: A privacy-friendly architecture for future cloud computing. International Journal of Grid and Utility Computing, 4(4), pp. 265-277 (2013)
24. Shao, J., Cao, Z.: CCA-secure proxy re-encryption without pairing. PKC 2009, LNCS 5443, pp. 357–376, Springer–Verlag (2009)
25. Wang, H., Cao, Z., Wang, L.: Multi-use and unidirectional identity-based proxy re-encryption schemes. Information Science, 180, pp. 4042–4059 (2010)
26. Shao, J., Cao, Z., Lin, P.: Generic construction for CCA-secure unidirectional proxy re-encryption. Security and Communication Networks, 2, pp. 1–16 (2009)
27. Weng, J., Deng, R., Chu, C., Ding, X., Lai J.: Conditional proxy re-encryption secure against chosen-ciphertext attack. ACM ASIACCS 2009, pp. 322–332 (2009)
28. Weng, J., Yang, Y., Tang, Q., Deng, R., Bao F.: Efficient conditional proxy re-encryption with chosen-ciphertext security. ISC 2009, LNCS 5735, pp. 151–166 (2009)
29. Weng, J., Zhao, Y., Hanaoka, G.: On the Security of a Bidirectional Proxy Re-encryption Scheme from PKC 2010. PKC 2011, pp. 284–295 (2011)
30. Weng, J., Chen, M., Yang, Y., Deng, R., Chen K., Bao, F.: CCA-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles. Science China Information Sciences, 53, pp. 593-606 (2010)
31. Chow, S., Weng, J., Yang, Y., Deng R.: Efficient unidirectional proxy re-encryption. AFRICACRYPT 2010, LNCS 6055, pp. 316–332 (2010)
32. Wang, X., Ma, J., Xhafa, F., Zhang, M., Luo, X.: Cost-effective secure E-health cloud system using identity based cryptographic techniques. Future Generation Computer System, 67, pp. 242–254 (2017)
33. Wang, X., Ma, J., Xhafa, F., Qin, B., Zhang, M.: New efficient chosen ciphertext secure Elgamal encryption schemes for secure Cloud storage service. International Journal of Web and Grid Services, 13(3), pp. 246–269 (2017)
34. Wang, X., Xhafa, F., Ma, J., Barolli, L., Ge. Y.: PRE+: dual of proxy re-encryption for secure cloud data sharing service. International Journal of Web and Grid Services, 14(1), pp. 44–69 (2018)
35. Guo, S., Xu, H.: A secure delegation scheme of large polynomial computation in multi-party cloud. International Journal of Grid and Utility Computing, 6(2), pp.1–7 (2015)
36. Wang, Y., Du, J., Cheng, X., Liu, Z., Lin, K.: Degradation and encryption for outsourced png images in cloud storage. International Journal of Grid and Utility Computing, 7(1), pp. 22–28 (2016)
37. Zhu, S., Yang, X.: Protecting data in cloud environment with attribute-based encryption. International Journal of Grid and Utility Computing, 6(2), pp. 91–97 (2015)
38. Meriem, T., Mahmoud, B., Fabrice, K.: An approach for developing an interoperability mechanism between cloud providers. International Journal of Space-Based and Situated Computing, 4(2), pp. 88–99 (2014)

39. Cristina, D., Elena, A., Catalin, L., Valentin, C.: A solution for the management of multimedia sessions in hybrid clouds. International Journal of Space-Based and Situated Computing, 4 (2), pp. 77-87 (2014)

Xu An Wang now is an associate professor in Engineering University of Chinese Armed Police Force. His main research interests include public key cryptography and cloud security. He has published about 150 papers in the field of cryptography, information security and computer science. He is an editor board member of several international journals and has been a guest editor for several special issues. He is also a TPC member or co-chair for several international conferences.

Xiaoyuan Yang now is a professor in Engineering University of Chinese Armed Police Force. His main research interests include public key cryptography and information hiding.

Cong Li now is a master student in Engineering University of Chinese Armed Police Force. His main research interests include attribute based encryption and public key cryptography.

Yudong Liu now is a master student in Engineering University of Chinese Armed Police Force. His main research interests include homomorphic encryption and cloud security.

Yong Ding now is a professor in Guilin University of Electronic Technology. His main research interests include cryptography and information security.

Received: December 18, 2017; Accepted: July 11, 2018.