A previous version of this paper appeared in *Science China: Information Science*, 2010, 53(3): 593-606

CCA-Secure Unidirectional Proxy Re-Encryption in the Adaptive Corruption Model without Random Oracles

Jian Weng[1,2], Minrong Chen[3], Yanjiang Yang[4], Robert H. Deng[2], Kefei Chen[5], Feng Bao[4]

[1] Department of Computer Science, Jinan University, Guangzhou 510632, China;
[2] School of Information Systems, Singapore Management University, Singapore 178902;
[3] College of Information Engineering, Shenzhen University, Shenzhen 518060, China
[4] Institute for Infocomm Research (I2R), Singapore 119613;
[5] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Abstract. Proxy re-encryption (PRE), introduced by Blaze, Bleumer and Strauss in Eurocrypt'98, allows a semi-trusted proxy to convert a ciphertext originally intended for Alice into an encryption of the same message intended for Bob. PRE has recently drawn great interest, and many interesting PRE schemes have been proposed. However, up to now, it is still an important question to come up with a chosen-ciphertext secure unidirectional PRE in the adaptive corruption model. To address this problem, we propose a new unidirectional PRE scheme, and prove its chosen-ciphertext security in the adaptive corruption model without random oracles. Compared with the best known unidirectional PRE scheme proposed by Libert and Vergnaud in PKC'08, our schemes enjoys the advantages of both higher efficiency and stronger security.

Keywords. unidirectional proxy re-encryption, adaptive corruption model, chosen-ciphertext attack

1 Introduction

Proxy re-encryption (PRE), introduced by Blaze, Bleumer and Strauss[1] in Eurocrypt'98, allows a semi-trust proxy to transform a ciphertext originally intended for Alice into an encryption of the same message intended for Bob. The proxy, however, cannot learn anything about the messages encrypted under either key. PRE turns out to be a very useful tool, and has found many practical applications, such as distributed file systems[2,3], outsourced filtering of encrypted spam[2,3], and encrypted email forwarding[1], etc. According to the direction of transformation, PRE can be categorized into *bidirectional* PRE and *unidirectional* PRE. In bidirectional PREs, the proxy can transform from Alice to Bob and vice versa. In contrast, the proxy in unidirectional PREs cannot transform ciphertexts in the opposite direction. According to another criterion, PRE systems can also be classified into *multi-hop* PRE, in which the ciphertexts can be transformed from Alice to Bob and then to Charlie and so on, and *single-hop* PRE, in

Received July 16, 2009; accepted September 27, 2009
doi: 10.1007/s11432-010-0047-3
Partially supported by the Office of Research, Singapore Management University. Also partially supported by the National Natural Science Foundation of China (Grant Nos. 60903178, 60970111, 60703030, 60873229), and the National High Technology Research and Development Program of China (863 Program) under Grants Nos 2009AA01Z418, 2008AA01Z403 and 2007AA01Z456.

which the ciphertexts can only be transformed once[1].

In their seminal paper, Blaze et al.[1] proposed the first bidirectional PRE scheme. In 2005, Ateniese et al.[2,3] presented unidirectional PRE schemes from bilinear pairings. All of these schemes are only secure against chosen-plaintext attack (CPA). However, applications often require security against chosen-ciphertext attacks (CCA). To fill this gap, in ACM CCS'07, Canetti and Hohenberger[4] presented the first CCA-secure bidirectional PRE scheme without random oracles. They left six open questions in ACM CCS'07. One of these questions is to come up with a CCA-secure unidirectional PRE scheme without random oracles. In PKC'08, Libert and Vergnaud[5] partially resolved this problem by presenting a single-hop unidirectional PRE scheme without random oracles. They proved that their scheme is secure against the replayable chosen-ciphertext attack (RCCA)[6] in the non-adaptive corruption model. Here RCCA-security is a weaker variant of the CCA-security in the sense that a harmless mauling of the challenge ciphertext is tolerated. In addition, the non-adaptive corruption model is somewhat weak, since the adversary is required to commit ahead of time which public key she wants to challenge. Thus, in their full paper[2], Libert and Vergnaud left an open question of constructing a CCA-secure PRE scheme in the adaptive corruption model.

To address this problem, in this paper, we propose a new single-hop unidirectional PRE scheme, and prove its CCA-security (instead of RCCA-security) in the adaptive corruption model without random oracles. In addition to the stronger security, our scheme is more efficient than Libert-Vergnaud's PRE scheme[5].

1.2 Related Work

In ACM CCS'07, Canetti and Hohenberger[4] left an open question of constructing a CCA-secure PRE scheme without pairings. In CANS'08, Deng et al.[8] successfully presented a CCA-secure bidirectional PRE scheme without pairings in the random oracle model. In PKC'09, Shao and Cao[9] proposed a unidirectional PRE without pairings, and claimed that their scheme is CCA-secure in the random oracle model. However, Chow et al. [24] pointed out that Shao-Cao's scheme is not CCA-secure by giving a concrete attack. In the same paper, Chow et al. also presented a more efficient CCA-secure unidirectional PRE scheme without pairings in the random oracle model. Recently, Zhang et al.[3] further showed that Shao-Cao's comparisons[9] between their scheme and Libert-Vergnaud's is unfair, since Shao-Cao's scheme is even not CPA-secure in Libert-Vergnaud's security model.

To control the proxy at a fine-grained level, Tang[10] and Weng et al.[11] independently introduced a variant of PRE named conditional proxy re-encryption (C-PRE). In such systems, ciphertexts are generated with respect to a certain condition, and the proxy can translate a ciphertext only if the associated condition is satisfied. Recently, Weng et al.[12] re-formalized the definition and security notions for C-PRE systems, and presented a more efficient C-PRE scheme. Chu et al.[13] introduced a generalized concept named conditional proxy broadcast re-encryption (CPBRE), and proposed a RCCA-secure CPBRE scheme.

Traceable proxy re-encryption, introduced by Libert and Vergnaud[14], attempts to solve the problem of disclosing re-encryption keys, by tracing the proxies who have done so. Proxy re-encryption has also been studied in identity-based scenarios, such as [15,16,17].

2 Preliminaries

[1]In refs. [2,3,5], for a single-hop unidirectional PRE scheme, the original ciphertext is named *second level ciphertext*, and the transformed ciphertext is called *first level ciphertext*. Through out this paper, we also follow these terminologies.

[2]Libert, B, Vergnaud, D. Unidirectional chosen-ciphertext secure proxy re-encryption. http://hal.inria.fr/inria-00339530/en/. This is the extended version of [5].

[3]Zhang, X, Chen, M.R, Li, X. Comments on Shao-Cao's proxy re-encryption scheme from PKC 2009. Cryptology ePrint Archive, Report 2009/344 (2009), http://eprint.iacr.org.

2.1 Notations

For a finite set S, $x \xleftarrow{\$} \mathcal{S}$ means choosing an element x from S with a uniform distribution. For a string x, we use $|x|$ to denote its bit-length, $[x]_\ell$ denote its first ℓ bits, and $[x]^\ell$ denote its last ℓ bits. We use $\mathcal{A}(x, y, \cdots)$ to indicate that $\mathcal{A}$ is an algorithm with the input $(x, y, \cdots)$. By $z \leftarrow \mathcal{A}(x, y, \cdots)$, we indicate the running of $\mathcal{A}(x, y, \cdots)$ letting z be the output. We use $\mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \cdots}(x, y, \cdots)$ to denote that $\mathcal{A}$ is an algorithm with the input $(x, y, \cdots)$ and can access to oracles $\mathcal{O}_1, \mathcal{O}_2, \cdots$. By $z \leftarrow \mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \cdots}(x, y, \cdots)$, we denote the running of $\mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \cdots}(x, y, \cdots)$, and letting z be the output.

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: (i). Bilinearity: $\forall g_1, g_2 \in \mathbb{G}, \forall a, b \in \mathbb{Z}_p$, we have $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$; (ii). Non-degeneracy: There exist $g_1, g_2 \in \mathbb{G}$ such that $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$; (iii). 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 Assumptions

The security of our proposed schemes is based on a complexity assumption named 3-weak Decisional Bilinear Diffie-Hellman Inversion (3-wDBDHI) assumption[4], which has been used by Libert and Vergnaud[5] to construct unidirectional PRE schemes. The 3-wDBDHI problem in groups $(\mathbb{G}, \mathbb{G}_T)$ is, given a tuple $(g, g^a, g^{(a^2)}, g^{(a^3)}, g^b, Q) \in \mathbb{G}^5 \times \mathbb{G}_T$ with unknown $a, b \xleftarrow{\$} \mathbb{Z}_p$, to decide whether $Q = e(g, g)^{b/a}$. Libert and Vergnaud[5] has proved that, the above problem is equivalent to how to decide whether Q is equal to $e(g, g)^{b/a^2}$ or a random value given $(g, g^{1/a}, g^a, g^{(a^2)}, g^b, Q)$.

A probabilistic polynomial-time (PPT) algorithm $\mathcal{B}$ has *advantage* ϵ in solving the 3-wDBDHI problem in groups $(\mathbb{G}, \mathbb{G}_T)$, if

$$\left|\Pr\left[\mathcal{B}\left(g, g^{\frac{1}{a}}, g^a, g^{(a^2)}, g^b, Q = e(g, g)^{\frac{b}{a^2}}\right) = 1\right] - \Pr\left[\mathcal{B}\left(g, g^{\frac{1}{a}}, g^a, g^{(a^2)}, g^b, Q = e(g, g)^c\right) = 1\right]\right| \geqslant \epsilon,$$

where the probability is taken over the random choices of a, b, c in $\mathbb{Z}_p$, the random choice of g in $\mathbb{G}$, and the random bits consumed by $\mathcal{B}$.

We say that the (t, ϵ)-3-wDBDHI assumption holds in groups $(\mathbb{G}, \mathbb{G}_T)$, if there exists no t-time algorithm $\mathcal{B}$ that has advantage ϵ in solving the 3-wDBDHI problem in $(\mathbb{G}, \mathbb{G}_T)$.

2.4 TCR Hash Function

The notion of target collision resistant (TCR) hash function family of hash functions was shown by Cramer and Shoup[18]. In a TCR hash function family, given a randomly chosen hash function H and a random element x from the definition domain of H, it is infeasible for a PPT adversary $\mathcal{H}$ to find $y \neq x$ such that $H(x) = H(y)$. Informally, we define the advantage of adversary $\mathcal{H}$ in attacking the target collision resistance of H as $\mathrm{Adv}_{H,\mathcal{H}}^{\mathsf{TCR}} \triangleq \Pr[\mathcal{A} \text{ succeeds}]$. A TCR family is said to be target collision resistant if the advantage $\mathrm{Adv}_{H,\mathcal{H}}^{\mathsf{TCR}}$ is negligible for any PPT adversary $\mathcal{H}$ and any hash function H chosen from this TCR hash function family.

In practice, to build a target collision resistant hash function H, one can use a dedicated cryptographic hash function such as SHA-1. For that reason and for simplicity of our presentation, hereinafter, we will consider the hash function H to be a fixed function.

2.5 Pseudorandom Function Family

We here review the definition of pseudorandom function family[19]. Let $F : \mathcal{K} \times D \to R$ be a function family, where $\mathcal{K}$ is the set of keys of F, and D is the domain and R is the range. Let $G : D \to R$ be a true random function family. Let $\mathcal{F}$ be a PPT adversary which outputs a bit. We consider the following

[4] In ref. [5], Libert and Vergnaud also called this assumption as 3-Quotient Decision Bilinear Diffie-Hellman (3-QDBDH) assumption.

two experiments:

Experiment $\mathbf{Exp}_F^{\text{PRF-1}}(\mathcal{F})$	Experiment $\mathbf{Exp}_F^{\text{PRF-0}}(\mathcal{F})$
$K \xleftarrow{\$} \mathcal{K}$	$g \xleftarrow{\$} G$
$b \xleftarrow{\$} \mathcal{F}^{F(K,\cdot)}$	$b \xleftarrow{\$} \mathcal{F}^{g(\cdot)}$
Return b	Return b

We define $\mathcal{F}$'s advantage $\mathrm{Adv}_{F,\mathcal{F}}^{\mathsf{PRF}}$ in attacking the pseudorandomness of the function family F as

$$\left|\Pr\left[\mathbf{Exp}_F^{\text{PRF-1}}(\mathcal{F})=1\right]-\Pr\left[\mathbf{Exp}_F^{\text{PRF-0}}(\mathcal{F})=1\right]\right|.$$

If for any PPT adversary $\mathcal{F}$, his advantage in attacking the pseudorandomness of the function family F is negligible, then we say that F is a pseudorandom function family.

3 Model of Unidirectional PRE

In this section, we review the definition and security notions for unidirectional PREs. Since it is still unknown how to construct a secure multi-hop unidirectional PRE scheme under proper security model, we here concentrate on single-hop unidirectional PREs. Formally, a single-hop unidirectional PRE scheme consists of the following algorithms[5]:

$\mathsf{Setup}(1^k)$: The setup algorithm takes as input a security parameter 1^k and outputs the global parameters $param$, which includes a description of the message space $\mathcal{M}$.

$\mathsf{KeyGen}(param)$: On input of the security parameter 1^k , all parties use this randomized algorithm to generate a public/private key pair (pk, sk).

For brevity, we assume that $param$ is implicitly included in the input of the other algorithms.

$\mathsf{ReKeyGen}(sk_i, pk_j)$: The re-encryption key generation algorithm takes as input a private key sk_i and another public key pk_j. It outputs a re-encryption key $rk_{i\to j}$.

$\mathsf{Enc}_2(pk, m)$: On input of a public key pk and a message $m \in \mathcal{M}$, this encryption algorithm outputs a second level ciphertext CT that can be re-encrypted into a first level one (intended for a possibly different receiver) using the suitable re-encryption key.

$\mathsf{Enc}_1(pk, m)$: On input of a public key pk and a message $m \in \mathcal{M}$, this encryption algorithm outputs a first level ciphertext that cannot be re-encrypted for another party.

$\mathsf{ReEnc}(rk_{i\to j}, \mathrm{CT}_i)$: On input a re-encryption key $rk_{i\to j}$ and a second level ciphertext CT_i under public key pk_i, this re-encryption algorithm outputs a first level ciphertext CT_j under public key pk_j.

$\mathsf{Dec}_2(sk, \mathrm{CT})$: On input a private key sk and a second level ciphertext CT, this decryption algorithm outputs a message $m \in \mathcal{M}$ or the error symbol $\perp$ if CT is invalid.

$\mathsf{Dec}_1(sk, \mathrm{CT})$: On input a private key sk and a first level ciphertext CT, this decryption algorithm outputs a message $m \in \mathcal{M}$ or the error symbol $\perp$ if CT is invalid.

Next, we review the game-based security definitions for unidirectional PRE systems derived from refs. [4,5]. Before giving these security notions, we consider the following oracles which together model the ability of an adversary:

- Public key oracle $\mathcal{O}_{pk}(i)$: Run $(pk_i, sk_i) \leftarrow \mathsf{KeyGen}(param)$, and return pk_i to $\mathcal{A}$.
- Secret key oracle $\mathcal{O}_{sk}(pk_i)$: Return the secret key sk_i with respect to pk_i to $\mathcal{A}$.

- Re-encryption key oracle $\mathcal{O}_{rk}(pk_i, pk_j)$: Run $rk_{i\to j} \leftarrow \mathsf{ReKeyGen}(sk_i, pk_j)$ and return $rk_{i\to j}$ to $\mathcal{A}$. Here sk_i is the private key with respect to pk_i.
- Re-encryption oracle $\mathcal{O}_{re}(pk_i, pk_j, \mathrm{CT}_i)$: Return $\mathrm{CT}_j \leftarrow \mathsf{ReEnc}(\mathsf{ReKeyGen}(sk_i, pk_j), \mathrm{CT}_i)$ to $\mathcal{A}$.
- First level decryption oracle $\mathcal{O}_{1d}(pk_j, \mathrm{CT}_j)$: Return the result of $\mathsf{Dec}_1(\mathrm{CT}_j, sk_j)$ to $\mathcal{A}$. Here CT_j is a first level ciphertext.

Note that for the last four oracles, we require that pk_i and pk_j are generated by oracle $\mathcal{O}_{pk}$ beforehand.

Security of second level ciphertexts. Libert and Vergnaud[5] defined the security notion for a single-hop unidirectional PRE scheme under replayable chose-ciphertext attacks at the second level. We here modify this security notion to allow *adaptive* corruptions of users. In addition, we consider the CCA-security instead of RCCA-security. We term this notion as IND-2PRE-CCA.

Definition 1 For a single-hop unidirectional PRE scheme $\mathcal{E}$ and a PPT adversary $\mathcal{A}$ running in two stages `find` and `guess`, we define $\mathcal{A}$'s advantage against the IND-2PRE-CCA security of $\mathcal{E}$ as

$$\mathrm{Adv}_{\mathcal{E},\mathcal{A}}^{\mathsf{IND\text{-}2PRE\text{-}CCA}}(1^k) = \left|\Pr\left[\delta' = \delta \,\middle|\, \begin{array}{l} param \leftarrow \mathsf{Setup}(1^k); (pk_{i^*}, (m_0, m_1), \mathtt{st}) \leftarrow \mathcal{A}_{\mathtt{find}}^{\mathcal{O}_{pk},\mathcal{O}_{sk},\mathcal{O}_{rk},\mathcal{O}_{re},\mathcal{O}_{1d}}(param); \\ \delta \xleftarrow{\$} \{0,1\}; \mathrm{CT}^* \leftarrow \mathsf{Enc}_2(pk_{i^*}, m_\delta); \delta' \leftarrow \mathcal{A}_{\mathtt{guess}}^{\mathcal{O}_{pk},\mathcal{O}_{sk},\mathcal{O}_{rk},\mathcal{O}_{re},\mathcal{O}_{1d}}(param, \mathrm{CT}^*, \mathtt{st}). \end{array}\right] - \frac{1}{2}\right|,$$

where `st` is some internal state information of adversary $\mathcal{A}$. Here it is mandated that $|m_0| = |m_1|$, and the following requirements are simultaneously satisfied: (i). $\mathcal{A}$ cannot issue the secret key query $\mathcal{O}_{sk}(pk_{i^*})$; (ii). For any public key pk_j, $\mathcal{A}$ cannot simultaneously issue the secret key query $\mathcal{O}_{sk}(pk_j)$ and the re-encryption key query $\mathcal{O}_{rk}(pk_{i^*}, pk_j)$; (iii). For any public key pk_j, $\mathcal{A}$ cannot simultaneously issue the secret key query $\mathcal{O}_{sk}(pk_j)$ and the re-encryption query $\mathcal{O}_{re}(pk_{i^*}, pk_j, \mathrm{CT}^*)$; (iv). For a first-level ciphertext CT_j output by the re-encryption oracle $\mathcal{O}_{re}(pk_{i^*}, pk_j, \mathrm{CT}^*)$, $\mathcal{A}$ cannot issue the first level decryption query $\mathcal{O}_{1d}(pk_j, \mathrm{CT}_j)$. We refer to the above adversary $\mathcal{A}$ as an IND-2PRE-CCA adversary. We say that a single-hop unidirectional PRE scheme $\mathcal{E}$ is $(t, q_{pk}, q_{sk}, q_{rk}, q_{re}, q_d, \epsilon)$-IND-2PRE-CCA secure, if for any t-time IND-2PRE-CCA adversary $\mathcal{A}$ that makes at most $q_{pk}, q_{sk}, q_{rk}, q_{re}$ and q_d queries to oracles $\mathcal{O}_{pk}, \mathcal{O}_{sk}, \mathcal{O}_{rk}, \mathcal{O}_{re}$ and $\mathcal{O}_{1d}$, respectively, we have $\mathrm{Adv}_{\mathcal{E},\mathcal{A}}^{\mathsf{IND\text{-}2PRE\text{-}CCA}}(1^k) \leqslant \epsilon$.

Remark. As said in ref. [5], explicitly providing adversary $\mathcal{A}$ with a second level decryption oracle is useless, since (i). for the challenge ciphertext CT^*, $\mathcal{A}$ is obviously not allowed to ask the second level decryption oracle to decrypt it; (ii). while for any other second level ciphertext CT', adversary $\mathcal{A}$ can first ask the re-encryption oracle to re-encrypt it into a first level ciphertext, and then ask the first level decryption oracle to decrypt it.

Security of first level ciphertexts. The above definition provides the adversary with a second level ciphertext in the challenge phase. In ref. [5], a complementary definition of security is defined to capture the inability to distinguish first level ciphertexts. We here review this definition (referred as IND-1PRE-CCA) as defined in ref. [5], with a slight modification to allow the adaptive corruptions of users and the CCA-security. In this definition, a first level ciphertext is provided for adversary $\mathcal{A}$ in the challenge phase. Note that, in a single-hop unidirectional PRE scheme, since the first level ciphertext cannot be re-encrypted, $\mathcal{A}$ should be allowed to obtain any re-encryption keys, even including those from the target public key pk_{i^*} to other public keys which are generated by oracle $\mathcal{O}_{sk}$. Furthermore, since $\mathcal{A}$ is allowed to obtain any re-encryption keys, there is no need to provide the re-encryption oracle $\mathcal{O}_{re}$ for him. As the aforementioned remark, the second level decryption oracle is also unnecessary.

Definition 2 For a single-hop unidirectional PRE scheme $\mathcal{E}$ and a PPT adversary $\mathcal{A}$ running in two stages `find` and `guess`, we define $\mathcal{A}$'s advantage against the IND-1PRE-CCA security of $\mathcal{E}$ as

$$\mathrm{Adv}_{\mathcal{E},\mathcal{A}}^{\mathsf{IND\text{-}1PRE\text{-}CCA}}(1^k) = \left|\Pr\left[\delta' = \delta \,\middle|\, \begin{array}{l} param \leftarrow \mathsf{Setup}(1^k); (pk_{i^*}, (m_0, m_1), \mathtt{st}) \leftarrow \mathcal{A}_{\mathtt{find}}^{\mathcal{O}_{pk},\mathcal{O}_{sk},\mathcal{O}_{rk},\mathcal{O}_{1d}}(param); \\ \delta \xleftarrow{\$} \{0,1\}; \mathrm{CT}^* \leftarrow \mathsf{Enc}_1(pk_{i^*}, m_\delta); \delta' \leftarrow \mathcal{A}_{\mathtt{guess}}^{\mathcal{O}_{pk},\mathcal{O}_{sk},\mathcal{O}_{rk},\mathcal{O}_{1d}}(param, \mathrm{CT}^*, \mathtt{st}). \end{array}\right] - \frac{1}{2}\right|,$$

where $\mathtt{st}$ is some internal state information of adversary $\mathcal{A}$. Here it is mandated that $|m_0| = |m_1|$, and $\mathcal{A}$ can issue neither $\mathcal{O}_{sk}(pk_{i^*})$ nor $\mathcal{O}_{1d}(pk_{i^*}, \mathrm{CT}^*)$. We refer to the above adversary $\mathcal{A}$ as an IND-1PRE-CCA adversary. We say that a single-hop unidirectional PRE scheme $\mathcal{E}$ is $(t, q_{pk}, q_{sk}, q_{rk}, q_d, \epsilon)$-IND-1PRE-CCA secure, if for any t-time IND-1PRE-CCA adversary $\mathcal{A}$ that makes at most q_{pk}, q_{pk}, q_{rk} and q_d queries to oracles $\mathcal{O}_{pk}, \mathcal{O}_{sk}, \mathcal{O}_{rk}$ and $\mathcal{O}_{1d}$, respectively, we have $\mathrm{Adv}_{\mathcal{E},\mathcal{A}}^{\textsf{IND-1PRE-CCA}}(1^k) \leqslant \epsilon$.

Remark. Note that there is no need to explicitly provide the re-encryption oracle for $\mathcal{A}$, since he can obtain *any* re-encryption key by resorting to oracle $\mathcal{O}_{rk}$, and thus he can re-encrypt *any* second level ciphertext himself.

Master Secret Security. In ref. [2], Ateniese et al. defined another important security notion, named delegator secret security, for unidirectional PRE. This security notion captures the intuition that, even if the dishonest proxy colludes with the delegatee, it is still impossible for them to derive the delegator's private key in full. We define this security notion (referred as MSS-PRE) in the following definition:

Definition 3 For a unidirectional PRE scheme $\mathcal{E}$ and a PPT adversary $\mathcal{A}$, we define $\mathcal{A}$'s advantage against the MSS-PRE security of $\mathcal{E}$ as

$$\mathrm{Adv}_{\mathcal{E},\mathcal{A}}^{\textsf{MSS-PRE}}(1^k) = \Pr\left[sk_{i^*} \text{is a valid secret key} \,\middle|\, param \leftarrow \mathsf{Setup}(1^k); sk_{i^*} \leftarrow \mathcal{A}^{\mathcal{O}_{pk},\mathcal{O}_{sk},\mathcal{O}_{rk}}(param)\right],$$

where it is mandated that the public key pk_{i^*} with respect to sk_{i^*} is generated by $\mathcal{O}_{pk}$, and $\mathcal{A}$ cannot issue the secret key query $\mathcal{O}_{sk}(pk_{i^*})$. We refer to the above adversary as a MSS-PRE adversary. We say that a unidirectional PRE scheme $\mathcal{E}$ is $(t, q_{pk}, q_{sk}, q_{rk}, \epsilon)$-MSS-PRE secure, if for any t-time MSS-PRE adversary $\mathcal{A}$ that makes at most q_{pk}, q_{sk} and q_{rk} queries to oracles $\mathcal{O}_{pk}, \mathcal{O}_{sk}$ and $\mathcal{O}_{rk}$, respectively, we have $\mathrm{Adv}_{\mathcal{E},\mathcal{A}}^{\textsf{MSS-PRE}}(1^k) \leqslant \epsilon$.

As indicated by the following Lemma, for single-hop unidirectional PREs, the master secret security is implied by the first level ciphertext security.

Lemma 1 For a single-hop unidirectional PRE, the master secret security is implied by the first level ciphertext security. That is, if there exists an adversary $\mathcal{A}$ who can break the MSS-PRE security of a single-hop unidirectional PRE scheme $\mathcal{E}$, then there also exists an adversary $\mathcal{B}$ who can also break $\mathcal{E}$'s IND-1PRE-CCA security.

The proof for Lemma 1 is straightforward, and hence is omitted here.

4 Our Unidirectional PRE Scheme

In this section, we first present our single-hop unidirectional PRE scheme, and then give the security proofs for our scheme. A comparison between our scheme and Libert-Vergnaud's scheme is also given.

4.1 Construction

Before presenting our scheme, some important and *necessary* principles for designing CCA-secure single-hop unidirectional PRE schemes should be mentioned: (i) the validity of the second level ciphertexts should be *publicly verifiable*; otherwise, it will suffer from a similar attack as illustrated in [8][5]; (ii) it should also be impossible for the adversary to maliciously manipulate the first level ciphertext. In addition, to ensure the first level ciphertext security, another principle should be kept in mind: (iii) for a first level ciphertext CT_j re-encrypted from a second level ciphertext CT_i, CT_j should not contains *all* the components of CT_i; otherwise, it will inevitably suffer from an attack as applied to Shao's scheme[7]. We will explain how our scheme follows these principles in the following description of our scheme. Our proposed scheme consists of the following algorithms:

[5]Weng, J, Deng, R.H, Liu, S, et al. Chosen-ciphertext secure proxy re-encryption without pairings. Cryptology ePrint Archive, Report 2008/509 (2008), http://eprint.iacr.org. This is the full paper of [8].

$\mathsf{Setup}(1^k)$: Given a security parameter 1^k, this setup algorithm works as follows:

1. Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^k$.
2. Pick generators $g, g_1, u, v, w \xleftarrow{\$} \mathbb{G}$, and set $Z = e(g, g)$.
3. Choose a collision-resistant hash function $H : \mathbb{G} \times \{0,1\}^\ell \to \mathbb{Z}_p^*$. Choose also a pseudo-random function (PRF) family $F : \mathbb{G}_T \times \mathbb{G} \to \{0,1\}^{\ell-\ell_1} \| \{0,1\}^{\ell_1}$ such that, given a seed in $\mathbb{G}_T$ and an input in $\mathbb{G}$, it outputs an ℓ-bit pseudorandom string. Here ℓ and ℓ_1 are security parameters.
4. Output the public parameters $param = (p, \mathbb{G}, \mathbb{G}_T, g, g_1, u, v, w, Z, H, F, \ell_1, \ell)$.

$\mathsf{KeyGen}(param)$: User i picks $x_i \xleftarrow{\$} \mathbb{Z}_p^*$, and sets his public key as $pk_i = g^{x_i}$ and private key as $sk_i = x_i$.

$\mathsf{ReKeyGen}(sk_i, pk_j)$: On input user i's private key $sk_i = x_i$ and user j's public key $pk_j = g^{x_j}$, this algorithm generates the re-encryption key $rk_{i\to j} = pk_j^{1/sk_i} = g^{x_j/x_i}$.

$\mathsf{Enc}_2(pk_i, m)$: To encrypt a message $m \in \{0,1\}^{\ell_1}$ under the public key pk_i at the second level, the sender proceeds as follows:

1. Pick $r \xleftarrow{\$} \mathbb{Z}_p^*$, and set $C_1 = g_1^r$ and $C_2 = pk_i^r$.
2. Compute $K = Z^r$ and set $C_3 = [F(K, C_1)]_{\ell-\ell_1} \| ([F(K, C_1)]^{\ell_1} \oplus m)$.
3. Pick $t \xleftarrow{\$} \mathbb{Z}_p^*$, and compute $h = H(C_1, C_3)$ and $C_4 = \left(u^h v^t w\right)^r$.
4. Output the second level ciphertext $\mathrm{CT}_i = (t, C_1, C_2, C_3, C_4)$.

Note that the validity of the second level ciphertext can be publicly verified as in eqs. (1) and (2) to be given. Hence it is impossible for the adversary to manipulate the second level ciphertext.

$\mathsf{Enc}_1(pk_j, m)$: To encrypt a message $m \in \{0,1\}^{\ell_1}$ under the public key pk_j at the first level, the sender proceeds as follows:

1. Pick $r \xleftarrow{\$} \mathbb{Z}_p^*$, and set $C_1 = g_1^r, C_2' = e(pk_j, g)^r, K = Z^r$, $C_3 = [F(K, C_1)]_{\ell-\ell_1} \| ([F(K, C_1)]^{\ell_1} \oplus m)$.
2. Pick $t \xleftarrow{\$} \mathbb{Z}_p^*$, and compute $h = H(C_1, C_3)$ and $C_4 = \left(u^h v^t w\right)^r$.
3. Output the first level ciphertext $\mathrm{CT}_i = (t, C_1, C_2', C_3, C_4)$.

Note that the validity of components t, C_1, C_3 and C_4 can be checked as in eq. (1), and the validity of C_2' can be verified as in eq. (4). So, it is also impossible for the adversary to maliciously manipulate the first level ciphertext.

$\mathsf{ReEnc}(rk_{i\to j}, \mathrm{CT}_i)$: On input a re-encryption key $rk_{i\to j}$, an second level ciphertext $\mathrm{CT}_i = (t, C_1, C_2, C_3, C_4)$ under public key pk_i, compute $h = H(C_1, C_3)$, and then check the validity of CT_i by testing whether the following equalities hold:

$$e(C_1, u^h v^t w) = e(C_4, g_1), \tag{1}$$

$$e(C_1, pk_i) = e(C_2, g_1). \tag{2}$$

If not, output $\perp$. Otherwise, compute $C_2' = e(C_2, rk_{i\to j})$, and output the first level ciphertext under public key pk_j as $\mathrm{CT}_j = (t, C_1, C_2', C_3, C_4)$.

Note that, the verification of eqs. (1) and (2) can be alternately done by picking $r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p^*$ and testing if

$$e\left(C_1, pk_i^{r_1}(u^h v^t w)^{r_2}\right) = e\left(C_2^{r_1} c_4^{r_2}, g_1\right). \tag{3}$$

In this way, the verification of the well-formedness of ciphertext CT_i reduces two pairings at the cost of only one more multi-exponentiation. Note that the multi-exponentiation can be computed more efficiently than the pairing.

$\mathsf{Dec}_2(sk_i, \mathrm{CT}_i)$: To decrypt a second level ciphertext $\mathrm{CT}_i = (t, C_1, C_2, C_3, C_4)$, user i with private key sk_i proceeds as follows:

1. First check the validity of the ciphertext as in eq. (3). If the verification fails, output "$\perp$".
2. Compute $K = e(C_2, g)^{1/sk_i}$. If $[F(K, C_1)]_{\ell-\ell_1} = [C_3]_{\ell-\ell_1}$ holds, output $m = [F(K, C_1)]^{\ell_1} \oplus [C_3]^{\ell_1}$; else output "$\perp$".

$\mathsf{Dec}_1(sk_j, \mathrm{CT}_j)$: On input a private key sk_j and a first level ciphertext $\mathrm{CT}_j = (t, C_1, C_2', C_3, C_4)$, user j with private key sk_j proceeds as follows:

1. First check the validity of the ciphertext as in eq. (1). If the verification fails, output "$\perp$".
2. Compute $K = C_2'^{1/sk_j}$. Output $m = [F(K, C_1)]^{\ell_1} \oplus [C_3]^{\ell_1}$ if the following equality holds:
$$[F(K, C_1)]_{\ell-\ell_1} = [C_3]_{\ell-\ell_1}. \tag{4}$$
Otherwise, output "$\perp$".

Remark: Libert-Vergnaud's scheme applies a technique introduced in Boyen-Boneh's selective-ID secure identity-based encryption scheme[20] (refer to the ciphertext component $C_4 = (u^{svk} \cdot v)^r$ in Libert-Vergnaud's scheme). Thus in the security proofs of Libert-Vergnaud's scheme, the adversary must commit ahead of time to the target user that it wants to attack; otherwise, the challenger will be unable to generate the challenge ciphertext for the adversary. In contrast, we use a technique inspired by Hohenberger-Waters' recent signatures scheme[21] (refer to the ciphertext component $C_4 = (u^h v^t w)^r$ in our scheme)[6]. As will be seen in our security proofs, this technique enables the challenger to successfully generate the challenge ciphertext for the adversary, even if the adversary is allowed to adaptively corrupt users.

4.2 Security Analysis

The chosen-ciphertext security at the second level for our scheme is asserted by the following theorem.

Theorem 1 Our scheme is IND-2PRE-CCA secure, assuming the hash function H is target collision resistant, F is a pseudorandom function family and the 3-wDBDHI assumption holds in groups $(\mathbb{G}, \mathbb{G}_T)$.

Proof Without loss of generality, assume that H is target collision resistant and F is a pseudorandom function family. Then suppose that there is an adversary $\mathcal{A}$ who can break the $(t, q_{pk}, q_{sk}, q_{rk}, q_{re}, q_d, \epsilon)$-IND-2PRE-CCA security of our scheme. We can construct an algorithm $\mathcal{B}$ which can break the (t', ϵ')-3-wDBDHI assumption in $(\mathbb{G}, \mathbb{G}_T)$ with

$$\epsilon' \geqslant \frac{\epsilon}{2\dot{e}(1 + q_{sk} + q_{rk})} - \frac{q_d + q_{re}}{p} - \mathrm{Adv}_{H,\mathcal{H}}^{\mathsf{TCR}} - \mathrm{Adv}_{F,\mathcal{F}}^{\mathsf{PRF}}, \quad t' \leqslant t + \mathcal{O}(\tau(q_{pk} + q_{rk} + 8q_{re} + 8q_d)),$$

where τ is the maximum over the time to compute an exponentiation, a multi-exponentiation and a pairing; $\dot{e}$ denotes the base of the natural logarithm.

Suppose algorithm $\mathcal{B}$ is given a 3-wDBDHI instance $(g, A_{-1} = g^{1/a}, A_1 = g^a, A_2 = g^{(a^2)}, B = g^b, Q) \in \mathbb{G}^5 \times \mathbb{G}_T$ with unknown $a, b \xleftarrow{\$} \mathbb{Z}_p^*$. $\mathcal{B}$'s goal is to decide whether $Q = e(g, g)^{b/a^2}$. $\mathcal{B}$ works by interacting with adversary $\mathcal{A}$ in the IND-2PRE-CCA game as follows:

`Initialize.` $\mathcal{B}$ provides $\mathcal{A}$ with public parameters including $g_1 = A_2^{\alpha_0}, u = A_1^{\alpha_1} A_2^{\beta_1}, v = A_1^{\alpha_2} A_2^{\beta_2}$ and $w = A_1^{\alpha_3} A_2^{\beta_3}$ for random $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3 \xleftarrow{\$} \mathbb{Z}_p^*$.

[6] Lai et al. has used this technique to construct an efficient CCA-secure public key encryption scheme [22].

`Find Stage.` $\mathcal{A}$ issues a series of queries as in the IND-2PRE-CCA game. $\mathcal{B}$ maintains a list L^{list}, and answers these queries for $\mathcal{A}$ as follows:

- *Public key oracle* $\mathcal{O}_{pk}(i)$: $\mathcal{B}$ picks $x_i \xleftarrow{\$} \mathbb{Z}_p^*$. Next, using the Coron's technique[23], it chooses a number $c_i \in \{0, 1, `-'\}$ such that $\Pr[c_i = 0] = Pr[c_i = 1] = \theta$ and $\Pr[c_i = `-'] = 1 - 2\theta$, where θ will be determined later. If $c_i = `-'$, it sets $pk_i = g^{x_i}$; if $c_i = 0$, it sets $pk_i = A_2^{x_i}$; if $c_i = 1$, it sets $pk_i = A_1^{x_i}$. Next, it adds the tuple (pk_i, x_i, c_i) to L^{list} and returns pk_i to $\mathcal{A}$.
- *Secret key oracle* $\mathcal{O}_{sk}(pk_i)$: $\mathcal{B}$ first recovers the tuple (pk_i, x_i, c_i) from L^{list}. If $c_i = `-'$, it returns $sk_i = x_i$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ outputs a random bit in $\{0, 1\}$ and **aborts**.
- *Re-encryption key oracle* $\mathcal{O}_{rk}(pk_i, pk_j)$: $\mathcal{B}$ first recovers tuples (pk_i, x_i, c_i) and (pk_j, x_j, c_j) from L^{list}, and then generate the re-encryption key $rk_{i\to j}$ for $\mathcal{A}$ according to the following cases:
  - $c_i = `-'$: It means that $sk_i = x_i$. $\mathcal{B}$ outputs $rk_{i\to j} = pk_j^{1/x_i}$.
  - $c_i = c_j$: $\mathcal{B}$ returns $rk_{i\to j} = g^{x_j/x_i}$, which is indeed a valid re-encryption key.
  - $c_i = 1 \wedge c_j = 0$: It means that $sk_i = ax_i$ and $sk_j = a^2x_j$. $\mathcal{B}$ returns $rk_{i\to j} = A_1^{\frac{x_j}{x_i}} = g^{\frac{a^2x_j}{ax_i}}$, which is indeed a valid re-encryption key.
  - $c_i = 1 \wedge c_j =`-'$: It means that $sk_i = ax_i$ and $sk_j = x_j$. $\mathcal{B}$ returns $rk_{i\to j} = A_{-1}^{x_j/x_i} = g^{\frac{x_j}{ax_i}}$, which is indeed a valid re-encryption key.
  - $c_i = 0 \wedge c_j = 1$: It means that $sk_i = a^2x_i$ and $sk_j = ax_j$. $\mathcal{B}$ returns $rk_{i\to j} = A_{-1}^{\frac{x_j}{x_i}} = g^{\frac{ax_j}{a^2x_i}}$, which is indeed a valid re-encryption key.
  - $c_i = 0 \wedge c_j =`-'$: $\mathcal{B}$ outputs a random bit in $\{0, 1\}$ and **aborts**.
- *Re-encryption oracle* $\mathcal{O}_{re}(pk_i, pk_j, \mathrm{CT}_i)$: $\mathcal{B}$ first parses CT_i as (t, C_1, C_2, C_3, C_4), and then checks the validity of the ciphertext as in eq. (3). If the verification fails, it returns "$\perp$" to $\mathcal{A}$. Otherwise, $\mathcal{B}$ proceeds to execute the following steps:
  1. Recover tuples (pk_i, x_i, c_i) and (pk_j, x_j, c_j) from L^{list}.
  2. If $(c_i = 0 \;\wedge\; c_j = `-')$ does not hold, first generate the re-encryption key $rk_{i\to j}$ as in the re-encryption key oracle $\mathcal{O}_{rk}$, and then return $\mathsf{ReEnc}(rk_{i\to j}, \mathrm{CT}_i)$ to $\mathcal{A}$.
  3. Else (it means that $sk_i = a^2x_i$ and $sk_j = x_j$), from $C_1 = g_1^r = A_2^{r\cdot\alpha_0}$ and $C_4 = (u^hv^tw)^r = \left(A_1^{\alpha_1h+\alpha_2t+\alpha_3}A_2^{\beta_1h+\beta_2t+\beta_3}\right)^r$ where $h = H(C_1, C_3)$, $\mathcal{B}$ can compute
$$A_1^r = \left(\frac{C_4}{C_1^{\frac{\beta_1h+\beta_2t+\beta_3}{\alpha_0}}}\right)^{\frac{1}{\alpha_1h+\alpha_2t+\alpha_3}}. \tag{5}$$
     Then compute $K = e(A_{-1}, A_1^r) = e(g, g)^r$, set $C_2' = K^{x_j} = e(pk_j, g)^r$, and return $\mathrm{CT}_j = (t, C_1, C_2', C_3, C_4)$ to $\mathcal{A}$.

  Recall that, in the public parameters $u = A_1^{\alpha_1}A_2^{\beta_1}$, $v = A_1^{\alpha_2}A_2^{\beta_2}$ and $w = A_1^{\alpha_3}A_2^{\beta_3}$, α_1 (α_2, α_3, resp.) is blinded by β_1 (β_2, β_3, resp.), and hence no information about α_1, α_2 and α_3 is leaked to the adversary. So, in eq. (5), the equality $\alpha_1h+\alpha_2t+\alpha_3 = 0 \mod p$ information-theoretically holds with probability at most $\frac{1}{p}$.
- *First level decryption oracle* $\mathcal{O}_{1d}(pk_j, \mathrm{CT}_j)$: Algorithm $\mathcal{B}$ first parses the first level ciphertext CT_j as (t, C_1, C_2', C_3, C_4). Next, it recovers tuple (pk_j, x_j, c_j) from L^{list}. If $c_j = `-'$, it means that $sk_j = x_j$, and $\mathcal{B}$ returns $\mathsf{Dec}_1(sk_j, \mathrm{CT}_j)$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ proceeds as follows:

1. Compute $h = H(C_1, C_3)$ and check the validity of the ciphertext as in eq. (1). If the verification fails, output "$\perp$" indicating an invalid ciphertext; else continue to execute the rest steps.
2. Compute $A_1^r = \left(\frac{C_4}{C_1^{\frac{\beta_1 h+\beta_2 t+\beta_3}{\alpha_0}}}\right)^{\frac{1}{\alpha_1 h+\alpha_2 t+\alpha_3}}$ and then $K = e(A_{-1}, A_1^r) = e(g,g)^r$. Note that, similarly to the analysis in the re-encryption oracle $\mathcal{O}_{re}$, the chance of $\alpha_1 h + \alpha_2 t + \alpha_3 = 0 \mod p$ holds with probability at most $\frac{1}{p}$.
3. If $[F(K, C_1)]_{\ell-\ell_1} \neq [C_3]_{\ell-\ell_1}$, output "$\perp$" indicating an invalid ciphertext. Otherwise, output $m = [F(K, C_1)]^{\ell_1} \oplus [C_3]^{\ell_1}$.

Note that even for the case of $\alpha_1 h + \alpha_2 t + \alpha_3 = 0 \mod p$, $\mathcal{B}$ is still able to reject invalid first level ciphertexts. For a first level ciphertext $\mathrm{CT}_j = (t, C_1, C_2', C_3, C_4)$ under public key pk_j, the validity of t, C_1, C_3 and C_4 can be ensured as in eq. (1). So, $\mathcal{B}$ needs only to check the validity of C_2'. Suppose $C_1 = g_1^r, C_3 = [F(K, C_1)]_{\ell-\ell_1} \| ([F(K, C_1)]^{\ell_1} \oplus m)$ and $C_4 = (u^{H(C_1,C_3)} v^t w)^r$, where $K = e(g,g)^r$. To check the validity of C_2', $\mathcal{B}$ needs to check whether $C_2' = e(pk_j, g)^r$ holds. Fortunately, $\mathcal{B}$ can do this, since she can compute $e(pk_j, g)^r$ according to the following cases:

– $c_j = 1$ (it meas that $pk_j = A_1^{x_j}$): $\mathcal{B}$ can obtain $e(pk_j, g)^r$ by computing

$$e(C_1, A_{-1})^{\frac{x_j}{\alpha_0}} = (g_1^r, A_{-1})^{\frac{x_j}{\alpha_0}} = (A_2^{\alpha_0 r}, A_{-1})^{\frac{x_j}{\alpha_0}} = e(A_1^{x_j}, g)^r = e(pk_j, g)^r.$$

– $c_j = 0$ (it meas that $pk_j = A_2^{x_j}$): $\mathcal{B}$ can obtain $e(pk_j, g)^r$ by computing

$$e(C_1, g)^{\frac{x_j}{\alpha_0}} = (g_1^r, g)^{\frac{x_j}{\alpha_0}} = (A_2^{\alpha_0 r}, g)^{\frac{x_j}{\alpha_0}} = e(A_2^{x_j}, g)^r = e(pk_j, g)^r.$$

`Challenge.` When $\mathcal{A}$ judges that `find` stage is over, it outputs a public key pk_{i^*} and messages $m_0, m_1 \in \{0,1\}^{\ell_1}$ with the restrictions specified in the IND-2PRE-CCA game. $\mathcal{B}$ responds as follows:

1. Recover tuple $(pk_{i^*}, x_{i^*}, c_{i^*})$ from L^{list}. If $c_{i^*} \neq 0$, $\mathcal{B}$ outputs a random bit in $\{0,1\}$ and **aborts**. Otherwise, it means that $pk_{i^*} = A_2^{x_{i^*}}$, and $\mathcal{B}$ proceeds to execute the rest steps.
2. Pick $\delta \xleftarrow{\$} \{0,1\}$. Define $C_1^* = B^{\alpha_0}, C_2^* = B^{x_{i^*}}, C_3^* = [F(Q, C_1^*)]_{\ell-\ell_1} \| ([F(Q, C_1^*)]^{\ell_1} \oplus m_\delta)$, $t^* = -\frac{\alpha_1 h^* + \alpha_3}{\alpha_2}$ and $C_4^* = B^{(\beta_1 h^* + \beta_2 t^* + \beta_3)}$ where $h^* = H(C_1^*, C_3^*)$. Return $\mathrm{CT}^* = (t^*, C_1^*, C_2^*, C_3^*, C_4^*)$ as the challenge ciphertext to $\mathcal{A}$.

Observe that, if $Q = e(g,g)^{\frac{b}{a^2}}$, CT^* is indeed a valid challenge ciphertext under public key pk_{i^*}. To see this, letting $r^* = \frac{b}{a^2}$, we have

$$\begin{aligned}
C_1^* &= B^{\alpha_0} = \left(g^{a^2}\right)^{\alpha_0 \cdot \frac{b}{a^2}} = (A_2^{\alpha_0})^{r^*} = g_1^{r^*}, \\
C_2^* &= B^{x_{i^*}} = \left(g^{a^2}\right)^{x_{i^*} \cdot \frac{b}{a^2}} = (A_2^{x_{i^*}})^{r^*} = pk_{i^*}^{r^*}, \\
C_3^* &= [F(Q, C_1^*)]_{\ell-\ell_1} \| ([F(Q, C_1^*)]^{\ell_1} \oplus m_\delta) = [F(Z^{r^*}, C_1^*)]_{\ell-\ell_1} \| ([F(Z^{r^*}, C_1^*)]^{\ell_1} \oplus m_\delta), \\
C_4^* &= B^{(\beta_1 h^* + \beta_2 t^* + \beta_3)} = \left(A_2^{\beta_1 h^* + \beta_2 t^* + \beta_3}\right)^{r^*} = \left(A_1^{\alpha_1 h^*} A_1^{-\frac{\alpha_1 h^* + \alpha_3}{\alpha_2} \cdot \alpha_2} A_1^{\alpha_3} \cdot A_2^{\beta_1 h^* + \beta_2 t^* + \beta_3}\right)^{r^*} \\
&= \left(A_1^{\alpha_1 h^*} A_1^{t^* \cdot \alpha_2} A_1^{\alpha_3} A_2^{\beta_1 h^* + \beta_2 t^* + \beta_3}\right)^{r^*} = \left(\left(A_1^{\alpha_1} A_2^{\beta_1}\right)^{h^*} \cdot \left(A_1^{\alpha_2} A_2^{\beta_2}\right)^{t^*} \cdot \left(A_1^{\alpha_3} A_2^{\beta_3}\right)\right)^{r^*} = \left(u^{h^*} v^{t^*} w\right)^{r^*}.
\end{aligned}$$

On the other hand, when Q is uniform and independent in $\mathbb{G}_T$, the challenge ciphertext CT^* is independent of δ in the adversary's view.

`Guess Stage.` Adversary $\mathcal{A}$ continues to issue the rest queries. $\mathcal{B}$ can respond these queries for $\mathcal{A}$ as in the `find stage`, since $\mathcal{A}$ has to follow the restrictions described in the IND-2PRE-CCA game and the hash function H is target collision resistant. Observe that, although the challenge ciphertext CT^* leaks the information $\alpha_1 h^* + \alpha_2 t^* + \alpha_3 = 0$ to $\mathcal{A}$, it can be seen that the probability of $\mathcal{A}$'s querying a ciphertext which can cause $\mathcal{B}$ to abort is still at most $\frac{1}{p}$.

`Output.` Eventually, $\mathcal{A}$ returns a guess $\delta' \in \{0,1\}$. If $\delta' = \delta$, $\mathcal{B}$ outputs $\beta' = 1$; else, outputs $\beta' = 0$.

This completes the description of the simulation. We next begin to analyze the simulation. It is clear that the simulations of oracle $\mathcal{O}_{pk}$ is perfect. Let Abort denote the event of $\mathcal{B}$'s aborting during the simulation of oracles $\mathcal{O}_{sk}, \mathcal{O}_{rk}$ or in `Challenge` phase. We have $\Pr[\neg\mathsf{Abort}] = (1-2\theta)^{q_{sk}}(1-\theta(1-2\theta))^{q_{rk}}\theta \geqslant (1-2\theta)^{q_{sk}+q_{rk}}\theta$, which is maximized at $\theta_{\mathrm{opt}} = \frac{q_{sk}+q_{rk}}{2(1+q_{sk}+q_{rk})}$. Using θ_{opt}, the probability $\Pr[\neg\mathsf{Abort}]$ is at least $\frac{1}{2\dot{e}(1+q_{sk}+q_{rk})}$. Note that, if even Abort does not happen during the simulation of oracles $\mathcal{O}_{sk}, \mathcal{O}_{rk}$ and the `Challenge` phase, the simulation for oracles $\mathcal{O}_{sk}, \mathcal{O}_{rk}$ and the challenge ciphertext are perfect.

We proceeds to analyze the simulation of the re-encryption oracle $\mathcal{O}_{re}$. The simulation of $\mathcal{O}_{re}$ is perfect, unless $\alpha_1 h + \alpha_2 t + \alpha_3 = 0 \mod p$ happens during the whole simulation (denote this event by ReEErr). However, as argued before, the equality $\alpha_1 h + \alpha_2 t + \alpha_3 = 0 \mod p$ holds in each query with probability at most $\frac{1}{p}$. So we have $\Pr[\mathsf{ReEErr}] \leqslant \frac{q_{re}}{p}$.

The simulation of decryption $\mathcal{O}_{1d}$ is also perfect, unless $\alpha_1 h + \alpha_2 t + \alpha_3 = 0 \mod p$ happens during the whole simulation (denote this event by DecErr). Similarly, we can have $\Pr[\mathsf{DecErr}] \leqslant \frac{q_d}{p}$.

Combining the above results and counting for the target collision resistant of the hash function H and the pseudorandomness of F, we can see that $\mathcal{B}$'s advantage against the 3-wDBDHI assumption is at least $\epsilon' \geqslant \frac{\epsilon}{2\dot{e}(1+q_{sk}+q_{rk})} - \frac{q_d+q_{re}}{p} - \mathrm{Adv}_{H,\mathcal{A}}^{\mathsf{TCR}} - \mathrm{Adv}_{F,\mathcal{F}}^{\mathsf{PRF}}$, and $\mathcal{B}$'s running time can be bounded by $t' \leqslant t + \mathcal{O}(\tau(q_{pk} + q_{rk} + 8q_{re} + 8q_d))$. This completes the proof of Theorem 1.

The CCA-security at the first level for our scheme is ensured by the following theorem:

Theorem 2 Our scheme is chosen-ciphertext secure at the first level, assuming the hash function H is target collision resistant, F is a pseudorandom function family and 3-wDBDHI assumption holds in groups $(\mathbb{G}, \mathbb{G}_T)$.

Proof Without loss of generality, assume that H is target collision resistant and F is a pseudorandom function family. Then, suppose there is an adversary $\mathcal{A}$ who can break the $(t, q_{pk}, q_{sk}, q_{rk}, q_d, \epsilon)$-IND-1PRE-CCA security of our scheme, we can construct an algorithm $\mathcal{B}$ which can break the (t', ϵ')-3-wDBDHI assumption in $(\mathbb{G}, \mathbb{G}_T)$ with

$$\epsilon' \geqslant \frac{\epsilon}{\dot{e}(1+q_{sk})} - \frac{q_d}{p} - \mathrm{Adv}_{H,\mathcal{A}}^{\mathsf{TCR}} - \mathrm{Adv}_{F,\mathcal{F}}^{\mathsf{PRF}}, \quad t' \leqslant t + \mathcal{O}(\tau(q_{pk} + q_{rk} + 8q_d)),$$

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

Suppose algorithm $\mathcal{B}$ is given a 3-wDBDHI instance $(g, A_{-1} = g^{1/a}, A_1 = g^a, A_2 = g^{(a^2)}, B = g^b, Q) \in \mathbb{G}^5 \times \mathbb{G}_T$ with unknown $a, b \xleftarrow{\$} \mathbb{Z}_p^*$. $\mathcal{B}$'s goal is to decide whether $Q = e(g,g)^{b/a^2}$. $\mathcal{B}$ works by interacting with adversary $\mathcal{A}$ in the IND-1PRE-CCA game as follows:

`Initialize.` $\mathcal{B}$ provides $\mathcal{A}$ with public parameters including $g_1 = A_2^{\alpha_0}, u = A_1^{\alpha_1} A_2^{\beta_1}, v = A_1^{\alpha_2} A_2^{\beta_2}$ and $w = A_1^{\alpha_3} A_2^{\beta_3}$ for random $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3 \xleftarrow{\$} \mathbb{Z}_p^*$.

`Find Stage.` $\mathcal{A}$ issues a series of queries as in the IND-1PRE-CCA game. $\mathcal{B}$ maintains a list L^{list}, and answers these queries for $\mathcal{A}$ as follows:

- *Public key oracle* $\mathcal{O}_{pk}(i)$: $\mathcal{B}$ picks $x_i \xleftarrow{\$} \mathbb{Z}_p^*$, and flips a random coin $c_i \in \{0,1\}$. If $c_i = 0$, it sets $pk_i = A_1^{x_i}$; else it sets $pk_i = g^{x_i}$. Next, it adds the tuple (pk_i, x_i, c_i) to L^{list} and returns pk_i to $\mathcal{A}$.

- *Secret key oracle* $\mathcal{O}_{sk}(pk_i)$: $\mathcal{B}$ first recovers (pk_i, x_i, c_i) from L^{list}. If $c_i = 1$, it returns $sk_i = x_i$ to $\mathcal{A}$; else it outputs a random bit in $\{0,1\}$ and **aborts**.

- *Re-encryption key oracle* $\mathcal{O}_{rk}(pk_i, pk_j)$: $\mathcal{B}$ first recovers tuples (pk_i, x_i, c_i) and (pk_j, x_j, c_j) from L^{list}, and then generates the re-encryption key $rk_{i\to j}$ for $\mathcal{A}$ according to the following cases:

- $c_i = 1$: It means that $sk_i = x_i$. $\mathcal{B}$ outputs $rk_{i\to j} = pk_j^{1/x_i}$.
- $c_i = c_j = 0$: $\mathcal{B}$ returns $rk_{i\to j} = g^{x_j/x_i}$, which is indeed a valid re-encryption key.
- $c_i = 0 \wedge c_j = 1$: $\mathcal{B}$ returns $rk_{i\to j} = A_{-1}^{x_j/x_i}$.

• *First level decryption oracle* $\mathcal{O}_{1d}(pk_j, \mathrm{CT}_j)$: Algorithm $\mathcal{B}$ first parses the first level ciphertext CT_j as (t, C_1, C_2', C_3, C_4). Next, it recovers tuple (pk_j, x_j, c_j) from L^{list}. If $c_j = 1$, it means that $sk_j = x_j$, and algorithm $\mathcal{B}$ returns $\mathsf{Dec}_1(sk_j, \mathrm{CT}_j)$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ proceeds as follows:

1. Compute $h = H(C_1, C_3)$ and then check the validity of the ciphertext as in eq. (1). If the verification fails, output "$\perp$"; else continue to execute the rest steps.
2. Compute $A_1^r = \left(\frac{C_4}{C_1^{\frac{\beta_1 h+\beta_2 t+\beta_3}{\alpha_0}}}\right)^{\frac{1}{\alpha_1 h+\alpha_2 t+\alpha_3}}$ and then $K = e(A_{-1}, A_1^r) = e(g, g)^r$. Note that, similarly to the analysis in the proof of Theorem 1, the chance of $\alpha_1 h + \alpha_2 t + \alpha_3 = 0 \mod p$ holds with probability at most $\frac{1}{p}$.
3. If $[F(K, C_1)]_{\ell-\ell_1} \neq [C_3]_{\ell-\ell_1}$, output "$\perp$" indicating an invalid ciphertext. Otherwise, output $m = [F(K, C_1)]^{\ell_1} \oplus [C_3]^{\ell_1}$.

Note that even for the case of $\alpha_1 h+\alpha_2 t+\alpha_3 = 0 \mod p$, challenger $\mathcal{B}$ is still able to reject invalid first level ciphertexts. For a first level ciphertext $\mathrm{CT}_j = (t, C_1, C_2', C_3, C_4)$ under public key $pk_j = A_1^{x_j}$, the validity of t, C_1, C_3 and C_4 can be ensured as in eq. (1). So, $\mathcal{B}$ needs only to check the validity of the component C_2'. Suppose $C_1 = g_1^r, C_3 = [F(K, C_1)]_{\ell-\ell_1}\|([F(K, C_1)]^{\ell_1} \oplus m)$ and $C_4 = \left(u^{H(C_1,C_3)}v^t w\right)^r$, where $K = e(g, g)^r$. To check the validity of C_2', $\mathcal{B}$ needs to check whether $C_2' = e(pk_j, g)^r$ holds. Fortunately, $\mathcal{B}$ can obtain $e(pk_j, g)^r$ by computing

$$e(C_1, A_{-1})^{\frac{x_j}{\alpha_0}} = (g_1^r, A_{-1})^{\frac{x_j}{\alpha_0}} = (A_2^{\alpha_0 r}, A_{-1})^{\frac{x_j}{\alpha_0}} = e(A_1^{x_j}, g)^r = e(pk_j, g)^r.$$

`Challenge.` When $\mathcal{A}$ judges that `find` stage is over, it outputs a public key pk_{i^*} and two equal-length messages $m_0, m_1 \in \{0,1\}^{\ell_1}$ with the restrictions specified in the $\mathsf{IND\text{-}1PRE\text{-}CCA}$ game. $\mathcal{B}$ first recovers tuple $(pk_{i^*}, x_{i^*}, c_{i^*})$ from L^{list}. If $c_{i^*} = 1$, $\mathcal{B}$ outputs a random bit and **aborts**. Otherwise, $\mathcal{B}$ picks $\delta \xleftarrow{\$} \{0,1\}$, defines $C_1^* = B^{\alpha_0}, C_2'^* = e(A_{-1}, B)^{x_{i^*}}, C_3^* = [F(Q, C_1^*)]_{\ell-\ell_1}\|([F(Q, C_1^*)]^{\ell_1} \oplus m_\delta)$, $t^* = -\frac{-\alpha_1 h^*+\alpha_3}{\alpha_2}$ and $C_4^* = B^{(\beta_1 h^*+\beta_2 t^*+\beta_3)}$, where $h^* = H(C_1^*, C_3^*)$, and returns $\mathrm{CT}^* = (t^*, C_1^*, C_2'^*, C_3^*, C_4^*)$ as the challenge ciphertext to $\mathcal{A}$.

Observe that, if $Q = e(g, g)^{\frac{b}{a^2}}$, CT^* is indeed a valid challenge ciphertext under public key pk_{i^*}. Letting $r^* = \frac{b}{a^2}$, the well-formedness of C_1^*, C_3^* and C_4^* can be seen as in the proof Theorem 1; while for $C_2'^*$, its well-formedness can be seen as below:

$$C_2'^* = e(A_{-1}, B)^{x_{i^*}} = e(g^{1/a}, g^b)^{x_{i^*}} = e(g^{a\cdot x_{i^*}}, g)^{\frac{b}{a^2}} = e(pk_{i^*}, g)^{r^*}.$$

On the other hand, when Q is uniform and independent in $\mathbb{G}_T$, the challenge ciphertext CT^* is independent of δ in the adversary's view.

`Guess Stage.` $\mathcal{A}$ continues to issue the rest queries. $\mathcal{B}$ can respond these queries for $\mathcal{A}$ as in the `find stage`, since $\mathcal{A}$ has to follow the restrictions described in the $\mathsf{IND\text{-}1PRE\text{-}CCA}$ game and the hash function H is target collision resistant.

`Output.` Eventually, $\mathcal{A}$ returns a guess $\delta' \in \{0,1\}$. If $\delta' = \delta$, $\mathcal{B}$ outputs $\beta' = 1$; else, outputs $\beta' = 0$.

This completes the description of the simulation. We next begin to analyze the simulation. It is clear that the simulations of oracles $\mathcal{O}_{pk}$ and $\mathcal{O}_{rk}$ are perfect. The simulation of decryption $\mathcal{O}_{1d}$ is also perfect, unless $\alpha_1 h + \alpha_2 t + \alpha_3 = 0 \mod p$ happens during the whole simulation (denote this event by DecErr). Similarly to the proof in Theorem 1, we can have $\Pr[\mathsf{DecErr}] \leqslant \frac{q_d}{p}$. Let Abort denote the event of $\mathcal{B}$'s aborting during the simulation of oracle $\mathcal{O}_{sk}$ or in `Challenge` phase. Similarly to the proof of Theorem 1, we have $\Pr[\neg\mathsf{Abort}] \leqslant \frac{1}{\dot{e}(1+q_{sk})}$.

Combining the above results and counting for the target collision resistant of the hash function H and the pseudorandomness of F, we can easily see that $\mathcal{B}$'s advantage against the 3-wDBDHI assumption is at least $\epsilon' \geqslant \frac{\epsilon}{\dot{e}(1+q_{sk})} - \frac{q_d}{p} - \mathrm{Adv}_{H,\mathcal{A}}^{\mathsf{TCR}} - \mathrm{Adv}_{F,\mathcal{F}}^{\mathsf{PRF}}$, and $\mathcal{B}$'s running time can be bounded by $t' \leqslant t + \mathcal{O}(\tau(q_{pk}+q_{rk}+8q_d))$. This completes the proof of Theorem 2.

From Lemma 1 and Theorem 2, we have

Theorem 3 Our scheme is MSS-PRE secure in the standard model, assuming the 3-wDBDHI assumption holds in groups $(\mathbb{G}, \mathbb{G}_T)$.

4.3 Comparisons

In Table 1, we compare our scheme with Livert-Vergnaud's scheme[5] (denoted by LV08). We first explain some notations used in Table 1. Here $|\mathbb{G}|$ and $|\mathbb{G}_T|$ denote the bit-length of an element in groups $\mathbb{G}$ and $\mathbb{G}_T$, respectively. $|\mathcal{M}|$ denotes the bit-length of the plaintext in LV08 scheme, and ℓ denotes the security parameter in our scheme. $|svk|$ and $|\sigma|$ denote the bit-length of the verification key and signature of one-time signature used in LV08 scheme[5], respectively. We use $t_p, t_e, t_{me}, t_s, t_v$ to represent the computational cost of a bilinear pairing, an exponentiation, a multi-exponentiation, one signing and one verifying a one-time signature, respectively.

Table 1: Comparisons between LV08 Scheme and Our Scheme

Schemes		Our Scheme	LV08 Scheme[5]
Ciphertext Length	2-level ciphtxt	$\lvert\mathbb{Z}_p\rvert+3\lvert\mathbb{G}\rvert+\ell$	$\lvert svk\rvert+2\lvert\mathbb{G}\rvert+1\lvert\mathbb{G}_T\rvert+\lvert\sigma\rvert$
	1-level ciphtxt	$\lvert\mathbb{Z}_p\rvert+2\lvert\mathbb{G}\rvert+1\lvert\mathbb{G}_T\rvert+\ell$	$\lvert svk\rvert+4\lvert\mathbb{G}\rvert+1\lvert\mathbb{G}_T\rvert+\lvert\sigma\rvert$
Computational Cost	Enc_1	$1t_{me}+3t_e$	$1t_{me}+4t_e+1t_s$
	Enc_2	$1t_{me}+3t_e$	$1t_{me}+2t_e+1t_s$
	ReEnc	$3t_p+2t_{me}$	$2t_p+4t_e+1t_v$
	Dec_2	$3t_p+2t_{me}+1t_e$	$3t_p+2t_e+1t_v$
	Dec_1	$2t_p+2t_{me}+1t_e$	$5t_p+2t_e+1t_v$
Security		CCA	RCCA
Corruption Model		Adaptive	Non-adaptive
Without RO?		Yes	Yes

The comparison results indicate that our scheme has a better overall performance than LV08 scheme in term of both ciphertext length and computational cost. Most importantly, our scheme achieve the CCA-security, while LV08 scheme only satisfies the RCCA-security. The latter is a weaker variant of CCA-security in the sense that it cannot withstand the attack by re-randomizing the challenge ciphertext. In addition, our scheme can be proved in the adaptive corruption model, while LV08 scheme cannot.

5 Conclusions

We presented a unidirectional proxy re-encryption scheme, and proved its CCA-security in the adaptive corruption model without random oracles. Compared with the best know unidirectional PRE scheme proposed by Libert and Vergnaud, our scheme enjoys the advantages of both higher efficiency and stronger security.

Many interesting questions still remain. For example, (1) how to construct a CCA-secure multi-hop unidirectional PRE scheme (either with or without random oracles); (2) study impossibility (or possibility) of constructing a CCA-secure (either bidirectional or unidirectional) PRE scheme without parings in the standard model; (3) how to construct a CCA-secure PRE scheme with key privacy[25].

Acknowledgement

We would like to thank Sherman C. M. Chow, Cheng-Kang Chu and Junzuo Lai for the helpful discussions on the topic of proxy re-encryption. We thank Jun Shao for his pointing out a problem in the simulation of the re-encryption key oracle in the proof of our journal version.

1 Blaze M, Bleumer G, Strauss M. Divertible protocols and atomic proxy cryptography. In: Nyberg K, ed. EUROCRYPT. Lecture Notes in Computer Science, Vol 1403. Berlin: Springer-Verlag, 1998. 127-144

2 Ateniese G, Fu K, Green M, et al. Improved proxy re-encryption schemes with applications to secure distributed storage. In: proceedings of NDSS'05, The Internet Society. 2005

3 Ateniese G, Fu K, Green M, et al. Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans. Inf. Syst. Secur, 2006, 9(1): 1-30

4 Canetti R, Hohenberger S. Chosen-ciphertext secure proxy re-encryption. In Ning P, di Vimercati S.D.C, Syverson P F, eds. ACM Conference on Computer and Communications Security, ACM, 2007, 185-194

5 Libert B, Vergnaud D. Unidirectional chosen-ciphertext secure proxy re-encryption. In Cramer R, ed. Public Key Cryptography. Lecture Notes in Computer Science, Vol 4939. Berlin: Springer-Verlag, 2008, 360-379 (full paper available at http://hal.inria.fr/inria-00339530/en/)

6 Canetti R, Krawczyk H, Nielsen J B. Relaxing chosenciphertext security. In Boneh D, ed. CRYPTO. Lecture Notes in Computer Science, Vol 2729. Berlin: Springer-Verlag, 2003, 565-582

7 Shao J. Proxy re-cryptography revisited (in chinese). PhD thesis. China Academic Journal Online Publication Integrated Database. 2008.

8 Deng R H, Weng J, Liu S, et al. Chosen-ciphertext secure proxy re-encryption without pairings. In Franklin M K, Hui L C K, Wong DS, eds. CANS. Lecture Notes in Computer Science, Vol 5339. Berlin: Springer-Verlag, 2008, 1-17

9 Shao J, Cao Z. CCA-secure proxy re-encryption without pairings. In Jarecki S, Tsudik G, eds. Public Key Cryptography. Lecture Notes in Computer Science, Vol 5443. Berlin: Springer-Verlag, 2009, 357-376

10 Tang Q. Type-based proxy re-encryption and its construction. In Chowdhury D R, Rijmen V, Das A, eds. INDOCRYPT. Lecture Notes in Computer Science, Vol 5365. Berlin: Springer-Verlag, 2008, 130-144

11 Weng J, Deng R H, Ding X, et al. Conditional proxy re-encryption secure against chosen-ciphertext attack. In Li W, Susilo W, Tupakula U K, Safavi-Naini R, Varadharajan V, eds. ASIACCS, ACM, 2009, 322-332

12 Weng J, Yang Y, Tang Q, et al. Efficient conditional proxy re-encryption with chosen-ciphertext security. In Samarati P, Yung M, Martinelli F, eds. ISC. Lecture Notes in Computer Science, Vol 5735. Berlin: Springer-Verlag, 2009, 151-166

13 Chu C K, Weng J, Chow S S M, et al. Conditional proxybroadcast re-encryption. In Boyd C, Nieto J M G, eds. ACISP. Lecture Notes in Computer Science, Vol 5594. Berlin: Springer-Verlag, 2009, 327-342

14 Libert B, Vergnaud D. Tracing malicious proxies in proxy re-encryption. In Galbraith S D, Paterson K G, eds. Pairing. Lecture Notes in Computer Science, Vol 5209. Berlin: Springer-Verlag, 2008, 332-353

15 Matsuo T. Proxy re-encryption systems for identity-based encryption. In Takagi T, Okamoto T, Okamoto E, Okamoto T, eds. Pairing. Lecture Notes in Computer Science, Vol 4575. Berlin: Springer-Verlag, 2007, 247-267

16 Green M, Ateniese G. Identity-based proxy re-encryption. In Katz J, Yung M, eds. ACNS. Lecture Notes in Computer Science, Vol 4521. Berlin: Springer-Verlag, 2007, 288-306

17 Chu C K, Tzeng W G. Identity-based proxy re-encryption without random oracles. In: Garay J A, Lenstra A K, Mambo M. et al. eds. ISC. Lecture Notes in Computer Science, Vol 4779. Berlin: Springer-Verlag, 2007, 189-202

18 Cramer R, Shoup V. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Comput, 2004, 33(1): 167-226

19 Goldreich O, Goldwasser S, Micali S. How to construct random functions. J. ACM, 1986, 33(4): 792-807

20 Boneh D, Boyen X. Efficient selective-ID secure identity-based encryption without Random Oracles. In: Cachin C, Camenisch J, eds. EUROCRYPT. Lecture Notes in Computer Science, Vol 3027. Berlin: Springer-Verlag, 2004, 223-238

21 Hohenberger S, Waters B. Realizing hash-and-sign signatures under standard assumptions. In: Joux A, ed. EUROCRYPT. Lecture Notes in Computer Science, Vol 5479. Berlin: Springer-Verlag, 2009, 333-350

22 Lai j, Deng R H, Liu S, Kou Wei. Efficient CCA-Secure PKE from Identity-Based Techniques. In: Pieprzyk J, ed. CT-RSA. Lecture Notes in Computer Science, Vol 5985. Berlin: Springer-Verlag, 2010, 132-147

23 Coron J S. On the exact security of full domain hash. In Bellare M, ed. CRYPTO. Lecture Notes in Computer Science, Vol 1880. Berlin: Springer-Verlag, 2000. 229-235

24 Chow S.S, Weng J, Yang Y, Deng R H. Efficient unidirectional proxy re-encryption. In Bernstein D J and T. Lang eds. AFRICACRYPT. Lecture Notes in Computer Science, Vol 6055. Berlin: Springer-Verlag, 2010. 316-332

25 G. Ateniese, K. Benson, S. Hohenberger. Key-Private Proxy Re-encryption. In M. Fischlin ed. CT-RSA 2009. Lecture Notes in Computer Science, Vol 5473. Berlin: Springer-Verlag, 2009. 279-294