A Timed-Release Proxy Re-Encryption Scheme and its Application to Fairly-Opened Multicast Communication

Keita Emura[1], Atsuko Miyaji[2], and Kazumasa Omote[2]

[1] Center for Highly Dependable Embedded Systems Technology
[2] School of Information Science
Japan Advanced Institute of Science and Technology, 1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan
{k-emura, miyaji, omote}@jaist.ac.jp

Abstract. Timed-Release Encryption (TRE) (proposed by May in 1993) prevents even a legitimate recipient decrypting a ciphertext before a semi-trusted Time Server (TS) sends trapdoor s_T assigned with a release time T of the encryptor's choice. Cathalo et al. (ICICS2005) and Chalkias et al. (ESORICS2007) have already considered encrypting a message intended to multiple recipients with the same release time. These schemes are efficient compared with previous TRE schemes with recipient-to-recipient encryption, since the most costly part (especially pairing computation) has only to be computed once, and this element is used commonly. One drawback of these schemes is the ciphertext size and computational complexity, which depend on the number of recipients N. In this paper, for the first time we propose Timed-Release Proxy Re-Encryption (TR-PRE) scheme. As in PRE, a semi-trusted proxy transforms a ciphertext under a particular public key (this can be regarded as a mailing list) into re-encrypted ciphertexts under each recipient (who can be regarded as mailing list members). *Even if the proxy transformation is applied to a TRE ciphertext, the release time is still effective.* An encryptor can transfer N-dependent computation parts to the proxy. This function can be applied to multicast communication with a release time indication. For example, in an on-line examination, an examiner sends encrypted e-mails to each examinee, and each examination can be fairly opened at the same time. Our TR-PRE scheme is provably secure under both chosen-time period chosen-ciphertext attack (IND-CTCA) and replayable chosen-ciphertext attack (IND-RCCA) without random oracles.

1 Introduction

Timed-Release Encryption (TRE) was proposed by May [26], where even a legitimate recipient cannot decrypt a ciphertext before a semi-trusted Time Server (TS) sends (or broadcasts) trapdoor s_T assigned with release time T of the encryptor's choice. Several kinds of TRE schemes have been proposed, e.g.,

TRE with Pre-open Capability (TRE-PC) [15, 16, 21], some generic constrictions [12–14, 28], and so on. Points worthy of special mention are Cathalo et al. [10] and Chalkias et al. [11][3], who have already considered encrypting a message intended to several recipients with the same release time. As in Cathalo et al. [10], we do not consider encrypting with distinct release times, since colluding receivers could decrypt the message without having the appropriate trapdoor. Schemes by Cathalo et al. and Chalkias et al. are efficient compared with previous TRE schemes with recipient-to-recipient encryption, since the most costly part (especially pairing computation $e(\cdot, \cdot)$) has only to be computed once, and this element is used commonly. Informally, for a common release time T and number of recipients N, the form of a ciphertext in the Cathalo et al. scheme is: $(C_1, C_2, \ldots, C_N, (M||\text{random nonce}) \oplus K), \mathit{RecipientList}, T)$, where $K = Hash(e(\cdot, \cdot))$ is a commonly used ephemeral key computed by both C_i and a user U_i's secret key. One drawback of this scheme is the ciphertext size, namely, the length of the ciphertext depends on the number of recipients N (See Fig.1). If each ciphertext (for a user U_i) is represented as $(C_i, (M||\text{random nonce}) \oplus K, T)$, then actual transferred ciphertext size is constant. Nevertheless, there is still a remaining problem, where computational complexity also depends on N. This can be a serious problem when N becomes large.

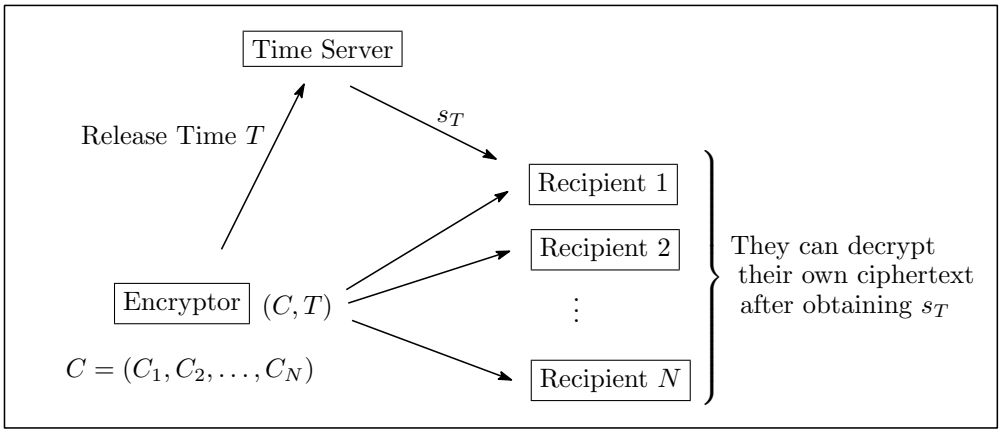

Fig. 1: Previous TRE for Multiple Recipients

Proxy Re-Encryption (PRE) was proposed in [4] by Blaze, Bleumer, and Stauss, where a semi-trusted proxy transforms a ciphertext encrypted by a delegator Alice's public key into a re-encrypted ciphertext that can be decrypted using a delegatee Bob's secret key. As application of PRE schemes, e-mail systems based on PRE have been proposed, such as [5, 22–24].

In cloud computing environments [30], users do not have to consider the actual data storage of some services, and therefore data management becomes

[3] Note that Chow et al. [15] showed that the Chalkias et al. scheme is vulnerable under the CCA attack.

more and more difficult. Usually, *access control of data* and *encryption of data* are different technologies. Therefore, TRE (attribute-based encryption [3, 20, 27] is also another example) is suitable in cloud computing environments, since the access control function is included in the encrypted data itself. In PRE, access control (namely, who has decryption rights) may be complicated and difficult to manage, when the number of users becomes large.

Our contribution : In this paper, for the first time we propose a Timed-Release Proxy Re-Encryption (TR-PRE) scheme. An encryptor computes a TRE ciphertext under a particular public key (this can be regarded as a mailing list), and a proxy translates this ciphertext into re-encrypted ciphertexts under each recipient (who can be regarded as mailing list members). *Even if the proxy transformation is applied to a TRE ciphertext, the release time is still effective.* To the best of our knowledge, no TR-PRE have considered this release time property, although there are several research papers on TRE and PRE. We illustrate our TR-PRE in Fig.2. In this situation, from the viewpoint of an encryptor, the number of ciphertexts (and computational complexity also) does not depend on the number of recipients N. In other words, the encryptor can transfer these N-dependent parts to the proxy. As in PRE, the proxy cannot decrypt ciphertexts.

Our TR-PRE can be applied to multicast communication with a release time indication. For example, in an on-line examination, an examiner sends encrypted e-mails to each examinee, and each examination can be opened at the same time. As another application, by using our scheme as a building tool for e-mail systems based on PRE [5, 22–24], we can achieve e-mail systems with release time indication. Our work is valuable in adding an access control function into encrypted (and re-encrypted) data itself.

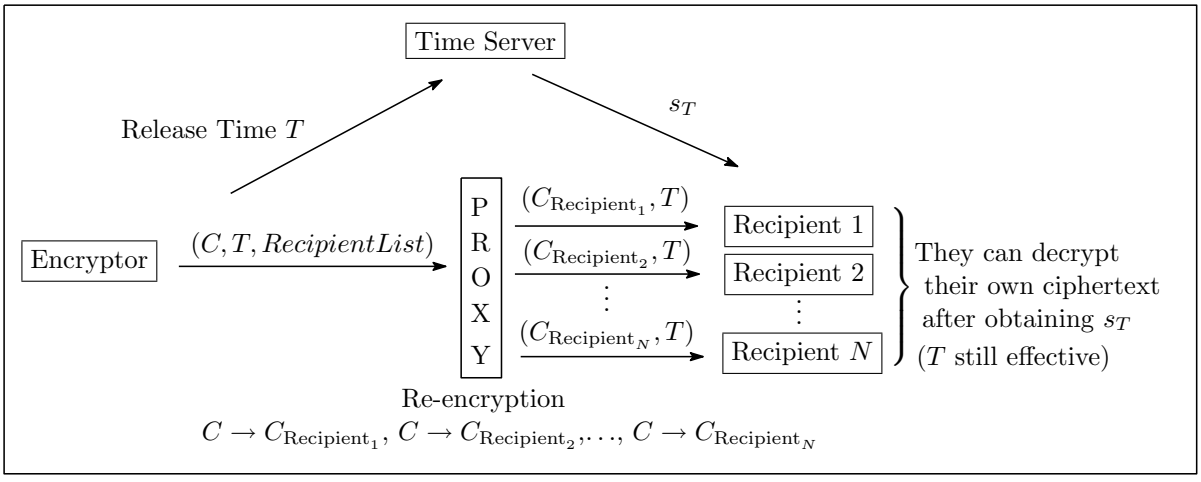

Fig. 2: Our TR-PRE with Multiple Recipients

The difficulty : The construction of TR-PRE is not trivial, even if a generic construction of TRE is given. For example, in Nakai et al.'s construction [28][4] (based on Identity-Based Encryption (IBE), Public Key Encryption (PKE), and Strongly Existential Unforgeable (SEU) one-time signature), a ciphertext is represented as $(K_v, T, c_1, c_2, \sigma)$, where K_v is a signature verification key (paired with a signing key K_s), T is a release time, $c_1 = \mathsf{PKE.Enc}(upk, (K_v||r))$, r is a random number chosen from the message space, upk is a user's public key, $c_2 = \mathsf{IBE.Enc}(T, (K_v||(M \oplus r)))$, and $\sigma = \mathsf{Sign}(K_s, (T||c_1||c_2))$. In this construction, T is regarded as the "identity" of IBE scheme. When simply exchanging underlying PKE scheme to a CCA-secure proxy re-encryption scheme (such as [25]), σ cannot work after the proxy translates c_1 into c_1' (which can be decrypted by another user), since a "signed-message" c_1 has already been changed. Other generic constructions [12, 13] require random oracles, since these constructions apply the Fujisaki-Okamoto transformation [18]. In [14], a general construction of TRE based on Security-Mediated Certificateless Encryption (SMCLE) was proposed. However, SMCLE is not a primitive tool (such as PKE, IBE, digital signatures, hash functions, and so on), and therefore "TRE combines PRE" is similar to "SMCLE combines PRE". From the above considerations, we need another structure to combine TRE and PRE schemes without random oracles.
Organization : The paper is organized as follows: Security definitions of TR-PRE are presented in Section 3. Our scheme is described in Section 4. The security analyses are presented in Section 5.

2 Preliminaries

Note that $x \xleftarrow{\$} S$ means that x is chosen uniformly from a set S. $y \leftarrow A(x)$ means that y is an output of an algorithm A under an input x.

2.1 Bilinear Groups and Complexity Assumption

Definition 1. (**Bilinear Groups**) *Bilinear groups and a bilinear map are defined as follows:*

1. $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *are cyclic groups of prime order* p.
2. g *is a generator of* $\mathbb{G}_1$.
3. e *is an efficiently computable bilinear map* $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ *with the following properties.*
   - *Bilinearity : for all* $u, u', v, v' \in \mathbb{G}_1$, $e(uu', v) = e(u, v)e(u', v)$ *and* $e(u, vv') = e(u, v)e(u, v')$.
   - *Non-degeneracy :* $e(g, g) \neq 1_{\mathbb{G}_2}$ ($1_{\mathbb{G}_2}$ *is the* $\mathbb{G}_2$ *unit*).

[4] This construction also handles pre-open capability. A recipient can decrypt a ciphertext by using a pre-open key before the release time has not been passed. We omit the explanation of this property.

Definition 2. (**3-QDBDH assumption**) *[25] The 3-Quotient Decision Bilinear Diffie-Hellman (3-QDBDH) problem is a problem, for input of a tuple* $(g, g^a, g^{a^2}, g^{a^3}, g^b, Z) \in \mathbb{G}_1^5 \times \mathbb{G}_2$ *to decide whether* $Z = e(g,g)^{b/a}$ *or not. An algorithm* $\mathcal{A}$ *has advantage* ϵ *in solving 3-QDBDH problem in* $\mathbb{G}_1$ *if* $Adv_{3\text{-}QDBDH}(\mathcal{A}) := |\Pr[\mathcal{A}(g, g^a, g^{a^2}, g^{a^3}, g^b, e(g,g)^{b/a}) = 0] - \Pr[\mathcal{A}(g, g^a, g^{a^2}, g^{a^3}, g^b, e(g,g)^z) = 0]| \geq \epsilon(k)$*, where* $e(g,g)^z \in \mathbb{G}_2 \setminus \{e(g,g)^{b/a}\}$*. We say that the 3-QDBDH assumption holds in* $\mathbb{G}_1$ *if no PPT algorithm has an advantage of at least* ϵ *in solving the 3-QDBDH problem.*

The hardness of the 3-QDBDH problem was discussed in [25], where the 3-QDBDH problem is not easier than the q–Decisional Bilinear Diffie-Hellman Inversion (q-DBDHI) problem [6]. The difficulty of q-DBDHI problem in generic groups was shown in [17], and this result implies the difficulty of the 3-QDBDH problem in generic groups. As in [25], we use the modified version of the 3-QDBDH (modified 3-QDBDH) problem, where for input of a tuple $(g, g^{1/a}, g^a, g^{a^2}, g^b, Z) \in \mathbb{G}_1^5 \times \mathbb{G}_2$ to decide whether $Z = e(g,g)^{b/a^2}$ or not. This modified 3-QDBDH problem is equivalent to the 3-QDBDH problem (See [25] Lemma 1).

Definition 3. (**Truncated decisional q-ABDHE assumption**) *[19] The truncated decisional q-Augmented Bilinear Diffie-Hellman (q-ABDHE) problem is a problem, for input of a tuple* $(g', g'^{(\alpha^{q+2})}, g, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^q}, Z) \in \mathbb{G}_1^{q+3} \times \mathbb{G}_2$ *to decide whether* $Z = e(g,g')^{\alpha^{q+1}}$ *or not. An algorithm* $\mathcal{A}$ *has advantage* ϵ *in solving truncated decisional q-ABDHE problem in* $\mathbb{G}_1$ *if* $Adv_{q\text{-}ABDHE}(\mathcal{A}) := |\Pr[\mathcal{A}(g', g'^{(\alpha^{q+2})}, g, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^q}, e(g,g')^{\alpha^{q+1}}) = 0] - \Pr[\mathcal{A}(g', g'^{(\alpha^{q+2})}, g, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^q}, e(g,g')^z) = 0]| \geq \epsilon(k)$*, where* $e(g,g')^z \in \mathbb{G}_2 \setminus \{e(g,g')^{\alpha^{q+1}}\}$*. We say that the truncated decisional q-ABDHE assumption holds in* $\mathbb{G}_1$ *if no PPT algorithm has an advantage of at least* ϵ *in solving the truncated decisional q-ABDHE problem.*

2.2 Strongly Existential Unforgeable One-Time Signatures

We apply the Libert-Vergnaud PRE [25], which applies the CHK transformation [8] to satisfy CCA security. Therefore, we also use an SEU one-time signature [2, 7, 29]. An SEU one-time signature Π consists of three algorithms, Sig.KeyGen, Sign and Verify. Sig.KeyGen is a probabilistic algorithm which outputs a signing/verification key pair (K_s, K_v). Sign is a probabilistic algorithm which outputs a signature σ from K_s, and a message $M \in \mathcal{M}_{Sig}$, where $\mathcal{M}_{Sig}$ is the message space of a signature scheme. Verify is a deterministic algorithm which outputs a bit from σ, K_v and M. "Verify outputs 1" indicates that σ is a valid signature of M, and 0, otherwise. The security experiment of SEU one-time signature under an adaptive Chosen Message Attack (one-time CMA-SEU) is defined as follows:

Definition 4. *one-time CMA-SEU*

$$Adv_{\Pi,\mathcal{A}}^{\text{one-time CMA-SEU}}(k) = \Big| \Pr\big[(K_s, K_v) \leftarrow \mathsf{Sig.KeyGen}(1^k); \\ (M, State) \leftarrow \mathcal{A}(K_v); \sigma \leftarrow \mathsf{Sign}(K_s, M);\ (M^*, \sigma^*) \leftarrow \mathcal{A}(K_v, \sigma, State); \\ (M^*, \sigma^*) \neq (M, \sigma); \mathsf{Verify}(K_v, \sigma^*, M^*) = 1\Big|$$

We say that a signature scheme is one-time CMA-SEU secure if the advantage $Adv_{\Pi,\mathcal{A}}^{\text{one-time CMA-SEU}}(k)$ is negligible for any polynomial-time adversary $\mathcal{A}$. Intuitively, an SEU one-time signature scheme is secure when an adversary $\mathcal{A}$ cannot issue a pair (M^*, σ^*) even if $\mathcal{A}$ has already obtained a signature of M^*.

3 Definitions of TR-PRE

3.1 System operations of TR-PRE

First, we define encryption levels (refer to [1]) as follows: A "first-level" ciphertext is a ciphertext which cannot be re-encrypted for another user. A "second-level" ciphertext is a ciphertext which can be re-encrypted into a first-level ciphertext using an appropriate re-encryption key. In this paper, we consider a single-hop scheme (such as [25]). A TR-PRE scheme Π consists of seven algorithms (Setup, KeyGen, Encrypt, TS-Release, RKGen, Re-Encrypt, Decrypt):

$\mathsf{Setup}(1^k)$: This algorithm takes as input the security parameter k, and returns the master public parameters $params$, the time server's public key TS_{pub}, and the time server's secret key ts_{priv}. We assume that $params$ includes TS_{pub}.

$\mathsf{KeyGen}(params)$: This algorithm takes as input $params$, and returns a public/secret key pair (upk, usk).

$\mathsf{TS\text{-}Release}(params, ts_{\mathsf{priv}}, T)$: This algorithm takes as input $params$, ts_{priv}, and a release time T, and returns a trapdoor s_T.

$\mathsf{Encrypt}(params, upk, M, T)$: This algorithm takes as input $params$, a user's public key upk, a plaintext M, and T, and returns a second-level ciphertext C which can be transformed into a first-level ciphertext using an appropriate re-encryption key.

$\mathsf{RKGen}(params, usk_i, upk_j)$: This algorithm takes as input $params$, a user U_i's secret key usk_i, and a user U_j's public key upk_j, and returns a re-encryption key R_{ij}.

$\mathsf{Re\text{-}Encrypt}(params, R_{ij}, upk_i, C)$: This algorithm takes as input $params$, R_{ij}, and upk_i, and a second-level ciphertext C encrypted by upk_i, and returns a first-level re-encrypted ciphertext C which can be decrypted by usk_j.

$\mathsf{Decrypt}(params, usk, s_T, C, T)$: This algorithm takes as input $params$, usk, s_T and C, and returns M or $\perp$.

3.2 Security Requirements

We define the chosen-ciphertext security of TR-PRE. This is naturally defined from the security definitions of TRE [10] and PRE [25]. Let (upk^*, usk^*) be the challenge public/secret key. For other honest parties, keys are subscripted by h or h'. For corrupted parties, keys are subscripted by c or c'. As in [25], an adversary is given all re-encryption keys, except from the target user to a corrupted user.

Oracles : $\mathcal{A}$ can issue re-encryption queries to the re-encryption oracle $\mathcal{O}_{RE\text{-}ENC}$, where for an input (upk_i, upk_j, C), if either "C is a first-level ciphertext" or "upk_j is a corrupted user and $C = C^{*}$", then $\mathcal{O}_{RE\text{-}ENC}$ returns $\perp$. Otherwise, $\mathcal{O}_{RE\text{-}ENC}$ returns a re-encrypted ciphertext C'. $\mathcal{A}$ can issue decryption queries to the decryption oracle $\mathcal{O}_{DEC}$, where for an input (upk, C, T), if either "upk was not produced by KeyGen" or "$(upk, C, T) = (upk^*, C^*, T^*)$", then $\mathcal{O}_{DEC}$ returns $\perp$. In addition, if C is a first-level ciphertext derived from C^*, and $upk \in \{upk_h, upk_{h'}\}$, then $\mathcal{O}_{DEC}$ returns $\perp$. Otherwise, $\mathcal{O}_{DEC}$ returns a decryption result M. We assume that C is a first-level ciphertext, since a second-level decryption oracle is useless. More precisely, $\mathcal{A}$ with R_{hc} and R_{*h} (these are defined in the following experiment) can re-encrypt second-level ciphertexts.

We say that a (single-hop) TR-PRE scheme is *replayable chosen-ciphertext* (IND-RCCA) secure if the advantage is negligible for any polynomial-time adversary $\mathcal{A}$ in the following experiment.

Definition 5. *IND-RCCA*

$$
\begin{aligned}
Adv^{IND\text{-}RCCA}_{\Pi,\mathcal{A}}(k) = \Big| \Pr\big[& (params, ts_{priv}) \leftarrow \mathsf{Setup}(1^k); (upk^*, usk^*) \leftarrow \mathsf{KeyGen}(params); \\
& (upk_h, usk_h) \leftarrow \mathsf{KeyGen}(params); (upk_{h'}, usk_{h'}) \leftarrow \mathsf{KeyGen}(params); \\
& (upk_c, usk_c) \leftarrow \mathsf{KeyGen}(params); (upk_{c'}, usk_{c'}) \leftarrow \mathsf{KeyGen}(params); \\
& Set\ Keys := \{upk^*, upk_h, upk_{h'}, (upk_c, usk_c), (upk_{c'}, usk_{c'})\}; \\
& R_{c*} \leftarrow \mathsf{RKGen}(params, usk_c, upk^*); R_{h*} \leftarrow \mathsf{RKGen}(params, usk_h, upk^*); \\
& R_{*h} \leftarrow \mathsf{RKGen}(params, usk^*, upk_h); R_{hc} \leftarrow \mathsf{RKGen}(params, usk_h, upk_c); \\
& R_{ch} \leftarrow \mathsf{RKGen}(params, usk_c, upk_h); R_{cc'} \leftarrow \mathsf{RKGen}(params, usk_c, upk_{c'}); \\
& R_{hh'} \leftarrow \mathsf{RKGen}(params, usk_h, upk_{h'}); \\
& Set\ ReKeys := \{R_{c*}, R_{h*}, R_{*h}, R_{hc}, R_{ch}, R_{cc'}, R_{hh'}\}; \\
& (M_0^*, M_1^*, T^*, State) \leftarrow \mathcal{A}^{\mathcal{O}_{RE\text{-}ENC}, \mathcal{O}_{DEC}}(params, Keys, ReKeys, ts_{priv}); \\
& \mu \xleftarrow{\$} \{0,1\}; C^* \leftarrow \mathsf{Encrypt}(params, upk^*, M_\mu^*, T^*); \\
& \mu' \leftarrow \mathcal{A}^{\mathcal{O}_{RE\text{-}ENC}, \mathcal{O}_{DEC}}(C^*, State); \mu = \mu' \big] - 1/2 \Big|
\end{aligned}
$$

In this experiment, $\mathcal{A}$ does not have to access the timed-release trapdoor extraction oracle $\mathcal{O}_{TS\text{-}Release}$ (defined in the IND-CTCA experiment), since $\mathcal{A}$ is given ts_{priv}. IND-RCCA security guarantees that even if the appropriate trapdoor is given, non-legitimate users (who do not have an appropriate secret key) cannot decrypt a ciphertext. This suggests $\mathcal{A}$ is an "honest but curious" time server.

As in [9, 25], we assume a static corruption model, which does not capture a scenario in which an adversary generates public/secret keys for all parties.

Next, we define *chosen-time period chosen-ciphertext* (IND-CTCA) security. $\mathcal{A}$ can issue key generation queries to the key generation oracle $\mathcal{O}_{KeyGen}$, where $\mathcal{O}_{KeyGen}$ returns (upk, usk). $\mathcal{A}$ can issue re-encryption queries to the re-encryption key generation oracle $\mathcal{O}_{RKGen}$, where for an input (usk_i, upk_j), $\mathcal{O}_{RKGen}$ returns R_{ij}. $\mathcal{A}$ can issue trapdoor extraction queries $\mathcal{O}_{TS\text{-}Release}$, where for an input time T, $\mathcal{O}_{TS\text{-}Release}$ returns a trapdoor s_T. Note that $\mathcal{A}$ cannot query T^* to $\mathcal{O}_{TS\text{-}Release}$, where T^* is the challenge time. $\mathcal{A}$ can issue decryption queries to the decryption oracle $\mathcal{O}_{DEC}$, where for an input (upk, C, T), if either "upk was not produced by KeyGen" or "$(upk, C, T) = (upk^*, C^*, T^*)$", then $\mathcal{O}_{DEC}$ returns $\perp$. Note that if C is a first-level ciphertext derived from C^*, then the decryption oracle $\mathcal{O}_{DEC}$ returns $\perp$. In our scheme, we can detect whether a first-level ciphertext C is derived from C^* or not, since a part of the ciphertext $(C_1^*, C_3^*, C_4^*, C_5^*, \sigma^*, T^*)$ is also included in C. We say that a (single-hop) TR-PRE scheme is IND-CTCA-secure if the advantage is negligible for any polynomial-time adversary $\mathcal{A}$ in the following experiment.

Definition 6. *IND-CTCA*

$$
\begin{aligned}
Adv_{\Pi,\mathcal{A}}^{IND\text{-}CTCA}(k) = \big| \Pr \big[& (params, ts_{priv}) \leftarrow \mathsf{Setup}(1^k); \\
& Set\ \mathcal{O} := \{\mathcal{O}_{KeyGen}, \mathcal{O}_{RE\text{-}ENC}, \mathcal{O}_{RKGen}, \mathcal{O}_{DEC}, \mathcal{O}_{TS\text{-}Release}\} \\
& (M_0^*, M_1^*, T^*, upk^*, State) \leftarrow \mathcal{A}^{\mathcal{O}}(params); \\
& \mu \xleftarrow{\$} \{0,1\}; C^* \leftarrow \mathsf{Encrypt}(params, upk^*, M_\mu^*, T^*); \\
& \mu' \leftarrow \mathcal{A}^{\mathcal{O}}(C^*, State); \mu = \mu' \big] - 1/2 \big|
\end{aligned}
$$

IND-CTCA security guarantees that even if the appropriate secret key is given, $\mathcal{A}$ cannot decrypt a ciphertext before the appropriate trapdoor is released. This suggests $\mathcal{A}$ is a malicious user in this experiment.

The above two definitions are secure for a second-level ciphertext, since the challenge ciphertext is a second-level one. We can define experiments for a first-level ciphertext in an orthogonal manner. We omit the details of definitions, since they are similar to the Libert-Vergnaud definitions [25].

4 Proposed Scheme

In this section, we propose our TR-PRE scheme. Our TR-PRE is based on the Libert-Vergnaud PRE [25], and the Gentry IBE [19].

Protocol 1. ***The proposed TR-PRE scheme***

$\mathsf{Setup}(1^k)$ *: Let n and m be polynomials in the security parameter k. Let $(\mathbb{G}_1, \mathbb{G}_2)$ be a bilinear group with prime order p, $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ a bilinear map, $g, u, v, h_1, h_2, h_3 \in \mathbb{G}_1$ generators, $\mathcal{M} = \{0,1\}^n$ a message space, and $\mathcal{T} = \{0,1\}^m$ a release time space. Select $s \xleftarrow{\$} \mathbb{Z}_p^*$, compute $TS_{pub} = g^s$, and*

output $ts_{priv} = s$ *and* $params = (g, u, v, h_1, h_2, h_3, TS_{pub}, H_1, H_2)$, *where* $H_1 : \{0,1\}^m \to \mathbb{Z}_p$ *and* $H_2 : \{0,1\}^* \to \mathbb{Z}_p$ *are cryptographic hash functions.*

$\mathsf{KeyGen}(params)$ *: For a user* U_i, *choose* $x_i \xleftarrow{\$} \mathbb{Z}_p$, *compute* $X_i = g^{x_i}$, *and output* $(upk_i, usk_i) = (X_i, x_i)$.

$\mathsf{TS\text{-}Release}(params, ts_{\mathsf{priv}}, T)$ *: For a release time* $T \in \mathcal{T}$, *choose* $r_{T,1}, r_{T,2}, r_{T,3} \xleftarrow{\$} \mathbb{Z}_p^*$, *compute* $s_T = \big((r_{T,1}, (h_1 \cdot g^{-r_{T,1}})^{\frac{1}{s-H_1(T)}}), (r_{T,2}, (h_2 \cdot g^{-r_{T,2}})^{\frac{1}{s-H_1(T)}}), (r_{T,3}, (h_3 \cdot g^{-r_{T,3}})^{\frac{1}{s-H_1(T)}})\big)$, *and output* s_T.

$\mathsf{Encrypt}(params, upk_i, M, T)$ *: Let* $upk_i = X_i$. *For* $M \in \mathcal{M}$ *and* $T \in \mathcal{T}$, *choose* $r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p$ *and a one-time signature key pair* $(K_s, K_v) \leftarrow \mathsf{Sig.KeyGen}(1^k)$, *set* $C_1 := K_v$, *compute* $C_2 = X_i^{r_1}$, $C_3 = M \cdot e(g,g)^{r_1} \cdot e(g, h_1)^{r_2}$, $C_4 = (u^{K_v} \cdot v)^{r_1}$, $C_5 = (g^{-H_1(T)} \cdot TS_{pub})^{r_2}$, $C_6 = e(g,g)^{r_2}$, $C_7 = (e(g,h_2) \cdot e(g,h_3)^{\beta})^{r_2}$, *where* $\beta = H_2(C_3, C_5, C_6)$, *and* $\sigma = \mathsf{Sign}(K_s, (C_3, C_4, C_5, C_6, C_7, T))$, *and output a second-level ciphertext* $C = (C_1, C_2, C_3, C_4, C_5, C_6, C_7, \sigma, T)$.

$\mathsf{RKGen}(params, usk_i, upk_j)$ *: Let* $usk_i = x_i$ *and* $upk_j = X_j$. *Compute* $R_{ij} = X_j^{\frac{1}{x_i}} = g^{\frac{x_j}{x_i}}$, *and output* R_{ij}.

$\mathsf{Re\text{-}Encrypt}(params, R_{ij}, upk_i, C)$ *: Let* $upk_i = X_i$. *From a second-level ciphertext* C, *a first-level ciphertext* C' *is computed as follows: Let* $C = (C_1, C_2, C_3, C_4, C_5, C_6, C_7, \sigma, T)$. *Check* $e(C_2, u^{C_1} \cdot v) \stackrel{?}{=} e(X_i, C_4)$ *and* $\mathsf{Verify}(C_1, (C_3, C_4, C_5, C_6, C_7, T)) \stackrel{?}{=} 1$. *If well-formed, choose* $t \xleftarrow{\$} \mathbb{Z}_p$, *compute* $C_2' = X_i^t$, $C_2'' = R_{ij}^{\frac{1}{t}}$, *and* $C_2''' = C_2^t$, *and output a first-level ciphertext* $C' = (C_1, C_2', C_2'', C_2''', C_3, C_4, C_5, C_6, C_7, \sigma, T)$.

$\mathsf{Decrypt}(params, usk, C, s_T)$ *:*

In the case of first-level ciphertext *: Let* $(C_1, C_2', C_2'', C_2''', C_3, C_4, C_5, C_6, C_7, \sigma, T)$ *be a first-level ciphertext, and* $s_T = \big((r_{T,1}, h_{T,1}), (r_{T,2}, h_{T,2}), (r_{T,3}, h_{T,3})\big)$ *be the trapdoor of* T. *Check* $e(C_2', C_2'') \stackrel{?}{=} e(X_j, g)$, $e(C_2''', u^{C_1} \cdot v) \stackrel{?}{=} e(C_2', C_4)$, $C_7 \stackrel{?}{=} e(C_5, h_{T,2} h_{T,3}^{\beta}) \cdot C_6^{r_{T,2} + r_{T,3}\beta}$, *where* $\beta = H_2(C_3, C_5, C_6)$, *and* $\mathsf{Verify}(C_1, (C_3, C_4, C_5, C_6, C_7, T)) \stackrel{?}{=} 1$. *If well-formed, compute*

$$\begin{aligned} e(C_2'', C_2''')^{\frac{1}{x_j}} &= e(g^{\frac{x_j}{t x_i}}, g^{t x_i r_1})^{\frac{1}{x_j}} = e(g,g)^{r_1}, \\ e(C_5, h_{T,1}) \cdot C_6^{r_{T,1}} &= e((g^{-H_1(T)} \cdot TS_{pub})^{r_2}, (h_1 \cdot g^{-r_{T,1}})^{\frac{1}{s-H_1(T)}}) \cdot e(g,g)^{r_{T,1} r_2} \\ &= e(g, h_1)^{r_2}, \text{ and} \\ C_3 / \{e(g,g)^{r_1} \cdot e(g, h_1)^{r_2}\} &= M, \text{ and output } M. \end{aligned}$$

In the case of second-level ciphertext *: Let* $(C_1, C_2, C_3, C_4, C_5, C_6, C_7, \sigma, T)$ *be a second-level ciphertext, and* $s_T = \big((r_{T,1}, h_{T,1}), (r_{T,2}, h_{T,2}), (r_{T,3}, h_{T,3})\big)$ *be the trapdoor of* T. *Check* $e(C_2, u^{C_1} \cdot v) \stackrel{?}{=} e(X_i, C_4)$, $C_7 \stackrel{?}{=} e(C_5, h_{T,2} h_{T,3}^{\beta}) \cdot C_6^{r_{T,2} + r_{T,3}\beta}$, *where* $\beta = H_2(C_3, C_5, C_6)$, *and* $\mathsf{Verify}(C_1, (C_3, C_4, C_5, C_6, C_7, T)) \stackrel{?}{=} 1$. *If well-formed, compute*

$$
\begin{aligned}
e(C_2, g)^{\frac{1}{x_i}} &= e(X_i^{r_1}, g)^{\frac{1}{x_i}} = e(g, g)^{r_1}, \\
e(C_5, h_{T,1}) \cdot C_6^{r_{T,1}} &= e((g^{-H_1(T)} \cdot TS_{pub})^{r_2}, (h_1 \cdot g^{-r_{T,1}})^{\frac{1}{s - H_1(T)}}) \cdot e(g, g)^{r_{T,1} r_2} \\
&= e(g, h_1)^{r_2}, \ \textit{and} \\
C_3 / \{e(g, g)^{r_1} \cdot e(g, h_1)^{r_2}\} &= M, \ \textit{and output } M.
\end{aligned}
$$

A first-level ciphertext can be computed directly: for an input $(param, upk_j = X_j, M, T)$, choose $t, r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p^*$, and a one-time signature key pair $(K_s, K_v) \leftarrow \mathsf{Sig.KeyGen}(1^k)$, set $C_1 := K_v$, and compute $C_2 = X_j^t$, $C_2'' = g^{1/t}$, $C_2''' = X_j^{r_1 t}$, $C_3 = M \cdot e(g, g)^{r_1} \cdot e(g, h_1)^{r_2}$, $C_4 = (u^{K_v} \cdot v)^{r_1}$, $C_5 = (g^{-H_1(T)} \cdot TS_{\mathrm{pub}})^{r_2}$, $C_6 = e(g, g)^{r_2}$, $C_7 = (e(g, h_2) \cdot e(g, h_3)^{\beta})^{r_2}$, where $\beta = H_2(C_3, C_5, C_6)$, and $\sigma = \mathsf{Sign}(K_s, (C_3, C_4, C_5, C_6, C_7, T))$. Then $C' = (C_1, C_2', C_2'', C_2''', C_3, C_4, C_5, C_6, C_7, \sigma, T)$ is a valid second-level ciphertext under U_j. We call this encryption "direct encryption method" (we use this in the proof of Theorem 1).

Intuitively, $e(g, g)^{r_1}$ is computed from a PRE section, and $e(g, h_1)^{r_2}$ is computed from a TRE section constructed by the Gentry IBE. Together with these elements, the cancel element $e(g, g)^{r_1} \cdot e(g, g)^{r_2}$ can be computed. (C_3, C_5, C_6, C_7) is a ciphertext for a message M' of the Gentry IBE scheme, where $M' := M \cdot e(g, g)^{r_1}$, and (C_1, C_2, C_3, C_4) is a (part of) ciphertext for a message M'' of the Libert-Vergnaud PRE scheme [25], where $M'' := M \cdot e(g, h_1)^{r_2}$.

5 Security Analysis

Theorem 1. *Our TR-PRE scheme is IND-RCCA-secure if the modified 3-QDBDH assumption holds, and the underlying one-time signature scheme is SEU.*

Proof. This proof is similar to that of the Libert-Vergnaud PRE scheme. However, we cannot directly use the challenger of the Libert-Vergnaud PRE scheme in a black-box manner, since the signature part of our scheme is different from that of the Libert-Vergnaud PRE scheme. Therefore, we have to write down the detailed proof: Let $(g, A_{-1} = g^{1/a}, A_1 = g^a, A_2 = g^{a^2}, B = g^b, Z)$ be a modified 3-QDBDH instance. We construct an algorithm $\mathcal{B}$ that can decide whether $Z = e(g, g)^{b/a^2}$ or not, by using an adversary $\mathcal{A}$ to break our TR-PRE scheme.

Before constructing $\mathcal{B}$, we explain two cases, in which we can break SEU of underlying one-time signature scheme: Let $C^* = (C_1^* = K_v^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*, C_7^*, \sigma^*, T^*)$ be the challenge ciphertext. Let event_1 be the event that $\mathcal{A}$ issues a decryption query $(K_v^*, C_2', C_2'', C_2''', C_3, C_4, C_5, C_6, C_7, T)$, where $\mathsf{Verify}(K_v^*, (C_3, C_4, C_5, C_6, C_7, T)) = 1$. Let event_2 be the event that $\mathcal{A}$ issues a re-encryption query $(K_v^*, C_2, C_3, C_4, C_5, C_6, C_7, T)$, where $\mathsf{Verify}(K_v^*, (C_3, C_4, C_5, C_6, C_7, T)) = 1$. If either event_1 or event_2 occurs, then we can construct an algorithm that breaks SEU of the underlying one-time signature scheme.

From now, we construct an algorithm $\mathcal{B}$ that outputs a random bit and aborts when either event_1 or event_2 occurs. $\mathcal{B}$ computes $(K_s^*, K_v^*) \leftarrow \mathsf{Sig.KeyGen}(1^k)$,

chooses $\alpha_1, \alpha_2 \xleftarrow{\$} \mathbb{Z}_p^*$, and computes $u := A_1^{\alpha_1} = g^{a\alpha_1}$ and $v := A_1^{-\alpha_1 \cdot K_v^*} \cdot A_2^{\alpha_2} = g^{-a\alpha_1 K_v^* + a^2\alpha_2}$. $u^{K_v} \cdot v = A_1^{\alpha_1(K_v - K_v^*)} \cdot A_2^{\alpha_2}$ will appear in a part of a ciphertext. $\mathcal{B}$ chooses $s \xleftarrow{\$} \mathbb{Z}_p$ as ts_{priv}, and $h_1, h_2, h_3 \xleftarrow{\$} \mathbb{Z}_p$, and computes $TS_{\text{pub}} = g^s$.

Public/Secret Key Generation: For the target user, $\mathcal{B}$ chooses $x^* \xleftarrow{\$} \mathbb{Z}_p$, and computes $upk^* = X^* := A_2^{x^*}$. For an honest user U_h (also in the case of $U_{h'}$), $\mathcal{B}$ chooses $x_h \xleftarrow{\$} \mathbb{Z}_p$, and computes $upk_h = X_h := A_1^{x_h}$. For a corrupted user U_c (also in the case of $U_{c'}$), $\mathcal{B}$ chooses $x_c \xleftarrow{\$} \mathbb{Z}_p$ as usk_c, and computes $upk_c = X_c := g^{x_c}$.

Re-encryption Key Generation: For R_{c*}, $\mathcal{B}$ can compute $R_{c*} = (X^*)^{1/x_c}$, since $\mathcal{B}$ knows $usk_c = x_c$. For R_{h*}, $\mathcal{B}$ can compute $R_{h*} = A_1^{x^*/x_h} = g^{x^* a^2/(x_h a)}$. For R_{*h}, $\mathcal{B}$ can compute $R_{*h} = A_{-1}^{x_h/x^*} = g^{x_h a/(x^* a^2)}$. Note that R_{h*} and R_{*h} are valid re-encryption keys, since $usk^* = x^* a^2$ and $usk_h = x_h a$. For R_{hc}, $\mathcal{B}$ can compute $R_{hc} = A_{-1}^{x_c/x_h} = g^{x_c/(x_h a)}$. For R_{ch}, $\mathcal{B}$ can compute $R_{ch} = A_1^{x_h/x_c} = g^{x_h a/x_c}$. For $R_{cc'}$, $\mathcal{B}$ can compute $R_{cc'} = g^{x_c/x_{c'}}$, since $\mathcal{B}$ knows $usk_c = x_c$ and $usk_{c'} = x_{c'}$. For $R_{hh'}$, $\mathcal{B}$ can compute $R_{hh'} = g^{x_{h'}/x_h} = g^{x_{h'} a/(x_h a)}$.

From the above considerations, $\mathcal{B}$ can send $params = (g, u, v, h_1, h_2, h_3, TS_{\text{pub}}, H_1, H_2)$, $Keys$, $ReKeys$, and ts_{priv} to $\mathcal{A}$, where $Keys := \{upk^*, upk_h, upk_{h'}, (upk_c, usk_c), (upk_{c'}, usk_{c'})\}$ and $ReKeys := \{R_{c*}, R_{h*}, R_{*h}, R_{hc}, R_{ch}, R_{cc'}, R_{hh'}\}$.

- If $\mathcal{A}$ issues $\mathcal{O}_{RE\text{-}ENC}$ with an input (upk_i, upk_j, C), where $C = (C_1, C_2, C_3, C_4, C_5, C_6, C_7, \sigma, T)$ is a second-level ciphertext, we consider the following three cases:
  - **i is the target user and j is an honest user** : $\mathcal{B}$ simply re-encrypts C by using R_{*h}.
  - **i is not the target user and j is an honest user** : $\mathcal{B}$ simply re-encrypts C by using $R_{hh'}$ or R_{ch}.
  - **i is the target user and j is a corrupted user** : If $C_1 = K_v^*$ (event_2), then $\mathcal{B}$ outputs a random bit, and aborts. Otherwise, $\mathcal{B}$ computes $C_2^{1/x^*} = ((X^*)^{r_1})^{1/x^*} = A_2^{r_1}$. Now $C_4 = (u^{K_v} \cdot v)^{r_1} = (A_1^{\alpha_1(K_v - K_v^*)} \cdot A_2^{\alpha_2})^{r_1}$. Therefore, $A_1^{r_1} = g^{ar_1} = \left(C_4/(C_2^{1/x^*})^{\alpha_2}\right)^{1/(\alpha_1(K_v - K_v^*))}$ holds. $\mathcal{B}$ chooses $t, r_2 \xleftarrow{\$} \mathbb{Z}_p$, sets $\tilde{t} := at/x_c$, and computes $C_2' = A_1^t = g^{at} = g^{x_c \cdot at/x_c} = g^{x_c \tilde{t}} = X_c^{\tilde{t}}$, $C_2'' = A_{-1}^{x_c/t} = g^{x_c/at} = g^{1/\tilde{t}}$, and $C_2''' = \{\left(C_4/(C_2^{1/x^*})^{\alpha_2}\right)^{1/(\alpha_1(K_v - K_v^*))}\}^t = g^{ar_1 t} = g^{r_1 x_c \tilde{t}} = (X_c^{r_1})^t$. Note that we use the direct encryption method.
- When $\mathcal{A}$ issues $\mathcal{O}_{DEC}$ with an input (upk_j, C, T), where C is a first-level ciphertext under upk_j, then if C is ill-formed, $\mathcal{B}$ returns $\perp$. If $upk_j = upk_c$, then $\mathcal{B}$ can decrypt C, since $\mathcal{B}$ knows usk_c. We consider the remaining two cases as follows:
  - **j is an honest user** : Since $X_j = g^{ax_j}$, $e(C_2'', C_2''') = e(X_j, g)^{r_1} = e(g, g)^{ar_1 x_j}$ hold. In addition, $C_4 = (u^{K_v} \cdot v)^{r_1} = (A_1^{\alpha_1(K_v - K_v^*)} \cdot A_2^{\alpha_2})^{r_1} = g^{a\alpha_1 r_1 (K_v - K_v^*)}$. $g^{a^2 \alpha_2 r_1}$ holds. Therefore $\left(\frac{e(C_4, A_{-1})}{e(C_2'', C_2''')^{\alpha_2/x_j}}\right)^{\frac{1}{\alpha_1(K_v - K_v^*)}} = e(g, g)^{r_1}$ holds. By

using x_j, $\mathcal{B}$ can compute $e(g,g)^{r_1}$. In addition, $\mathcal{B}$ can compute s_T, and $e(g,h_1)^{r_2}$ from (C_5, C_6, C_7). $\mathcal{B}$ returns $M = C_3/\{e(g,g)^{r_1} \cdot e(g,h_1)^{r_2}\}$ to $\mathcal{A}$.

- **j is the target user** : If $C_1 = K_v^*$ (event_1), then $\mathcal{B}$ outputs a random bit, and aborts. Now $X_j = g^{x^* a^2}$. Therefore, $e(C_2'', C_2''') = (e, X_j, g)^{r_1} = e(g,g)^{a^2 r_1 x^*}$ hold. Since $C_4 = g^{a\alpha_1 r_1 (K_v - K_v^*)} \cdot g^{a^2 \alpha_2 r_1}$ holds, $e(C_4, g) = e(g,g)^{a\alpha_1 r_1 (K_v - K_v^*)} \cdot e(g,g)^{a^2 \alpha_2 r_1}$ holds. Therefore $\left(\frac{e(C_4,g)}{e(C'', C_2''')^{\alpha_2 / x_j}}\right)^{\frac{1}{\alpha_1 (K_v - K_v^*)}} = e(g,g)^{a r_1}$ holds. In addition, $e(C_4, A_{-1}) = e(g,g)^{\alpha_1 r_1 (K_v - K_v^*)} \cdot e(g,g)^{a\alpha_2 r_1}$ holds. $\mathcal{B}$ computes $\left(\frac{e(g,g)^{\alpha_1 r_1 (K_v - K_v^*)} \cdot e(g,g)^{a\alpha_2 r_1}}{(e(g,g)^{a r_1})^{\alpha_2}}\right)^{\frac{1}{\alpha_1 (K_v - K_v^*)}} = e(g,g)^{r_1}$. In addition, $\mathcal{B}$ can compute s_T, and $e(g,h_1)^{r_2}$ from (C_5, C_6, C_7). $\mathcal{B}$ returns $M = C_3/\{e(g,g)^{r_1} \cdot e(g,h_1)^{r_2}\}$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ sends (M_0^*, M_1^*, T^*) to $\mathcal{B}$. $\mathcal{B}$ chooses $r_2^* \overset{\$}{\leftarrow} \mathbb{Z}_p$, sets $C_1^* = K_v^*$, and computes $C_2^* = B^{x^*}$, $C_3 = M_\mu^* \cdot Z \cdot e(g,h_1)^{r_2^*}$, $C_4^* = B^{\alpha_2}$, $C_5^* = (g^{-H_1(T^*)} \cdot TS_{\mathrm{pub}})^{r_2^*}$, $C_6 = e(g,g)^{r_2^*}$, and $C_7 = e(g,h_2)^{r_2^*} \cdot e(g,h_3)^{r_2^* \beta}$, where $\beta = H_2(C_3^*, C_5^*, C_6^*)$, and $\sigma^* = \mathsf{Sign}(K_s^*, (C_3^*, C_4^*, C_5^*, C_6^*, C_7^*, T^*))$. When $Z = e(g,g)^{b/a^2}$, $C^* = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*, C_7^*, \sigma^*, T^*)$ is a valid ciphertext of M_μ^* with $r_1^* := b/a^2$. Otherwise, if Z is a random value, M_μ^* is perfectly hidden by Z. Therefore, $\mathcal{B}$ decides $Z = e(g,g)^{b/a^2}$ when $\mu' = \mu$, and Z is a random value, otherwise. □

Theorem 2. *Our TR-PRE scheme is IND-CTCA-secure if the truncated decisional q-ABDHE assumption holds.*

Proof. This proof clearly holds, since we can use the challenger of the Gentry IBE scheme $\mathcal{C}$ in a black-box manner, and the Gentry IBE scheme is secure under the truncated decisional q-ABDHE assumption. More precisely, the simulator $\mathcal{B}$ chooses all PRE-related parameters (incl. all user's secret keys), and can use $\mathcal{C}$ when $\mathcal{O}_{TS\text{-}Release}$ and $\mathcal{O}_{DEC}$ are issued by $\mathcal{A}$. Note that $\mathcal{B}$ can decrypt (C, T) if an element (canceled by the TRE section) $e(g,h_1)^{r_2}$ is computed by $\mathcal{C}$, since $\mathcal{B}$ knows all user's secret keys. For the challenge phase, $\mathcal{B}$ can set $((M_0')^*, (M_1')^*) := (M_0^* \cdot e(g,g)^{r_1^*}, M_1^* \cdot e(g,g)^{r_1^*})$ as the challenge message, and sends this to $\mathcal{C}$. $\mathcal{B}$ can compute the challenge ciphertext C^* to add the PRE section into the challenge ciphertext of the IBE scheme given by $\mathcal{C}$. Note that $\mathcal{B}$ cannot decrypt the challenge ciphertext C^*, since the TRE part of the C^* is the challenge ciphertext of the Gentry IBE scheme. $\mathcal{B}$ outputs μ' to $\mathcal{C}$ as the guessing bit, where μ' is the output result of the IND-CTCA adversary $\mathcal{A}$. □

Note that the above proofs only cover a second-level ciphertext, since the challenge ciphertext is a second-level one. As in [25], we can prove the security of the first-level ciphertext in the IND-RCCA (resp. IND-CTCA) experiment in the same manner as the proof of Theorem 1 (resp. Theorem 2).

6 Conclusion

In this paper, for the first time we propose a TR-PRE scheme. Even if the proxy transformation is applied to a TRE ciphertext, the release time is still effective. Our construction is based on the Libert-Vergnaud PRE [25] and the Gentry IBE [19]. We modified Nakai et al.'s construction [28] (which is a generic construction of TRE) to combine PRE and TRE. Our work is valuable in adding an access control function into encrypted (and re-encrypted) data itself. This feature is suitable for data management in cloud computing environments.

Acknowledgements

The authors would like to thank anonymous reviewers of ProvSec 2010 for their invaluable comments. The first author Keita Emura is supported by the Center for Highly Dependable Embedded Systems Technology as a Postdoc researcher.

References

1. Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, 2006.
2. Mihir Bellare and Sarah Shoup. Two-tier signatures, strongly unforgeable signatures, and fiat-shamir without random oracles. In *Public Key Cryptography*, pages 201–216, 2007.
3. John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.
4. Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In *EUROCRYPT*, pages 127–144, 1998.
5. Rakeshbabu Bobba, Joe Muggli, Meenal Pant, Jim Basney, and Himanshu Khurana. Usable secure mailing lists with untrusted servers. In *IDtrust*, pages 103–116, 2009.
6. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.
7. Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *J. Cryptology*, 21(2):149–177, 2008.
8. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, pages 207–222, 2004.
9. Ran Canetti and Susan Hohenberger. Chosen-ciphertext secure proxy re-encryption. In *ACM Conference on Computer and Communications Security*, pages 185–194, 2007.
10. Julien Cathalo, Benoît Libert, and Jean-Jacques Quisquater. Efficient and non-interactive timed-release encryption. In *ICICS*, pages 291–303, 2005.
11. Konstantinos Chalkias, Dimitrios Hristu-Varsakelis, and George Stephanides. Improved anonymous timed-release encryption. In *ESORICS*, pages 311–326, 2007.
12. Jung Hee Cheon, Nicholas Hopper, Yongdae Kim, and Ivan Osipkov. Timed-release and key-insulated public key encryption. In *Financial Cryptography*, pages 191–205, 2006.

13. Jung Hee Cheon, Nicholas Hopper, Yongdae Kim, and Ivan Osipkov. Provably secure timed-release public key encryption. *ACM Trans. Inf. Syst. Secur.*, 11(2), 2008.
14. Sherman S. M. Chow, Volker Roth, and Eleanor G. Rieffel. General certificateless encryption and timed-release encryption. In *SCN*, pages 126–143, 2008.
15. Sherman S. M. Chow and Siu-Ming Yiu. Timed-release encryption revisited. In *ProvSec*, pages 38–51, 2008.
16. Alexander W. Dent and Qiang Tang. Revisiting the security model for timed-release encryption with pre-open capability. In *ISC*, pages 158–174, 2007.
17. Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In *Public Key Cryptography*, pages 416–431, 2005.
18. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO*, pages 537–554, 1999.
19. Craig Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.
20. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.
21. Yong Ho Hwang, Dae Hyun Yum, and Pil Joong Lee. Timed-release encryption with pre-open capability and its application to certified e-mail system. In *ISC*, pages 344–358, 2005.
22. Himanshu Khurana and Hyung-Seok Hahm. Certified mailing lists. In *ASIACCS*, pages 46–58, 2006.
23. Himanshu Khurana, Jin Heo, and Meenal Pant. From proxy encryption primitives to a deployable secure-mailing-list solution. In *ICICS*, pages 260–281, 2006.
24. Himanshu Khurana, Adam J. Slagell, and Rafael Bonilla. SELS: a secure e-mail list service. In *SAC*, pages 306–313, 2005.
25. Benoît Libert and Damien Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. In *Public Key Cryptography*, pages 360–379, 2008.
26. Timothy C. May. Time-release crypto. Unpublished manuscript, 1993.
27. Takeo Mizuno and Hiroshi Doi. Hybrid proxy re-encryption scheme for attribute-based encryption. In *INSCRYPT*, pages 385–399, 2009.
28. Yasumasa Nakai, Takahiro Matsuda, Wataru Kitada, and Kanta Matsuura. A generic construction of timed-release encryption with pre-open capability. In *IWSEC*, pages 53–70, 2009.
29. Isamu Teranishi, Takuro Oyama, and Wakaha Ogata. General conversion for obtaining strongly existentially unforgeable signatures. *IEICE Transactions*, 91-A(1):94–106, 2008.
30. Wikipedia. Cloud computing. `http://en.wikipedia.org/wiki/Cloud_computing`.