Computer Security Symposium 2014
22 - 24 October 2014

条件付き代理人再暗号化における代理人の信頼の軽減

翁 健 †‡　　穴田 啓晃 ♯　　櫻井 幸一 ‡♯

† 暨南大学計算機科学科
510632 中国広東省広州市黄埔大道西 601 号
cryptjweng@gmail.com

‡ 九州大学大学院システム情報科学研究院
819-0395　福岡市西区元岡 744 番地
sakurai@inf.kyushu-u.ac.jp

♯ 公益財団法人九州先端科学技術研究所
814-0001　福岡県福岡市早良区 2-1-22 福岡 SRP センタービル 7 階
anada@isit.or.jp

あらまし 条件付き代理人再暗号化 (CPRE) は有用な暗号プリミティブである．CPRE は，指定された代理人が，何らかの具体的な条件を満足する委託者の暗号文を，受任者向け暗号文へと変換することを可能にする．ただし代理人は根底にある平文については何ら知識を知ることはない．しかしながら，現在ある CPRE は代理人によってなされる変換の正当性を保証しえない．また，単一故障点の問題を被る．CPRE における単一の代理人への信用を軽減するため，本論文では，我々は閾値条件付き代理人再暗号化 (TCPRE) という新しい変種を導入する．TCPRE では，委託者のため，n 人の代理人の内の t 人が（何らかの明示された条件を満足する）暗号文を変換できる（t-1 人まででではできない）．なおかつ，変換の正当性はパブリックに検証されうる．我々はまた，提案スキームが選択暗号文攻撃に対し安全であることを証明する．

Alleviating the Trust of the Proxy in Conditional Proxy Re-Encrypton

Jian Weng†‡　　Hiroaki Anada♯　　Kouichi Sakurai‡♯

†Department of Computer Science, Jinan University
601 Huangpu W Ave, Tianhe, Guangzhou, Guangdong, 510632, CHINA
cryptjweng@gmail.com

‡Faculty of Information Science and Electrical Engineering, Kyushu University
744, Motooka, Nishi-ku, Fukuoka-city, 819-0395, JAPAN
sakurai@inf.kyushu-u.ac.jp

♯ Institute of Systems, Information Technologies and Nanotechnologies (ISIT)
Fukuoka SRP Center Building 7F, 2-1-22, Momochihama, Sawara-ku, Fukuoka, 814-0001, JAPAN
anada@isit.or.jp

Abstract Conditional proxy re-encryption (CPRE) is a useful cryptographic primitive, which allows a designated proxy to transform the delegator's ciphertexts satisfying some specific conditions into the ciphertexts intended for the delegatee, while the proxy knows nothing about the underlying plaintexts. However, existing CPREs cannot guarantee the correctness of the transformation done by the proxy, and they also suffer from the problem of single point of failure. To alleviate the trust on the single proxy in CPRE, in this paper, we introduce a new primitive named threshold conditional proxy re-encryption (TCPRE), in which t out of n proxies can transform ciphertexts (satisfying some specified conditions) for the delegator (while up to $t-1$ proxies cannot), and the correctness of the transformation can be publicly verified. We formalize the security models for TCPRE, and propose a concrete TCPRE scheme. We also prove that the proposed scheme is secure against chosen-ciphertext attacks.

1 Introduction

Proxy re-encryption (PRE), introduced by Blaze et al. [3] in Eurocrypt'98, is a cryptographic primitive which enables a proxy to transform a ciphertext under a delegator's public key into another ciphertext under the delegatee's public key, without learning anything about the content of the encrypted message. In the past recent years, PRE has found many applications, such as encrypted email forwarding, secure distributed file systems, and outsourced filtering of encrypted spam.

To illustrate the useful applications of PRE, let's take the data sharing in cloud computing as an example. Nowadays, many users store data to the cloud, e.g., Dropbox and Google Docs. To protect the secrecy of their data, the users might first encrypt the data and then upload the ciphertext to the cloud. However, if the users use traditional public key encryption schemes to encrypt the data, there might exist some shortcomings. For example, suppose Alice wants to share the data with Bob, then she has to first download the ciphertexts from the cloud, decrypts them with her own secret key, encrypts again the data under Bob's public key, and finally uploads the new ciphertext to the cloud. Then Bob can access the data with his secret key. However, such a solution is highly unsatisfactory, since it introduces heavy computational cost and communication overhead. Fortunately, PRE schemes can be used to efficiently resolve this problem: Alice uses a PRE scheme to encrypt her data and then upload the ciphertext to the cloud. When she wants to share the data with Bob, she can simply give a re-encryption key $rk_{A\to B}$ to a proxy in the cloud, and then the latter can efficiently transform these ciphertexts into the ciphertexts intended for Bob, who can then decrypt the ciphertext to obtain the data with his own secret key.

Nevertheless, there exist some situations which are hard for traditional PRE to tackle. Let's take again the above cloud application as an example. Suppose some of Alice's ciphertexts are *highly secret*, and she wants to decrypt these ciphertexts *only* by herself. Unfortunately, in the above scenario, with the re-encryption key $rk_{A\to B}$, the proxy can transform *all* of Alice's ciphertexts, including the highly secret ones, and thus Bob can decrypt them to obtain these highly secret data. To address this problem, conditional proxy re-encryption (CPRE) were introduced in [18, 20]. In a CPRE scheme, ciphertexts are generated associated with a certain condition, and the proxy can translate those ciphertexts satisfying the specified condition. As to the above cloud application, with CPREs, Alice can control the proxy to transform only those non-highly-secret ciphertexts.

Our Motivations: Compared with traditional PRE, CPRE enables the delegator to implement fine-grained delegation of decryption rights. However, there still exist some problems hard for CPRE to deal with. One problem is that CPRE cannot guarantee the correctness of the transformation done by the proxy. This is indeed a challenge in applications. For example, in the pay-per-use model cloud computing service, the proxy charges the customer (e.g., Alice) for the transformation numbers. For saving the time and the computational cost, the proxy might simply return ciphertexts which are not really generated via the re-encryption algorithm. Unfortunately, existing CPREs cannot enable the users to check such a malicious behavior of the proxy. In addition, existing CPREs only involve a single proxy. This inevitably faces with the single point of failure problem: if the proxy is out of work, the delegatee cannot access the data any more. Thus some solutions should be introduced to deal with these problems for CPRE.

Our Contributions: To deal with the above problems, in this paper we introduce a variant of CPRE named threshold conditional proxy re-encryption (TCPRE). The ciphertext in TCPRE is also associated with specified conditions, and the proxies can only successfully transform those ciphertexts satisfying specified conditions. Unlike CPRE, TCPRE involves a number n of proxies, and t out of n proxies can successfully transform ciphertexts, while up to $t-1$ proxies cannot. In addition, the correctness of the transformation done by each proxy can be publicly verified. We formalize the chosen-ciphertext security models for TCPRE with respect to two types of ciphertexts (i.e., original ciphertexts and transformed ciphertexts). Then we present a concrete TCPRE scheme, and prove its chosen-ciphertext security under the formalized security model.

1.1 Related Work

In 1997, Mambo and Okamoto [15] initially introduced the concept of delegation of decryption rights, as a better-performance alternative to the trivial approach of decrypting-then-encrypting of ciphertexts. In Eurocrypt'98, Blaze et al. [3] introduced the concept of proxy re-encryption, and presented the first bidirectional PRE scheme [1]. In NDSS'05, Ateniese *et al.* [2] presented unidirectional PRE schemes secure against chosen-plaintext attacks (CPA). The first chosen-ciphertext secure bidirectional PRE scheme and unidirectional PRE scheme are proposed by Canetti and Hohenberger [5] and Libert and Vergnaud [14] respectively, and both schemes rely on the bilinear pairings. Deng et al. [9, 21] proposed a CCA-secure bidirectional PRE scheme without pairings. Shao and Cao [17] tried to propose a unidirectional PRE scheme without pairings, and later was improved by Chow et al. [7]. Subsequently, several bidirectional and unidirectional PRE schemes are proposed, e.g., [11, 16, 19, 22]. Proxy re-encryption has also been studied in identity-based scenarios, such as [8, 10].

Several variants of PRE have also been proposed in the past few years. Libert and Vergnaud [13] introduced the notion of traceable proxy re-encryption, in which a proxy who leaks its re-encryption key can be identified by the delegator. Ateniese et al. [1] introduced the concept of key-private proxy re-encryption, in which the anonymity of the sender and receiver's identities can be protected. In TCC'12, Chandran et al [6] introduced the notion of functional re-encryption, which can transform an encryption of a message m under an "input public key" pk into an encryption of the same message m under one of the n output public keys, namely the public key index by function $F(m)$.

2 Preliminaries

2.1 Lagrange Interpolation

Let $f(x) = \sum_{i=0}^{t-1} a_i x^i$ be a $(t-1)$-degree polynomial over $\mathbb{Z}_q$. Then given t pairwise distinct points $\{(z, f(z))\}_{z\in S}$ with $|S| = t$, one can reconstruct this polynomial $f(x)$ as

$$f(x) = \sum_{z\in S} (f(z) \cdot \lambda_{z,S}(x)),$$

where $\lambda_{z,S}(x) = \prod_{\substack{v\in S\\ v\neq z}} \frac{x-v}{z-v}$.

2.2 Bilinear Pairings

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic multiplicative groups with the same prime order p. A bilinear pairing is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties: (1) Bilinearity: $\forall g_1, g_2 \in \mathbb{G}, \forall a, b \in \mathbb{Z}_p^*$, we have $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$; (2) Non-degeneracy: There exist $g_1, g_2 \in \mathbb{G}$ such that $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ is the identity element in group $\mathbb{G}_T$; (3) Computability: There exists an efficient algorithm to compute $e(g_1, g_2)$ for $\forall g_1, g_2 \in \mathbb{G}$.

2.3 Complexity Assumption

The q-weak decisional bilinear Diffie-Hellman inversion (q-wDBDHI) problem [4] in groups $(\mathbb{G}, \mathbb{G}_T)$ is, given $(g, g^a, \cdots, g^{a^q}, g^b, Z) \in \mathbb{G}^{q+2} \times \mathbb{G}_T$ with unknown $a, b \xleftarrow{\$} \mathbb{Z}_p^*$, to decide whether $Z = e(g, g)^{b/a}$. Below we give an equivalent formulation of the q-wDBDHI problem. Due to the space limit, please refer to our full paper for the proof of the equivalence.

Lemma 1 *The q-wDBDHI problem is equivalent to, given* $(g, g^{1/a}, g^a, \cdots, g^{a^{q-1}}, g^b, Z) \in \mathbb{G}^{q+2} \times \mathbb{G}_T$ *as input, decide whether Z equals* $e(g, g)^{\frac{b}{a^2}}$ *or a random value.*

Definition 1 *For an algorithm* $\mathcal{B}$*, we define its advantage* $\mathrm{Adv}_{\mathcal{B}}^{q\text{-wDBDHI}}$ *in solving the* q-wDBDHI *problem as*

$$\Big| \Pr[\mathcal{B}(g, g^{1/a}, g^a, \cdots, g^{a^{q-1}}, g^b, e(g, g)^{\frac{b}{a^2}}) = 1] \\ -\Pr[\mathcal{B}(g, g^{1/a}, g^a, \cdots, g^{a^{q-1}}, g^b, e(g, g)^z) = 1]\Big|,$$

where the probability is over the random choices of $g \in \mathbb{G}, a, b, z \in \mathbb{Z}_p^*$*, and the random bits consumed by* $\mathcal{B}$*. We say that the* (t, ϵ)-q-wDBDHI *assumption holds in* $(\mathbb{G}, \mathbb{G}_T)$ *if no t-time algorithm* $\mathcal{B}$ *has advantage at least* ϵ *in solving the* q-wDBDHI *problem in* $(\mathbb{G}, \mathbb{G}_T)$.

[1] In bidirectional PRE, the delegation from Alice to Bob also allows re-encryption from Bob to Alice. In contrast, unidirectional PRE only allows the delegation of one direction.

Based on the q-wDBDHI assumption, Boneh et al. [4] have constructed a hierarchical identity-based encryption scheme with constant ciphertext size. Libert and Verganaud [12, 14] also constructed a unidirectional proxy re-encryption scheme under the 3-wDBDHI assumption. In this paper, to prove the security for our proposed TCPRE scheme, we also only use the above assumption for constant values of q, i.e., we only use the 1-wDBDHI and 4-wDBDHI assumptions.

3 Framework of TCPRE

3.1 Definition

A TCPRE scheme is defined by the following algorithms:

$\mathsf{Setup}(\kappa)$: The global setup algorithm takes as input a security parameter κ, and outputs the global parameters $param$.

$\mathsf{KeyGen}(param)$: Each user i uses this key generation algorithm to generate a public/private key pair $(\mathtt{pk}_i, \mathtt{sk}_i)$.

$\mathsf{ReKeyGen}(\mathtt{sk}_i, \mathtt{pk}_j, \mathtt{w}, t, n)$: On input the delegator's secret key $\mathtt{sk}_i$, the delegatee's public key $\mathtt{pk}_j$, a condition $\mathtt{w}$, a number n of proxies and a threshold t with $1 \le t \le n$, this algorithm generates n shares of re-encryption keys $\{\mathtt{rk}_{i\overset{\mathtt{w}}{\to}j,v}\}_{v=1}^{n}$ and verification keys $\{\mathtt{vk}_{i\overset{\mathtt{w}}{\to}j,v}\}_{v=1}^{n}$. For each $v \in \{1, \cdots, n\}$, return the v-th re-encryption key share $\mathtt{rk}_{i\overset{\mathtt{w}}{\to}j,v}$ to the v-th proxy, and make all the verification shares $\{\mathtt{vk}_{i\overset{\mathtt{w}}{\to}j,v}\}_{v=1}^{n}$ public.

$\mathsf{Encrypt}(\mathtt{pk}_i, m, \mathtt{w})$: On input a public key $\mathtt{pk}_i$ and a plaintext $m \in \mathcal{M}$ (here $\mathcal{M}$ denotes the plaintext space) and a condition $\mathtt{w}$, this encryption algorithm outputs an original ciphertext CT_i.

$\mathsf{ReEncShare}(\mathtt{rk}_{i\overset{\mathtt{w}}{\to}j,v}, \mathrm{CT}_i)$: On input an original ciphertext CT_i, and a v-th re-encryption key share $\mathtt{rk}_{i\overset{\mathtt{w}}{\to}j,v}$, this algorithm outputs a v-th re-encryption share $\theta_{i\overset{\mathtt{w}}{\to}j,v}$.

$\mathsf{ShareVerify}(\mathrm{CT}_i, \mathtt{vk}_{i\overset{\mathtt{w}}{\to}j,v}, \theta_{i\overset{\mathtt{w}}{\to}j,v})$: On input an original ciphertext CT_i, the v-th verification key $\mathtt{vk}_{i\overset{\mathtt{w}}{\to}j,v}$ and the v-th re-encryption share $\theta_{i\overset{\mathtt{w}}{\to}j,v}$, this algorithm outputs 1 if $\theta_{i\overset{\mathtt{w}}{\to}j,v}$ is a valid re-encryption share; otherwise, it outputs 0 indicating $\theta_{i\overset{\mathtt{w}}{\to}j,v}$ invalid.

$\mathsf{ShareCombine}(\mathrm{CT}_i, \{\theta_{i\overset{\mathtt{w}}{\to}j,v}\}_{v\in S}, \{\mathtt{vk}_{i\overset{\mathtt{w}}{\to}j,v}\}_{v\in S})$: Given an original ciphertext CT_i, and a set $\{\theta_{i\overset{\mathtt{w}}{\to}j,v}\}_{v\in S}$ of valid re-encryption shares with $|S| \ge t$ (for convenience, we assume $|S| = t$), this algorithm outputs a transformed ciphertext CT_j.

$\mathsf{Decrypt}(\mathtt{sk}_i, \mathrm{CT}_i)$: On input a private key $\mathtt{sk}_i$ and a (original or transformed) ciphertext CT_i, this algorithm outputs m if CT_i is a valid ciphertext; otherwise, it outputs $\perp$.

Roughly speaking, the correctness requires that, for any $m \in \mathcal{M}$, any condition $\mathtt{w}$ in proper space and any $(\mathtt{pk}_i, \mathtt{sk}_i) \leftarrow \mathsf{KeyGen}(param)$, $(\mathtt{pk}_j, \mathtt{sk}_j) \leftarrow \mathsf{KeyGen}(param)$, $\{\mathtt{rk}_{i\overset{\mathtt{w}}{\to}j,v}, \mathtt{vk}_{i\overset{\mathtt{w}}{\to}j,v}\}_{v=1}^{n} \leftarrow \mathsf{ReKeyGen}(\mathtt{sk}_i, \mathtt{pk}_j, \mathtt{w}, t, n)$ and $\mathrm{CT}_i \leftarrow \mathsf{Encrypt}(\mathtt{pk}_i, m, \mathtt{w})$, the equality $\mathsf{Decrypt}(\mathtt{sk}_i, \mathrm{CT}_i) = m$ should hold. Also, for $\theta_{i\overset{\mathtt{w}}{\to}j,v} \leftarrow \mathsf{ReEncShare}(\mathtt{rk}_{i\overset{\mathtt{w}}{\to}j,v}, \mathrm{CT}_i)$ with $v \in \{1, \cdots, n\}$, $\mathsf{ShareVerify}(\mathrm{CT}_i, \mathtt{vk}_{i\overset{\mathtt{w}}{\to}j,v}, \theta_{i\overset{\mathtt{w}}{\to}j,v}) = 1$ should hold. In addition, for any set $S \subseteq \{1, \cdots, n\}$ such that $|S| = t$, $\mathsf{Decrypt}(\mathtt{sk}_j, \mathsf{ShareCombine}(\mathrm{CT}_i, \left\{\theta_{i\overset{\mathtt{w}}{\to}j,v}\right\}_{v\in S}, \{\mathtt{vk}_{i\overset{\mathtt{w}}{\to}j,v}\}_{v\in S})) = m$ should hold.

3.2 Security Models for TCPRE

Before giving the security notions for TCPRE, we here introduce the following oracles which will be used to model the abilities of an adversary:

- *Uncorrupted key generation query* $\mathcal{O}_{\mathrm{u}}(\cdot)$: On input an index i, this oracle runs algorithm KeyGen to obtain a public/private key pair $(\mathtt{pk}_i, \mathtt{sk}_i)$, and returns $\mathtt{pk}_i$.
- *Corrupted key generation query* $\mathcal{O}_{\mathrm{c}}(\cdot)$: On input an index j, this oracle runs algorithm KeyGen to obtain a public/private key pair $(\mathtt{pk}_j, \mathtt{sk}_j)$, and returns $(\mathtt{pk}_j, \mathtt{sk}_j)$ to $\mathcal{A}$.
- *Re-encryption key share query* $\mathcal{O}_{\mathrm{rks}}(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot)$: On input $(\mathtt{pk}_i, \mathtt{pk}_j, \mathtt{w}, t, n, U)$ with $1 \le t \le n$ and $U \subseteq \{1, \cdots, n\}$, it runs $\{\mathtt{rk}_{i\overset{\mathtt{w}}{\to}j,v}, \mathtt{vk}_{i\overset{\mathtt{w}}{\to}j,v}\}_{v=1}^{n} \leftarrow \mathsf{ReKeyGen}(\mathtt{sk}_i, \mathtt{pk}_j, \mathtt{w}, t, n)$, and returns $\{\mathtt{rk}_{i\overset{\mathtt{w}}{\to}j,v}\}_{v\in U}$ and $\{\mathtt{vk}_{i\overset{\mathtt{w}}{\to}j,v}\}_{v=1}^{n}$.
- *Re-encryption share query* $\mathcal{O}_{\mathrm{res}}(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot,\cdot)$: On input $(\mathtt{pk}_i, \mathtt{pk}_j, \mathtt{w}, t, n, \mathrm{CT}_i, L)$ with $L \subseteq \{1, \cdots, n\}$, it returns $\{\mathsf{ReEncShare}(\mathtt{rk}_{i\overset{\mathtt{w}}{\to}j,v}, \mathrm{CT}_i)\}_{v\in L}$.

- *Decryption query* $\mathcal{O}_{\rm dec}(\cdot,\cdot)$: On input $(\mathtt{pk}_z, \mathrm{CT}_z)$, it returns the result of $\mathsf{Decrypt}(\mathtt{sk}_z, \mathrm{CT}_z)$ to $\mathcal{A}$. Here $\mathtt{sk}_z$ is the private key with respect to $\mathtt{pk}_z$.

3.2.1 Original Ciphertext Security

The chosen-ciphertext security for a TCPRE scheme can be defined via the following experiment between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

For an adversary $\mathcal{A}$ running in stages $\mathtt{find}$ and $\mathtt{guess}$, we define its advantage $\mathrm{Adv}^{\textsf{IND-oCCA}}_{\mathrm{TCPRE},\mathcal{A}}(k)$ as $\big|\Pr[\mathbf{Exp}^{\textsf{IND-oCCA-1}}_{\mathrm{TCPRE},\mathcal{A}}(k) = 1] - \Pr[\mathbf{Exp}^{\textsf{IND-oCCA-0}}_{\mathrm{TCPRE},\mathcal{A}}(k) = 1]\big|$, where $\mathbf{Exp}^{\textsf{IND-oCCA-}\delta}_{\mathrm{PRE},\mathcal{A}}$ is defined by the following game:

```
**Experiment** $\mathbf{Exp}^{\textsf{IND-oCCA-}\delta}_{\mathrm{TCPRE},\mathcal{A}}(k)$
    $param \leftarrow \mathsf{setup}(1^k)$;
    $(m_0, m_1, \mathtt{pk}_{i^*}, \mathtt{w}^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\rm u},\mathcal{O}_{\rm c},\mathcal{O}_{\rm rks},\mathcal{O}_{\rm res},\mathcal{O}_{\rm dec}}(\mathtt{find}, param)$;
    $\mathrm{CT}^* \leftarrow \mathsf{Encrypt}(\mathtt{pk}_{i^*}, m_\delta, \mathtt{w}^*)$;
    $\delta' \leftarrow \mathcal{A}^{\mathcal{O}_{\rm u},\mathcal{O}_{\rm c},\mathcal{O}_{\rm rks},\mathcal{O}_{\rm res},\mathcal{O}_{\rm dec}}(\mathtt{guess}, param, \mathrm{CT}^*)$;
    return $\delta'$.
```

During the above experiment, it is required that the following requirements should be simultaneously satisfied: (1) $\mathtt{pk}_{i^*}$ is generated by oracle $\mathcal{O}_{\rm u}$; (2) For a public key $\mathtt{pk}_j$ generated by $\mathcal{O}_{\rm c}$, a number n and the threshold t with $1 \le t \le n$, $\mathcal{A}$ cannot issue $\mathcal{O}_{\rm rks}(\mathtt{pk}_{i^*}, \mathtt{pk}_j, \mathtt{w}^*, t, n, U)$ with $|U| \ge t$; (3) For a public key $\mathtt{pk}_j$ generated by $\mathcal{O}_{\rm c}$, a number n and the threshold t with $1 \le t \le n$, $\mathcal{A}$ cannot issue queries $\mathcal{O}_{\rm rks}(\mathtt{pk}_{i^*}, \mathtt{pk}_j, \mathtt{w}^*, t, n, U)$ and $\mathcal{O}_{\rm res}(\mathtt{pk}_{i^*}, \mathtt{pk}_j, \mathtt{w}^*, t, n, \mathrm{CT}^*, L)$ such that $|U \cup L| \ge t$; (4) $\mathcal{A}$ cannot issue query $\mathcal{O}_{\rm dec}(\mathtt{pk}_{i^*}, \mathrm{CT}^*)$; (5) For a public key $\mathtt{pk}_j$ generated by $\mathcal{O}_{\rm u}$, a number n and the threshold t with $1 \le t \le n$, if $\mathcal{A}$ has issued queries $\mathcal{O}_{\rm rks}(\mathtt{pk}_{i^*}, \mathtt{pk}_j, \mathtt{w}^*, t, n, U)$ and $\mathcal{O}_{\rm res}(\mathtt{pk}_{i^*}, \mathtt{pk}_j, \mathtt{w}^*, t, n, \mathrm{CT}^*, L)$ such that $|U \cup L| \ge t$, then $\mathcal{A}$ cannot issue query $\mathcal{O}_{\rm dec}(\mathtt{pk}_j, \mathrm{CT}_j)$ such that $\mathsf{Decrypt}(\mathtt{sk}_j, \mathrm{CT}_j) \in \{m_0, m_1\}$.

Definition 2 *A bidirectional PRE scheme is said to be* $(t, q_{\rm u}, q_{\rm c}, q_{\rm rks}, q_{\rm res}, q_{\rm d}, \epsilon)$-$\textsf{IND-oCCA}$-*secure, if for any t-time adversary* $\mathcal{A}$ *who asks at most* $q_{\rm u}, q_{\rm c}, q_{\rm rks}, q_{\rm res}$ *and* $q_{\rm d}$ *queries to oracles* $\mathcal{O}_{\rm u}, \mathcal{O}_{\rm c}, \mathcal{O}_{\rm rks}, \mathcal{O}_{\rm res}$ *and* $\mathcal{O}_{\rm d}$*, respectively, we have* $\mathrm{Adv}^{\textsf{IND-oCCA}}_{\mathrm{TCPRE},\mathcal{A}}(k) \le \epsilon$.

3.2.2 CCA-security of transformed ciphertexts

For an adversary $\mathcal{A}$ running in stages $\mathtt{find}$ and $\mathtt{guess}$, we define its advantage $\mathrm{Adv}^{\textsf{IND-tCCA}}_{\mathrm{TCPRE},\mathcal{A}}(k)$ as $\big|\Pr[\mathbf{Exp}^{\textsf{IND-tCCA-1}}_{\mathrm{TCPRE},\mathcal{A}}(k) = 1] - \Pr[\mathbf{Exp}^{\textsf{IND-tCCA-0}}_{\mathrm{TCPRE},\mathcal{A}}(k) = 1]\big|$, where $\mathbf{Exp}^{\textsf{IND-tCCA-}\delta}_{\mathrm{PRE},\mathcal{A}}$ is defined by the following game:

```
**Experiment** $\mathbf{Exp}^{\textsf{IND-tCCA-}\delta}_{\mathrm{TCPRE},\mathcal{A}}(k)$
    $param \leftarrow \mathsf{setup}(1^k)$;
    $(m_0, m_1, \mathtt{pk}_i, \mathtt{pk}_{j^*}, \mathtt{w}^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\rm u},\mathcal{O}_{\rm c},\mathcal{O}_{\rm rks},\mathcal{O}_{\rm dec}}(\mathtt{find}, param)$;
    $\mathrm{CT} \leftarrow \mathsf{Encrypt}(\mathtt{pk}_i, m_\delta, \mathtt{w}^*)$;
    $\{\theta_{i\xrightarrow{\mathtt{w}^*}j^*,v} \leftarrow \mathsf{ReEncShare}(\mathtt{rk}_{i\xrightarrow{\mathtt{w}^*}j^*,v}, \mathrm{CT})\}^t_{v=1}$
    $\mathrm{CT}^* \leftarrow \mathsf{ShareCombine}(\mathrm{CT}_i, \{\theta_{i\xrightarrow{\mathtt{w}^*}j^*,v}\}^t_{v=1}, \{\mathtt{rk}_{i\xrightarrow{\mathtt{w}^*}j^*,v}\}^t_{v=1})$;
    $\delta' \leftarrow \mathcal{A}^{\mathcal{O}_{\rm u},\mathcal{O}_{\rm c},\mathcal{O}_{\rm rks},\mathcal{O}_{\rm dec}}(\mathtt{guess}, param, \mathrm{CT}^*)$;
    return $\delta'$.
```

During the above experiment, it is required that the following requirements should be simultaneously satisfied: (1) $\mathtt{pk}_{j^*}$ is generated by oracle $\mathcal{O}_{\rm u}$; (2) $\mathcal{A}$ cannot issue query $\mathcal{O}_{\rm dec}(\mathtt{pk}_{j^*}, \mathrm{CT}^*)$.

Definition 3 *A bidirectional PRE scheme is said to be* $(t, q_{\rm u}, q_{\rm c}, q_{\rm rks}, q_{\rm d}, \epsilon)$-$\textsf{IND-tCCA}$-*secure, if for any t-time adversary* $\mathcal{A}$ *who asks at most* $q_{\rm u}, q_{\rm c}, q_{\rm rks}$ *and* $q_{\rm d}$ *queries to oracles* $\mathcal{O}_{\rm u}, \mathcal{O}_{\rm c}, \mathcal{O}_{\rm rks}$ *and* $\mathcal{O}_{\rm d}$*, respectively, we have* $\mathrm{Adv}^{\textsf{IND-tCCA}}_{\mathrm{TCPRE},\mathcal{A}}(k) \le \epsilon$.

4 Proposed TCPRE Scheme

4.1 Construction

Our proposed TCPRE scheme consists of the following algorithms:

$\mathsf{Setup}(\kappa)$: Given a security parameter κ, choose bilinear map groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $q > 2^\kappa$, and pick generator $g \overset{\$}{\leftarrow} \mathbb{G}$. In addition, choose hash functions $H_1 : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \mathbb{Z}_p^*, H_2 : \mathbb{G}_T \to \{0,1\}^{l_1} \times \{0,1\}^{l_2}, H_3 : \mathbb{G} \times \{0,1\}^* \to \mathbb{G}$ and $H_4 : \{0,1\}^{l_1+l_2} \times \mathbb{G} \to \mathbb{G}$. The global parameters are $param = (\mathbb{G}, \mathbb{G}_T, g, H_1, H_2, H_3, H_4)$.

$\mathsf{KeyGen}(param)$: To generate a public/private key pair for user i, this algorithm picks $x_i \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, and sets $\mathtt{pk}_i = g^{x_i}$ and $\mathtt{sk}_i = x_i$.

$\mathsf{ReKeyGen}(\mathtt{sk}_i, \mathtt{pk}_j, \mathtt{w}, t, n)$: On input the delegator's secret key $\mathtt{sk}_i$, the delegatee's public key $\mathtt{pk}_j$, a condition $\mathtt{w}$, a number n of proxies and a threshold t with $1 \le t \le n$, this algorithm performs the following steps:

1. For each index $v \in \{1, \cdots, t-1\}$, pick $\alpha_v, \beta_v \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, and set the v-th re-encryption key share

$$\mathtt{rk}_{i\xrightarrow{\mathtt{w}}j,v} = g^{\alpha_v} H_3(\mathtt{pk}_i, \mathtt{w})^{\beta_v}, \qquad (1)$$

and the v-th verification key share $\mathtt{vk}_{i\xrightarrow{\mathtt{w}}j,v}$ to be

$$(vk_{v,1}, vk_{v,2}) = (\mathtt{pk}_i^{\alpha_v}, \mathtt{pk}_i^{\beta_v}). \tag{2}$$

2. Let $S' = \{0, 1, \cdots, t-1\}$. Pick $\beta \xleftarrow{\$} \mathbb{Z}_p^*$. For each remaining index $v \in \{t, \cdots, n\}$, set the v-th re-encryption key share $\mathtt{rk}_{i\xrightarrow{\mathtt{w}}j,v}$ to be

$$\left(\mathtt{pk}_j^{\frac{1}{\mathtt{sk}_i}} H_3(\mathtt{pk}_i, \mathtt{w})^{\beta}\right)^{\lambda_{0,S'}(v)} \prod_{z=1}^{t-1} \mathtt{rk}_{i\xrightarrow{\mathtt{w}}j,z}^{\lambda_{z,S'}(v)}, \tag{3}$$

and the v-th verification key share $\mathtt{vk}_{i\xrightarrow{\mathtt{w}}j,v} = (vk_{v,1}, vk_{v,2})$ to be

$$vk_{v,1} = \mathtt{pk}_j^{\lambda_{0,S'}(v)} \prod_{z=1}^{t-1} vk_{z,1}^{\lambda_{z,S'}(v)}, \tag{4}$$

$$vk_{v,2} = \left(\mathtt{pk}_i^{\beta}\right)^{\lambda_{0,S'}(v)} \prod_{z=1}^{t-1} vk_{z,2}^{\lambda_{z,S'}(v)}. \tag{5}$$

3. For each $v \in \{1, \cdots, n\}$, give the v-th re-encryption key share $\mathtt{rk}_{i\xrightarrow{\mathtt{w}}j,v}$ to the v-th proxy, and make the verification key shares $\{\mathtt{vk}_{i\xrightarrow{\mathtt{w}}j,v}\}_{v=1}^{n}$ public. Note that the proxy can check the validity of the v-th re-encryption key share $\mathtt{rk}_{i\xrightarrow{\mathtt{w}}j,v}$ by testing whether $e(\mathtt{rk}_{i\xrightarrow{\mathtt{w}}j,v}, \mathtt{pk}_i) = e(g, vk_{v,1})$ $e(H_3(\mathtt{pk}_i, \mathtt{w}), vk_{v,2})$ holds.

Note that the re-encryption key shares and verification key shares in Eq. (3)-(5) have the same form as those in Eqs. (1)-(2). Due to the space limit, please refer to our full paper for the proof.

$\mathsf{Encrypt}(\mathtt{pk}_i, m, \mathtt{w})$: Given a plaintext $m \in \{0,1\}^{l_1}$, a public key $\mathtt{pk}_i$ and a condition $\mathtt{w}$, the sender picks $r' \xleftarrow{\$} \{0,1\}^{l_2}$, computes $r = H_1(m, r')$, and outputs the original ciphertext $\mathrm{CT}_i = (C_1, C_2, C_3, C_4)$ as

$$C_1 = \mathtt{pk}_i^r, C_2 = H_2(e(g,g)^r) \oplus (m \| r'),$$
$$C_3 = H_3(\mathtt{pk}_i, \mathtt{w})^r, C_4 = H_4(C_2, C_3)^r.$$

$\mathsf{ReEncShare}(\mathtt{rk}_{i\xrightarrow{\mathtt{w}}j,v}, \mathrm{CT}_i)$: To generate the v-th re-encryption share of an original ciphertext CT_i, the v-th proxy with the v-th re-encryption key $\mathtt{rk}_{i\xrightarrow{\mathtt{w}}j,v}$ performs the following steps:

1. Check whether the following equalities hold. If no, output $\perp$ and terminate indicating CT_i invalid.

$$e(C_1, H_3(\mathtt{pk}_i, \mathtt{w})) = e(\mathtt{pk}_i, C_3), \tag{6}$$
$$e(C_1, H_4(C_2, C_3)) = e(\mathtt{pk}_i, C_4). \tag{7}$$

2. Pick $s \xleftarrow{\$} \mathbb{Z}_p^*$ and compute $\bar{C}_{v,1} = \mathtt{pk}_i^s, \bar{C}_{v,2} = \mathtt{rk}_{i\xrightarrow{\mathtt{w}}j,v}^{\frac{1}{s}}, \bar{C}_{v,3} = C_1^s$. Finally, output the v-th re-encryption share $\theta_{i\xrightarrow{\mathtt{w}}j,v} = (\bar{C}_{v,1}, \bar{C}_{v,2}, \bar{C}_{v,3})$.

$\mathsf{ShareVerify}(\mathrm{CT}_i, \mathtt{vk}_{i\xrightarrow{\mathtt{w}}j,v}, \theta_{i\xrightarrow{\mathtt{w}}j,v})$: On input an original ciphertext $\mathrm{CT}_i = (C_1, C_2, C_3, C_4)$, a verification key $\mathtt{vk}_{i\xrightarrow{\mathtt{w}}j,v} = (vk_{i,1}, vk_{i,2})$ and a re-encryption share $\theta_{i\xrightarrow{\mathtt{w}}j,v} = (\bar{C}_{v,1}, \bar{C}_{v,2}, \bar{C}_{v,3})$, this algorithm checks whether the following equalities hold:

$$e(\bar{C}_{v,1}, C_1) = e(\mathtt{pk}_i, \bar{C}_{v,3}),$$
$$e(\bar{C}_{v,2}, \bar{C}_{v,1}) = e(g, vk_{v,1}) e(H_3(\mathtt{pk}_i, \mathtt{w}), vk_{v,2}).$$

If yes, output 1 indicating $\theta_{i\xrightarrow{\mathtt{w}}j,v}$ valid; otherwise, output 0 indicating $\theta_{i\xrightarrow{\mathtt{w}}j,v}$ invalid.

$\mathsf{ShareCombine}(\mathrm{CT}_i, \{\theta_{i\xrightarrow{\mathtt{w}}j,v}\}_{v\in S}, \{\mathtt{vk}_{i\xrightarrow{\mathtt{w}}j,v}\}_{v\in S})$: Given an original ciphertext $\mathrm{CT}_i = (C_1, C_2, C_3, C_4)$, a set of valid re-encryption shares $\{\theta_{i\xrightarrow{\mathtt{w}}j,v}\}_{v\in S}$ and verification key shares $\{\mathtt{vk}_{i\xrightarrow{\mathtt{w}}j,v}\}_{v\in S} = \{(vk_{v,1}, vk_{v,2})\}_{v\in S}$ with $|S| \geq t$ (for convenience, we assume $|S| = t$), this algorithm computes $C_1' = \prod_{v\in S}\left(\frac{e(\bar{C}_{v,2}, \bar{C}_{v,3})}{e(C_3, vk_{v,2})}\right)^{\lambda_{v,S}(0)}$, and outputs $\mathrm{CT}_j = (C_1', C_2)$.

Observe that C_1' is in fact the form of $e(\mathtt{pk}_j, g)^r$. Due to the space limit, please refer to our full paper for the explanation.

$\mathsf{Decrypt}(\mathtt{sk}_i, \mathrm{CT}_i)$: On input a private key $\mathtt{sk}_i$ and ciphertext CT_i, this algorithm works according to two cases:

- CT_i is an original ciphertext $\mathrm{CT}_i = (C_1, C_2, C_3, C_4)$: First check whether Eqs. (6)and (7) hold. If no, output $\perp$ and terminate indicating CT_i invalid. Else, compute $(m\|r') = H_2(e(C_1, g)^{\frac{1}{\mathtt{sk}_i}}) \oplus C_2$, and check whether $C_4 = H_4(C_2, C_3)^{H_1(m,r')}$ holds. If no, output $\perp$ indicating CT_i invalid; otherwise, output m.
- CT_i is a transformed ciphertext $\mathrm{CT}_i = (C_1', C_2)$: Compute $(m\|r') = C_2 \oplus H_2(C_1'^{\frac{1}{\mathtt{sk}_i}})$, and check whether $C_1' = e(g^{\mathtt{sk}_i}, g)^{H_1(m,r')}$ holds. If no, output $\perp$; otherwise, output m.

4.2 Security Analysis

Theorem 1 *Our proposed scheme is* $\mathsf{IND\text{-}oCCA}$ *secure in the random oracle model, assuming the 4-wDBDHI assumption holds in groups* $(\mathbb{G}, \mathbb{G}_T)$. *Concretely, if there exists an* $\mathsf{IND\text{-}oCCA}$ *adversary* $\mathcal{A}$, *who asks at most q_{H_i} random oracle queries to H_i with $i \in \{1, \cdots, 4\}$, and breaks the $(t, q_{\rm u}, q_{\rm c}, q_{\rm rks}, q_{\rm res}, q_{\rm d}, \epsilon)$-*IND-oCCA *security of our proposed scheme, then there exists an algorithm $\mathcal{B}$ which can solve the (t', ϵ')-4-wDBDHI problem in groups $(\mathbb{G}, \mathbb{G}_T)$ with*

$$\begin{aligned}\epsilon' &\geq \frac{\epsilon}{e(1+q_{\rm rks})} - \frac{q_{H_1}}{2^{l_2}} - \frac{q_{\rm rks}+q_{\rm res}+q_{\rm d}}{p},\\ t' &\leq t + \mathcal{O}\big(\tau(q_{H_3}+q_{H_4}+q_{\rm u}+q_{\rm c}+3n_{\max}q_{\rm rks}\\ &\qquad + (4q_{H_1}+3n_{\max})q_{\rm res}+4q_{H_1}q_{\rm d})\big).\end{aligned}$$

where τ is the maximum time among time for computing a multi-exponentiation and a pairing in $\mathbb{G}$,$\mathbb{G}_T$, and $n_{\max}$ denotes the maximal number of proxies.

Theorem 2 *Our proposed scheme is* IND-tCCA *secure in the random oracle model, assuming the 1-wDBDHI assumption holds in groups $(\mathbb{G}, \mathbb{G}_T)$. Concretely, if there exists an* IND-tCCA *adversary $\mathcal{A}$, who asks at most q_{H_i} random oracle queries to H_i with $i \in \{1, \cdots, 4\}$, and breaks the $(t, q_{\rm u}, q_{\rm c}, q_{\rm rks}, q_{\rm d}, \epsilon)$-*IND-tCCA *security of our proposed scheme, then there exists an algorithm $\mathcal{B}$ which can solve the (t', ϵ')-1-wDBDHI problem in groups $(\mathbb{G}, \mathbb{G}_T)$ with*

$$\begin{aligned}\epsilon' &\geq \epsilon - \frac{q_{H_1}}{2^{l_2}} - \frac{q_{\rm d}}{p},\\ t' &\leq t + \mathcal{O}\big(\tau(q_{H_3}+q_{H_4}+q_{\rm u}+q_{\rm c}\\ &\qquad + 3n_{\max}q_{\rm rks}+4q_{H_1}q_{\rm d})\big).\end{aligned}$$

where τ and $n_{\max}$ have the same meaning as in Theorem 1.

Due to the space limit, please refer to our full paper for the proofs of Theorem 1 and 2.

5 Conclusions

To alleviate the trust on the single proxy in conditional proxy re-encryption, we introduced the notion of threshold conditional proxy re-encryption (TCPRE), in which t out of n proxies can successfully transform ciphertexts, while up to $t-1$ proxies cannot. In addition, the correctness of the transformation done by each proxy can be publicly verified. We gave the formal definition and security models for TCPRE, and presented a concrete TCPRE scheme. The chosen-ciphertext security of the proposed scheme can be proved in the random oracle model.

6 Acknowledgements

The first autuhor was partially supported by the Invitation Programs for Foreign-based Researchers provided by NICT. The last author was partially supported by Grants-in-Aid for Scientific Research; Research Project Number: 25540004 of Japan Society for the Promotion of Science.

参考文献

[1] G. Ateniese, K. Benson, and S. Hohenberger. Key-Private Proxy Re-encryption. In M. Fischlin, editor, *CT-RSA*, volume 5473 of *LNCS*, pages 279–294. Springer, 2009.

[2] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage. In *NDSS*. The Internet Society, 2005.

[3] M. Blaze, G. Bleumer, and M. Strauss. Divertible Protocols and Atomic Proxy Cryptography. In *EUROCRYPT*, pages 127–144, 1998.

[4] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical Identity Based Encryption with Constant Size Ciphertext. In R. Cramer, editor, *EUROCRYPT*, volume 3494 of *LNCS*, pages 440–456. Springer, 2005.

[5] R. Canetti and S. Hohenberger. Chosen-Ciphertext Secure Proxy Re-Encryption. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM Conference on Computer and Communications Security*, pages 185–194. ACM, 2007.

[6] N. Chandran, M. Chase, and V. Vaikuntanathan. Functional re-encryption and collusion-resistant obfuscation. In R. Cramer, editor, *TCC*, volume 7194 of *Lecture Notes in Computer Science*, pages 404–421. Springer, 2012.

[7] S. S. M. Chow, J. Weng, Y. Yang, and R. H. Deng. Efficient unidirectional proxy re-encryption. In D. J. Bernstein and T. Lange,

editors, *AFRICACRYPT*, volume 6055 of *Lecture Notes in Computer Science*, pages 316–332. Springer, 2010.

[8] C.-K. Chu and W.-G. Tzeng. Identity-Based Proxy Re-encryption without Random Oracles. In J. A. Garay, A. K. Lenstra, M. Mambo, and R. Peralta, editors, *ISC*, volume 4779 of *LNCS*, pages 189–202. Springer, 2007.

[9] R. H. Deng, J. Weng, S. Liu, and K. Chen. Chosen-Ciphertext Secure Proxy Re-encryption without Pairings. In M. K. Franklin, L. C. K. Hui, and D. S. Wong, editors, *CANS*, volume 5339 of *LNCS*, pages 1–17. Springer, 2008.

[10] M. Green and G. Ateniese. Identity-Based Proxy Re-encryption. In J. Katz and M. Yung, editors, *ACNS*, volume 4521 of *LNCS*, pages 288–306. Springer, 2007.

[11] G. Hanaoka, Y. Kawai, N. Kunihiro, T. Matsuda, J. Weng, R. Zhang, and Y. Zhao. Generic Construction of Chosen Ciphertext Secure Proxy Re-Encryption. In O. Dunkelman, editor, *CT-RSA*, volume 7178 of *Lecture Notes in Computer Science*, pages 349–364. Springer, 2012.

[12] B. Libert and D. Vergnaud. Unidirectional Chosen-Ciphertext Secure Proxy Re- Encryption. *IEEE Transactions on Information Theory*, 57:1786–1802.

[13] B. Libert and D. Vergnaud. Tracing Malicious Proxies in Proxy Re-encryption. In S. D. Galbraith and K. G. Paterson, editors, *Pairing*, volume 5209 of *LNCS*, pages 332–353. Springer, 2008.

[14] B. Libert and D. Vergnaud. Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption. In R. Cramer, editor, *Public Key Cryptography*, volume 4939 of *LNCS*, pages 360–379. Springer, 2008.

[15] M. Mambo and E. Okamoto. Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E80-A(1):54–63, 1997.

[16] T. Matsuda, R. Nishimaki, and K. Tanaka. CCA Proxy Re-Encryption without Bilinear Maps in the Standard Model. In P. Nguyen and D. Pointcheval, editors, *PKC*, volume 6056 of *Lecture Notes in Computer Science*, page 261C278. Springer, 2010.

[17] J. Shao and Z. Cao. CCA-Secure Proxy Re-encryption without Pairings. In S. Jarecki and G. Tsudik, editors, *Public Key Cryptography*, volume 5443 of *LNCS*, pages 357–376. Springer, 2009.

[18] Q. Tang. Type-based proxy re-encryption and its construction. In D. R. Chowdhury, V. Rijmen, and A. Das, editors, *INDOCRYPT*, volume 5365 of *Lecture Notes in Computer Science*, pages 130–144. Springer, 2008.

[19] J. Weng, M.-R. Chen, Y. Yang, R. H. Deng, K. Chen, and F. Bao. Cca-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles. *SCIENCE CHINA Information Sciences*, 53(3):593–606, 2010.

[20] J. Weng, R. H. Deng, X. Ding, C.-K. Chu, and J. Lai. Conditional proxy re-encryption secure against chosen-ciphertext attack. In W. Li, W. Susilo, U. K. Tupakula, R. Safavi-Naini, and V. Varadharajan, editors, *ASIACCS*, pages 322–332. ACM, 2009.

[21] J. Weng, R. H. Deng, S. Liu, and K. Chen. Chosen-ciphertext secure bidirectional proxy re-encryption schemes without pairings. *Inf. Sci.*, 180(24):5077–5089, 2010.

[22] J. Weng, Y. Zhao, and G. Hanaoka. On the security of a bidirectional proxy re-encryption scheme from pkc 2010. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 284–295. Springer, 2011.