An Efficient Non-transferable Proxy Re-Encryption Scheme

S. Sharmila Deva Selvi[⋆], Arinjita Paul[⋆⋆] and C. Pandu Rangan[⋆⋆]

Theoretical Computer Science Lab,
Department of Computer Science and Engineering,
Indian Institute of Technology Madras, Chennai, India.
{sharmila,arinjita,prangan}@cse.iitm.ac.in

Abstract. Proxy re-encryption (PRE) allows re-encryption of a ciphertext for Alice (delegator) into a ciphertext for Bob (delegatee) via a semi-trusted proxy, who should not obtain the underlying plaintext. Alice generates a re-encryption key (re-key) for the proxy using which, the proxy transforms the ciphertexts. The basic notion of PRE provides security against the proxy from learning anything about the encrypted message given the re-encryption key. However, this is not sufficient in all situations as the proxy can collude with Bob and re-delegate Alice's decryption rights. Hence, non-transferability is a desirable property in real-time scenarios wherein an illegal attempt to transfer Alice's decryption rights exposes Bob's private key as a penalty. In Pairing 2010, Wang et al. presented a CPA secure non-transferable Identity Based PRE scheme in the random oracle model. However, we show that the scheme violates the non-transferable property. Also, we present the first construction of a non-transferable unidirectional PRE scheme in the PKI setting using bilinear maps which meets CCA security under a variant of the decisional Diffie-Hellman hardness assumption in the random oracle model.

Keywords: Proxy Re-Encryption, Bilinear Maps, Public Key, Unidirectional, Non-transferable.

1 Introduction

Blaze et al. [2] in 1998 first proposed the concept of proxy re-encryption, which allows a proxy with specific information (re-encryption key) to translate a ciphertext for Alice into another ciphertext for Bob, without knowing the underlying plaintext. PRE has many useful applications, such as ensuring security of shared data in the cloud computing setting, enabling a data owner to encrypt shared data in the cloud in his public key and store them, which can be transformed by a

[⋆] Postdoctoral researcher supported by Project No. CCE/CEP/22/VK&CP/CSE/14-15 on Information Security & Awareness(ISEA) Phase-II by Ministry of Electronics & Information Technology, Government of India

[⋆⋆] Work supported by Project No. CCE/CEP/22/VK&CP/CSE/14-15 on ISEA-Phase II

proxy-server into a ciphertext for a legitimate recipient. This consigns the costly burden of secure data sharing to the resource-abundant semi-trusted proxy. PRE offers promising solutions to encrypted email forwarding, digital rights management, outsourced encrypted spam filtering among others [3], [1], [14].
PRE schemes are classified into bidirectional and unidirectional schemes based on the direction of delegation. They are also classified into single-hop and multi-hop schemes. In this paper, we focus on unidirectional single-hop PRE schemes. The existing PRE schemes assume that the proxy is semi-trusted and does not collude with Bob to acquire Alice's private key or re-delegate Alice's decryption rights to a malicious user Carol, failing to provide the non-transferable property which was first proposed by Ateniese et al. [1]. A PRE scheme is said to be non-transferable when the colluding proxy and delegatees should not be able to re-delegate decryption rights to other parties without compromising the private keys of the delegatees or the privacy of the delegatees. Note that Bob can always decrypt and forward the message to the malicious user Carol, but this would require Bob to be online. The notion of non-transferability is to prevent the colluding proxy and Bob to provide Carol with a secret value that can be used to decrypt Alice's ciphertexts when Bob is offline. Hence, the only way for Bob to transfer decryption capabilities to Carol is to reveal his own private key.

1.1 Related Work

While several protocols achieving PRE in various models are available, only a few provides the non-transferable property as well. In this section, we focus on PRE schemes supporting non-transferability. Illegal delegation of decryption rights would cause unauthorised sharing of data and financial losses which marks non-transferability as an important property in practice, such as the cloud service security scenario. Libert et al. [9] stated the difficulty in preventing such collusions and proposed a CPA secure scheme to trace the malicious proxies after a collusion. Even though penalising the colluders after an unauthorised transferance is a possible strategy to attain non-transferability, it is more desirable to prevent collusion than discouraging it. In the ID-based PRE scheme given by Wang et al. [13] in the random oracle model, a PKG generates the re-encryption keys and this is undesirable as it requires the PKG to be online for the re-encryption keys generation and introduces the *key-escrow problem* and *key-despotism problem.* He et al. [7] proposes a non-transferable ID-based PRE scheme in the random oracle model that addresses the previous problems but involves multiple rounds of interactions for partial-key generations and key-validations which makes their scheme less practical. Hayashi et al. [6] introduces a partial solution to non-transferability as their schemes are shown to achieve unforgeability of re-encryption keys against collusion attack (UFReKey-CA), assuming the hardness of the variants of the Diffie-Hellman inversion problem in the standard model, which was later shown vulnerable to forgeability attack on the re-encryption keys by Isshiki et al. [8]. Guo et al. [5] uses indistinguishability obfuscation ($i\mathcal{O}$), a highly complex primitive, to resolve the problem of non-transferability in PRE.

1.2 Our Contributions

In 2005, Ateniese et al. [1] stated that *"achieving a proxy scheme that is non-transferable, in the sense that the only way for Bob to transfer offline decryption capabilities to Carol is to expose his own secret key, seems to be the main open problem left for proxy re-encryption"*. Guo et al. [5] achieves non-transferability using indistinguishability obfuscation ($i\mathcal{O}$), a highly complex and impractical primitive. Our major contribution lies in providing a non-transferable unidirectional single-hop PRE scheme in the random oracle model that uses bilinear maps and group operations, and is much more practical.To the best of our knowledge, there are no known PRE schemes satisfying non-transferability in the PKI setting based on group theoretic operations. Wang et al.[13] proposed an uni-directional non-transferable PRE scheme in the random oracle model in the identity-based setting, in which the fully trusted PKG generates the re-encryption keys. We present an attack on their scheme, by showing that the colluders can indeed construct an illegal decryption function that can be used by any malicious third party to decrypt the delegator's second level ciphertexts, without any compromise of the delegatees private keys.

2 Preliminaries

2.1 Bilinear Pairings

Our PRE scheme is based on bilinear pairings. Let G_1 and G_2 be an additive and multiplicative cyclic groups respectively of prime order q. G_1 is generated by P. $\mathbb{G}_1$ has an admissible bilinear mapping into $\mathbb{G}_2$, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, if the following three conditions hold:

1. *Bilinear* :$\forall P, Q, R \in G_1$, $\forall a, b \in \mathbb{Z}_q^*$
   (a) $\hat{e}(P + Q, R) = \hat{e}(P, R) \cdot \hat{e}(Q, R)$
   (b) $\hat{e}(P, Q + R) = \hat{e}(P, Q) \cdot \hat{e}(P, R)$
   (c) $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
2. *Non-degenerate* :$\exists P, Q \in \mathbb{G}_1$ such that, $\hat{e}(P, Q) \neq 1_{\mathbb{G}_2}$.
3. *Computable* : $\forall P, Q \in \mathbb{G}_1$, there is an efficient algorithm to compute $\hat{e}(P, Q)$.

2.2 Hardness Assumptions

In this section, we state the computational hardness assumptions used to establish the security of the schemes.

Modified Decisional Bilinear Diffie-Hellman (m-DBDH) assumption [12] : The m-DBDH assumption is said to hold if, given the elements $\{P, aP, bP, cP, a^{-1}P\} \in \mathbb{G}_1$ and $T \in \mathbb{G}_2$, there exists no probabilistic polynomial-time adversary which can determine whether $T = \hat{e}(P, P)^{abc}$ or a random element from $\mathbb{G}_2$ with a non-negligible advantage, where P is a generator of $\mathbb{G}_1$ and $a, b, c \in_R \mathbb{Z}_q^*$.

1-weak Decisional Bilinear Diffie-Hellman Inversion (1-wDBDHI) assumption [10] : The 1-wDBDHI assumption is said to hold if, given the elements $\{P, \frac{1}{a}P, bP\} \in \mathbb{G}_1$ and $T \in \mathbb{G}_2$, there exists no probabilistic polynomial-time adversary which can determine whether $T = \hat{e}(P, P)^{ab}$ or a random element from $\mathbb{G}_2$ with a non-negligible advantage, where P is a generator of $\mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q^*$.

3 Definition and Security Model

3.1 Definition

We describe the syntactical definition of unidirectional proxy re-encryption[13] and its security notion. A PRE scheme consists of the following seven algorithms:

- Global setup(λ): returns a set of public parameters $params$, which is shared by all the users in the system.
- KeyGen($params$): returns the public key and private key pair (pk_i, sk_i) of a user i.
- ReKeyGen($sk_i, pk_i, pk_j, params$): returns a re-encryption key $RK_{i \to j}$.
- Encrypt($m, pk_i, params$): returns the ciphertext C_i corresponding to m which is allowed to be re-encrypted for another user. The ciphertext C_i generated is called as the second level ciphertext.
- Re-Encrypt($C_i, RK_{i \to j}, params$): returns a ciphertext C'_j, re-encryption of C_i, now encrypted under the public key pk_j. The re-encrypted ciphertext C'_j is called as the first level ciphertext.
- Decrypt($C_i, sk_i, params$): returns a plaintext m or the error symbol $\perp$ if the ciphertext is invalid.
- Re-Decrypt($C'_j, sk_j, params$): returns a plaintext m or the error symbol $\perp$ if the ciphertext is invalid.

The consistency of a PRE scheme for any given public parameters $params$ and a public-private key pair $\{(pk_i, sk_i), (pk_j, sk_j)\}$ is defined as follows:

1. Consistency between encryption and decryption; i.e.,
$$Decrypt(Encrypt(m, pk_i), sk_i) = m, \forall m \in \mathcal{M}$$
2. Consistency between encryption, proxy re-encryption and decryption; i.e.,
$$Re\text{-}Decrypt(Re\text{-}Encrypt(RK_{i \to j}, Encrypt(m, pk_i)), sk_j) = m, \forall m \in \mathcal{M}$$

3.2 Security Model

Since there exists two types of ciphertexts namely first level and second level ciphertexts in PRE, it is necessary to prove the security of each of these two levels as defined in [9]. As in [4], in our model, the adversary $\mathcal{A}$ can only obtain the uncorrupted public keys $pk_{i:i \in HU}$ and corrupted public-private key pairs $\{pk_i, sk_i\}_{i:i \in CU}$ from the challenger $\mathcal{C}$ and cannot determine which parties will be compromised adaptively. $\mathcal{A}$ is provided with re-encryption keys he is entitled to know but can adaptively query the re-encryption and decryption oracles which $\mathcal{C}$ answers as below and simulates an environment running PRE for $\mathcal{A}$.
– Re-encryption oracle $\mathcal{O}_{ReEnc}(C_i, pk_i, pk_j) : \mathcal{C}$ runs $C'_j \leftarrow ReEnc(C_i, RK_{i \to j})$, where $RK_{i \to j} = ReKeyGen(sk_i, pk_i, pk_j)$ and returns C'_j to $\mathcal{A}$.
– Second level decryption oracle $\mathcal{O}_{Dec}(C_i, pk_i) : \mathcal{C}$ runs $Decrypt(C_i, sk_i)$ and returns the result to $\mathcal{A}$.
– First level decryption oracle $\mathcal{O}_{ReDec}(C'_j, pk_j) : \mathcal{C}$ runs $ReDecrypt(C'_j, sk_j)$ and returns the result to $\mathcal{A}$.

Second level ciphertext security. It models the scenario that the adversary $\mathcal{A}$ is challenged with a second level ciphertext C^*, where C^* is the challenge ciphertext under the targeted public key pk_{i^*} where we use the index i^* to denote the targeted user. $\mathcal{C}$ responds to the queries issued by $\mathcal{A}$ to the above defined oracles considering that they do not allow $\mathcal{A}$ to decrypt the challenge ciphertext trivially. For example, $\mathcal{A}$ is not allowed to obtain a re-encryption key $RK_{i^* \rightarrow j}$ where sk_j was already compromised. In such a case, $\mathcal{A}$ can trivially decrypt the challenge ciphertext by first re-encrypting it into a first level ciphertext and then decrypting it with sk_j. Also, for a first level ciphertext $C'_j = \mathit{Re–Encrypt}(C^*_i, RK_{i^* \rightarrow j})$, querying on $\mathcal{O}_{ReDec}(C'_j, pk_j)$ by $\mathcal{A}$ is not permitted.

Below is given the formal definition for second level ciphertexts semantic security under chosen ciphertext attack (IND-PRE-CCA).

Definition 1. *Given a single-hop unidirectional PRE scheme, the advantage of any PPT adversary $\mathcal{A}$ denoted by $Adv_{\mathcal{A}}$ in the game shown below is defined by the probability:*

$$\begin{aligned}
Pr[\{(pk_i, sk_i) \leftarrow KeyGen(\lambda)\}_{i \in CU \cup HU}, (pk^*_i, sk^*_i) \leftarrow KeyGen(\lambda);\\
\{RK_{i^* \rightarrow j} \leftarrow ReKeyGen(sk^*_i, pk_j)\}_{j \in HU};\\
\{RK_{i \rightarrow j} \leftarrow ReKeyGen(sk_i, pk_j)\}_{i \in HU, j \in CU \cup HU \cup \{i^*\}},\\
(m_0, m_1, St) \leftarrow \mathcal{A}^{\mathcal{O}_{ReEnc}, \mathcal{O}_{ReDec}}(pk^*_i, \{pk_j, sk_j\}_{j \in CU},\\
\{pk_j\}_{j \in HU}, \{RK_{i^* \rightarrow j}\}_{j \in HU}; \{RK_{i \rightarrow j}\}_{i \in HU, j \in CU \cup HU \cup \{i^*\}});\\
b \epsilon_R \{0,1\}, C^* \leftarrow Encrypt(pk^*_i, m_b); b' \leftarrow \mathcal{A}^{\mathcal{O}_{ReEnc}, \mathcal{O}_{ReDec}}(C^*, St) : b' = b]
\end{aligned}$$

Note that $|m_0| = |m_1|$. St is the state information maintained by $\mathcal{A}$. A single hop unidirectional PRE scheme is IND-PRE-CCA secure for second level ciphertext if for any IND-PRE-CCA adversary $\mathcal{A}$, $|Adv_{\mathcal{A}} - \frac{1}{2}|$ is negligibly small.

First level ciphertext security. In the first-level ciphertext security, $\mathcal{A}$ is allowed to obtain the re-encryption keys for *any* user, since the first level ciphertext cannot be further re-encrypted in a given *single hop* PRE scheme. This also justifies the fact that there is no need for any second-level decryption or re-encryption oracle as all the re-encryption keys are available to $\mathcal{A}$.

Definition 2. *Given a single-hop unidirectional PRE scheme, the advantage of any PPT adversary $\mathcal{A}$ denoted by $Adv_{\mathcal{A}}$ in the game shown below is defined by the probability :*

$$\begin{aligned}
Pr[\{(pk_i, sk_i) \leftarrow KeyGen(\lambda)\}_{i \in CU \cup HU}, (pk^*_i, sk^*_i) \leftarrow KeyGen(\lambda);\\
\{RK_{i \rightarrow j} \leftarrow ReKeyGen(sk_i, pk_j)\}_{i,j \in CU \cup HU \cup \{i^*\}},\\
(m_0, m_1, St) \leftarrow \mathcal{A}^{\mathcal{O}_{ReDec}}(pk^*_i, \{pk_j, sk_j\}_{j \in CU},\\
\{pk_j\}_{j \in HU}, \{RK_{i \rightarrow j}\}_{i,j \in CU \cup HU \cup \{i^*\}});\\
b \epsilon_R \{0,1\}, C^* \leftarrow Re–Encrypt(Encrypt(m_b, pk_i), RK_{i \rightarrow i^*})_{i \in HU \cup CU};\\
b' \leftarrow \mathcal{A}^{\mathcal{O}_{ReDec}}(C^*, St) : b' = b]
\end{aligned}$$

Note that $|m_0| = |m_1|$ and St is the state information maintained by $\mathcal{A}$. A single hop unidirectional PRE scheme is said to be IND-PRE-CCA secure for first level ciphertext if for any IND-PRE-CCA adversary $\mathcal{A}$, $|Adv_{\mathcal{A}} - \frac{1}{2}|$ is negligibly small.

4 Non-transferability

In order to achieve non-transferability, Alice's ciphertext must possess the property that if a malicious user has the private key of Bob and the re-encryption key, only then it can obtain the plaintext, else it shall obtain nothing useful. Our security definition of non-transferability follows from the definition of non-transferability proposed in [6].
In the following definition, we use the following subscripts $i^*, h \in HU, c_i \in CU, j$ to denote a target honest delegator, an honest user, a corrupted delegatee and a malicious user respectively, where $i \in \{1, \cdots L\}$ and L is polynomially bounded.
Definition 3.[6] *Non-transferability: A single-hop unidirectional PRE scheme is non-transferable if there exists a polynomial time algorithm $\mathcal{J}'$, such that*

$$\begin{aligned}
Pr[(pk_i^*, sk_i^*) \leftarrow Keygen(1^\lambda); (pk_h, sk_h) \leftarrow Keygen(1^\lambda);\\
\{(pk_{c_i}, sk_{c_i} \leftarrow Keygen(1^\lambda)\}; (pk_j, sk_j) \leftarrow Keygen(1^\lambda);\\
\{RK_{i^* \rightarrow c_i} \leftarrow ReKeyGen(sk_i^*, pk_{c_i})\}; \{RK_{h \rightarrow c_i} \leftarrow ReKeyGen(sk_h, pk_{c_i})\};\\
m \leftarrow \mathcal{M}; C^* \leftarrow Encrypt(m, pk_i^*); \{m_i \leftarrow \mathcal{M}\}; \{C_i \leftarrow Encrypt(m_i, pk_{c_i})\};\\
\{m_i' \leftarrow \mathcal{M}\}; \{C_i' \leftarrow Re\text{–}Encrypt(RK_{h \rightarrow c_i}, Encrypt(m_i', pk_h))\};\\
X \leftarrow \mathcal{C}(pk_i^*, \{(pk_{c_i}, sk_{c_i})\}, \{RK_{i^* \rightarrow c_i}\}); m_{\mathcal{J}} \leftarrow \mathcal{J}(X, (pk_j, sk_j), C^*);\\
m_{\mathcal{J}'} \leftarrow \mathcal{J}'(X, (pk_j, sk_j), \{C_i\}, \{C_i'\})\\
: m \neq m_{\mathcal{J}} \vee m_{\mathcal{J}'} \in \{m_i\} \cup \{m_i'\}]
\end{aligned}$$

is overwhelming for any polynomial time algorithm $\mathcal{C}$, $\mathcal{J}$ and polynomial L.

In the above definition, $\mathcal{C}$ denote the set of colluders and $\mathcal{J}, \mathcal{J}'$ denotes the malicious users. The definition states that, if $\mathcal{C}$ tries to construct an illegal decryption box X for the second level ciphertext of the target honest user i^* to re-delegate the decryption rights to $\mathcal{J}$, then $\mathcal{J}'$ can exploit X to compromise the decryption capabilities of $\mathcal{C}$. Informally, the colluders should not be able to generate a decryption-box to decrypt the delegator's ciphertext, without compromising the private keys of the delegatee. The main challenge for constructing such a scheme lies in extracting the decryption capability of the delegatee from this illegal decryption box.

5 Analysis of a CPA-Secure Non-Transferable PRE Scheme by Wang et al.[13]

5.1 Review of the scheme

- **Setup(λ):** $\mathbb{G}_1$ and $\mathbb{G}_2$ are multiplicative groups of order p. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear map. PKG computes $g_1 = g^\alpha \in \mathbb{G}_1$ where g is a generator of $\mathbb{G}_1$ and $\alpha \in \mathbb{Z}_p^*$. Also, $g_2, \eta \in \mathbb{G}_1$ are chosen at random. $H : \{0,1\}^l \rightarrow \mathbb{G}_1$ is a cryptographic hash function. the system parameters are $params = \{\mathbb{G}_1, \mathbb{G}_2, p, \hat{e}, g, g_1, g_2, \eta, H\}$, and $msk = g_2^\alpha$.
- **Extract(id):** Choose $u \in \mathbb{Z}_p^*$, set $sk_{id} = (d_0, d_1) = (g_2^\alpha H(id)^u, g^u)$, where $u = h_{msk}(id)$. Validation of key by user id with sk sk_{id} is done by

$$\hat{e}(d_0, g) \stackrel{?}{=} \hat{e}(g_1, g_2)\hat{e}(H(id), d_1)$$

- **ReKeyGen(id, id'):** PKG returns seed of re-key to delegator id:

$$\tilde{rk}_{id \to id'} = \left(\frac{H(id)}{H(id')} \right)^{u'}$$

  Here, u' is selected by PKG to generate private key of id'. User id selects $\delta \in \mathbb{Z}_p^*$ at random and computes rekey as:

$$rk_{id \to id'} = (rk_1, rk_2) = \left(\eta^{\delta} \Big(\frac{H(id)}{H(id')} \Big)^{u'}, g^{\delta} \right)$$

- **Encryption($m \in \mathbb{G}_2, id$):** Encryptor chooses $r \in \mathbb{Z}_p^*$ and computes

$$C = (C_1, C_2, C_3, C_4) = (m.\hat{e}(g_1, g_2)^r, g^r, H(id)^r, \eta^r)$$

- **Re-Encryption(m, id'):** The proxy conducts a consistency check for the received 2^{nd} level ciphertext: $\hat{e}(C_2, \eta) \overset{?}{=} \hat{e}(C_4, g)$. If it holds, compute:

$$C' = (C'_1, C_2, C_3) = \left(C_1 . \frac{\hat{e}(C_4, rk_2)}{\hat{e}(C_2, rk_1)}, C_2, C_3 \right)$$

- **Decryption(C, sk_{id}):** m is obtained from the second level ciphertext by computing:

$$m = C_1 . \frac{\hat{e}(C_3, d_1)}{\hat{e}(C_2, d_0)}$$

- **Re-Decryption($C', sk_{id'}$):** m is obtained from the first level ciphertext by computing:

$$m = C'_1 . \frac{\hat{e}(C_3, d'_1)}{\hat{e}(C_2, d'_0)}$$

5.2 Attack on the Scheme

We show an attack on the non-transferable property of the ID-PRE scheme proposed in [13]. As per the definition of non-transferability in Section 4, the adversary is allowed to obtain one pair of keys $(rk_{id_{i*} \to id_j}, sk_{id_j})$ wherein the delegatee id_j is a corrupt user. So, consider the following attack where the adversary queries for a re-encryption key $(rk_{id_i \to id_j}) = (rk_1, rk_2)$ and a private key for id_j to obtain the corresponding private key $sk_{id_j} = (d_0, d_1) = (g_2^{\alpha} H(id_j)^{u_j}, g^{u_j})$. Now, given the second level ciphertext $C = (C_1, C_2, C_3, C_4)$, the adversary does the following computation:

1. Pick $\beta \in \mathbb{Z}_q^*$.
2. Define $d' \overset{\Delta}{=} d_1 \cdot g^{\beta} = g^{u_j + \beta}$.
3. Compute the value of a partial decryption key $psk_{id_i} = (rk_1 \cdot d_0 \cdot H(id_i)^{\beta})$
   $= \eta^{\delta} \big(\frac{H(id_i)}{H(id_j)} \big)^{u_j} \cdot g_2^{\alpha} H(id_j)^{u_j} \cdot H(id_i)^{\beta}$
   $= \eta^{\delta} \cdot H(id_i)^{u_j + \beta} \cdot g_2^{\alpha}$ (Note that this gives the adversary a function of the private key of user id_i which can be used to compute a decryption box for ciphertexts encrypted under id_i)

4. Construct a decryption box for a second level ciphertext of id_i as :

$$m = \frac{C_1}{\hat{e}(C_2, psk_{id_i}) \cdot \hat{e}(C_3, d')^{-1} \cdot \hat{e}(C_4, rk_2)^{-1}}$$

The malicious users can obtain the second level ciphertext $C = (C_1, C_2, C_3, C_4)$ of user id_i and obtain obtain the plaintext m as follows:

$$\begin{aligned}
&\frac{C_1}{\hat{e}(C_2, psk_{id_i}) \cdot \hat{e}(C_3, d')^{-1} \cdot \hat{e}(C_4, rk_2)^{-1}} \\
&= \frac{C_1}{\hat{e}(C_2, \eta^{\delta} \cdot H(id_i)^{u_j+\beta} \cdot g_2^{\alpha}) \cdot \hat{e}(C_3, d')^{-1} \cdot \hat{e}(C_4, rk_2)^{-1}} \\
&= \frac{m \cdot \hat{e}(g_1, g_2)^r}{\hat{e}(g_1, g_2)^r \cdot \hat{e}(C_4, rk_2) \cdot \hat{e}(d', C_3) \cdot \hat{e}(C_4, rk_2)^{-1} \cdot \hat{e}(d', C_3)^{-1}} \\
&= m.
\end{aligned}$$

Note that the private key of the delegatee (d_0, d_1) is not compromised and the second level encrypted message of user id_i is exposed to the malicious users violating the non-transferable property of Proxy Re-encryption.

6 A CCA-secure Non-transferable Scheme

6.1 Our Scheme

- *Setup*(λ): Let λ be the security parameter, $\mathbb{G}_1, \mathbb{G}_2$ are two groups of prime order q, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a bilinear map. Let P be a generator of the group $\mathbb{G}_1$ and randomly choose $Q \in \mathbb{G}_1$. Set $\alpha = \hat{e}(P, P)$. Choose five hash functions $\tilde{H} : \mathbb{G}_1 \leftarrow \mathbb{Z}_q^*, H_1 : \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{Z}_q^*, H_2 : \mathbb{G}_2 \to \{0,1\}^{l_m+l_\omega}, H_3 : \{0,1\}^{l_m+l_\omega} \to \mathbb{Z}_q^*, H_4 : \mathbb{G}_1 \times \mathbb{G}_1 \times \{0,1\}^{l_m+l_\omega} \times \mathbb{G}_1 \to \mathbb{G}_1$, where l_m, l_ω denote the message space $\mathcal{M}$. The hash functions are modelled as random oracles in the security proof. The global parameters are:

$$params := \{\mathbb{G}_1, \mathbb{G}_2, q, P, Q, \tilde{H}, H_1, H_2, H_3, H_4, \alpha\}$$

- *KeyGen*$(\lambda$,params): Pick $x_i, y_i, z_i \leftarrow \mathbb{Z}_q^*$, set the private key $sk_i = (x_i, y_i, z_i)$, public key $pk_i = (X_i, Y_i, Z_i, Q_i) = (x_iP, y_iP, z_iP, y_iQ)$ and set $h_i = H_1(pk_i)$.
- *ReKeyGen*$(sk_i, pk_i, pk_j$,params): Given as input the public key $pk_j = (X_i, Y_i, Z_i, Q_i)$ and private key $sk_i = (x_i, y_i, z_i)$ of user i and the public key $pk_j = (X_j, Y_j, Z_j, Q_j)$ of user j, pick $s, \delta, \beta \leftarrow \mathbb{Z}_q$ at random, and compute the re-encryption key as follows:

$$\begin{aligned}
T &= \frac{z_i + h_i}{\delta + \beta} \in \mathbb{Z}_q^*, \\
R &= {x_i}^{-1}(\delta Y_j + sP) + {x_i}^{-1}\tilde{H}(X_j)Q \\
&= {x_i}^{-1}(\delta y_j + s)P + {x_i}^{-1}\tilde{H}(X_j)Q \in \mathbb{G}_1, \\
S &= {y_i}^{-1}(\beta Y_j - sP) + {y_i}^{-1}Q_j \\
&= {y_i}^{-1}(\beta y_j - s)P + {y_i}^{-1}Q_j \in \mathbb{G}_1.
\end{aligned}$$

Return the re-encryption key $RK_{i\rightarrow j} = (R, S, T)$.

- *Encrypt*(m, pk_i):Given a message $m \in \mathcal{M}$ and a public key $pk_i = (X_i, Y_i, Z_i, Q_i)$ as input:
  - Choose $\omega \in_R \mathbb{Z}_q^*$.
  - Set $r = H_3(m, \omega) \in Z_q^*$.
  - Compute $C_1 = rX_i \in \mathbb{G}_1$.
  - Compute $C_2 = rY_i \in \mathbb{G}_1$.
  - Compute $C_3 = (m||\omega) \oplus H_2(\hat{e}(Z_i + h_i P, P)^r) = (m||\omega) \oplus H_2(\hat{e}(P, P)^{(z_i+h_i)r})$.
  - Compute $C_4 = r \cdot H_4(C_1, C_2, C_3, C_5) \in \mathbb{G}_1$.
  - Compute $C_5 = r \cdot Q \in \mathbb{G}_1$.

  The second level ciphertext $C = (C_1, C_2\ ,C_3\ ,C_4, C_5)$ is returned.
- *Re-Encrypt*$(C, RK_{i\rightarrow j})$: On input of a second level ciphertext $C = (C_1, C_2\ ,C_3\ ,C_4, C_5)$ and a re-key $RK_{i\rightarrow j} = (R, S, T)$, check the validity of C by testing if condition (1) and (2) holds:

$$\hat{e}(C_4, X_i) \overset{?}{=} \hat{e}(H_4(C_1, C_2, C_3, C_5), C_1) \tag{1}$$

$$\hat{e}(X_i + Y_i, C_5) \overset{?}{=} \hat{e}(C_1 + C_2, Q) \tag{2}$$

  If the above check fails, return *invalid*, else compute

$$D_1 = \Big[\frac{\hat{e}(C_1, R) \cdot \hat{e}(C_2, S)}{\hat{e}(\tilde{H}(X_j)P, C_5) \cdot \hat{e}(Y_j, C_5)}\Big]^T = \hat{e}(P, P)^{(z_i+h_i)ry_j} \in \mathbb{G}_2, \tag{3}$$

  Set $D_2 = C_3, D_3 = C_5$; return $D = (D_1, D_2, D_3)$ as the first level ciphertext.
- *Decrypt*(C, sk_i): Given as input the private key sk_i and second level ciphertext $C = (C_1, C_2\ ,C_3\ ,C_4, C_5)$, first check if conditions (1) and (2) hold. If they do not hold, return "invalid", else compute

$$(m||\omega) = H_2(\hat{e}((C_1 + C_2), \frac{1}{(x_i + y_i)}P)^{(z_i+h_i)}) \oplus C_3 \tag{4}$$

  Remark 1. Conditions (1) and (2) allow for the public verifiability of the ciphertext C. After conditions (1) and (2) are checked, recover $(m||\omega)$ and it suffices to verify any one of the conditions from (6) to (9) in $Verify(pk_i, (m||\omega), C)$.

  Remark 2. To avoid checking conditions (1) and (2) as it incurs heavy computation cost as indicated in Table 2 due to bilinear pairing, recover $(m||\omega)$, ensure if C is well-formed by checking if $Verify(pk_i, (m||\omega), C) = valid$ and return $(m||\omega)$, else return *invalid.*
- *Re-Decrypt*$(D, sk_j,)$: Given as input a private key sk_j and first level ciphertext $D = (D_1, D_2, D_3)$, compute

$$(m||\omega) = H_2(D_1^{y_j^{-1}}) \oplus D_2 \tag{5}$$

  Return $(m||\omega)$.
- *Verify*$((pk_i, m||\omega, C))$: Given as input a second level ciphertext $C = (C_1, C_2\ ,C_3\ ,C_4, C_5)$, a public key pk_i and a message $(m||\omega)$, compute $r = H_3(m||\omega)$ and check if the following conditions hold:

$$C_1 \overset{?}{=} r \cdot X_i \tag{6}$$

$$C_2 \overset{?}{=} r \cdot Y_i \tag{7}$$

$$C_4 \overset{?}{=} r \cdot H_4(C_1, C_2, C_3, C_5) \tag{8}$$

$$C_5 \overset{?}{=} r \cdot Q \tag{9}$$

  If all the conditions $(6) - (9)$ are satisfied, return *valid* else return *invalid.*

6.2 Security Proof

We prove the second level security under a variant of the m-DBDH assumption.

Lemma 1. *The variant of the modified decisional bilinear diffie-hellman (m-DBDH) assumption is said to hold if, given the elements* $(P, aP, a^{-1}P, a^{-2}P, bP, cP)$ *and* $T \in \mathbb{G}_2$*, there exists no probabilistic polynomial-time adversary which can determine whether* $T = \hat{e}(P,P)^{abc}$ *or a random element from* $\mathbb{G}_2$ *with a non-negligible advantage, where* P *is a generator of* $\mathbb{G}_1$ *and* $a, b, c \in_R \mathbb{Z}_q^*$.

Theorem 1. *Our proposed scheme is CCA-secure for the second level ciphertext under the variant of the m-DBDH assumption.*

Theorem 2. *Our proposed scheme is CCA-secure for the first level ciphertext under the 1-wDBDHI assumption.*

Remark 3. The proof of **Lemma 1**, **Theorem 1** and **Theorem 2** is shown in the full version of this paper [11].

Remark 4. The proposed scheme is non-transferable as the proxy and a set of colluding delegatees cannot re-delegate decryption rights to a third party. We can observe this from the following. In order to re-delegate decryption rights to an illegal user, the colluding delegatee will construct the decryption box $(D_1' \oplus C_3)$ by defining $D_1' \overset{\Delta}{=} D_1^{y_j^{-1}} = e(P,P)^{(z_i+h_i)r\cdot y_j \cdot y_j^{-1}} = e(P,P)^{(z_i+h_i)r}$. Given $C = (C_1, C_2, C_3, C_4, C_5)$, which is the second level ciphertext encrypted under the public key of the delegator, any malicious user can decrypt C by computing $D_1' \oplus C_3$. However, this re-delegation will only succeed when the delegatee sends his private key component y_j explicitly to the malicious user as y_j^{-1} must be used to exponentiate D_1 to compute D_1' and extract $(m||\omega)$. Since the value of D_1 changes in every delegation as a fresh random element $\omega \in \mathbb{Z}_q^*$ is used for every encryption, the value of $D^{y_j^{-1}}$ cannot be computed offline and hence must be explicitly provided by the delegatee to the malicious users. Hence, the delegatee must expose his private key for the illegal transference of decryption rights to a third party. Therefore, as per the definition in Section 4, non-transferable property is achieved in our scheme.

7 Comparison

We give a comparison of our scheme with the existing single-hop PRE schemes studied in the literature with respect to the non-transferable property. In Table 1, we show the various properties of a PRE scheme which are satisfied by the existing schemes alongside our scheme. In Table 2, we show the computational efficiency of a few well-known PRE schemes. Note that we use t to denote the time required for the various computations subscripted with bp, e, et, me, s, v to denote the time taken for a bilinear pairing, exponentiation in $\mathbb{G}_1$, exponentiation in $\mathbb{G}_2$, multi-exponentiation in group $\mathbb{G}_1$, signing algorithm and verification algorithm respectively. The comparisons show that our proposed design is the first scheme that achieves non-transferability with minimal efficiency loss and satisfies all the properties of an unidirectional single-hop PRE scheme.

Property	[13]	[7]	[6]	[5]	Our Scheme
Model	Random Oracle	Random Oracle	Standard	Standard	Random Oracle
Security	CCA	CCA	RCCA	CPA	CCA
Non-interactive	No	No	Yes	Yes	Yes
Proxy invisibility	Yes	Yes	Yes	Yes	Yes
Collusion-safe	Yes	Yes	Yes	Yes	Yes
Non-transitive	Yes	Yes	Yes	Yes	Yes
Non-transferable	No	Yes	No	Yes	Yes
Non-key escrow	Yes	No	Yes	Yes	Yes

Table 1. Comparative analysis of the properties of uni-directional single-hop PRE schemes studied in the literature and our scheme.

Scheme	Encrypt	Decrypt	Re-Encrypt	Re-Decrypt
[5]	$5t_e + 5t_{et} + 8t_{bp}$	$t_e + 6t_{et} + 4t_{bp}$	$t_e + t_{et} + t_{bp}$	$2t_e + 2t_{et} + 3t_{bp}$
[9]	$((n+2)t_e + t_{et})^*$	$t_e + t_{bp}$	$2t_{bp}$	t_{et}
[6]	$t_s + 4t_e + t_{et} + t_{bp} + t_{me}$	$t_e + t_{et} + 9t_{bp} + t_v$	$t_e + 8t_{bp} + t_v$	$t_e + 2t_{et} + 18t_{bp} + t_v$
[13]	$3t_e + t_{et} + t_{bp}$	$2t_{bp}$	$4t_{bp}$	$2t_{bp}$
Our Scheme	$5t_e + t_{et} + t_{bp}$	$5t_e + t_{et} + t_{bp}$ or $(2t_e + t_{et} + 5t_{bp})^{**}$	$t_{et} + 8t_{bp}$	t_{et}

Table 2. The Efficiency comparisons among unidirectional schemes in the literature with our scheme. * $O(n) = O(logN)$, where N is the maximum number of delegatees for each delegator in [9]. ** denotes the computation complexity for decrypt algorithm when conditions (1) and (2) are used for public verification along with any one of conditions (6) to (9) of the $Verify()$ algorithm.

8 Conclusion

Although there are several protocols achieving PRE in the literature, only two schemes [7] (ID-based settings) and [5] have reported the non-transferable property. To resolve the problem of non-transferability in PRE, [5] uses indistinguishability obfuscation ($i\mathcal{O}$), which involves very complex operations and is highly impractical. In [7], the IB-PRE protocol involves multiple rounds of interaction for partial-key generations and key validations which incurs computational overhead as indicated in the comparison Table 2. Our non-transferable PRE scheme is practical, based on direct manipulation in groups. Our scheme is shown to be CCA secure in the random oracle model for both the first and second level ciphertext and meets the non-transferability definition wherein the colluders (delegatee and proxy) cannot re-delegate the decryption rights of the delegator. An attempt to construct an illegal decryption box to decrypt the second level ciphertexts of the delegator reveals the private key components of the colluding delegatee. We have proposed an efficient non-transferable PRE scheme that affirmatively resolves the problem of illegal transference of decryption rights.

References

1. Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2005, San Diego, California, USA*, pages 29–43, 2005.
2. Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In *Advances in Cryptology - EUROCRYPT '98, International*

Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding, pages 127–144, 1998.

3. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 207–222, 2004.
4. Sherman S. M. Chow, Jian Weng, Yanjiang Yang, and Robert H. Deng. Efficient unidirectional proxy re-encryption. In *Progress in Cryptology - AFRICACRYPT 2010, Third International Conference on Cryptology in Africa, Stellenbosch, South Africa, May 3-6, 2010. Proceedings*, pages 316–332, 2010.
5. Hui Guo, Zhenfeng Zhang, and Jing Xu. Non-transferable proxy re-encryption. *IACR Cryptology ePrint Archive*, 2015:1216, 2015.
6. Ryotaro Hayashi, Tatsuyuki Matsushita, Takuya Yoshida, Yoshihiro Fujii, and Koji Okada. Unforgeability of re-encryption keys against collusion attack in proxy re-encryption. In *Advances in Information and Computer Security - 6th International Workshop, IWSEC 2011, Tokyo, Japan, November 8-10, 2011. Proceedings*, pages 210–229, 2011.
7. Yi Jun He, Tat Wing Chim, Lucas Chi Kwong Hui, and Siu-Ming Yiu. Non-transferable proxy re-encryption scheme. In *5th International Conference on New Technologies, Mobility and Security, Istanbul, Turkey, NTMS 2012, May 7-10, 2012*, pages 1–4, 2012.
8. Toshiyuki Isshiki, Manh Ha Nguyen, and Keisuke Tanaka. Attacks to the proxy re-encryption schemes from IWSEC2011. In *Advances in Information and Computer Security - 8th International Workshop on Security, IWSEC 2013, Okinawa, Japan, November 18-20, 2013, Proceedings*, pages 290–302, 2013.
9. Benoît Libert and Damien Vergnaud. Tracing malicious proxies in proxy re-encryption. In *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings*, pages 332–353, 2008.
10. Benoît Libert and Damien Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. *IEEE Trans. Information Theory*, 57(3):1786–1802, 2011.
11. S Sharmila Deva Selvi, Arinjita Paul, and C. Pandu Rangan. An efficient non-transferable proxy re-encryption scheme (full version). *Cryptology ePrint Archive*, May 2017.
12. S Sree Vivek, S Sharmila Deva Selvi, V Radhakishan, and C Pandu Rangan. Efficient conditional proxy re-encryption with chosen ciphertext security. *International Journal of Network Security & Its Applications*, 4(2):179–199, 2012.
13. Lihua Wang, Licheng Wang, Masahiro Mambo, and Eiji Okamoto. New identity-based proxy re-encryption schemes to prevent collusion attacks. In *Pairing-Based Cryptography - Pairing 2010 - 4th International Conference, Yamanaka Hot Spring, Japan, December 2010. Proceedings*, pages 327–346, 2010.
14. Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Attribute based data sharing with attribute revocation. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2010, Beijing, China, April 13-16, 2010*, pages 261–270, 2010.