Collusion-Resistant Identity-based Proxy Re-encryption

Woo Kwon Koo\textsuperscript{1}, Jung Yeon Hwang\textsuperscript{2}, and Dong Hoon Lee\textsuperscript{1,*}

\textsuperscript{1} Graduate School of Information Security, Korea University, Seoul, Korea
\textsuperscript{2} Electronics and Telecommunications Research Institute(ETRI), Daejeon, Korea
kwk4386@korea.ac.kr, videmot@etri.re.kr, donghlee@korea.ac.kr

Abstract. An identity-based proxy re-encryption scheme allows the semi-trusted proxy to convert ciphertexts encrypted under Alice’s public identity into the ciphertexts encrypted under Bob’s public identity. The proxy cannot learn any information of the underlying message. As in other delegation schemes, identity-based proxy re-encryption scheme should be secure against collusion, especially of the proxy and a delegatee. Unfortunately, no collusion-resistant identity-based proxy re-encryption with CCA-security and non-interactivity has not been proposed yet. In this paper we propose a novel collusion-resistant scheme which enjoys useful properties such as unidirectionality, non-interactivity and non-transferability. We show that our approach using two secret keys is sufficient to achieve the CCA-security and actually minimizes a security loss from the collusion attack. This result affirmatively resolves the open problem posed by Koo et al..

Keywords: Identity-based proxy re-encryption, collusion-resistance, unidirectionality, non-interactivity, non-transferability

1 Introduction

Identity-based proxy re-encryption (IBPRE) is identity-based encryption (IBE) which permits delegation of decryption capability \cite{5}. In IBPRE, a user who has a secret key corresponding to his/her public identity can decrypt a ciphertext encrypted with the identity as in IBE. In addition the user can delegate his/her decryption capability to another user by providing a “proxy” with a re-encryption key and requesting the proxy to convert a ciphertext into a ciphertext that the delegatee can decrypt. Due to this delegation of decrypting capability, IBPRE can be used for various applications such as email forwarding, DRM, law enforcement, and secure network file storage \cite{1–4,7}.

In the delegation process above, we must carefully consider a security property related to the proxy. Obviously the collusion of the proxy and a delegatee

\textsuperscript{*}Corresponding author. Tel.: +82 2 3290 4892; fax: +82 2 928 9109.

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST)(No. 2011-0029831) and IT R&D program of MKE/KEIT. [KI002113, Development of Security Technology for Car-Healthcare]
may result in the decryption of ciphertexts for a delegator if the ciphertexts are re-encrypted. This (potential) security threat from the collusion is inevitable because of the intrinsic delegation structure of an IBPRE scheme. However malicious behaviors by the colluding adversaries must be confined into such decryption. For example, the secret keys of honest delegators must not be revealed from the collusion because these secret keys can be used for a variety of their own security tasks.

Novel IBPRE schemes were proposed in [5, 4, 8] to provide useful properties such as unidirectionality and non-interactivity. Unidirectionality means that a delegator can delegate only his decryption capability to a delegatee, without permitting the delegator to decrypt the delegatee’s ciphertexts. Non-interactivity means that a delegator can generate a re-encryption key without the participation of a delegatee or any other third party. However, these schemes are known to be vulnerable to a collusion attack which reveals an honest user’s secret key [6]. This security weakness further induces a failure of its achievement for the CCA-security and so, undesirably strong trust in the proxy is required.

Besides the exposure of the delegator’s secret key through collusion of the proxy and the delegatee, we can also consider another security property that the proxy and the delegatee cannot re-delegate decryption rights. For example, given a re-encryption key $RK_{id \rightarrow \hat{id}}$ that delegates from $id$ to $\hat{id}$, and a delegatee’s secret key $sk_{\hat{id}}$, and a public identity $\overline{id}$, they cannot produce a new re-encryption key $RK_{\hat{id} \rightarrow \overline{id}}$ which is supposed to be generated by the delegator. This property is known as “non-transferability” in [1, 9]. It is important in a sense that the colluding adversaries must not misuse capabilities for the delegator beyond decryption.

**Our Contributions.** In this paper, we present a collusion-resistant and non-transferable IBPRE scheme which improves the scheme of [5] while inheriting all useful properties such as unidirectionality and non-interactivity.

Our improvement idea is simple but sufficient to achieve the robustness to a collusion attack of the proxy and a delegatee. In our scheme a user has two secret keys which must be used to decrypt the user’s ciphertexts regardless of re-encryption. Although the two secret keys are also used to generate a proxy re-encryption key, any individual key of the two keys cannot be computed through the collusion attack. Thus our scheme provides a kind of one-wayness for key derivation from a delegator’s secret key to a proxy re-encryption key. Note that most of unidirectional and non-interactive IBPRE schemes [5, 4, 8] fail to guarantee this one-wayness and so a delegator’s secret key can be easily computed with a delegatee’s secret key and a proxy re-encryption key. Actually, this security breach is intrinsically caused by the structure that generates re-encryption key. When the delegator generates the re-encryption key, the delegator’s secret key must be included to cancel out the terms of ciphertext that the delegatee cannot decrypt and then this secret key must be hidden by terms that the delegatee can decrypt not to be revealed to the proxy. This re-encryption structure is good method to achieve the unidirectional and non-interactive delegation structure of IBPRE, but it has intrinsic vulnerability to collusion attack between the del-
egatee and the proxy. In fact, our approach for collusion-resistance minimizes a security loss from the collusion attack in the sense that the colluding adversaries have only delegator’s decryption capability. We show that our approach is enough to achieve the CCA-security.

2 Preliminaries

2.1 Identity-Based Proxy Re-encryption

An Identity-Based Proxy Re-Encryption scheme (IBPRE) consists of six PPT algorithms Setup, KeyGen, Encrypt, RKGen, Reencrypt, Decrypt:

- **Setup(1^\lambda)** takes as input the security parameter \(\lambda\) and outputs the public parameters \(\text{params}\) and a master secret key \(\text{msk}\).
- **KeyGen(\text{params, msk, id})** takes as inputs the public parameter \(\text{params}\), an identity \(id \in \{0, 1\}^*\) and the master secret key \(\text{msk}\). It outputs a secret key \(sk_{id}\) corresponding to that identity.
- **Encrypt(\text{params, id, M})** takes as inputs the public parameters \(\text{params}\), an identity \(id \in \{0, 1\}^*\) and a plaintext \(M\). It outputs a first-level ciphertext \(CT_{id}\), the encryption of \(M\) under identity \(id\).
- **RKGen(\text{params, sk}_{id}, id, \hat{id})** takes as inputs the public parameters \(\text{params}\), a secret key \(sk_{id}\) (derived via the KeyGen algorithm) and identities \((id, \hat{id}) \in \{0, 1\}^*\). It outputs a re-encryption key from \(id\) to \(\hat{id}\) as \(RK_{id \rightarrow \hat{id}}\).
- **Reencrypt(\text{params, RK}_{id \rightarrow \hat{id}}, CT_{id})** takes as inputs a first-level ciphertext \(CT_{id}\) under identity \(id\) and a re-encryption key \(RK_{id \rightarrow \hat{id}}\) (generated by the RKGen algorithm). It outputs a second-level ciphertext \(CT_{\hat{id}}\) under identity \(\hat{id}\) or \(\bot\) (a symbol indicating an invalid ciphertext).
- **Decrypt(\text{params, sk}_{id}, CT_{id})** takes as inputs a first level ciphertext or a second level ciphertext \(CT_{id}\) under identity \(id\) and a secret key \(sk_{id}\). It output a plaintext \(M\) or \(\bot\) (a symbol indicating an invalid ciphertext).

**Correctness.** For all \((\text{params, msk})\) properly generated by **Setup(1^\lambda)**, any identities \(id, \hat{id} \in \{0, 1\}^*\), where the corresponding secret keys are \(sk_{id} \leftarrow \text{KeyGen(\text{params, msk, id})}\), \(sk_{\hat{id}} \leftarrow \text{KeyGen(\text{params, msk, \hat{id}})\)}, and \(RK_{id \rightarrow \hat{id}} \leftarrow \text{RKGen(\text{params, X}_{id, id, \hat{id})\)}\), the following equations hold. Let \(CT_{id} \leftarrow \text{Encrypt(\text{params, id, M})}\) be the first-level ciphertext and \(CT_{\hat{id}} \leftarrow \text{Reencrypt(\text{params, RK}_{id \rightarrow \hat{id}}, CT_{id})}\) be the second-level ciphertext.

- **Decrypt(\text{params, sk}_{id}, CT_{id}) = M, Decrypt(\text{params, sk}_{\hat{id}}, CT_{\hat{id}}) = M**

2.2 Bilinear Maps and Complexity Assumptions

**Bilinear Maps.** Let \(G\) and \(G_T\) be two multiplicative cyclic groups of prime order \(q\). Assume that \(g\) is a generator of \(G\). Let \(e\) be a bilinear map \(e : G \times G \rightarrow G_T\) which has the following properties:
1. Bilinearity: For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q^*$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.
3. Computability: There is an efficient algorithm to compute the map $e$.

We say that $\mathbb{G}$ is a bilinear group and $e$ is a bilinear pairing in $\mathbb{G}$.

The Decisional Bilinear Diffie-Hellman (DBDH) Problem. The DBDH problem [5] is defined as follows: given $(g, g^a, g^b, g^c, T) \in \mathbb{G}^4 \times \mathbb{G}^T$ as input, determine whether $T = e(g, g)^{abc}$ or $T$ is random in $\mathbb{G}^T$.

3 Identity-based Proxy Re-encryption Scheme

Setup. Given a security parameter $\lambda$, the algorithm selects generator $g \in \mathbb{G}$ and $\alpha, \beta \in \mathbb{Z}_q^*$ at random, and then computes $w = g^{\alpha - \beta}$. The public parameters $\text{params}$ and the master secret key $\text{msk}$ are defined by $\text{params} = (e, G, T, g, w, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4, \mathcal{H}_5)$, and $\text{msk} = (\alpha, \beta)$, where $\mathcal{H}_i : \{0,1\}^* \rightarrow \mathbb{G}$ for $i = 1, 2, 3, 4, 5 : \mathbb{G}_T \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$, $\mathcal{H}_5 : \mathbb{G}_T \rightarrow \{0,1\}^n$ for $n = \text{poly}(\lambda)$. Assume that all the hash functions are independently defined.

KeyGen. On inputs $\text{msk}$ and an identity $id \in \{0,1\}^*$, the algorithm returns the corresponding secret key $sk_{id} = (X_{id,\alpha}, X_{id,\beta}) = (\mathcal{H}_1(id)^{\alpha}, \mathcal{H}_1(id)^{\beta})$.

Encrypt. On inputs $\text{params}$, an identity $id$, and a plaintext $M \in \{0,1\}^n$, the algorithm works as follows:

1. Select a random $\sigma \in \mathbb{G}_T$ and compute $Q_{id} = \mathcal{H}_1(id), r = \mathcal{H}_4(\sigma, M)$.
2. Compute the ciphertext $CT_{id} = (id, C_1, C_2, C_3, C_4)$ as:
   
   $C_1 = g^r, C_2 = \sigma \cdot e(w, Q_{id})^r, C_3 = M \oplus \mathcal{H}_5(\sigma), C_4 = \mathcal{H}_5(id||C_1||C_2||C_3)^r$.

RKGen. On inputs $\text{params}$, a secret key $sk_{id} = (X_{id,\alpha}, X_{id,\beta})$, the algorithm selects a random $R \in \{0,1\}^n$ and computes the re-encryption key $RK_{id \rightarrow \hat{id}}$ as:

$$RK_{id \rightarrow \hat{id}} = \left(\frac{X_{id,\beta} \cdot \mathcal{H}_2(e(X_{id,\alpha}, Q_{\hat{id}})||id||\hat{id}||R)}{X_{id,\alpha} \cdot \mathcal{H}_2(e(X_{id,\beta}, Q_{\hat{id}})||id||\hat{id}||R)}, R\right).$$

Reencrypt. On inputs $\text{params}$, a (first-level) ciphertext $CT_{id}$ and a re-encryption key $RK_{id \rightarrow \hat{id}}$, the algorithm (run by Proxy) works as follows.

1. Parse $CT_{id}$ and $RK_{id \rightarrow \hat{id}}$ as $(id, C_1, C_2, C_3, C_4)$ and $(A, R)$, respectively.
2. Compute $h = \mathcal{H}_3(id||C_1||C_2||C_3)$.
3. If the equality $e(g, C_4) = e(C_1, h)$ does not hold, output $\perp$.
4. Otherwise, select a random $t \in \mathbb{Z}_q^*$, compute $\tilde{C}_2 = C_2 \cdot e(C_1, A \cdot h^t) \cdot e(g^t, C_4)^{-1}$, and return the re-encrypted ciphertext $CT_{id \rightarrow \hat{id}} = (id, \hat{id}, C_1, \tilde{C}_2, C_3, R)$.

Decrypt. This decryption algorithm works differently according to the level of a ciphertext.

- Decrypt$(1)$. On input $\text{params}$, a first-level ciphertext $CT_{id} = (id, C_1, C_2, C_3, C_4)$, and a secret key $sk_{id} = (X_{id,\alpha}, X_{id,\beta})$, the algorithm works as follows:
1. Compute $h = \mathcal{H}_3(id||C_1||C_2||C_3)$.
2. Select a random $t \in \mathbb{Z}_q^*$ and compute

$$
\sigma' = C_2 \cdot e(C_1, X_{id,\beta} \cdot X_{id,\alpha}^{-1} \cdot h^t) \cdot e(g^t, C_4)^{-1},
$$

$M' = C_3 \oplus \mathcal{H}_5(\sigma')$ and $r' = \mathcal{H}_4(\sigma', M')$.
3. Check if $C_4 \triangleq h^r$ and $C_1 \triangleq g^r$. If these equalities do not hold, output $\bot$, and otherwise, $M'$.

**Decrypt(2).** On input $\text{params}$, a second-level ciphertext $CT_{id \rightarrow \hat{id}} = (id, \hat{id}, C_1, \hat{C}_2, C_3, R)$, a secret key $(X_{\hat{id},\alpha}, X_{\hat{id},\beta})$, the algorithm computes as follows:

1. $K_1 = \mathcal{H}_2(e(Q_{\hat{id}}, X_{\hat{id},\alpha})||id||id||R)$, $K_2 = \mathcal{H}_2(e(Q_{\hat{id}}, X_{\hat{id},\beta})||id||\hat{id}||R)$.
2. $\sigma' = C_2 \cdot e(C_1, K_1^{-1} \cdot K_2)$, $M' = C_3 \oplus \mathcal{H}_5(\sigma')$, $r' = \mathcal{H}_4(\sigma', M')$.
3. Check if $C_1 \triangleq g^r$. If it does not hold, output $\bot$, and otherwise, $M'$.

**CCA Security.** As our scheme is similar to that of [5], our proof idea essentially follows that of [5]. We omit the details here due to the page limit.

### 4 Conclusion

We first constructed a collusion-resistant IBPRE scheme by using the two secret keys. This scheme achieves CCA-security while satisfying useful properties such as unidirectionality and non-interactivity. Our IBPRE scheme is an answer to the open problem presented in [6].

### References