Analysis on the Security of an Identity Based Proxy Re-encryption

Xuan Wang and Xiaoyuan Yang
Key Laboratory of Information and Network Security
Engineering College of Chinese Armed Police Force, P. R. China
E-mail: wangxahq@yahoo.com.cn, wangxuaan@gmail.com

Abstract—In Pairing’07, Matsuo proposed two proxy re-encryption schemes: proxy re-encryption from CBE to IBE and IBE to IBE. Now both of the schemes have been standardized by P1363.3工作组. In this paper, we show that the proxy re-encryption scheme from IBE to IBE is not as secure as its author claimed. We give two attacks to this scheme. The first attack shows that the proxy can re-encrypt any IBE user’s ciphertext to be the delegatee’s ciphertext. The second attack implies that, if the proxy colludes with any delegatee, the proxy and this delegatee can derive any other IBE user’s secret key.

I. INTRODUCTION

The concept of proxy re-encryption comes from the work of Blaze et al. in 1998 [2]. The goal of proxy re-encryption is to securely enable the re-encryption of ciphertexts from one key to another, without relying on trusted parties. In 2005, Ateniese et al proposed few new proxy re-encryption schemes and discussed its several potential applications [1]. In Pairing’07, Green et al. proposed the first identity based proxy re-encryption schemes [6]. In Pairing’07, Matsuo proposed new proxy re-encryption schemes in identity based setting [8]. Interestingly, they proposed the concept of four types of proxy re-encryption: IBE to IBE, IBE to CBE, CBE to CBE and CBE to IBE, which can help the ciphertext circulate smoothly in the network. They constructed two proxy re-encryption schemes: one is the hybrid proxy re-encryption from CBE to IBE, the other is the proxy re-encryption from IBE to IBE. Now both of the schemes have been standardized by P1363.3工作组 [7].

In this letter, we show that Matsuo’s proxy re-encryption from IBE to IBE 1 is not as secure as its author claimed. We first review the definition and security model for identity based proxy re-encryption proposed by Matsuo in Pairing’07, and then we review their scheme. At last we give two attacks to their scheme in their security model.

II. REVIEW OF IDENTITY BASED PROXY RE-ENCRYPTION PROPOSED IN PAIRING’07

A. Definition and Security Model

There are five entities involved in an identity based proxy re-encryption system, delegator, proxy, delegatee, PKG and Re-encryption Key Generator RKG 2.

1Proxy re-encryption scheme from IBE to IBE is actually the identity based proxy re-encryption scheme.
2The PKG and the RKG might be operated by one entity.

Definition 1: An identity-based proxy re-encryption system consists of: 1) the four algorithms making up an IBE system SetupIBE, KeyGenIBE, EncIBE, and DecIBE, 2) and five algorithms for re-encryption, which are

1) EGen(skID, params). Given an IBE secret key skID for ID with params, generate eID for re-encryption key generation.
2) KeyGenRKG(mk, params). Given an IBE master-secret key mk with params, generate a secret key skR for re-encryption.
3) KeyGenPRO(skR, eID’, params, ID, ID’). Given sk, eID’, the delegator’s identity ID and the delegatee’s identity ID’ with params, generate a re-encryption key rkID′→ID.
4) ReEnc(rkID′→ID, params, CID, ID, ID’). Given the delegator’s identity ID, the delegatee’s identity ID’, the re-encryption key rkID′→ID, and an IBE ciphertext CID with params, re-encrypt CID into the different IBE ciphertext CID’.
5) Check(params, CID, ID). Given the delegator’s identity ID and an IBE ciphertext CID with params, output 0 if CID is a malformed ciphertext for ID. Otherwise, output 1.

Furthermore, the PKG deploys the digital signature scheme (KeyGenSIG, Sign, Verify) to sign eID for authenticating eID.

Definition 2: IND-IND-CPA security for identity based proxy re-encryption is defined as a game between the adversary A and challenger C like following:

Setup The challenger C selects a digital signature scheme (KeyGenSIG, Sign, Verify). C generates

1) (skSIG, vkSIG) by running KeyGenSIG.
2) (params, mk) by running SetupIBE and
3) skR by running KeyGenRKG. C gives (params, vk) to calA, keeping (mk, skSIG, skR) to itself.

Phase 1 Given (params, vkSIG), A adaptively queries C. When A queries C, it responds as following:

• Secret key queries. When A queries C at a point ID, C generates a secret key skID for ID by running KeyGenIBE. C computes eID by running EGen with the input skID. C generates a signature σc for ID by running Sign, and C returns (skID, ID||eID, σc) to A.
• Type-1 re-encryption key queries. When A queries C about IDi → ID′, C generates an IBE secret key skID′.
by running $\text{KeyGen}_{IBE}$, and computes $e_{1D'}$ by running $\text{EGen}$ with the input $sk_{ID'}$. $C$ generates a signature $\sigma_c$ for $ID'[e_{1D'}]$ by running $\text{Sign}$. $C$ runs $\text{KeyGen}_{PRO}$ with the inputs $e_{1D'}$, and returns the resulting re-encryption key $rk_{ID',-1D'}$ with $(ID'[e_{1D'}], e_{1D'})$ to $A$.

- **Type-2 re-encryption key queries.** Suppose that $A$ queries $C$ about $(ID_i \rightarrow ID'_i, ID'_i[e_{1D'}], \sigma_c)$. If $ID'_i[e_{1D'}]$ has already generated in the answering for secret key query, then $C$ rejects the query. Otherwise $C$ verifies $(ID'_i[e_{1D'}], \sigma_c)$ by running $\text{Verify}$ with $vk$, and works as follows:

  1) If it is valid then $C$ runs $\text{KeyGen}_{PRO}$ with the input $e_{1D'}$, and returns the resulting re-encryption key $rk_{ID',-1D'}$.

  2) Otherwise $C$ rejects the query.

- **Re-encryption queries.** Suppose that $A$ queries $C$ about $(sk_{ID'}, C_{ID'}, ID_i \rightarrow ID'_i)$. If $sk_{ID'}$ has never issued to $A$ then $C$ rejects the query. Otherwise $C$ runs $\text{Check}$ with the input $(params, C_{ID'}, ID_i)$.

  1) If $\text{Check}$ outputs 0 then $C$ rejects the query.

  2) Otherwise, $C$ generates $e_{1D'}$ by running $\text{EGen}$ with $sk_{ID'}$ as input. $C$ generates $rk_{ID',-1D'}$ by running $\text{KeyGen}_{PRO}$ with the input $e_{1D'}$. $C$ re-encrypts $C_{ID'}$ into $C'_{ID'}$ by running $\text{ReEnc}$ with the input $rk_{ID',-1D'}$. $C$ returns $C'_{ID'}$ to $A$.

- **Challenge.** After some queries, $A$ selects two equal length plaintexts $M_0, M_1 \in M$ and a target identity $ID^*$ which no secret key for $ID^*$ has issued, and sends them to $C$. Given $(M_0, M_1, ID^*)$, $C$ selects $d \rightarrow R \{0,1\}$ and computes $C_{ID^*} = \text{Enc}_{IBE}(ID^*, \text{params, } M_d)$. $C$ returns $C_{ID^*}$ to $A$.

- **Phase 2.** $A$ continues to issue queries as in Phase 1, and $C$ responds as before except the following case.

  1) If $A$ makes the secret key query at the point $ID^*$, then $C$ rejects.

  2) If $A$ makes the re-encryption query such that $ID_i = ID^*$, then $C$ rejects.

- **Guess.** Finally, $A$ outputs a guess $d' \in \{0, 1\}$. The adversary $A$ wins if $d' = d$. An identity-based proxy re-encryption system is secure in the sense of IND-ID-CPA if $|Pr[d' = d] - 1/2|$ is negligible.

### B. Matsuo’s Scheme

The underlying IBE scheme ($BB_1$-IBE scheme): Let $G$ be a bilinear group of prime order $p$ (the security parameter determines the size of $G$). Let $e : G \times G \rightarrow G_1$ be the bilinear map. For now, we assume public keys ($ID$) is element in $Z_p^*$. We later extend the construction to public keys over $\{0, 1\}^*$ by first hashing $ID$ using a collision resistant hash $H : \{0, 1\}^* \rightarrow Z_p$. We also assume messages to be encrypted are elements in $G$. The IBE system works as follows:

1) **SetUpIBE($k$).** Given a security parameter $k$, select a random generator $g \in G$ and random elements $g_2, h \in G$. Pick a random $\alpha \in Z_p^*$. Set $g_1 = g^\alpha$, $mk = g_2^2$, and $params = (g, g_1, g_2, h)$. Let $mk$ be the master-secret key and let $params$ be the public parameters.

2) **KeyGen_{IBE}(mk, params, ID).** Given $mk = g_2^2$ and $ID$ with $params$, the $PKG$ pick a random $u \in Z_p^*$. Set $sk_{ID} = (d_0, d_1) = (g_2(g_1^{d_1}h)^u, g^u)$. 

3) **Enc_{IBE}(ID, params, M).** To encrypt a message $M \in G_1$ under the public key $ID \in Z_p^*$, pick a random $r \in Z_p^*$ and compute $C_{ID} = (g^r, (g_1^{d_1}h)^r, Me(g_1, g_2)^r)$. 

4) **Dec_{IBE}(sk_{ID}, params, C_{ID}).** Given ciphertext $C_{ID} = (C_1, C_2, C_3)$ and the secret key $sk_{ID} = (d_0, d_1)$ with $params$, compute $M = \frac{C_3e(d_0, C_2)}{C_1e(d_0, C_1)}$.

The delegation scheme:

1) **EGen(sk_{ID}, params).** Given $sk_{ID} = (d_0, d_1) = (g_2^2(g_1^{d_1}h)^u, g^u)$ for $ID$ with $params$, set $e_{1D} = d_1 = g^u$.

2) **KeyGen_{PKG}(mk, params).** Given $mk = \alpha$ with $params$, set $sk_R = \alpha$.

3) **KeyGen_{PRO}(sk_R, e_{1D'}, params, ID, ID').** Given $sk_R = \alpha, e_{1D'} = g^u$ with $params$, set $rk_{1D',-1D'} = (ID \rightarrow ID', g^{u\alpha})$.

4) **Check(params, C_{ID'}).** Given the delegator’s identity $ID$ and $C_{ID'} = (C_1, C_2, C_3)$ with $params$, compute $v_0 = e(C_1, g_1^{ID'}h)$ and $v_1 = e(C_2, g)$. If $v_0 = v_1$ then output 1. Otherwise output 0.

5) **ReEnc(rk_{1D',-1D'}, params, C_{ID'}, ID').** Given identities $ID, ID', rk_{1D',-1D'} = (ID \rightarrow ID', g^{u\alpha})$, $C_{ID'} = (C_1, C_2, C_3)$ with $params$, the proxy re-encrypt the ciphertext $C_{ID}$ into $C_{ID'}$ as following. First it runs “Check”, if output 0, then return “Reject”. Else it computes $C_{ID'} = (C_1', C_2', C_3') = (C_1, C_2, C_3e(C_1^{ID'-ID'}h, g^{u\alpha}))$.

### III. Attacks

In this section, we give two attacks to this scheme in their security model.

#### A. Attack 1

Suppose adversary $A$’s target identity is $ID^*$, he attacks as following:

1) **First $A$ makes secret key query on a randomly chosen identity $ID$.** For $ID \neq ID^*$, Challenger $C$ returns $sk_{ID} = (d_0, d_1) = (g_2^2(g_1^{ID'}h)^u, g^u)$.

2) **Next $A$ makes Type-1 re-encryption key queries on $ID \rightarrow ID$ where $ID$ is a randomly chosen identity.** For $ID \neq ID^*$, Challenger $C$ returns $rk_{ID,-ID} = (rk_1, rk_2) = (ID \rightarrow ID, g^{u\alpha})$.

3) **When $A$ receives a ciphertext $C_{ID} = (C_1, C_2, C_3)$ for $ID^*$ with $params$, he re-encrypts this ciphertext to the ciphertext for $ID$ as following.** $C_{ID} = (C_1, C_2, C_3) = (C_1^*, C_2^*, C_3^*e(C_1^{ID-ID'}h, g^{u\alpha}))$ where $g^{u\alpha} = rk_2$. We can verify this ciphertext is a valid ciphertext for $ID$.

4) **$A$ can decrypt the ciphertext $C_{ID}$ by the secret key $sk_{ID}$**
for the following.

\[
C_3 e(d_1, C_2) = \frac{e(d_0, C_1)}{e(g_1^{(ID - ID^*)}, g_2^{u\alpha}) e(g^u, (g_1^{ID^*} h^u)^r)} = \frac{e(g_2^2 (g_1^{ID} h)^u, g^r)}{e(g_1, g_2)^r e((g_1^{ID} h)^r, g^u)} M^* \cdot e(g_1, g_2)^r e((g_1^{ID} h)^r, g^u)
\]

And thus he can decrypt every ciphertext for \( ID^* \).

**Remark 1:** The first attack shows that the proxy can re-encrypt any IBE user’s ciphertext to be the delegatee’s ciphertext, which beyond the ability supposed to give proxy, that is, just re-encrypting the delegator’s ciphertext to be the delegatee’s ciphertext.

**B. Attack II**

Suppose adversary \( A \)’s target identity is \( ID^* \), he attacks as following:

1) First \( A \) makes secret key query on a randomly chosen identity \( ID \). For \( ID \neq ID^* \), Challenger \( C \) returns \( sk_{ID} = (g_2^2 (g_1^{ID} h)^u, g^u) \) where \( u \) is randomly chosen from \( \mathbb{Z}_q \).

2) Next \( A \) makes Type-1 re-encryption key queries on \( \tilde{ID} \rightarrow ID \) where \( \tilde{ID} \) is a randomly chosen identity. For \( \tilde{ID} \neq ID^* \), Challenger \( C \) returns \( r_{k, \tilde{ID} \rightarrow ID} = (r_k_1, r_k_2) = (\tilde{ID} \rightarrow ID, g^{u\alpha}) \).

3) Now \( A \) can compute valid private keys for \( ID^* \).

\[
g_2^2 (g_1^{ID} h)^u \cdot g^{u\alpha ID^*} = g_2^2 (g_1^{ID^*} h)^u
\]

and we can see \( (g_2^2 (g_1^{ID^*} h)^u, g^u) \) is a valid private key for \( ID^* \).

**Remark 2:** The second attack implies that, if the proxy colludes with any delegatee, the proxy and this delegatee can derive any other IBE user’s secret key.

**IV. Conclusion**

In this letter, we give two attacks to an identity based proxy re-encryption scheme [8]. The reason why their scheme is not secure is that their re-encryption key is of the form \( g^{\alpha/\alpha} \) where \( \alpha \) is the master – key. This re-encryption key can give the adversary more power than just doing re-encryption from the delegator to the delegatee.

**V. Acknowledgements**

The authors would like to express their gratitude thanks to Dr. Whyte Willam and Dr. Luther Martin for providing us the chance to present this work in the workgroup teleconference.