A Type-and-Identity-based Proxy Re-Encryption Scheme and its Application in Healthcare

Luan Ibraimi[1], Qiang Tang[1], Pieter Hartel[1], Willem Jonker[1,2]

[1] Faculty of EWI, University of Twente, the Netherlands
[2] Philips Research, the Netherlands

Abstract. Proxy re-encryption is a cryptographic primitive developed to delegate the decryption right from one party (the delegator) to another (the delegatee). In a proxy re-encryption scheme, the delegator assigns a key to a proxy to re-encrypt all messages encrypted with his public key such that the re-encrypted ciphertexts can be decrypted with the delegatee's private key. We propose a type-and-identity-based proxy re-encryption scheme based on the Boneh-Franklin Identity Based Encryption (IBE) scheme. In our scheme, the delegator can categorize messages into different types and delegate the decryption right of each type to the delegatee through a proxy. Our scheme enables the delegator to provide the proxy fine-grained re-encryption capability. As an application, we propose a fine-grained Personal Health Record (PHR) disclosure scheme for healthcare service by applying the proposed scheme.

Keywords: Proxy re-encryption, Identity-Based Encryption, Personal Health Record

1 Introduction

Proxy re-encryption is a cryptographic method developed to delegate the decryption right from one party (the delegator) to another (the delegatee). In a proxy re-encryption scheme, the delegator assigns a key to a proxy to re-encrypt all messages encrypted with his public key such that the re-encrypted ciphertexts can be decrypted with the delegatee's private key. Since Mambo and Okamoto first proposed the concept [1], a number of proxy re-encryption schemes have been proposed [2,3,4,5,6]. Proxy re-encryption has many promising applications including access control in file storage [7], email forwarding [8], and law enforcement [3]. With the increasing privacy concerns over personal data, proxy re-encryption, in particular IBE proxy re-encryption schemes (due to their benefits [9]), will find more and more applications. For example, in the healthcare domain, many regulations, such as HIPPA [10], require that the patient is the owner of his personal health record and should control the disclosure policy for his Personal Health Record (PHR). As we show in Section 5, proxy re-encryption is a powerful tool for patient to enforce his PHR disclosure policies.

1.1 Motivations and contributions

An observation on the existing proxy re-encryption schemes is that the proxy is able to re-encrypt all ciphertexts from the delegator to the delegatee. As a result, it is difficult for the delegator to implement any further fine-grained cryptographically enforced access control policy for multiple delegation services. Suppose the delegator wants delegatees Bob and Charlie to recover different subsets of his messages. In this case, the delegator can only trust the proxy to enforce his policies by re-encrypting the legitimate ciphertexts. In practice, this trust assumption might be unrealistic (for example, the proxy can be corrupted). To solve this problem, an alternative solution would be that the delegator chooses a different key pair for each delegatee, which is also unrealistic.

Contribution We propose a type-and-identity-based proxy re-encryption scheme based on the Boneh-Franklin IBE scheme to enable the delegator to implement different access control policies for his ciphertexts against his delegatees. To achieve our goal, in the proposed scheme, the delegator can categorize his messages into different types, and delegate the decryption right of each type to the delegatee through a proxy. One benefit of our scheme is that the delegator only needs one key pair to provide fine-grained re-encryption capability to his proxy. In other words, the delegator only needs one key pair to provide fine-grained access control policies for his ciphertexts against his delegatees. The other benefit is that there is no further trust assumption on the proxy compared to existing proxy re-encryption schemes. However, the proposed scheme works only for the ciphertexts generated by the delegator. As an application, we propose a fine-grained PHR disclosure scheme for a healthcare service by applying the proposed scheme.

1.2 Organization

The rest of the paper is organized as follows. In Section 2 we introduce related work in proxy re-encryption. In Section 3 we briefly review the preliminaries of pairing and IBE. In Section 4 we present our new scheme which enables the delegator to offer fine-grained re-encryption capability to the proxy and prove its security. In Section 5 we propose a fine-grained PHR disclosure scheme as an application of our proxy re-encryption scheme. The last section concludes the paper.

2 Related work

Mambo and Okamoto [1] first propose the concept of delegation of decryption right in the context of speeding up decryption operations. Blaze *et al.* [2] introduce the concept of atomic proxy cryptography which is the current concept of proxy re-encryption. In a proxy re-encryption scheme, the proxy can transform ciphertexts encrypted with the delegator's public key into ciphertexts that can

be decrypted with the delegatee's private key. Blaze *et al.* propose a proxy re-encryption scheme based on the ElGamal encryption scheme [11]. One property of this scheme is that, with the same proxy key, the proxy can transform the ciphertexts not only form the delegator to the delegatee but also from the delegatee to the delegator. This is called the "bi-directional" property in the literature. Bi-directionality might be a problem in some applications, but it might also be a desirable property in some other applications. Jacobsson [4] addresses this "problem" using a quorum controlled asymmetric proxy re-encryption where the proxy is implemented with multiple servers and each of them performs partial re-encryption.

Dodis and Ivan [3] propose a generic construction method for proxy re-encryption schemes and also provide a number of example schemes. Their constructions are based on the concept of secret splitting, which means that the delegator splits his private key into two parts and sends them to the proxy and the delegatee separately. During the re-encryption process the proxy performs partial decryption of the encrypted message using the first part of the delegator's private key, and the delegatee can recover the message by performing partial decryption using the second part of the delegator's private key. One disadvantage of this method is that it is not collusion-safe, i.e. the proxy and the delegatee together can recover the delegator's private key. Another disadvantage of this scheme is that the delegatee's public/private key pair can only be used for dealing with the delegator's messages. If this key pair is used by the delegatee for other encryption services, then the delegator can always decrypt the ciphertexts.

Ateniese *et al.* [7] propose several proxy re-encryption schemes based on the ElGamal scheme. In their schemes, the delegator does not have to interact and share his private key with the delegatee. The delegator stores two secret keys, a master secret key and a "weak" secret key. The ciphertext can be fully decrypted using either of the two distinct keys. Their scheme is collusion safe, since only the "weak" secret key is exposed if the delegatee and the proxy collude but the master key remains safe. The disadvantage of this scheme is that the delegator has to perform two levels of encryptions, the first level encryption encrypts messages that can be decrypted by the delegator, and the second level encryption encrypts messages that can be decrypted by the delegator and his delegatees. In addition, Ateniese *et al.* also discuss a number of properties for proxy re-encryption schemes in [7].

The concept of IBE is proposed by Shamir [12]. Unlike a traditional public key encryption scheme, an IBE does not require a digital certificate to certify the public key because the public key of any user in an IBE can be an arbitrary string such as an email address, IP address, etc. IBE becomes practical and popular after Boneh and Fraklin [9] propose the first pairing-based scheme. Recently, two IBE proxy re-encryption schemes were proposed by Matsuo [6] and Green and Atteniese [5], respectively. The Matsuo scheme assumes that the delegator and the delegatee belong to the same Key Generation Center (KGC) and use the Boneh-Boyen encryption scheme [13]. The Green-Atteniese scheme assumes that

the delegator and the delegatee can belong to different KGCs but the delegatee posesses the public parameter of the delegator's KGC.

3 Preliminary

In this section we briefly review the pairing technique and the concept of IBE.

3.1 Review of pairing

We briefly review the basis of pairing and the related assumptions. More detailed information can be found in the seminal paper [9]. A pairing (or, bilinear map) satisfies the following properties:

1. $\mathbb{G}$ and $\mathbb{G}_1$ are two multiplicative groups of prime order p;
2. g is a generator of $\mathbb{G}$;
3. $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is an efficiently-computable bilinear map with the following properties:
   - Bilinear: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$, we have $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$.
   - Non-degenerate: $\hat{e}(g, g) \neq 1$.

As defined in [9], $\mathbb{G}$ is said to be a bilinear group if the group action in $\mathbb{G}$ can be computed efficiently and if there exists a group $\mathbb{G}_1$ and an efficiently-computable bilinear map $\hat{e}$ as defined above.

The Bilinear Diffie-Hellman (BDH) problem in $\mathbb{G}$ is as follows: given $g, g^a, g^b, g^c \in \mathbb{G}$ as input, output $\hat{e}(g, g)^{abc} \in \mathbb{G}_1$. An algorithm $\mathcal{A}$ has advantage ϵ in solving BDH in $\mathbb{G}$ if:

$$\Pr[\mathcal{A}(g, g^a, g^b, g^c) = \hat{e}(g, g)^{abc}] \geq \epsilon.$$

Similarly, we say that an algorithm $\mathcal{A}$ has advantage ϵ in solving the decision BDH problem in $\mathbb{G}$ if:

$$|\Pr[\mathcal{A}(g, g^a, g^b, g^c, g^{abc}) = 0] - \Pr[\mathcal{A}(g, g^a, g^b, g^c, T) = 0]| \geq \epsilon.$$

Here the probability is over the random choice of $a, b, c \in \mathbb{Z}_p^*$, the random choice of $T \in \mathbb{G}_1$, and the random bits of $\mathcal{A}$ (the adversary is a nondeterministic algorithm).

Definition 1. *We say that the (decision) (t, ϵ)-BDH assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ in solving the (decision) BDH problem in $\mathbb{G}$.*

As in the general group, the Computational Diffie-Hellman (CDH) problem in $\mathbb{G}$ is as follows: given $g, g^a, g^b \in \mathbb{G}$ as input, output $g^{ab} \in \mathbb{G}$. An algorithm $\mathcal{A}$ has advantage ϵ in solving CDH in $\mathbb{G}$ if:

$$\Pr[\mathcal{A}(g, g^a, g^b) = g^{ab}] \geq \epsilon.$$

Definition 2. *We say that the (t, ϵ)-CDH assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ in solving the CDH problem in $\mathbb{G}$.*

Given a security parameter k, a problem (say, BDH) is believed to be intractable if any adversary has only negligible advantage in reasonable time. We usually define a scheme to be secure if any adversary has only a negligible advantage in the underlying security model. The time parameter is usually be ignored.

Definition 3. *The function $P(k) : \mathbb{Z} \rightarrow \mathbb{R}$ is said to be negligible if, for every polynomial $f(k)$, there exists an integer N_f such that $P(k) \leq \frac{1}{f(k)}$ for all $k \geq N_f$.*

3.2 Review of Identity Based Encryption

We briefly review the Boneh-Fraklin scheme, which, compared with the original scheme [9], is slightly modified in the definition of the message domain and the encryption/decryption procedures (as we show below). Nonetheless, we still call it the Boneh-Franklin scheme.

1. $\mathsf{Setup}(k)$: Run by the KGC, given a security parameter k, the algorithm generates two cyclic groups $\mathbb{G}$ and $\mathbb{G}_1$ of prime order p, a generator g of $\mathbb{G}$, a bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$, a master secret key $\alpha \in \mathbb{Z}_p^*$, and a hash function $\mathsf{H}_1 : \{0,1\}^* \rightarrow \mathbb{G}$. The public parameter is $params = (\mathbb{G}, \mathbb{G}_1, p, g, \mathsf{H}_1, \hat{e}, pk)$, where $pk = g^\alpha$ is the public key of the KGC.
   In the original Boneh-Franklin scheme, the plaintext space is $\{0,1\}^n$ where n is an integer and there is an additional hash function $\mathsf{H}_2 : \mathbb{G}_1 \rightarrow \{0,1\}^n$.
2. $\mathsf{Extract}(id)$: Run by the KGC, given an identifier id, the algorithm outputs the private key $sk_{id} = pk_{id}^\alpha$, where $pk_{id} = \mathsf{H}_1(id)$.
3. $\mathsf{Encrypt}(m, id)$: Run by the message sender, given a message $m \in \mathbb{G}_1$ and an identifier $id \in \{0,1\}^*$ the algorithm outputs the ciphertext $c = (c_1, c_2)$ where $c_1 = g^r$, $c_2 = m \cdot \hat{e}(pk_{id}, pk)^r$, and $r \in \mathbb{Z}_p^*$.
   In the original Boneh-Franklin scheme, $c_2 = m \oplus \mathsf{H}_2(\hat{e}(pk_{id}, pk)^r)$.
4. $\mathsf{Decrypt}(c, sk_{id})$: Run by the receiver with identifier id, given a ciphertext $c = (c_1, c_2)$ and sk_{id}, the algorithm outputs the message $m = \frac{c_2}{\hat{e}(sk_{id}, c_1)}$.
   In the original Boneh-Franklin scheme, $m = c_2 \oplus \mathsf{H}_2(\hat{e}(sk_{id}, c_1))$.

The same modifications are also made in in [5] and they are essential for us to construct proxy re-encryption schemes. Implied by the security proof of the scheme IBP1 in [5], the Boneh-Franklin scheme is semantically secure against an adaptive chosen plaintext attack (IND-ID-CPA) based on the decision BDH assumption in the random oracle model. The IND-ID-CPA security is defined as follows.

The semantic security against an adaptive chosen ciphertext attack (IND-ID-CCA) is modelled by an IND-ID-CPA game. The game is carried out between a challenger and an adversary, where the challenger simulates the protocol execution and answers the queries from the adversary. Specifically, the game is as follows.

1. Game setup: The challenger takes a security parameter k and runs the Setup algorithm to generate the public system parameter $params$ and the master key mk.
2. Phase 1: The adversary takes $params$ as input and is allowed to issue two type of queries:
   (a) $\mathsf{Extract}$ query with any identifier id: The challenger returns the private key sk_{id} corresponding to id.
   (b) $\mathsf{Decrypt}$ query with any ciphertext c and any identifier id: The challenger runs $\mathsf{Extract}$ to generate the private key sk_{id} corresponding to id, and then returns the value of $\mathsf{Decrypt}(c, sk_{id})$.

   Once the adversary decides that Phase 1 is over, it outputs two equal length plaintexts m_0, m_1 and an identifier id^* on which it wishes to be challenged. The only constraint is that id^* has not been the input to any $\mathsf{Extract}$ query.
3. Challenge: The challenger picks a random bit $b \in \{0, 1\}$ and returns $c^* = \mathsf{Encrypt}(m_b, id^*)$ as the challenge to the adversary.
4. Phase 2: The adversary is allowed to continue issuing the same types of queries as in Phase 1. However, it is not allowed to ask a $\mathsf{Extract}$ query with the input id^* and a $\mathsf{Decrypt}$ query with the input (c^*, id^*).
5. Guess (game ending): the adversary outputs a guess $b' \in \{0, 1\}$.

Definition 4. *An IBE scheme is said to be semantically secure against an adaptive chosen ciphertext attack (IND-ID-CCA) if no polynomial-time adversary has a non-negligible advantage against the challenger in the IND-ID-CCA game, where the adversary's advantage is defined to be* $|\Pr[b' = b] - \frac{1}{2}|$.

Definition 5. *An IBE scheme is said to be semantically secure against an adaptive chosen plaintext attack (IND-ID-CPA) if any polynomial time IND-ID-CCA adversary's advantage is negligible when it makes no* $\mathsf{Decrypt}$ *query in the game.*

Apart from semantic security, we can also define the one-wayness for IBE. Formally, we have the following attack game.

1. Game setup: The challenger takes a security parameter k and runs the Setup algorithm to generate the public system parameter $params$ and the master key mk.
2. Extraction: The adversary takes $params$ as input and is allowed to issue any number of $\mathsf{Extract}$ query with any identifier id: The challenger returns the private key sk_{id} corresponding to id. Once the adversary decides that this phase is over, it outputs an identifier id^* on which it wishes to be challenged. The only constraint is that id^* has not been the input to any $\mathsf{Extract}$ query.
3. Challenge: The challenger picks a random message m and returns $c^* = \mathsf{Encrypt}(m, id^*)$ as the challenge to the adversary.
4. Guess (game ending): the adversary outputs a guess m'.

Definition 6. *An IBE scheme is said to be one-way if any polynomial time adversary's advantage is negligible in the above game, where the adversary's advantage is defined to be* $\Pr[m' = m]$.

4 A type-and-identity-based proxy re-encryption scheme

In this section we propose a type-and-identity-based proxy re-encryption scheme based on the Boneh-Franklin scheme described in Section 3.2. In our scheme, the delegator and the delegatee are allowed to be from different domains, which nonetheless share some public parameters.

- Suppose that the delegator is registered at KGC_1 in a modified Boneh-Franklin IBE scheme $(\mathsf{Setup}_1, \mathsf{Extract}_1, \mathsf{Encrypt}_1, \mathsf{Decrypt}_1)$. Users categorize their messages into different types, say $\{t \in \{0,1\}^*\}$; the IBE algorithms are defined as follows.
  - Setup_1 and $\mathsf{Extract}_1$ are the same as in the Boneh-Franklin scheme, except that Setup_1 outputs an additional hash function $\mathsf{H}_2 : \{0,1\}^* \to \mathbb{Z}_p^*$. The public parameter is $params_1 = (\mathbb{G}, \mathbb{G}_1, p, g, \mathsf{H}_1, \mathsf{H}_2, \hat{e}, pk_1)$, and the master key is $mk_1 = \alpha_1$.
  - $\mathsf{Encrypt}_1(m, t, id)$: Given a message m, a type t, and an identifier id, the algorithm outputs the ciphertext $c = (c_1, c_2, c_3)$ where $r \in_R \mathbb{Z}_p^*$,
  $$c_1 = g^r, \;\; c_2 = m \cdot \hat{e}(pk_{id}, pk)^{r \cdot \mathsf{H}_2(sk_{id}||t)}, \;\; c_3 = t.$$
  - $\mathsf{Decrypt}_1(c, sk_{id})$: Given a ciphertext $c = (c_1, c_2, c_3)$, the algorithm outputs the message
  $$m = \frac{c_2}{\hat{e}(sk_{id}, c_1)^{\mathsf{H}_2(sk_{id}||c_3)}}$$

  Without loss of generality, suppose the delegator holds the identity id_i and the corresponding private key sk_{id_i}. Apart from the delegator, another party cannot run the $\mathsf{Encrypt}_1$ algorithm under the delegator's identity id_i since he does not know sk_{id_i}.
- Suppose that the delegatee (with identity id_j) possesses private key sk_{id_j} registered at KGC_2 in the Boneh-Franklin IBE scheme, where the public parameter is $params_2 = (\mathbb{G}, \mathbb{G}_1, p, g, \mathsf{H}_1, \hat{e}, pk_2)$, the master key is $mk_2 = \alpha_2$, and $sk_{id_j} = \mathsf{H}_1(id_j)^{\alpha_2}$. For the ease of comparison, we denote the IBE scheme as $(\mathsf{Setup}_2, \mathsf{Extract}_2, \mathsf{Encrypt}_2, \mathsf{Decrypt}_2)$ although these algorithms are identical to those described in Section 3.2.

4.1 The delegation process

If the delegator wants to delegate his decryption right for messages with type t to the delegatee, the algorithms of the proxy re-encryption scheme are as follows.

- $\mathsf{Pextract}(id_i, id_j, t, sk_{id_i})$: Run by the delegator, this algorithm takes the delegator's identifier id_i, the delegatee's identifier id_j, the type t, and the delegator's private key sk_{id_i} as input and outputs the proxy key $rk_{id_i \to id_j}$, where $X \in_R \mathbb{G}_1$ and
$$rk_{id_i \to id_j} = (t, sk_{id_i}^{-\mathsf{H}_2(sk_{id_i}||t)} \cdot \mathsf{H}_1(X), \mathsf{Encrypt}_2(X, id_j)).$$

– $\mathsf{Preenc}(c_i, rk_{id_i \to id_j})$: Run by the proxy, this algorithm, takes a ciphertext $c_i = (c_{i1}, c_{i2}, c_{i3})$ and the proxy key $rk_{id_i \to id_j}$ as input where $t = c_{i3}$, and outputs a new ciphertext $c_j = (c_{j1}, c_{j2}, c_{j3})$, where $c_{j1} = c_{i1}$ and

$$\begin{aligned} c_{j2} &= c_{i2} \cdot \hat{e}(c_{i1}, sk_{id_i}^{-\mathsf{H}_2(sk_{id_i}||c_{i3})} \cdot \mathsf{H}_1(X)) \\ &= m \cdot \hat{e}(g^{\alpha_1}, pk_{id_i}^{r\mathsf{H}_2(sk_{id_i}||t)}) \cdot \hat{e}(g^r, sk_{id_i}^{-\mathsf{H}_2(sk_{id_i}||t)} \cdot \mathsf{H}_1(X)) \\ &= m \cdot \hat{e}(g^r, \mathsf{H}_1(X)), \end{aligned}$$

and $c_{j3} = \mathsf{Encrypt}_2(X, id_j)$.

Given a re-encrypted ciphertext c_j, the delegatee can obtain the plaintext m by computing

$$\begin{aligned} m' &= \frac{c_{j2}}{\hat{e}(c_{j1}, \mathsf{H}_1(\mathsf{Decrypt}_2(c_{j3}, sk_{id_j})))} \\ &= \frac{m \cdot \hat{e}(g^r, \mathsf{H}_1(X))}{\hat{e}(g^r, \mathsf{H}_1(X))} \\ &= m. \end{aligned}$$

4.2 Threat model

We assume that both KGC_1 and KGC_2 are semi-trusted in the following sense: they will behave honestly all the time except that they might be curious about the plaintexts for either the delegator or the delegatee; in addition, they are passive attackers. As mentioned in [14], the key escrow problem of IBE can be avoided by applying some standard techniques (such as secret sharing) to the underlying scheme, hence, we skip any further discussion in this paper. The proxy is assumed to be semi-trusted in the following sense: it will honestly convert the delegator's ciphertexts using the proxy key; however, it might act actively to obtain some information about the plaintexts for the delegator and the delegatee. The delegatee may be curious in the sense that it may try to obtain some information about the plaintexts corresponding to the delegator's ciphertexts which have not been re-encrypted by the proxy.

As a standard practice, we describe an attack game for modeling the semantic security against an adaptive chosen plaintext attack for the delegator (IND-ID-DR-CPA security) for our scheme. The IND-ID-DR-CPA game is carried out between a challenger and an adversary, where the challenger simulates the protocol execution and answers the queries from the adversary. Note that the allowed queries for the adversary reflect the adversary's capability in practice. Specifically, the game is as follows.

1. Game setup: The challenger takes a security parameter k as input, runs the Setup_1 algorithm to generate the public system parameter $params_1$ and the master key mk_1, and runs the Setup_2 algorithm to generate the public system parameter $params_2$ and the master key mk_2.

2. Phase 1: The adversary takes $params_1$ and $params_2$ as input and is allowed to issue the following types of queries:
   (a) $\mathsf{Extract}_1$ query with any identifier id: The challenger returns the private key sk corresponding to id.
   (b) $\mathsf{Extract}_2$ query with any identifier id': The challenger returns the private key sk' corresponding to id'.
   (c) $\mathsf{Pextract}$ query with (id, id', t): The challenger returns the proxy key $rk_{id \to id'}$ for the type t.
   (d) $\mathsf{Preenc}^{\dagger}$ query with (m, t, id, id'): The challenger first computes $c = \mathsf{Encrypt}_1(m, t, id)$ and then returns a new ciphertext c' which is obtained by applying the delegation key $rk_{id \to id'}$ to c, where $rk_{id \to id'}$ is issued for type t.

   Once the adversary decides that Phase 1 is over, it outputs two equal length plaintexts m_0, m_1, a type t^*, and an identifier id^*. At the end of Phase 1, there are three constraints here:
   (a) id^* has not been the input to any $\mathsf{Extract}_1$ query.
   (b) For any id', if (id^*, id', t^*) has been the input to a $\mathsf{Pextract}$ query then id' has not been the input to any $\mathsf{Extract}_2$ query.
   (c) If there is a $\mathsf{Preenc}^{\dagger}$ query with (m, t, id, id'), then (id, id', t) has not been queried to $\mathsf{Pextract}$.
3. Challenge: The challenger picks a random bit $b \in \{0, 1\}$ and returns $c^* = \mathsf{Encrypt}_1(m_b, t^*, id^*)$ as the challenge to the adversary.
4. Phase 2: The adversary is allowed to continue issuing the same types of queries as in Phase 1. At the end of Phase 2, there are the same constraints At the end of Phase 1.
5. Guess (game ending): the adversary outputs a guess $b' \in \{0, 1\}$.

At the end of the game, the adversary's advantage is defined to be $|\Pr[b' = b] - \frac{1}{2}|$. Compared with the CPA security formalizations in [5,6], in our case, we also take into account the categorization of messages for the delegator. The $\mathsf{Preenc}^{\dagger}$ query reflects the fact that a curious delegatee has access to the the delegator's plaintexts.

4.3 Security analysis of our scheme

We first briefly prove the IND-ID-DR-CPA security of our scheme and then show some other security properties.

Theorem 1. *For the type-and-identity-based proxy re-encryption scheme described in Section 4.1, any adversary's advantage is negligible.*

Proof sketch. We suppose that the total number of queries issued to H_1 and H_2 is bounded by integer q_1 and q_2, respectively[3]. Suppose an adversary $\mathcal{A}$ has the

[3] For simplicity of description, it is reasonable to assume that the total number is counted for queries with different inputs.

non-negligible advantage ϵ in the IND-ID-DR-CPA game. The security proof is done through a sequence of games.

Game_0: In this game, $\mathcal{B}$ faithfully answers the oracle queries from $\mathcal{A}$. Specifically, $\mathcal{B}$ simulates the random oracle H_1 as follows: $\mathcal{B}$ maintains a list of vectors, each of them containing a request message, an element of $\mathbb{G}$ (the hash-code for this message), and an element of $\mathbb{Z}_p^*$. After receiving a request message, $\mathcal{B}$ first checks its list to see whether the request message is already in the list. If the check succeeds, $\mathcal{B}$ returns the stored element of $\mathbb{G}$; otherwise, $\mathcal{B}$ returns g^y, where y a randomly chosen element of $\mathbb{Z}_p^*$, and stores the new vector in the list. $\mathcal{A}'$ simulates the random oracle H_2 as follows: $\mathcal{B}$ maintains a list of vectors, each of them containing a request message and an element of $\mathbb{Z}_p^*$ (the hash-code for this message). After receiving a request message, $\mathcal{B}$ first checks its list to see whether the request message is already in the list. If the check succeeds, $\mathcal{B}$ returns the stored element of $\mathbb{Z}_p^*$; otherwise, $\mathcal{B}$ returns u which is a randomly chosen element of $\mathbb{Z}_p^*$, and stores the new vector in the list.

Let $\delta_0 = \Pr[b' = b]$, as we assumed at the beginning, $|\delta_0 - \frac{1}{2}| = \epsilon$.

Game_1: In this game, $\mathcal{B}$ answers the oracle queries from $\mathcal{A}$ as follows.

1. Game setup: $\mathcal{B}$ faithfully simulates the setup phase.
2. Phase 1: $\mathcal{B}$ randomly selects $j \in \{1, 2, \cdots, q_1 + 1\}$. If $j = q_1 + 1$, $\mathcal{B}$ faithfully answers the oracle queries from $\mathcal{A}$. If $1 \leq j \leq q_1$, we assume the j-th input to H_1 is $\tilde{id}$ and $\mathcal{B}$ answers the oracle queries from $\mathcal{A}$ as follows: Answer the queries to $\mathsf{Extract}_1$, $\mathsf{Extract}_2$, $\mathsf{Pextract}$, and $\mathsf{Preenc}^\dagger$ faithfully, except that $\mathcal{B}$ aborts as a failure when $\tilde{id}$ is the input to a $\mathsf{Extract}_1$ query.
3. Challenge: After receiving (m_0, m_1, t^*, id^*) from the adversary, if one of the following events occurs, $\mathcal{B}$ aborts as a failure.
   (a) id^* has been issued to H_1 as the i-th query and $i \neq j$,
   (b) id^* has not been issued to H_1 and $1 \leq j \leq q_1$.
   Note that, if the adversary does not abort then either $1 \leq j \leq q_1$ and $id^* = \tilde{id}$ is the input to j-th H_1 query or $j = q_1 + 1$ and id^* has not been the input to any H_1 query. $\mathcal{B}$ faithfully returns the challenge.
4. Phase 2: $\mathcal{B}$ answers the oracle queries faithfully.
5. Guess (game ending): the adversary outputs a guess $b' \in \{0, 1\}$.

The probability that $\mathcal{B}$ successfully ends is $\frac{1}{q_1+1}$, i.e. the probability that $\mathcal{B}$ does not abort in its execution is $\frac{1}{q_1+1}$. Let $\delta_1 = \Pr[b' = b]$ when $\mathcal{B}$ successfully ends, in which case $|\delta_1 = \delta_0|$. Let θ_1 be the probability that $\mathcal{B}$ successfully ends and $b' = b$. We have $\theta_1 = \frac{\delta_1}{q_1+1}$.

Game_2: In this game, $\mathcal{B}$ simulates the protocol execution and answers the oracle queries from $\mathcal{A}$ in the following way.

1. Game setup: $\mathcal{B}$ faithfully simulates the setup phase. Recall that $pk_1 = g^{\alpha_1}$.
2. Phase 1: $\mathcal{B}$ randomly selects $j \in \{1, 2, \cdots, q_1 + 1\}$. If $j = q_1 + 1$, $\mathcal{B}$ faithfully answers the oracle queries from $\mathcal{A}$. If $1 \leq j \leq q_1$, $\mathcal{B}$ answers j-th query to H_1 with g^β where $\beta \in_R \mathbb{Z}_p^*$, and answers the oracle queries from $\mathcal{A}$ as follows. Suppose the input of the j-th query to H_1 is $\tilde{id}$.

(a) Answer $\mathsf{Extract}_1$ and $\mathsf{Extract}_2$ faithfully, except that $\mathcal{B}$ aborts as a failure when $\tilde{id}$ is the input to a $\mathsf{Extract}_1$ query.
(b) $\mathsf{Pextract}$ query with (id, id', t): If $id = \tilde{id}$, $\mathcal{B}$ returns the proxy key $rk_{id \to id'}$, where

$$g_{t \sim id'} \in_R \mathbb{G},\ X_{t \sim id'} \in_R \mathbb{G}_1,\ rk_{id \to id'} = (t, g_{t \sim id'}, \mathsf{Encrypt}_2(X_{t \sim id'}, id')).$$

Otherwise, $\mathcal{B}$ answers the query faithfully. If id' has been queried to $\mathsf{Extract}_2$, when $X_{t \sim id'}$ is queried to H_1 then $\mathcal{B}$ returns $g_{t \sim id'} \cdot h_{t \sim id'}^{-1}$ where $h_{t \sim id'} \in_R \mathbb{G}$.
(c) $\mathsf{Preenc}^{\dagger}$ query with (m, t, id, id'): If $id = \tilde{id}$, $\mathcal{B}$ returns

$$r \in_R \mathbb{Z}_p^*, X_{t \sim id'} \in_R \mathbb{G}_1, c' = (g^r, \hat{e}(g^r, \mathsf{H}_1(X_{t \sim id'})), \mathsf{Encrypt}_2(X_{t \sim id'}, id')).$$

Otherwise, $\mathcal{B}$ answers the query faithfully.

3. Challenge: After receiving (m_0, m_1, t^*, id^*) from the adversary, if one of the following events occurs, $\mathcal{B}$ aborts as a failure.
(a) id^* has been issued to H_1 as the i-th query and $i \neq j$,
(b) id^* has not been issued to H_1 and $1 \leq j \leq q_1$.
Note that, if the adversary does not abort then either $1 \leq j \leq q_1$ and $id^* = \tilde{id}$ is the input to j-th H_1 query or $j = q_1 + 1$ and id^* has not been the input to any H_1 query. In the latter case, $\mathcal{B}$ sets $\mathsf{H}_1(id^*) = g^{\beta}$ where $\beta \in_R \mathbb{Z}_p^*$, and returns $c^* = (c_1^*, c_2^*, c_3^*)$ as the challenge to the adversary, where:

$$b \in_R \{0, 1\},\ r \in_R \mathbb{Z}_p^*,\ T \in_R \mathbb{G}_1,\ c_1^* = g^r,\ c_2^* = m_b \cdot T,\ c_3^* = t^*.$$

4. Phase 2: $\mathcal{B}$ answers the oracle queries from $\mathcal{A}$ as in Phase 1.
5. Guess (game ending): the adversary outputs a guess $b' \in \{0, 1\}$.

Let θ_2 be the probability that $\mathcal{B}$ successfully ends and $b' = b$. We have $\theta_2 = \frac{1}{2(q_1+1)}$ since $T \in_R \mathbb{G}_1$. Let E_1 be the event that, for some id' and t, the adversary issues a H_2 query with the input $g^{\alpha_1 \cdot \beta} || t$ or $X_{t \sim id'}$ is issued to H_1 while id' has not been issued to $\mathsf{Extract}_2$. Compared with Game_1, Game_2 differs when E_1 occurs. From the difference lemma [15], we have $|\delta_2 - \delta_1| \leq \epsilon_2$ which is negligible in the random oracle model based on the BDH assumption. Note that $(\mathsf{Setup}_2, \mathsf{Extract}_2, \mathsf{Encrypt}_2, \mathsf{Decrypt}_2)$ is one-way based on the BDH assumption and BDH implies CDH.

From $|\theta_2 - \theta_1| \leq \epsilon_2$ and $\theta_2 = \frac{1}{2(q_1+1)}$, we have $|\frac{1}{2(q_1+1)} - \theta_1| \leq \epsilon_2$. In addition, from $|\delta_0 - \frac{1}{2}| = \epsilon$, $|\delta_1 - \delta_0| \leq \epsilon_1$ and $\theta_1 = \frac{\delta_1}{q_1+1}$, we have $\frac{\epsilon}{q_1+1} \leq \frac{\epsilon_1}{q_1+1} + \epsilon_2$. Because ϵ_i $(1 \leq i \leq 2)$ are negligible and ϵ is assumed to be non-negligible, we get a contradiction. As a result, the proposed scheme is IND-ID-DR-CPA secure based on the CDH assumption in the random oracle model, given that $(\mathsf{Setup}_2, \mathsf{Extract}_2, \mathsf{Encrypt}_2, \mathsf{Decrypt}_2)$ is one-way. □

Recall that Ateniese *et al.* describe a number of properties for proxy re-encryption schemes [7]. Our scheme possesses the following properties:

– Uni-directional. In our scheme, the delegation key is generated by the delegator, hence it is clear that the delegation is only from the delegator to the delegatee but not from the delegatee to the delegator.

– Non-Interactive. In our scheme, the delegator creates the re-encryption key by himself, neither the delegatee nor any other party is involved.
– Collusion Safe. In our scheme, the delegatee and the proxy together can recover the private key for the type t if the delegator wants to delegate his decryption right for t to the delegatee. We cannot see any damage here since the delegatee is allowed to see the messages encrypted under this key. Apart from this, the delegatee and the proxy together cannot recover the delegator's private key sk_{id_i}; in particular, they cannot recover any key for other message types from Theorem 1.

5 Fine-grained PHR disclosure

As mentioned in [16], a Personal Health Record (PHR) contains all kinds of health-related information about an individual (say, Alice). Firstly, the PHR contains medical data from various medical service providers, for example about surgery, illness, family history, vaccinations, laboratory test results, allergies, drug reactions, etc. Secondly, the PHR may also contain information collected by Alice herself, for example weight change, food statistics, and any other information connected with her health. The PHR is helpful for Alice to obtain health care services and monitor her health status, however, a PHR is sensitive information. Inappropriate disclosure of the PHR may cause an individual serious problems. For example, if Alice has some disease and a prospective employer obtains this, then she might be discriminated in finding a job. Alice needs to protect her PHR. It is worth stressing that PHR data may have different levels of privacy concerns. For example, Alice may not be seriously concerned about disclosing her food statistics to other persons, but she might wish to keep her illness history as a top secret and only disclose it to the appropriate person.

There are some possible solutions to guarantee the privacy of Alice's PHR. In one solution, Alice could make her own access control policies for her PHR, store her PHR in plaintext in a database, and rely on this database to enforce her policies. In this case, Alice needs to trust the database fully. Once the database is corrupted all Alice's PHR will be disclosed. As an alternative, Alice could encrypt her PHR and store the ciphertext in a database, and then decrypt the ciphertext on demand. In this case, Alice only needs to assume that the database will properly store her encrypt data, and even if the database is corrupted Alice's PHR will not be disclosed. The problem with this solution is that Alice needs to be involved in every request and perform the decryption. Yet another solution is to use a traditional proxy re-encryption scheme, in which Alice assigns a re-encryption key to the database which re-encrypts the encrypted PHR into encrypted PHR with the requester's public key. In this case, Alice must assume that the database will properly store her encrypt data and that the database performs the re-encryption. If the database is corrupted, some of Alice's PHR may be disclosed to an illegitimate entity based on the fact that the proxy key can re-encrypt all Alice's encrypted PHR. To avoid this problem, Alice needs to have as many key pairs as there are categories of her PHR data.

Using our type-and-identity-based proxy re-encryption scheme, we can construct a fine-grained PHR disclosure scheme for Alice as follows:

1. Alice categorizes her PHR according to her privacy concerns. For instance, she can set her illness history as type t_1, her food statistics as type t_2, and the necessary PHR data in case of emergency as type t_3.
2. For each type of PHR, Alice finds a proxy and stores each type of her PHR in encrypted form using our scheme, and assigns a re-encryption key to the proxy. In practice, this could be a dynamic process. For example, if Alice wishes to travel to the US, then she can find a proxy there and store her encrypted PHR data for emergency case (type t_3) there. Then if Alice needs emergency help in the US, the PHR data can be disclosed on demand by the proxy.

In this solution Alice only needs one key pair to protect her PHR data and can choose the proxy for each category of her PHR data according to her trust and privacy concerns. Since Alice chooses a different proxy for every type of PHR, even if the proxies for certain types of PHR are corrupted, other types of PHR cannot be illegitimately disclosed from Theorem 1.

6 Conclusion

In this paper we propose a type-and-identity-based proxy re-encryption scheme based on the Boneh-Franklin scheme which has been proved semantically secure against a chosen plaintext attack. Our scheme enables the delegator to provide different re-encryption capabilities to the proxy while using the same key pair. This property is showed to be useful in our PHR disclosure scheme, where an individual can easily implement fine-grained access control policies to his PHR data. For future work, it would be interesting to construct type-and-identity-based proxy re-encryption schemes with chosen ciphertext security and to investigate new applications for this primitive.

References

1. M. Mambo and E. Okamoto. Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 80(1):54–63, 1997.
2. M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In K. Nyberg, editor, *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques*, volume 1403 of *Lecture Notes in Computer Science*, pages 127–144. Springer, 1998.
3. A. Ivan and Y. Dodis. Proxy cryptography revisited. In *Proceedings of the Network and Distributed System Security Symposium*. The Internet Society, 2003.
4. Markus Jakobsson. On quorum controlled asymmetric proxy re-encryption. In H. Imai and Y. Zheng, editors, *Public Key Cryptography, Second International Workshop on Practice and Theory in Public Key Cryptography*, volume 1560 of *Lecture Notes in Computer Science*, pages 112–121. Springer, 1999.

5. M. Green and G. Ateniese. Identity-based proxy re-encryption. In J. Katz and M. Yung, editors, *Applied Cryptography and Network Security, 5th International Conference*, volume 4521 of *Lecture Notes in Computer Science*, pages 288–306. Springer, 2007.
6. T. Matsuo. Proxy re-encryption systems for identity-based encryption. In T. Takagi, T. Okamoto, E. Okamoto, and T. Okamoto, editors, *Pairing-Based Cryptography - Pairing 2007, First International Conference*, volume 4575 of *Lecture Notes in Computer Science*, pages 247–267. Springer, 2007.
7. G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):1–30, 2006.
8. L. Wang, Z. Cao, T. Okamoto, Y. Miao, and E. Okamoto. Authorization-Limited Transformation-Free Proxy Cryptosystems and Their Security Analyses*. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, (1):106–114, 2006.
9. D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In J. Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.
10. The US Department of Health and Human Services. Summary of the HIPAA Privacy Rule, 2003.
11. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology, Proceedings of CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 1985.
12. A. Shamir. Identity-based cryptosystems and signature schemes. *Proceedings of CRYPTO 84 on Advances in cryptology table of contents*, pages 47–53, 1985.
13. D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.
14. L. Chen. An interpretation of identity-based cryptography. In A. Aldini and R. Gorrieri, editors, *Foundations of Security Analysis and Design IV, FOSAD 2006/2007 Tutorial Lectures*, volume 4677 of *Lecture Notes in Computer Science*, pages 183–208. Springer, 2007.
15. V. Shoup. Sequences of games: a tool for taming complexity in security proofs. http://shoup.net/papers/, 2006.
16. P.C. Tang, J.S. Ash, D.W. Bates, J.M. Overhage, and D.Z. Sands. Personal Health Records: Definitions, Benefits, and Strategies for Overcoming Barriers to Adoption. *Journal of the American Medical Informatics Association*, 13(2):121–126, 2006.