PRE: Stronger Security Notions and Efficient Construction with New Property

Jiang Zhang, Zhenfeng Zhang, and Yu Chen

Institute of Software, Chinese Academy of Sciences, Beijing, 100190, China
jiangzhang09@gmail.com,zfzhang@is.iscas.ac.cn,cycosmic@gmail.com

Abstract. In a proxy re-encryption (PRE) scheme, a proxy is given a re-encryption key and has the ability to translate a ciphertext under one key into a ciphertext of the same message under a different key, without learning anything about the message encrypted under either key. PREs have been widely used in many exciting applications, such as email forwarding and law enforcement. Based on a good observation on the applications of PREs, we find that a PRE receiver needs an ability, just like what is provided by public-key encryption with non-interactive opening, to non-interactively and efficiently convince third parties of what he obtains from a particular (transformed) ciphertext, while still keeping the security of his secret key and other ciphertexts.

To meet such a practical requirement, we first introduce proxy re-encryption with non-interactive opening (PRENO), and formally define the notions of security against *chosen ciphertext attacks* (CCA) and *proof soundness*. Our security model is natural and strong since we allow the CCA adversary to adaptively choose public keys for malicious users (i.e., a chosen key model), and a scheme secure in previous models (i.e., knowledge of secret key models) is not necessarily secure in our model. Then, we present an efficient PRENO scheme which satisfies our security notions based on the decisional bilinear Diffie-Hellman (DBDH) assumption in the standard model. Compared with two previous PRE schemes, our scheme is competitive in several aspects. First, its CCA security is proved in a strong security model under a well-studied assumption in the standard model. Second, it has a good overall performance in terms of ciphertext length and computational cost. Third, it first provides non-interactive opening for PRE schemes.

1 Introduction

In 1998, Blaze, Bleumer, and Strauss [5] proposed the notion of "atomic proxy re-encryption", in which a semi-trusted proxy is given a re-encryption key that allows it to translate a ciphertext under one key into a ciphertext of the same message under a different key, without seeing the underlying plaintext. Blaze et al. [5] called a proxy re-encryption (PRE) scheme is *bidirectional* if a re-encryption key $rk_{1,2}$ allows the proxy to translate ciphertexts under the delegator's public key pk_1 to ciphertexts under the delegatee's public key pk_2 and *vice versa*, and called a scheme is *unidirectional* if a re-encryption key $rk_{1,2}$ only allows the proxy to translate ciphertexts under pk_1 to ciphertexts under pk_2. They also gave another method to classify PRE schemes, namely, a scheme is *single-hop* [34,44,35] if the ciphertext can only be transformed once, otherwise it is *multi-hop* [9].

In the last decade, proxy re-encryption has attracted many researchers' attention [5,9,34,39] and has plenty of exciting applications in key management [5], email forwarding [30,27], law enforcement [24], publish/subscribe systems [29], multicast [40], secure file systems [2,3], telemedicine [26], digital right management [43] and so on.

When examining those applications, we find that a PRE receiver has a critical requirement to efficiently convince third parties of that the decryption of some particular ciphertext is what he claims. A naive solution of this requirement is to reveal the receiver's secret key. However, such a solution compromises the security of all other ciphertexts of the receiver, and may not be necessary for practical applications. Actually, a similar requirement has been introduced in the public-key encryption setting, and been satisfied by public-key encryption with non-interactive opening (PKENO) proposed by Damgård and Thorbek [14].

We try to introduce the "non-interactive opening" ability to proxy re-encryption, while keeping the security of the receiver's secret key and other ciphertexts. We call this new primitive "proxy re-encryption with non-interactive opening" (PRENO). At first, we prefer to recall two applications where PREs are used and PRENOs are very useful.

Key Escrow System. In a key escrow system (e.g., [38,18,32,24]), a trusted party is set to mediate the conflicts between users and law enforcement agencies. The problem is to allow the law enforcement agency (without knowing users' secret keys) to read messages encrypted for a set of users, for a limited period of time. In a traditional solution, a key escrow agent holds all users' secret keys, such that whenever the law enforcement agency wants to read a message belonging to a user, the key escrow agent first decrypts the message and then re-encrypts it with the law enforcement agency's public key. In order to prevent the key escrow agent from knowing users' secret keys and plaintexts, Ivan and Dodis [24] used proxy re-encryption scheme in their system, namely, the key escrow agent only holds proxy keys between users and the law enforcement agency, and transforms ciphertexts under users' public keys into ciphertexts under the law enforcement agency's public key when necessary.

Ivan and Dodis's [24] solution for a key escrow system works perfectly so far. However, if we go a step further, it is easy to find that there are still other problems needed to be concerned. For example, whenever finding some criminal messages of a user, say Eve, the law enforcement agency may take those messages as evidence to charge Eve in court. To convince the judge and the public, it is very necessary for the law enforcement agency to prove those illegal messages are indeed the results of decrypting Eve's ciphertexts.

Telemedicine. A telemedicine system (e.g., [17,41,26]) involves patients, doctors and electronic medical records (EMR) servers. In such a system, patients' health information such as blood pressure and blood sugar readings which is collected by portable devices, is sent to the EMR server in encrypted form under the server's public key. Then, the server decrypts the data, and re-encrypts it to corresponding doctors. After making an evaluation, doctors will provide consultations to patients. To protect patients' private information from the EMR server, Kailasam et al. [26] used proxy re-encryption in their telemedicine system–Arogyasree.

A practical telemedicine system can bring good healthcare to patients, especially for those living in an area lacking of doctors. However, we should take care of the doctor-patient relationship, since there are always conflicts between patients and doctors in the real world (e.g., the recent widely concerned trial of pop star Michael Jackson's doctor). Thus, both types of users (i.e., doctors and patients) in telemedicine systems (e.g., [26]) may wish to effectively protect their own interests in probable medical accidents. The most important thing for each user is to keep all the transcripts, and to prove those are actually the decryption results of his received ciphertexts if necessary.

Note that, all receivers in the above applications have to convince others of what they obtain from a ciphertext alone, since the sender might be absent or be unwilling to help prove a claim that would result in negative effects on his own interests. Thus, it is very important to give a mechanism that provides a receiver the ability to non-interactively achieve such a goal. Also, if taking account of practical factors, the mechanism should be efficient, and still keep the security of the receiver's other ciphertexts. As one of our contributions, PRENO is an ideal candidate to handle this problem[1]. Besides, there are many other promising applications for PRENOs, such as Mailing lists [30], publish-subscribe systems [29], multicast key management [40], and securing distributed storage [28].

1.1 Related Work.

In Eurocrypt 2007, Damgård and Thorbek [14] introduced the notion of public-key encryption with non-interactive opening (PKENO). Later, two generic constructions [13,45] from identity-based encryption (IBE) to PKENO have been proposed. Galindo et al. [19] gave another generic construction from any robust non-interactive threshold encryption (TPKE) to PKENO. Most recently, Lai et al. [31] constructed a PKENO scheme by combining the identity-based techniques in [8] and [7].

As for proxy re-encryption, Mambo and Okamoto [37] proposed the idea of delegating decryption rights. Blaze et al. [5] later introduced the notion of "atomic proxy re-encryption", and gave a bidirectional scheme. Since then, many works of handling different problems appeared in the literature [25,24,46,2,3]. However, almost all the constructions above are only secure against *chosen plaintext attacks (CPA)*, which may not meet the security requirement in some applications (e.g., encrypted email forwarding [9]).

In CCS 2007, Canetti and Hohenbergery [9] studied the security against *chosen ciphertext attacks (CCA)* for bidirectional PRE schemes, and proposed a scheme that satisfied their security definition in the standard model. They also left an open problem to construct a CCA secure unidirectional PRE scheme (in the standard model). Later, Libert and Vergnaud [34,35] considered the security for single-hop unidirectional proxy re-encryption scheme in the sense of *replayable chosen ciphertext attacks (RCCA)* [10]. Informally, in the RCCA security game, the challenger rejects any decryption query that the underlying plaintext is either m_0 or m_1 in the second phase[2]. Libert et al. [34] proposed the first RCCA secure single-hop unidirectional PRE scheme based on the 3-Quotient Decision Bilinear Diffie-Hellman (3-QDBDH) assumption in the standard model. As in [9], they used the CHK technique [8] to achieve RCCA security, with a price of increasing computational cost and ciphertext length.

There are several unidirectional PRE schemes [2,42,11] in the random oracle model. In PKC 2009, Shao et al. [42] constructed a unidirectional single-hop PRE scheme without pairings. Concretely, they constructed a scheme based on the DDH assumption over $\mathbb{Z}_{N^2}$ in the random oracle model. As the big size of the modulus N is needed (at least 1024 bits), Shao et al. noticed that their scheme needs more time for computation and more storage for ciphertext than the scheme using pairings in the standard model [34][3].

In 2010, Weng et al. [44] proposed a single-hop unidirectional scheme also based on the 3-QDBDH assumption. They tried to use pseudorandom functions (PRFs) to achieve CCA security, and a possibly implicit assumption in their security proof is that the outputs of PRFs with two related keys and the same inputs are mutually independent and still pseudorandom. As far as we

[1] Non-interactive Zero Knowledge proof (NIZK) is another possible solution, but as far as we know, NIZKs (in the standard model) are too inefficient to be practical.

[2] m_0, m_1 are the two equal length messages submitted by the adversary in the challenge phase.

[3] Shao et al. [42] claimed their scheme is CCA secure, but a following work [11] pointed out that Shao et al.'s scheme is vulnerable to chosen ciphertext attacks. However, the attack can be fixed with one more check in the decryption algorithm.

know, it is unclear whether PRFs can provide such a guarantee. Moreover, they claimed that their scheme could allow the adversary to adaptively corrupt users, but their scheme essentially follows the routine of [34]. The strategy of using Coron's technique [12], namely determining whether a key is honestly generated or not beforehand by taking random coins, may not work in the PRE setting due to the second phase re-encryption key queries especially from the challenge public key to honestly generated ones (i.e., predetermined for users' corruption but the adversary did not corrupt).

Libert et al. [35] considered a more natural and realistic security model for PREs (also in the sense of RCCA), namely, the chosen key (CK) model wherein the adversary itself can adaptively choose malicious users' public keys. They also noted that almost all previous schemes only considered security in the knowledge of secret key (KOSK) model, which they conjectured to be weaker than the CK model although they "did not find a strict separation in the context of proxy re-encryption". Besides, they proposed a temporary PRE scheme proved RCCA secure in the CK model.

More recently, Hanaoka et al. [22] gave a generic construction from threshold public key encryption (TPKE) with special property (i.e., what they called resplittable TPKE [22]) to PRE. Their strategy is clear, but the instantiation is somewhat inefficient. We also note that Hanaoka et al.'s generic construction [22] seems not to give light on how to obtain a PRENO scheme. Besides, there are many other fruitful results on proxy cryptography, such as identity-based PRE schemes [21,36], key-private PRE schemes [1], and proxy re-signature schemes (PRS) [4,33].

1.2 Our Contribution.

In this paper, we first consider a strong CCA security model for PRE(NO) schemes, namely, the CK model wherein the adversary can adaptively choose public keys for malicious users. Then, we give a single-hop unidirectional PRENO scheme, which satisfies our security notions under DBDH assumption in the standard model. Our PRENO scheme is very efficient and very useful in practice.

Many previous PRE schemes (e.g.,[34,42,44]) are only proved secure in the KOSK model, where the challenger generates all users' public keys and can (and perhaps must) make use of the malicious users' secret keys to go through the security proof. Libert et al. [35] noted that the KOSK model might be somewhat unrealistic in practice. In fact, the "knowledge of secret key" requirement is relatively strong in the context of unidirectional PRE, since it is hard and not necessary for the delegator to check whether a given delegatee's public key is properly generated (e.g., when generating the re-encryption key). Thus, our CK model seems more natural and powerful. Actually, we show that the scheme in [34] proved secure in the KOSK model is not secure in our CK model, which gives a strict separation between KOSK and CK.

As for non-interactive opening, maybe the simplest way is to give third parties the ability to decrypt the particular ciphertext. Several previous PKENO schemes are constructed from identity-based encryption (IBE) schemes where a user's secret key is the IBE scheme's master secret key, and each ciphertext is associated with a unique identity. Then, the proof of a particular ciphertext is given by the secret key corresponding to the unique identity in the ciphertext.

We try to follow the idea from IBE to PKENO, but in the single-hop unidirectional PRENO setting, this becomes somewhat difficult. Since there are two types of ciphertext at two levels, and we require that both levels' ciphertexts can be treated as "identity-based" ciphertexts. Unfortunately, we find no existing PRE schemes that already have the desired form. To achieve our goals, we imagine the re-encryption algorithm plays a similar role as the "bootstrapping strategy" in Fully

Homomorphic Encryption (FHE) schemes introduced by Gentry [20]. Concretely, the re-encryption algorithm "decrypts" the incoming ciphertext and "encrypts" the resulting message under another public key at the same time by using the re-encryption key[4]. Finally, we obtain a PRENO scheme that is CCA secure in our CK model. Compared with two well-known single-hop unidirectional PRE schemes [34,42], our scheme is competitive in several aspects. This probably shows that our new treatment for re-encryption algorithm could be used to design other efficient and secure PRE schemes with/without random oracles.

Stronger security. Libert et al.'s scheme [34] is proved secure under the 3-QDBDH assumption, which the authors claimed to be hard in generic groups according to the results of Dodis and Yampolskiy [16]. But given an oracle solving DBDH problem, it is easy to solve the 3-QDBDH problem. Thus, it seems more confident to construct cryptographic schemes based on the DBDH assumption rather than the 3-QDBDH assumption. We also note that their scheme [34] is only secure in the sense of RCCA which is strictly weaker than CCA. The scheme in [42] is based on the decision Diffie Hellman assumption over $\mathbb{Z}_{N^2}^*$ in the random oracle model, where N is a big modulus. But it is well-known that a security proof in the random oracle model may not guarantee its security in the standard model, thus, should be avoided whenever possible. Besides, both above schemes [34,42] are only proved secure in the KOSK model.

We employ the "decryption and check" idea to achieve CCA security in a fashion similar to [6] by using collision resistant hash functions and universal hash functions, and finally establish our scheme's CCA security in the CK model based on the collision resistance of hash functions, the generalized leftover hash lemma [15], as well as the DBDH assumption in the standard model. One byproduct of our "decryption and check" strategy is that the proof algorithm only gives proofs for valid ciphertexts (i.e., the decryption results are not equal to $\perp$), which might not be necessary for "non-interactive opening". However, we note that our scheme is still useful and satisfactory for the applications mentioned before where only valid ciphertexts make sense (e.g., the law enforcement agency is not likely to use '$\perp$' as an evidence to charge someone in court, and the doctor may not be willing to give any consultation as a response to an incoming message '$\perp$').

Better efficiency. We use the technique introduced by Hohenberger and Waters in [23] to avoid using one-time signatures. Briefly, we attach "two tags" to each ciphertext, one is randomly chosen and the other is uniquely determined by other elements in the ciphertext. This greatly reduces our scheme's overhead both in computation and communication. In Section 5, we give a detailed comparison between our scheme and two well-known single-hop unidirectional PRE schemes in [34,42], and the result shows that our scheme has a competitive overall performance in terms of computational cost and ciphertext size.

1.3 Organization.

The rest of this paper is organized as follows. After a brief preliminaries section, we give definitions and security models for single-hop unidirectional PRENOs in Section 3. In Section 4, we propose our PRENO scheme. A comparison of related single-hop unidirectional PRE schemes is given in Section 5, and finally we conclude in Section 6.

[4] Note that unlike in FHE schemes, the re-encryption key cannot simply be a ciphertext of the delegator's secret key under the delegatee's public key due to the security of PREs.

2 Preliminaries

2.1 Notation

If x is a string, $|x|$ denotes its length, and if S is a set, $|S|$ denotes its size. Denote $x\|y$ as the bit concatenation of two strings $x, y \in \{0,1\}^*$. We use 1^k to denote the string of k ones for some positive integer k. We use the notation $\leftarrow$ to denote randomly choosing an element from some set (distribution) or indicate the output of some algorithm. For example, $s \leftarrow S$ means that we randomly choose an element s from the set (distribution) S, and $z \leftarrow \mathcal{A}(x, y, \dots)$ means that the output of algorithm $\mathcal{A}$ with inputs $x, y, \dots,$ is z. We say a function $f(n)$ is negligible if for every $c > 0$, there exists a N such that $f(n) < 1/n^c$ for all $n > N$. Usually, we denote an unspecific negligible function by $negl(n)$. We say a probability is overwhelming if it is $1 - negl(n)$.

2.2 Bilinear Groups and Assumptions

Let positive integer k be the security parameter, $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order $p \geq 2^k$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear pairing. The decisional bilinear Diffie-Hellman (DBDH) assumption states that the two distributions $(g^x, g^y, g^z, e(g,g)^{xyz})$ and $(g^x, g^y, g^z, e(g,g)^r)$, where x, y, z, r are randomly and independently chosen from $\mathbb{Z}_p$, are indistinguishable for any polynomial time adversary. Formally, for any probabilistic polynomial time adversary $\mathcal{A}$, its advantage

$$\mathrm{Adv}^{\mathrm{dbdh}}_{\mathcal{PG},\mathcal{A}}(k) = |\Pr[\mathcal{A}(g^x, g^y, g^z, e(g,g)^{xyz}) = 1] - \Pr[\mathcal{A}(g^x, g^y, g^z, e(g,g)^r) = 1]|$$

is negligible in security parameter k, where the probability is over the random choices of x, y, z, r in $\mathbb{Z}_p$ and the random bits of $\mathcal{A}$.

2.3 Families of Hash Functions

Let $\mathcal{H} = \{H : X \to Y\}_{H \in \mathcal{H}}$ be a family of hash functions. We say $\mathcal{H}$ is collision resistant (CR), if there is no polynomial-time algorithm that given a randomly chosen $H \leftarrow \mathcal{H}$, outputs an element x_1, x_2 in X and $x_1 \neq x_2$, such that $H(x_1) = H(x_2)$ holds with non-negligible probability, where the probability is over the random choices of H, and the random bits of the algorithm. We say $\mathcal{H}$ is universal if for all $x_1 \neq x_2$ we have $\Pr_{H \leftarrow \mathcal{H}}[H(x_1) = H(x_2)] = 1/|Y|$.

For a pair of (possibly correlated) random variables A and B, the average min-entropy of A given B is defined as

$$\tilde{H}_\infty(A|B) = -\log\left(\mathbf{Exp}_{b \leftarrow B}\left[\max_a \Pr[A = a|B = b]\right]\right) = -\log\left(\mathbf{Exp}_{b \leftarrow B}\left[2^{-\tilde{H}_\infty(A|B=b)}\right]\right).$$

Lemma 1 (Generalized Leftover Hash Lemma [15]). *If $\mathcal{H} = \{H : X \to Y\}_{H \in \mathcal{H}}$ is a family of universal hash functions. Then, for any random variables W, I, the statistical difference between $(H, H(W), I)$ and (H, U, I) is at most $\frac{1}{2} \cdot \sqrt{2^{-\tilde{H}_\infty(W|I)}|Y|}$, where H and U are uniformally distributed over $\mathcal{H}$ and Y, respectively.*

3 Proxy Re-encryption with Non-interactive Opening

Several works have studied single-hop unidirectional PRE schemes [34,42,22]. In this paper, we also concentrate on single-hop unidirectional PRENO scheme, which consists of the following algorithms:

$\texttt{Setup}(1^k)$: Given the security parameter 1^k, return a public parameter $param$, which specifies plaintext space $\mathcal{P}$, ciphertext space $\mathcal{C}$ and randomness space $\mathcal{R}$.
$\texttt{KeyGen}(param)$: Given the parameter $param$, return a public-secret key pair (pk, sk).
$\texttt{ReKeyGen}(sk_i, pk_j)$: Given a secret key sk_i of user i (i.e., delegator) and a public key pk_j of user j (i.e., delegatee), return the re-encryption key $rk_{i\to j}$.
$\texttt{Enc}(pk_i, m, \delta_i)$: Given a public key pk_i, a message $m \in \mathcal{P}$ and a ciphertext type $\delta_i \in \{1st, 2nd\}$, return a δ_i level ciphertext C_i.
$\texttt{ReEnc}(rk_{i\to j}, C_i)$: Given a re-encryption key $rk_{i\to j}$ and a $2nd$ level ciphertext C_i, return a $1st$ level ciphertext C_j under pk_j or an error symbol $\perp$.
$\texttt{Dec}(sk_i, C_i, \delta_i)$: Given a secret key sk_i and a $\delta_i \in \{1st, 2nd\}$ level ciphertext C_i, return a message $m \in \mathcal{P}$ or an error symbol $\perp$.
$\texttt{Prove}(sk_i, C_i, \delta_i)$: Given a secret key sk_i and a $\delta_i \in \{1st, 2nd\}$ level ciphertext C_i, return a proof π_i of the decryption result of C_i.
$\texttt{Ver}(pk_i, C_i, \delta_i, m_i, \pi_i)$: Given a public key pk_i, a $\delta_i \in \{1st, 2nd\}$ level ciphertext C_i, a message m and a proof π_i, return 1 if π_i is a valid proof for $(pk_i, C_i, \delta_i, m_i)$, else return 0.

For correctness, we require that for any key pair (pk, sk), any message $m \in \mathcal{P}$, and any $\delta \in \{1st, 2nd\}$ level ciphertext of m under public key pk that is output by either `Enc` or `ReEnc`, the following conditions hold with overwhelming probability.

- $\texttt{Dec}(sk, C, \delta) = m$.
- $\texttt{Ver}(pk, C, \delta, m, \texttt{Prove}(sk, C, \delta)) = 1$.

Remark. Previous works usually define two encryption (decryption, resp.) algorithms for two levels ciphertexts. Here, we use a parameter δ to specify which case is in related algorithms for compactness.

3.1 Chosen-Ciphertext Security

Unlike previous KOSK models in [34,42], we consider the CK model, where the adversary can adaptively choose public keys for malicious users. The CK model is strictly stronger than previous KOSK models since in Appendix A, we show that the scheme in [34] is not secure in our CK model. But as previous models [34,35], we also disallow the adversary to adaptively corrupt users[5]. We provide a PRENO adversary the following oracles:

- Honest key generation oracle $\mathcal{O}_{\text{hkg}}(i)$: Compute $(pk_i, sk_i) \leftarrow \texttt{KeyGen}(i)$, and return pk_i.
- Re-encryption key oracle $\mathcal{O}_{\text{rek}}(pk_i, pk_j)$: Given public keys pk_i and pk_j, return the re-encryption key $rk_{i\to j} \leftarrow \texttt{ReKeyGen}(sk_i, pk_j)$.
- Re-encryption oracle $\mathcal{O}_{\text{rec}}(pk_i, pk_j, C_i)$: Given public keys pk_i, pk_j, and a $2nd$ level ciphertext C_i, compute $rk_{i\to j} \leftarrow \texttt{ReKeyGen}(sk_i, pk_j)$, and return a $1st$ level ciphertext $C_j \leftarrow \texttt{ReEnc}(rk_{i\to j}, C_i)$.

[5] Note that the CK model does not necessarily mean the adversary can adaptively corrupt users, but it does imply the adversary can statically corrupt users. Thus, there is no need to explicitly provide a malicious key generation oracle as in previous models. Besides, it is still an open problem to handle fully adaptive corruptions [35].

– Decryption oracle $\mathcal{O}_{\text{dec}}(pk_i, C_i, \delta_i)$: Given a public key pk_i and a $\delta_i \in \{1st, 2nd\}$ level ciphertext C_i, return $m \leftarrow \texttt{Dec}(sk_i, C_i, \delta_i)$.
– Proof oracle $\mathcal{O}_{\text{pf}}(pk_i, C_i, \delta_i)$: Given a public key pk_i and a $\delta_i \in \{1st, 2nd\}$ level ciphertext C_i, return $\pi_i \leftarrow \texttt{Prove}(sk_i, C_i, \delta_i)$.

Libert et al. [34] claimed that it is useless to explicitly provide a $2nd$ level decryption oracle to the adversary. However, Hanaoka et al. [22] gave a counterexample that is insecure in the model with a $2nd$ level decryption oracle while still secure in the model without. Thus, we allow the adversary to make $2nd$ level decryption queries for stronger security.

For convenience, we say a public key pk_i is *good* if it is output by $\mathcal{O}_{\text{hkg}}(i)$, otherwise we say it is *bad* (i.e., chosen by the adversary). Throughout the paper, we require all the public keys used as the first inputs in $\mathcal{O}_{\text{rek}}(\cdot)$ and $\mathcal{O}_{\text{rec}}(\cdot)$, as well as the public keys in $\mathcal{O}_{\text{dec}}(\cdot)$ and $\mathcal{O}_{\text{pf}}(\cdot)$ are output by $\mathcal{O}_{\text{hkg}}(\cdot)$. This is natural, since other public keys are chosen by the adversary, the challenger does not know the corresponding secret keys (it is very possible that the adversary does not know them either). Moreover, we simply use the integer (i.e., i) as the unique identifier of a user. Note that in our model, the adversary can arbitrarily assign a "public key" to a user even through he may not know the corresponding secret key. Especially, the adversary can set the public key of user j exactly equal to the public key of user i by setting $pk_j = pk_i$. In this case, it is natural to treat pk_j as a *bad* public key no matter whether pk_i is *bad* or not.

To capture the CCA security for single-hop unidirectional PRENO schemes, we follow a routine in [34,42,44] to separably consider the security of ciphertexts at two levels. First, we associate to a PRENO adversary $\mathcal{A}$ the following **PRENO-CCA** abstract experiment with parameters $(\mathcal{O}', \delta)$, where $\mathcal{O}'$ is a set of oracles provided to $\mathcal{A}$, and $\delta \in \{1st, 2nd\}$ specifies which level ciphertext that $\mathcal{A}$ attacks. Both parameters will be instantiated in Definition 1 and Definition 2.

```
**Experiment** $\mathbf{Exp}^{\text{preno-cca}}_{\Pi_s,\mathcal{A}}(1^k)$
    $param \leftarrow \texttt{Setup}(1^k)$
    $(pk_{i^*}, m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}'}(param)$
    $b \leftarrow \{0,1\}$
    $C^* \leftarrow \texttt{Enc}(pk_{i^*}, m_b, \delta)$
    $b' \leftarrow \mathcal{A}^{\mathcal{O}'}(param, C^*)$
    If $b = b'$ return 1, else return 0
```

The advantage of $\mathcal{A}$ is defined as $\text{Adv}^{\text{preno-cca}}_{\Pi_s,\mathcal{A}}(k) = |\Pr[\mathbf{Exp}^{\text{preno-cca}}_{\Pi_s,\mathcal{A}}(1^k) = 1] - \frac{1}{2}|$.

Definition 1 (CCA Security for the 2nd Level Ciphertext). *For any single-hop unidirectional PRENO Π_s, we instantiate the* **PRENO-CCA** *experiment with $\mathcal{O}' = \{\mathcal{O}_{hkg}, \mathcal{O}_{rek}, \mathcal{O}_{rec}, \mathcal{O}_{dec}, \mathcal{O}_{pf}\}$ and $\delta = 2nd$. We require that pk_{i^*} is good, $|m_0| = |m_1|$, and $\mathcal{A}$ never makes a query $\mathcal{O}_{rek}(pk_{i^*}, pk_j)$ where pk_j is bad. $\mathcal{A}$ is also disallowed to make the following queries after seeing C^*:*

– *Decryption query $\mathcal{O}_{dec}(pk_{i^*}, C^*, 2nd)$ or proof query $\mathcal{O}_{pf}(pk_{i^*}, C^*, 2nd)$.*
– *Re-encryption query $\mathcal{O}_{rec}(pk_{i^*}, pk_j, C^*)$ where pk_j is bad.*
– *Decryption query $\mathcal{O}_{dec}(pk_j, C', 1st)$ or proof query $\mathcal{O}_{pf}(pk_j, C', 1st)$ where*
  - *$C' = \mathcal{O}_{rec}(pk_{i^*}, pk_j, C^*)$,* *[C1]*
  - *or $\mathcal{A}$ has made a query $rk_{i^*\to j} \leftarrow \mathcal{O}_{rek}(pk_{i^*}, pk_j)$ and $C' = \texttt{ReEnc}(rk_{i^*\to j}, C^*)$.* *[C2]*

We say Π_s is secure against chosen-ciphertext attacks at the 2nd level if for any polynomial-time adversary $\mathcal{A}$, the advantage function $\mathrm{Adv}_{\Pi_s,\mathcal{A}}^{\textit{preno-cca}}(k)$ *is negligible in k.*

Remark. The power of the CK model is implicitly covered by Definition 1, and one can figure it out from the following two points: (1) The adversary can obtain the re-encryption key from any *good* public key pk_i to any public key pk_j by using a query $\mathcal{O}_{\text{rek}}(pk_i, pk_j)$, only with a restriction that pk_j cannot be chosen by the adversary if $pk_i = pk_{i^*}$; (2) The adversary can transform any 2*nd* level ciphertext C_i under *good* public key pk_i to a 1*st* level ciphertext under any public key pk_j by using a query $\mathcal{O}_{\text{rec}}(pk_i, pk_j, C_i)$, only with a restriction that $(pk_i, C_i) \neq (pk_{i^*}, C^*)$ if pk_j is chosen by the adversary. This means the adversary still can use either pk_{i^*} or C^* (but not both) in any re-encryption query. Moreover, all the restrictions in our definition only aim at ruling out the trivial attacks, thus should be considered as the minimal requirements for non-trivial definitions.

In single-hop proxy re-encryption schemes, a 1*st* level ciphertext cannot be re-encrypted any more, thus there is no need to keep re-encryption keys secret when considering ciphertexts' security at the 1*st* level. In fact, we give adversaries all re-encryption keys by providing a re-encryption key oracle without restrictions (thus re-encryption oracle is useless). Formally,

Definition 2 (CCA Security for the 1st Level Ciphertext). *For any single-hop unidirectional PRENO Π_s, we instantiate the* **PRENO-CCA** *experiment with $\mathcal{O}' = \{\mathcal{O}_{hkg}, \mathcal{O}_{rek}, \mathcal{O}_{dec}, \mathcal{O}_{pf}\}$ and $\delta = 1st$. We require that pk_{i^*} is good, $|m_0| = |m_1|$, and that $\mathcal{A}$ is disallowed to make decryption query $\mathcal{O}_{dec}(pk_{i^*}, C^*, 1st)$ or proof query $\mathcal{O}_{pf}(pk_{i^*}, C^*, 1st)$ after seeing ciphertext C^*. We say Π_s is secure against chosen-ciphertext attacks at the 1st level if for any polynomial-time adversary $\mathcal{A}$, the advantage function* $\mathrm{Adv}_{\Pi_s,\mathcal{A}}^{\textit{preno-cca}}(k)$ *is negligible in k.*

3.2 Proof Soundness

Informally, the *proof soundness* says that a dishonest receiver cannot convince others of a fake claim, that the decryption result of a ciphertext c with underlying plaintext m is another message m' (i.e., $m' \neq m$). In our definition, we allow the adversary to output a target ciphertext at either level under any public key it chooses, as well as two proofs for two different messages respectively. The adversary wins the experiment if and only if the verification algorithm accepts the two proofs simultaneously. A formal definition is given in the following definition:

Definition 3 (Proof Soundness). *A single-hop unidirectional PRENO scheme Π_s is proof sound if for any PPT adversary $\mathcal{C}$, its advantage defined below is negligible*

$$\mathrm{Adv}_{\Pi_s,\mathcal{C}}^{\textit{preno-pfsnd}}(k) = \Pr[\textit{param} \leftarrow \mathtt{Setup}(1^k), (pk, C, \delta, m, \pi, m', \pi') \leftarrow \mathcal{C}(\textit{param}) : \\ \mathtt{Ver}(pk, C, \delta, m, \pi) = 1 = \mathtt{Ver}(pk, C, \delta, m', \pi') \wedge m \neq m']$$

Our *proof soundness* definition is similar to the *committing property* in the public key encryption with non-interactive opening setting [19], which is stronger than the original *proof soundness* definition for PKENO [13].

4 A Unidirectional PRENO Scheme

In this section, we give a single-hop unidirectional PRENO scheme. Our scheme is very efficient, and its CCA security is guaranteed by the DBDH assumption in the standard model.

4.1 Our Construction

Let k be the security parameter, $\mathbb{G}$ and $\mathbb{G}_T$ be groups of prime order p, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. Let $\mathcal{H}_0 : \mathbb{G} \times \{0,1\}^l \to \mathbb{Z}_p$ and $\mathcal{H}_1 : \mathbb{G}_T \to \{0,1\}^{l_1}$ be families of collision resistant hash functions, and $\mathcal{H}_2 : \mathbb{G}_T \to \{0,1\}^{l_2}$ be a family of universal hash functions, where $l = l_1 + l_2$ is a parameter related to the security parameter[6]. The message space is $\mathcal{P} = \{0,1\}^{l_2}$. The description of our single-hop unidirectional PRENO scheme Π_s is given below.

$\texttt{Setup}(1^k)$: Given the security parameter 1^k, randomly choose $g, h_1, h_2, u, v, w \leftarrow \mathbb{G}$, $H_0 \leftarrow \mathcal{H}_0$, $H_1 \leftarrow \mathcal{H}_1$ and $H_2 \leftarrow \mathcal{H}_2$, and compute $Y = e(h_1, h_2)$. Return the system parameter $param = (g, h_1, h_2, u, v, w, Y, H_0, H_1, H_2)$.

$\texttt{KeyGen}(param)$: Given the system parameter $param$, randomly choose $\alpha_i, \beta_i \leftarrow \mathbb{Z}_p$, compute public key $pk_i = (pk_{i,1} = h_1^{\alpha_i}, pk_{i,2} = g^{\beta_i})$, and set secret key $sk_i = (sk_{i,1} = \alpha_i, sk_{i,2} = \beta_i)$. Return the key pair (pk_i, sk_i).

$\texttt{ReKeyGen}(sk_i, pk_j)$: Given a secret key sk_i and a public key pk_j, compute $rk_{i\to j} = (h_2 \cdot pk_{j,2})^{1/sk_{i,1}}$. Return the re-encryption key $rk_{i\to j}$.

$\texttt{Enc}(pk_i, m, \delta_i)$: Given a public key pk_i, a message $m \in \mathcal{P}$, and a ciphertext type $\delta_i \in \{1st, 2nd\}$, randomly choose $r, s \leftarrow \mathbb{Z}_p$, compute $c_0 = g^r, c_2 = H_1(Y^r)||H_2(Y^r) \oplus m$, and $c_3 = (u^t v^s w)^r$, where $t = H_0(c_0, c_2)$. Finally set $c_1 = pk_{i,1}^r$ if $\delta_i = 2nd$, else $c_1 = Y^r \cdot e(h_1, pk_{j,2})^r$, return the δ_i level ciphertext $C_i = (s, c_0, c_1, c_2, c_3)$.
For convenience, given a $\delta_i \in \{1st, 2nd\}$ level ciphertext $C_i = (s, c_0, c_1, c_2, c_3)$ and a public key pk_i, we define a function $\texttt{CheckCCA}(pk_i, C_i, \delta_i)$, which first computes $t = H_0(c_0, c_2)$ and returns 1 iff (1) $e(c_3, g) = e(u^t v^s w, c_0)$ and (2) $e(c_1, g) = e(pk_{i,1}, c_0)$ if $\delta_i = 2nd$.

$\texttt{ReEnc}(rk_{i\to j}, C_i)$: Given a re-encryption key $rk_{i\to j}$ and a $2nd$ level ciphertext $C_i = (s, c_0, c_1, c_2, c_3)$, if $\texttt{CheckCCA}(pk_i, C_i, 2nd) \neq 1$, return $\perp$. Otherwise, compute $c_1' = e(c_1, rk_{i\to j}) = Y^r \cdot e(h_1, pk_{j,2})^r$, return the $1st$ level ciphertext $C_j = (s, c_0, c_1', c_2, c_3)$.

$\texttt{Dec}(sk_i, C_i, \delta_i)$: Given a secret key sk_i and a $\delta_i \in \{1st, 2nd\}$ level ciphertext $C_i = (s, c_0, c_1, c_2, c_3)$, return $\perp$ if $\texttt{CheckCCA}(pk_i, C_i, \delta_i) \neq 1$. Otherwise, parse $c_2 = \tau_1||\tau_2$, and compute $K = e(c_1, h_2^{1/sk_{i,1}})$ if $\delta_i = 2nd$, else $K = c_1/e(h_1^{sk_{j,2}}, c_0)$. If $\tau_1 = H_1(K)$, return $m = \tau_2 \oplus H_2(K)$, else return $\perp$.

$\texttt{Prove}(sk_i, C_i, \delta_i)$: Given a secret key sk_i and a $\delta_i \in \{1st, 2nd\}$ level ciphertext $C_i = (s, c_0, c_1, c_2, c_3)$, if $\texttt{Dec}(sk_i, C_i, \delta_i) = \perp$, return $\perp$. Otherwise, randomly choose $\gamma \leftarrow \mathbb{Z}_p$, if $\delta_i = 2nd$, compute $d_1 = pk_{i,1}^{\gamma}, d_2 = h_2^{1/sk_{i,1}}(u^t v^s w)^{\gamma}$, else compute $d_1 = g^{\gamma}, d_2 = h_1^{sk_{j,2}}(u^t v^s w)^{\gamma}$, where $t = H_0(c_0, c_2)$. Finally, return the proof $\pi_i = (d_1, d_2)$.

$\texttt{Ver}(pk_i, C_i, \delta_i, m_i, \pi_i)$: Given a message m, a $\delta_i \in \{1st, 2nd\}$ level ciphertext $C_i = (s, c_0, c_1, c_2, c_3)$, a public key pk_i and a proof $\pi_i = (d_1, d_2)$, if $\pi_i = \perp$ or $m = \perp$, return 0. Else the algorithm returns 0 if $\texttt{CheckCCA}(pk_i, C_i, \delta_i) \neq 1$, or $\delta_i = 2nd$ and $e(pk_{i,1}, d_2) \neq Y \cdot e(d_1, u^t v^s w)$, or $\delta_i = 1st$ and $e(d_2, g) \neq e(h_1, pk_{j,2}) \cdot e(u^t v^s w, d_1)$, where $t = H_0(c_0, c_2)$. Otherwise, parse $c_2 = \tau_1||\tau_2$, compute $K = \frac{e(c_1, d_2)}{e(d_1, c_3)}$ if $\delta_i = 2nd$, else $K = \frac{c_1 \cdot e(c_3, d_1)}{e(d_2, c_0)}$, and $m_i' = \tau_2 \oplus H_2(K)$. If $H_1(K) = \tau_1$ and $m_i' = m_i$, return 1, else return 0.

Remark 1. We employ the "decryption and check" technique to ensure that a $1st$ level ciphertext is properly produced. This seems a little limited if compared to ciphertexts' (fully) public verifiability, especially when the proof algorithm only outputs a $\perp$ for each invalid ciphertext[7].

[6] l_1 and l_2 must be set such that the generalized leftover hash lemma holds in respect to the security parameter k in our security proof.

[7] In fact, our scheme can provide proofs for any $2nd$ level ciphertexts, we omit it just for simple description. Besides, one can consider our scheme has partial public verifiability of ciphertexts due to the $\texttt{CheckCCA}$ function.

However, for those applications (we mentioned before) where proofs are useful only for meaningful message m (instead of $\perp$), our scheme is still satisfactory.

Remark 2. Several optimizations are available. First, one can replace the two equations when $\delta_i = 2nd$ in `CheckCCA` with $e(c_3^x c_1^y, g) = e((u^t v^s w)^x pk_{i,1}^y, c_0)$ by randomly choosing $x, y \in \mathbb{Z}_p^*$ as in [44]. Thus, two pairings are eliminated by only adding one multi-exponentiation. Second, the pairing computation of c_1 when $\delta_i = 1st$ in algorithm `Enc` can be eliminated if one pre-computes the value $Y_2 = e(h_1, pk_{j,2})$, and instead computes $c_1 = (Y \cdot Y_2)^r$. Third, by the collision resistance of $\mathcal{H}_1$, one can omit $e(c_3, g) = e(u^t v^s w, c_0)$ in algorithm `Dec` and `Prove` when $\delta_i = 1st$, and compute $K = \frac{c_1 \cdot e(c_3, g^x)}{e((u^t v^s w)^x \cdot h_1^{sk_{j,2}}, c_0)}$ using randomly chosen $x \in \mathbb{Z}_p^*$ instead, thus another pairing is eliminated.

Remark 3. Ateniese et al. [2] discussed several useful properties for proxy re-encryption schemes. It's easy to check that our scheme is *unidirectional, non-interactive, proxy-invisible, key optimal*, and *collusion-"safe" (or master secret secure)*, where the *master secret security* is implied by the CCA security at the first level [34], we omit the details.

Correctness. A $2nd$ level ciphertext C_i under pk_i has a form of $(s, c_0 = g^r, c_1 = pk_{i,1}^r, c_2 = H_1(Y^r) || H_2(Y^r) \oplus m, c_3 = (u^t v^s w)^r)$ where $t = H_0(c_0, c_2)$. Given a re-encryption key $rk_{i \to j}$, C_i can be transformed into a $1st$ level ciphertext C_j under pk_j with a form $(s, c_0 = g^r, c_1 = Y^r \cdot e(h_1, pk_{j,2})^r, c_2 = H_1(Y^r) || H_2(Y^r) \oplus m, c_3 = (u^t v^s w)^r)$. Also, we note that ciphertexts output by `ReEnc` and $\texttt{Enc}(\cdot, \cdot, 1st)$ have the same distribution.

Correctness for Decryption. Note that `CheckCCA` always returns 1 for properly generated $1st$ or $2nd$ level ciphertexts. Also, the decryption algorithm computes a temporary value K for either level ciphertext, and if $K = Y^r$, the correctness of our decryption algorithm is obvious. In fact, for the $2nd$ level ciphertext, we have $K = e(c_1, h_2^{1/sk_{i,1}}) = e(pk_{i,1}^r, h_2^{1/sk_{i,1}}) = Y^r$. For the $1st$ level ciphertext, we have $K = c_1 / e(h_1^{sk_{j,2}}, c_0) = Y^r \cdot (h_1, pk_{j,2})^r / e(h_1^{sk_{j,2}}, g^r) = Y^r$. This proves that our decryption algorithm is correct.

Correctness for Proof. Our proof algorithm essentially outputs a temporary "decryption key" π for each corresponding ciphertext. The verification algorithm first decrypts a ciphertext by using π, and compares the resulting message m' with a claimed message m, accepts the proof if and only if $m' = m$. Thus we just have to show that given π, the temporary value K is indeed equal to Y^r. For a $2nd$ level proof $\pi_i = (d_1, d_2)$, we have $e(pk_{i,1}, d_2) = Y \cdot e(d_1, u^t v^s w)$ and $K = e(c_1, d_2)/e(d_1, c_3) = (pk_{i,1}^r, d_2)/e(d_1, (u^s v^t w)^r) = Y^r$. For a $1st$ level proof $\pi_j = (d_1, d_2)$, we have $e(d_2, g) = e(h_1, pk_{j,2}) \cdot e(u^t v^s w, d_1)$ and $K = c_1 \cdot e(c_3, d_1)/e(d_2, c_0) = Y^r \cdot e(h_1, pk_{j,2})^r \cdot e((u^s v^t w)^r, d_1)/e(d_2, g^r) = Y^r$. This proves that our verification algorithm is correct.

Security. Our PRENO scheme's security is formally summarized in the following theorem,

Theorem 1. *Assume DBDH problem is hard, $\mathcal{H}_0$ is a family of collision-resistant hash functions, $\mathcal{H}_1$ is a family of collision-resistant and one-way hash functions, and $\mathcal{H}_2$ is a family of universal hash functions, our single-hop unidirectional PRENO scheme Π_s is CCA secure at the $1st$ and $2nd$ level ciphertext, and is proof sound.*

Due to space reason, we defer the proof to Appendix B.

5 Comparisons

In this section, we give a comparison among our scheme and two well-known single-hop unidirectional PRE schemes in literatures. At first, we introduce some notations used in Table 1. We use

Schemes		Our scheme	LV08 [34]	SC09 [42]
Ciphertext Length	1*st* level	$\|\mathbb{Z}_p\| + 3\|\mathbb{G}\| + l$	$\|svk\| + 4\|\mathbb{G}\| + \|\mathbb{G}_T\| + \|\sigma\|$	$2k' + 3\|N_X^2\| + \|m\|$
	2*nd* level	$\|\mathbb{Z}_p\| + 2\|\mathbb{G}\| + \|\mathbb{G}_T\| + l$	$\|svk\| + 2\|\mathbb{G}\| + \|\mathbb{G}_T\| + \|\sigma\|$	$k_1 + 3\|N_X^2\| + 2\|N_Y^2\| + \|m\|$
Computational Cost	2*nd* level Enc	$t_{me} + 3t_e$	$t_{me} + 2t_e + t_s$	$5t_{eN}$
	1*st* level Enc	$t_{me} + 3t_e$	$t_{me} + 4t_e + t_s$	$5t_{eN}$
	ReEnc	$3t_p + 2t_{me}$	$2t_p + 4t_e + t_v$	$2t_{meN} + 1t_{eN}$
	2*nd* level Dec	$3t_p + 2t_{me} + t_e$	$3t_p + 2t_e + t_v$	$2t_{meN} + 2t_{eN}$ or $3t_{meN} + t_{eN}$
	1*st* level Dec	$2t_p + t_{me} + t_e$	$5t_p + 2t_e + t_v$	$5t_{eN}$ or $1t_{meN} + 4t_{eN}$
Security		CCA	RCCA	CCA*
Assumption		DBDH	3-QDBDH	DDH over $\mathbb{Z}_{N^2}$
Standard model		$\surd$	$\surd$	$\times$
Chosen key model		$\surd$	$\times$	$\times$
PRENO		$\surd$	$\times$	$\times$

Table 1. Comparisons of unidirectional PRE(NO) schemes in the standard model (where k' and k_1 are two security parameters in [42]. Besides, the first level encryption algorithm is not directly given in [42].)

$|svk|$, $|\sigma|$, $|\mathbb{G}|$, $|\mathbb{G}_T|$, and $|\mathbb{Z}_p|$ to denote the bit-length of a verification key (of one-time signature scheme in [34]), a signature for a ciphertext, an element in $\mathbb{G}$, an element in $\mathbb{G}_T$, and an integer in $\mathbb{Z}_p$ respectively, where p is the order of $\mathbb{G}$ and $\mathbb{G}_T$. The notation $l = l_1 + l_2$ in our scheme, namely, the total length of H_0's output length and the bit-length of the plaintext space. We also denote t_p, t_e, t_{me}, t_s and t_v as the time for computing a bilinear pairing, an exponentiation, a multi-exponentiation in the bilinear group, a signing algorithm and a verification algorithm, respectively. And N_X, N_Y are two safe-prime modulus with size at least 1024 bits [42], and t_{eN} and t_{meN} represent the time for computing an exponentiation and multi-exponentiation in the $\mathbb{Z}^*_{N^2}$, respectively. The possibly different time for decryption in [42] is due to different secret key input.

Libert and Vergnaud [34] proposed a RCCA secure single-hop unidirectional scheme based on the 3-QDBDH assumption. Later, Shao et al. [42] tried to construct a CCA secure unidirectional single-hop PRE scheme based on the DDH assumption over $\mathbb{Z}_{N^2}$ in the random oracle model, but later their scheme was shown vulnerable to CCA attacks [11]. Due to the big modulus N, Shao et al. observed that their scheme is less efficient than Libert et al.'s scheme [34] in both computation cost and ciphertext length.

We note that all the notations in Table 1 may have different meanings when taking account of details of each scheme in implementations, since the choices of all the parameters are closely related to the security parameter, the hardness of the underlying assumption, the tightness of the security reduction and so on. We only try to use the rough comparison in Table 1 to give some intuitive impression, but the results may not be necessarily right for each time implementation. However, we believe our scheme is competitive in terms of both ciphertext length and computational cost when considering all possible differences among those schemes. Moreover, our construction is a PRENO scheme and its CCA security is proved in the strong CK model based on the well-studied DBDH assumption in the standard model.

6 Conclusion

In this paper, security notions of *CCA security* and *proof soundness* for single-hop unidirectional PRENO scheme are considered. The proposed security model is a CK model, which is showed stronger than previous KOSK models. Then, an efficient scheme which is proved secure in the given models is proposed. The scheme is based on the DBDH assumption, and is efficient compared with two well-known single-hop unidirectional PRE schemes. Also, the definitions for unidirectional PRENO scheme can be extended to the bidirectional setting.

Acknowledgments. We thank Hui Guo, Yanfei Guo, Xusheng Zhang and the anonymous reviewers of CCS 2012 for their helpful comments and suggestions on an earlier version of this paper.

References

1. Giuseppe Ateniese, Karyn Benson, and Susan Hohenberger. Key-private proxy re-encryption. In Marc Fischlin, editor, *Topics in Cryptology - CT-RSA 2009*, volume 5473 of *LNCS*, pages 279–294. Springer Berlin / Heidelberg, 2009.
2. Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In *Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS)*, pages 29–44, 2005.
3. Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9:1–30, February 2006.
4. Giuseppe Ateniese and Susan Hohenberger. Proxy re-signatures: new definitions, algorithms, and applications. In *Proceedings of the 12th ACM conference on Computer and communications security*, CCS '05, pages 310–319, New York, NY, USA, 2005. ACM.
5. Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT'98*, volume 1403 of *LNCS*, pages 127–144. Springer Berlin / Heidelberg, 1998.
6. Dan Boneh and Jonathan Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In Alfred Menezes, editor, *Topics in Cryptology - CT-RSA 2005*, volume 3376 of *LNCS*, pages 87–103. Springer Berlin / Heidelberg, 2005.
7. Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In *Proceedings of the 12th ACM conference on Computer and communications security*, CCS '05, pages 320–329, New York, NY, USA, 2005. ACM.
8. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer Berlin / Heidelberg, 2004.
9. Ran Canetti and Susan Hohenberger. Chosen-ciphertext secure proxy re-encryption. In *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, pages 185–194, New York, NY, USA, 2007. ACM.
10. Ran Canetti, Hugo Krawczyk, and Jesper Nielsen. Relaxing chosen-ciphertext security. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *LNCS*, pages 565–582. Springer Berlin / Heidelberg, 2003.
11. Sherman Chow, Jian Weng, Yanjiang Yang, and Robert Deng. Efficient unidirectional proxy re-encryption. In Daniel Bernstein and Tanja Lange, editors, *Progress in Cryptology - AFRICACRYPT 2010*, volume 6055 of *LNCS*, pages 316–332. Springer Berlin / Heidelberg, 2010.
12. Jean-Sbastien Coron. On the exact security of full domain hash. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000*, volume 1880 of *LNCS*, pages 229–235. Springer Berlin / Heidelberg, 2000.
13. Ivan Damgård, Dennis Hofheinz, Eike Kiltz, and Rune Thorbek. Public-key encryption with non-interactive opening. In Tal Malkin, editor, *Topics in Cryptology - CT-RSA 2008*, volume 4964 of *LNCS*, pages 239–255. Springer Berlin / Heidelberg, 2008.
14. Ivan Damgård and Rune Thorbek. Non-interactive proofs for integer multiplication. In Moni Naor, editor, *Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 412–429. Springer Berlin / Heidelberg, 2007.
15. Yevgeniy Dodis, Ostrovsky Rafail, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38:97–139, 2008.
16. Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In Serge Vaudenay, editor, *Public Key Cryptography - PKC 2005*, volume 3386 of *LNCS*, pages 416–431. Springer Berlin / Heidelberg, 2005.

17. M.V.M. Figueredo and J.S. Dias. Mobile telemedicine system for home care and patient monitoring. In *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, volume 2, pages 3387 – 3390, sept. 2004.
18. Yair Frankel and Moti Yung. Escrow encryption systems visited: Attacks, analysis and designs. In Don Coppersmith, editor, *Advances in Cryptology - CRYPT0'95*, volume 963 of *LNCS*, pages 222–235. Springer Berlin / Heidelberg, 1995.
19. David Galindo, Benoît Libert, Marc Fischlin, Georg Fuchsbauer, Anja Lehmann, Mark Manulis, and Dominique Schröder. Public-key encryption with non-interactive opening: New constructions and stronger definitions. In Daniel Bernstein and Tanja Lange, editors, *Progress in Cryptology - AFRICACRYPT 2010*, volume 6055 of *LNCS*, pages 333–350. Springer Berlin / Heidelberg, 2010.
20. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM.
21. Matthew Green and Giuseppe Ateniese. Identity-based proxy re-encryption. In Jonathan Katz and Moti Yung, editors, *Applied Cryptography and Network Security - ACNS 2007*, volume 4521 of *LNCS*, pages 288–306. Springer Berlin / Heidelberg, 2007.
22. Goichiro Hanaoka, Yutaka Kawai, Noboru Kunihiro, Takahiro Matsuda, Jian Weng, Rui Zhang, and Yunlei Zhao. Generic construction of chosen ciphertext secure proxy re-encryption. In Orr Dunkelman, editor, *Topics in Cryptology - CT-RSA 2012*, volume 7178 of *LNCS*, pages 349–364. Springer Berlin / Heidelberg, 2012.
23. Susan Hohenberger and Brent Waters. Realizing hash-and-sign signatures under standard assumptions. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 333–350. Springer Berlin / Heidelberg, 2009.
24. Anca-Andreea Ivan and Yevgeniy Dodis. Proxy cryptography revisited. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2003*, San Diego, California, USA. The Internet Society.
25. Markus Jakobsson. On quorum controlled asymmetric proxy re-encryption. In *Public Key Cryptography - PKC'99*, volume 1560 of *LNCS*, pages 632–632. Springer Berlin / Heidelberg, 1999.
26. Sriram Kailasam, Santosh Kumar, and Janakiram Dharanipragada. Arogyasree: an enhanced grid-based approach to mobile telemedicine. *Int. J. Telemedicine Appl.*, 2010:2:1–2:11, January 2010.
27. Apu Kapadia, Patrick P. Tsang, and Sean W. Smith. Attribute-based publishing with hidden credentials and hidden policies. In *Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS)*, pages 179–192, 2007.
28. Vishal Kher and Yongdae Kim. Securing distributed storage: challenges, techniques, and systems. In *Proceedings of the 2005 ACM workshop on Storage security and survivability*, StorageSS '05, pages 9–25, New York, NY, USA, 2005. ACM.
29. Himanshu Khurana. Scalable security and accounting services for content-based publish/subscribe systems. In *Proceedings of the 2005 ACM symposium on Applied computing*, SAC '05, pages 801–807, New York, NY, USA, 2005. ACM.
30. Himanshu Khurana, Adam Slagell, and Rafael Bonilla. SELS: a secure e-mail list service. In *Proceedings of the 2005 ACM symposium on Applied computing*, SAC '05, pages 306–313, New York, NY, USA, 2005. ACM.
31. Junzuo Lai, Robert Deng, Shengli Liu, and Weidong Kou. Efficient CCA-secure PKE from identity-based techniques. In Josef Pieprzyk, editor, *Topics in Cryptology - CT-RSA 2010*, volume 5985 of *LNCS*, pages 132–147. Springer Berlin / Heidelberg, 2010.
32. Arjen Lenstra, Peter Winkler, and Yacov Yacobi. A key escrow system with warrant bounds. In Don Coppersmith, editor, *Advances in Cryptology - CRYPT0'95*, volume 963 of *LNCS*, pages 197–207. Springer Berlin / Heidelberg, 1995.
33. Benoît Libert and Damien Vergnaud. Multi-use unidirectional proxy re-signatures. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 511–520, New York, NY, USA, 2008. ACM.
34. Benoît Libert and Damien Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. In Ronald Cramer, editor, *Public Key Cryptography - PKC 2008*, volume 4939 of *LNCS*, pages 360–379. Springer Berlin / Heidelberg, 2008.
35. Benoît Libert and Damien Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. *Information Theory, IEEE Transactions on*, 57(3):1786 –1802, march 2011.
36. Song Luo, Jianbin Hu, and Zhong Chen. New construction of identity-based proxy re-encryption. In *Proceedings of the tenth annual ACM workshop on Digital rights management*, DRM '10, pages 47–50, New York, NY, USA, 2010. ACM.
37. Mambo Masahiro and Okamoto Eiji. Proxy cryptosystems: Delegation of the power to decrypt ciphertexts. *IEICE transactions on fundamentals of electronics, Communications and computer sciences*, 80(1):54–63, 1997.

38. Silvio Micali. Fair public-key cryptosystems. In Ernest Brickell, editor, *Advances in Cryptology - CRYPTO'92*, volume 740 of *LNCS*, pages 113–138. Springer Berlin / Heidelberg, 1993.
39. Edna Milgo. A secure unidirectional proxy re-encryption using identity and secret key exchange. In *Proceedings of the 47th Annual Southeast Regional Conference*, ACM-SE 47, pages 24:1–24:5, New York, NY, USA, 2009. ACM.
40. Ritesh Mukherjee and J. William Atwood. Proxy encryptions for secure multicast key management. In *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks*, LCN '03, pages 377–384, Washington, DC, USA, 2003. IEEE Computer Society.
41. C.O. Rolim, F.L. Koch, M. Assuncao, and C.B. Westphall. Towards a grid of sensors for telemedicine. In *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, pages 485 –490, 2006.
42. Jun Shao and Zhenfu Cao. CCA-secure proxy re-encryption without pairings. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography - PKC 2009*, volume 5443 of *LNCS*, pages 357–376. Springer Berlin / Heidelberg, 2009.
43. Gelareh Taban, Alvaro A. Cárdenas, and Virgil D. Gligor. Towards a secure and interoperable drm architecture. In *Proceedings of the ACM workshop on Digital rights management*, DRM '06, pages 69–78, New York, NY, USA, 2006. ACM.
44. Jian Weng, MinRong Chen, YanJiang Yang, Robert Deng, KeFei Chen, and Feng Bao. CCA-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles. *SCIENCE CHINA Information Sciences*, 53:593–606, 2010.
45. Jiang Zhang, Xiang Xie, Rui Zhang, and Zhenfeng Zhang. A generic construction from selective-IBE to public-key encryption with non-interactive opening. Information Security and Cryptology - 7th International Conference, INSCRYPT'2011, Beijing, China, November 30 - Dec 3 2011.
46. Lidong Zhou, Michael A. Marsh, Fred B. Schneider, and Anna Redz. Distributed blinding for distributed elgamal re-encryption. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems, ICDCS 2005*, pages 815–824, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

A The CK Model is Stronger Than the KOSK Model

In this section, we show that the CK model is stronger than the KOSK model by considering the scheme in [34], which is secure in the KOSK model and is insecure in the CK model. For completeness, we first recall the scheme in [34].

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of prime order p, and let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. Let k be the security parameter and $p \geq 2^k$.

`Setup`(1^k): Take the security parameter 1^k as input, randomly chooses $g, u, v \leftarrow \mathbb{G}$ and a strongly unforgeable one-time signature scheme $\mathbf{Sig} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$. Return the public parameters $param = (g, u, v, \mathbf{Sig})$. The plaintext space $\mathcal{P} = \mathbb{G}_T$.

`KeyGen`$(param)$: Take the parameter $param$ as input, randomly choose $x_i \leftarrow \mathbb{Z}_p$, return the public key and secret key pair $(pk_i, sk_i) = (g^{x_i}, x_i)$.

`ReKeyGen`(sk_i, pk_j): Take a secret key sk_i and a public key pk_j as inputs, return the re-encryption key $rk_{i\to j} = pk_j^{1/x_i} = g^{x_j/x_i}$.

`Enc`$_2(pk_i, m)$: Take a public key pk_i and a message $m \in \mathcal{P}$ as inputs, randomly choose a one-time signature key pair $(ssk, svk) \leftarrow \mathcal{G}(1^k)$, and $r \leftarrow \mathbb{Z}_p^*$, compute

$$c_1 = svk, c_2 = pk_i^r, c_3 = e(g, g)^r \cdot m, c_4 = (u^{svk} \cdot v)^r, c_5 = \mathcal{S}(ssk, (c_3, c_4)).$$

Return a $2nd$ level ciphertext $C_i = (c_1, c_2, c_3, c_4, c_5)$.

`Enc`$_1(pk_j, m)$: Take a public key pk_j and a message $m \in \mathcal{P}$ as inputs, randomly choose a one-time signature key pair $(ssk, svk) \leftarrow \mathcal{G}(1^k)$, and $r, t \leftarrow \mathbb{Z}_p^*$, compute

$$c_1 = svk, c_2 = pk_j^t, c_3 = e(g, g)^r \cdot m, c_4 = (u^{svk} \cdot v)^r, c_5 = \mathcal{S}(ssk, (c_3, c_4)), c_6 = g^{1/t}, c_7 = pk_j^{rt}.$$

Return a 1st level ciphertext $C_i = (c_1, c_2, c_3, c_4, c_5, c_6, c_7)$.

$\texttt{ReEnc}(rk_{i\to j}, C_i)$: Take a re-encryption key $rk_{i\to j} = g^{x_j/x_i}$ and a 2nd level ciphertext $C_i = (c_1, c_2, c_3, c_4, c_5)$ as inputs, check whether $e(c_2, u^{c_1}v) = e(pk_i, c_4)$ and $\mathcal{V}(c_1, c_5, (c_3, c_4)) = 1$. If not, return $\perp$. Otherwise, randomly choose $t \leftarrow \mathbb{Z}_p$, and compute

$$c_2' = pk_i^t, c_6 = rk_{i\to j}^{1/t}, c_7 = c_2^t.$$

return the 1st level ciphertext $C_j = (c_1, c_2', c_3, c_4, c_5, c_6, c_7)$.

$\texttt{Dec}_2(sk_i, C_i)$: Take a secret key sk_i and a 2nd level ciphertext $C_i = (c_1, c_2, c_3, c_4, c_5)$ as inputs, If $e(c_2, u^{c_1}v) = e(pk_i, c_4)$ and $\mathcal{V}(c_1, c_5, (c_3, c_4)) = 1$, return $m = c_3/e(c_2, g^{1/x_i})$, else return $\perp$.

$\texttt{Dec}_1(sk_j, C_j)$: Take a secret key sk_j and a 1st level ciphertext $C_j = (c_1, c_2, c_3, c_4, c_5, c_6, c_7)$ as inputs, If $e(c_2, c_6) = e(pk_j, g)$, $e(c_7, u^{c_1}v) = e(c_2, c_4)$ and $\mathcal{V}(c_1, c_5, (c_3, c_4)) = 1$, return $m = c_3/e(c_6, c_7)^{1/x_j}$, else return $\perp$.

Now, we show that the above scheme is insecure in the CK model. As for the CCA security at the 1st level, the adversary can make any re-encryption key generation queries from the challenge public key to any other public keys. After seeing the challenge ciphertext, the adversary only needs to make one re-encryption key queries to decrypt the challenge ciphertext. Concretely, assuming the challenge ciphertext has a form of $C_j = (c_1, c_2, c_3 = e(g, g)^r \cdot m, c_4, c_5, c_6 = g^{1/t}, c_7 = pk_j^{rt})$ where pk_j is the challenge public key. Now the adversary can set $c_6 = g^{1/t}$ as the public key $pk_{j'}$ of another user j' (i.e., $pk_{j'} = g^{1/t}$), and ask a re-encryption key $rk_{j\to j'}$ from pk_j to $pk_{j'}$. Note that $rk_{j\to j'} = pk_{j'}^{1/x_j} = g^{1/(x_j t)}$, thus the adversary can decrypt the ciphertext by computing $e(c_7, rk_{j\to j'}) = e(g, g)^r$, thus the 1st level ciphertext can not be secure in the CK model.

As for the 2nd level ciphertext, the adversary only needs one more re-encryption queries to transform a 2nd level ciphertext under the challenge public key to any honest public key since he cannot directly ask a re-encryption key from the challenge public key to malicious public key. Specifically, given a 2nd level challenge ciphertext $C_i = (c_1, c_2 = pk_i^r, c_3 = e(g, g)^r \cdot m, c_4, c_5)$, the adversary first makes a re-encryption queries from C_i to C_j under public key pk_j of any honest user j (this query is always allowed). Then, we have $C_j = (c_1, c_2, c_3 = e(g, g)^r \cdot m, c_4, c_5, c_6 = g^{x_j/(x_i t)}, c_7 = c_2^t = pk_i^{rt})$ for some unknown t. Now, the adversary can attack the resulting ciphertext and set $c_6 = g^{x_j/(x_i t)}$ as the public key $pk_{j'}$ of another user j' (i.e., $pk_{j'} = g^{x_j/(x_i t)}$), and ask a re-encryption key $rk_{j\to j'}$ from pk_j to $pk_{j'}$ (this query is also allowed). Note that $rk_{j\to j'} = pk_{j'}^{1/x_j} = g^{1/(x_i t)}$, thus the adversary can decrypt the ciphertext by computing $e(c_7, rk_{j\to j'}) = e(g, g)^r$, thus the 2st level ciphertext can not be secure in the CK model. Finally, we have shown the CK model is strictly stronger than the KOSK model.

B Proof of Theorem 1

In this section, we give the proof of Theorem 1. Equivalently, we prove the following three lemmas.

Lemma 2. *Assume $\mathcal{H}_1$ is a family of collision resistant hash functions, our single-hop unidirectional PRENO scheme is proof sound.*

Proof. Note that given a δ level ciphertext $C = (s, c_0, c_1, c_2 = \tau_1 \| \tau_2, c_3)$ under public key pk, and a proof-message pair (π, m), the verification algorithm treats π as a "decryption key" and computes a temporary value K, and accepts (π, m) (i.e., $\texttt{Ver}(pk, C, \delta, m, \pi) = 1$) if and only if $H_1(K) = \tau_1$

and $m = \tau_2 \oplus H_2(K)$. Thus, for the same ciphertext $C = (s, c_0, c_1, c_2 = \tau_1 \| \tau_2, c_3)$, and any two proof-message pairs (π, m) and (π', m'). Let K be the temporary value for (π, m), and K' be the temporary value for (π', m'). If the verification algorithm accepts both (π, m) and (π', m'), we must have $H_1(K) = \tau_1 = H_1(K')$. By the collision resistance of H_1, we have $K = K'$. Finally, we have $m = \tau_2 \oplus H_2(K) = \tau_2 \oplus H_2(K') = m'$. Thus, for any ciphertext C, all the valid proofs (that the verification algorithm accepts) for C must be corresponding to the same message m, which completes our proof.

Lemma 3. *Assume DBDH problem is hard, $\mathcal{H}_0$ is a family of collision-resistant hash functions, $\mathcal{H}_1$ is a family of collision-resistant and one-way hash functions, and $\mathcal{H}_2$ is a family of universal hash functions, our single-hop unidirectional PRENO scheme is CCA secure at the 2nd level.*

Proof. We assume there is a polynomial-time algorithm $\mathcal{A}$ that makes at most q_{hkg} honest key generation queries, attacks our unidirectional PRENO scheme at the $2nd$ level with non-negligible advantage ϵ. We construct an algorithm $\mathcal{B}$ that solves the DBDH problem. Note that $\mathcal{B}$ is given (g, g^a, g^b, g^c, Z), and he wants to decide whether $Z = e(g, g)^{abc}$ or random. The description of $\mathcal{B}$ that simulates the **PRENO-CCA** game at the second level for $\mathcal{A}$ is given below:

Setup. Randomly choose $H_0 \leftarrow \mathcal{H}_0$, $H_1 \leftarrow \mathcal{H}_1$ and $H_2 \leftarrow \mathcal{H}_2$. Set $h_1 = g^a, h_2 = g^b, Y = e(h_1, h_2)$. Randomly choose $x_v \leftarrow \mathbb{Z}_p^*$, and $x_w, y_u, y_v, y_w \leftarrow \mathbb{Z}_p$. Then compute $u = h_1 g^{y_u}$,$v = h_1^{x_v} g^{y_v}$,$w = h_1^{x_w} g^{y_w}$. Finally, set the system parameter $param = (g, h_1, h_2, u, v, w, Y, H_0, H_1, H_2)$. $\mathcal{B}$ also chooses integer $k^* \leftarrow \{1, \ldots, q_{hkg}\}$ randomly, and hopes $\mathcal{A}$ will use the public key produced by the k^*-th honest key generation query as the challenge public key. It is easy to know the probability that $\mathcal{B}$ guesses right is at least $1/q_{hkg}$.

Phase 1. $\mathcal{B}$ answers $\mathcal{A}$'s queries as follows:

- $\mathcal{O}_{\text{hkg}}(i)$: Randomly choose $\alpha_i, \beta_i \leftarrow \mathbb{Z}_p^*$. If it is the k^*-th honest key generation query, let $i^* = i$, and compute $pk_{i^*,1} = g^{\alpha_{i^*}}, pk_{i^*,2} = g^{\beta_{i^*}}$ ($\mathcal{B}$ implicitly sets $sk_{i^*,1} = \alpha_{i^*}/a, sk_{i^*,2} = \beta_{i^*}$). Otherwise, compute $pk_{i,1} = h_1^{\alpha_i}, pk_{i,2} = h_2^{-1} g^{\beta_i}$ ($\mathcal{B}$ implicitly sets $sk_{i,1} = \alpha_i, sk_{i,2} = \beta_i - b$). Finally, return $pk_i = (pk_{i,1}, pk_{i,2})$.
- $\mathcal{O}_{\text{rek}}(pk_i, pk_j)$: $\mathcal{B}$ distinguishes the following cases:
  Case a1 $pk_i \neq pk_{i^*}$: Run `ReKeyGen` by using $sk_{i,1}$, and return whatever it outputs.
  Case a2 $pk_i = pk_{i^*}$ and pk_j is *good*: Compute $rk_{i \to j} = (h_2 \cdot pk_{j,2})^{1/sk_{i^*,1}} = (g^a)^{\beta_j/\alpha_{i^*}}$
  Case a3 $pk_i = pk_{i^*}$ and pk_j is *bad*: Return a random bit and `abort`.
- $\mathcal{O}_{\text{rec}}(pk_i, pk_j, C_i)$: Parse $C_i = (s, c_0, c_1, c_2, c_3)$, if $\texttt{CheckCCA}(pk_i, C_i, 2nd) \neq 1$, return $\perp$. Otherwise, $\mathcal{B}$ distinguishes the following cases:
  Case b1 $pk_i \neq pk_{i^*}$ or pk_j is *good*: Run `ReEnc` by using $rk_{i \to j}$, and return whatever it outputs.
  Case b2 $pk_i = pk_{i^*}$ and pk_j is *bad*: Compute $t = H_0(c_0, c_2)$, if $t + sx_v + x_w = 0$, output a random bit and `abort`. Else compute $h_1^r = (c_3/c_0^{ty_u + sy_v + y_w})^{1/(t + sx_v + x_w)}$, and $c_1' = e(h_1^r, h_2 \cdot pk_{j,2}) = Y^r \cdot e(h_1, pk_{j,2})^r$. Return $C_j = (s, c_0, c_1', c_2, c_3)$ to $\mathcal{A}$.
  Note that $c_0 = g^r$, $c_1 = pk_{i^*,1}^r$, $c_3 = (u^t v^s w)^r = (h_1^{(t + sx_v + x_w)} g^{ty_u + sy_v + y_w})^r$ for some unknown r, we have $h_1^r = (c_3/c_0^{ty_u + sy_v + y_w})^{1/(t + sx_v + x_w)}$.
- $\mathcal{O}_{\text{dec}}(pk_i, C_i, \delta_i)$: Parse $C_i = (s, c_0, c_1, c_2, c_3)$, if $\texttt{CheckCCA}(pk_i, C_i, \delta_i) \neq 1$, return $\perp$. Otherwise, $\mathcal{B}$ distinguishes the following cases:
  Case c1 $\delta_i = 2nd$ and $pk_i \neq pk_{i^*}$: Run `Dec` by using $sk_{i,1}$, and return whatever it outputs.
  Case c2 $\delta_i = 1st$ and $pk_i = pk_{i^*}$: Run `Dec` by using $sk_{i,2}$, and return whatever it outputs.

Case c3 Otherwise: Compute $t = H_0(c_0, c_2)$, if $t + sx_v + x_w = 0$, output a random bit and `abort`. Else, compute $h_1^r = (\frac{c_3}{c_0^{ty_u+sy_v+y_w}})^{1/(t+sx_v+x_w)}$, and $K = e(h_1^r, h_2)$ if $\delta_i = 2nd$, else $K = \frac{c_1}{e(h_1^r, pk_{j,2})}$. Parse $c_2 = \tau_1 || \tau_2$, if $\tau_1 = H_1(K)$, return $m = \tau_2 \oplus H_2(K)$, else return $\perp$.

– $\mathcal{O}_{\mathrm{pf}}(pk_i, C_i, \delta_i)$: Parse $C_i = (s, c_0, c_1, c_2, c_3)$, if `CheckCCA`$(pk_i, C_i, \delta_i) \neq 1$, return $\perp$. Else, $\mathcal{B}$ distinguishes the following cases:

Case d1 $\delta_i = 2nd$ and $pk_i \neq pk_{i^*}$: Run `Prove` by using $sk_{i,1}$, and return whatever it outputs.

Case d2 $\delta_i = 1st$ and $pk_i = pk_{i^*}$: Run `Prove` by using $sk_{i,2}$, and return whatever it outputs.

Case d3 Otherwise: Compute $t = H_0(c_0, c_2)$, if $t + sx_v + x_w = 0$, output a random bit and `abort`. Else, compute $h_1^r = (\frac{c_3}{c_0^{ty_u+sy_v+y_w}})^{1/(t+sx_v+x_w)}$, and $K = e(h_1^r, h_2)$ if $\delta_i = 2nd$, else $K = \frac{c_1}{e(h_1^r, pk_{j,2})}$. Parse $c_2 = \tau_1 || \tau_2$, if $\tau_1 \neq H_1(K)$, return $\perp$. Else, randomly choose $\gamma \leftarrow \mathbb{Z}_p$, if $\delta_i = 2nd$, compute

$$d_1 = (g^b)^{-1/(t+sx_v+x_w)} {pk_{i,1}}^{\gamma}, d_2 = (g^b)^{-\frac{ty_u+sy_v+y_w}{\alpha_i(t+sx_v+x_w)}} (u^t v^s w)^{\gamma},$$

else, compute

$$d_1 = (g^b)^{\frac{1}{(t+sx_v+x_w)}} \cdot g^{\gamma}, d_2 = h_1^{\beta_i} \cdot (g^b)^{\frac{ty_u+sy_v+y_w}{t+sx_v+x_w}} \cdot (u^t v^s w)^{\gamma},$$

Finally, $\mathcal{B}$ returns $\pi_i = (d_1, d_2)$ to $\mathcal{A}$.

Note that if $\delta_i = 2nd$, we have $pk_i = pk_{i^*}$, if let $\hat{\gamma} = \gamma - \frac{b}{\alpha_{i^*}(t+sx_v+x_w)}$, we have $d_1 = pk_{i^*,1}^{\hat{\gamma}}, d_2 = h_2^{1/sk_{i^*,1}} (u^t v^s w)^{\hat{\gamma}}$. Else, if let $\hat{\gamma} = \gamma + \frac{b}{(t+sx_v+x_w)}$, we have $d_1 = g^{\hat{\gamma}}, d_2 = h_1^{sk_{i,2}} (u^t v^s w)^{\hat{\gamma}}$. Thus, the proof is perfectly simulated in both cases.

Challenge. Once $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal-length messages $m_0, m_1 \in \mathcal{P}$ and a target public key pk^*. If $pk^* \neq pk_{i^*}$, $\mathcal{B}$ outputs a random bit and `aborts`. Otherwise, $\mathcal{B}$ picks a random coin $b \in \{0, 1\}$, and computes $c_0^* = g^c, c_1^* = (g^c)^{\alpha_{i^*}}$, and $c_2^* = H_1(Z) || H_2(Z) \oplus m_b$ (implicitly set $r^* = c$). Then compute $t^* = H_0(c_0^*, c_2^*)$ and $s^* = -\frac{t^*+x_w}{x_v}$ (thus $t^* + s^* x_v + x_w = 0$). Finally, $\mathcal{B}$ computes $c_3^* = (g^c)^{t^* y_u + s^* y_v + y_w}$, returns $C^* = (s^*, c_0^*, c_1^*, c_2^*, c_3^*)$ to $\mathcal{A}$.

Phase 2. $\mathcal{B}$ answers $\mathcal{A}$'s queries as follows:

– $\mathcal{O}_{\mathrm{hkg}}(i)$: $\mathcal{B}$ responds as in Phase 1.

– $\mathcal{O}_{\mathrm{rek}}(pk_i, pk_j)$: $\mathcal{B}$ responds as in Phase 1.

– $\mathcal{O}_{\mathrm{rec}}(pk_i, pk_j, C_i)$: Parse $C_i = (s, c_0, c_1, c_2, c_3)$, if `CheckCCA`$(pk_i, C_i, 2nd) \neq 1$, return $\perp$. Otherwise, $\mathcal{B}$ distinguishes the following cases:

Case e1 $pk_i \neq pk_{i^*}$ or pk_j is *good*: Run `ReEnc` by using $rk_{i \to j}$, and return whatever it outputs.

Case e2 $pk_i = pk_{i^*}$ and pk_j is *bad* and $C_i = C^*$: Return $\perp$.

Case e3 $pk_i = pk_{i^*}$ and pk_j is *bad* and $C_i \neq C^*$: Compute $t = H_0(c_0, c_2)$, if $t + sx_v + x_w = 0$, output a random bit and `abort`. Else compute $h_1^r = (c_3 / c_0^{ty_u+sy_v+y_w})^{1/(t+sx_v+x_w)}$, and $c_1' = e(h_1^r, h_2 \cdot pk_{j,2}) = Y^r \cdot e(h_1, pk_{j,2})^r$. Return $C_j = (s, c_0, c_1', c_2, c_3)$ to $\mathcal{A}$.

– $\mathcal{O}_{\mathrm{dec}}(pk_i, C_i, \delta_i)$: Parse $C_i = (s, c_0, c_1, c_2, c_3)$, if `CheckCCA`$(pk_i, C_i, \delta_i) \neq 1$, return $\perp$. Otherwise, $\mathcal{B}$ distinguishes the following cases:

Case f1 $\delta_i = 2nd$ and $pk_i \neq pk_{i^*}$: Run `Dec` by using $sk_{i,1}$, and return whatever it outputs.

Case f2 $\delta_i = 2nd$ and $(pk_i, C_i) = (pk_{i^*}, C^*)$: Return $\perp$.

Case f3 $\delta_i = 1st$ and $pk_i = pk_{i^*}$: Run `Dec` by using $sk_{i,2}$, and return whatever it outputs.

Case f4 $\delta_i = 1st$ and $pk_i \neq pk_{i^*}$ and $(s, c_0, c_2, c_3) = (s^*, c_0^*, c_2^*, c_3^*)$: Return $\perp$.

Case f5 Otherwise: Compute $t = H_0(c_0, c_2)$, if $t + sx_v + x_w = 0$ holds, output a random bit and `abort`. Else compute $h_1^r = (\frac{c_3}{c_0^{ty_u+sy_v+y_w}})^{1/(t+sx_v+x_w)}$, and $K = e(h_1^r, h_2)$ if $\delta_i = 2nd$, else $K = \frac{c_1}{e(h_1^r, pk_{i,2})}$. Parse $c_2 = \tau_1 || \tau_2$, if $\tau_1 = H_1(K)$, return $m = \tau_2 \oplus H_2(K)$, else return $\perp$.

- $\mathcal{O}_{\text{pf}}(pk_i, C_i, \delta_i)$: Parse $C_i = (s, c_0, c_1, c_2, c_3)$, if `CheckCCA`$(pk_i, C_i, \delta_i) \neq 1$, return $\perp$. Otherwise, $\mathcal{B}$ distinguishes the following cases:

Case g1 $\delta_i = 2nd$ and $pk_i \neq pk_{i^*}$: Run `Prove`$_2$ by using $sk_{i,1}$, and return whatever it outputs.

Case g2 $\delta_i = 2nd$ and $(pk_i, C_i) = (pk_{i^*}, C^*)$: Return $\perp$.

Case g3 $\delta_i = 1st$ and $pk_i = pk_{i^*}$: Run `Prove`$_1$ by using $sk_{j,2}$, and return whatever it outputs.

Case g4 $\delta_i = 1st$ and $pk_i \neq pk_{i^*}$ and $(s, c_0, c_2, c_3) = (s^*, c_0^*, c_2^*, c_3^*)$: Return $\perp$.

Case g5 Otherwise: Compute $t = H_0(c_0, c_2)$, if $t + sx_v + x_w = 0$, output a random bit and `abort`. Else, compute $h_1^r = (\frac{c_3}{c_0^{ty_u+sy_v+y_w}})^{1/(t+sx_v+x_w)}$, and $K = e(h_1^r, h_2)$ if $\delta_i = 2nd$, else $K = \frac{c_1}{e(h_1^r, pk_{i,2})}$. Parse $c_2 = \tau_1 || \tau_2$, if $\tau_1 \neq H_1(K)$, return $\perp$. Else, randomly choose $\gamma \leftarrow \mathbb{Z}_p$, if $\delta_i = 2nd$, compute

$$d_1 = (g^b)^{-1/(t+sx_v+x_w)} {pk_{i,1}}^{\gamma}, d_2 = (g^b)^{-\frac{ty_u+sy_v+y_w}{\alpha_i(t+sx_v+x_w)}} (u^t v^s w)^{\gamma},$$

else, compute

$$d_1 = (g^b)^{\frac{1}{(t+sx_v+x_w)}} \cdot g^{\gamma}, d_2 = h_1^{\beta_i} \cdot (g^b)^{\frac{ty_u+sy_v+y_w}{t+sx_v+x_w}} \cdot (u^t v^s w)^{\gamma},$$

Finally, $\mathcal{B}$ returns $\pi_i = (d_1, d_2)$ to $\mathcal{A}$.

Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. If $b = b'$, $\mathcal{B}$ outputs 1, else outputs 0.

Analysis. To give a final analysis, we first define the following events:

- $\mathbf{E}_1$: $\mathcal{B}$ aborts in **Case a3** or in the **Challenge** phase.
- $\mathbf{E}_2$: $\mathcal{B}$ aborts in any of the **Cases b2, c3, d3, e3, f5, g5** due to $t + sx_v + x_w = 0$.
- $\mathbf{E}_3$: $\mathcal{A}$ submits a ciphertext C_j under pk_j in either **Cases f4, g4** such that C_j is a valid re-encryption ciphertext of C^*, and $\mathcal{A}$ never makes a query $\mathcal{O}_{\text{rek}}(pk^*, pk_j)$ and C_j is not an output of a query $\mathcal{O}_{\text{rec}}(pk^*, pk_j, C^*)$.

Note that if Events $\mathbf{E_1}$ and $\mathbf{E_2}$ never happen, then $\mathcal{B}$ will not `abort` in the simulation. And if $\mathbf{E_3}$ does not happen, the simulation for **Cases f4, g4** is perfect according to our security model.

Thus, if all the events above do not happen throughout the simulation, then $\mathcal{B}$ almost perfectly simulates an attack environment for $\mathcal{A}$. In fact, if $Z = e(g, g)^{abc}$, we have C^* is a real ciphertext for m_b with randomness c. Thus, $\mathcal{A}$ will guess $b' = b$ with the same probability as in the real game. If Z is random, since H_2 is a universal hash function, we have $H_2(Z)$ is statistically close to uniform even given $H_1(Z)$, thus c_2^* statistically hides m_b, and the probability that $\mathcal{A}$ guesses $b' = b$ is $1/2 + negl(k)$.

The remaining work is to estimate the probability for each events. Actually, since k^* is randomly chosen, thus $1 - \Pr[\mathbf{E}_1] = \Pr[pk^* = pk_{i^*}] \geq 1/q_{hkg}$. Besides, since $\mathcal{A}$ has no information about x_v, x_w in the first phase, thus the probability $\Pr[t + sx_v + x_w = 0] \leq 1/p$. Though $\mathcal{A}$ could obtain the information $t^* + s^* x_v + x_w = 0$ from the challenge ciphertext C^*, there are still p possible pair

(x_v, x_w) satisfying the equation. Thus, $\Pr[t + sx_v + x_w = 0 | (s,t) \neq (s^*, t^*)]$ is still at most $1/p$. Note that in **Cases e3, f5, g5**, we almost always have $(s,t) \neq (s^*, t^*)$, or else $\mathcal{B}$ could find collisions for H_0. Hence, we have $\Pr[\mathbf{E}_2]$ is negligible.

Intuitively, $\mathbf{E}_3$ says that $\mathcal{A}$ can transform a valid $2nd$ level ciphertext C_i under pk_i to a valid $1st$ level ciphertext C_j under pk_j without knowing the re-encryption key $rk_{i\to j}$, the information of sk_i, sk_j, and without accessing the corresponding re-encryption proxy. Thus, one can informally consider this as the security of the re-encryption key. Assuming the challenge ciphertext $C^* = (s^*, c_0^* = g^{r^*}, c_1^* = pk_{i^*,1}^{r^*}, c_2^* = H_1(Y^{r^*})||H_2(Y^{r^*}) \oplus m, c_3^* = (u^{t^*} v^{s^*} w)^{r^*})$ for some $r^* \in \mathbb{Z}_p$ and $t^* = H_0(c_0^*, c_2^*)$. If $\mathcal{A}$ makes $\mathbf{E}_3$ happens, he must output a ciphertext $C' = (s^*, c_0^*, c_1, c_2^*, c_3^*)$ under public key $pk_{i'} \neq pk_{i^*}$. Since both the decryption and proof algorithm would check whether $H_1(c_1/e(h_1^{sk_{i',2}}, c_0^*)) = H_1(Y^{r^*})$, and by the collision resistance of H_1, we have C' is valid if and only if $c_1 = Y^{r^*} \cdot e(h_1, pk_{i',2})^{r^*}$. Note that $\mathcal{A}$ never makes a re-encryption key query from pk_{i^*} to $pk_{i'}$, and C' is not output by the re-encryption query. Now, we construct an algorithm $\mathcal{C}$ that makes good use of the above two conditions to solve computational BDH problem by interacting with $\mathcal{A}$ that makes $\mathbf{E}_3$ happen with non-negligible probability.

Technically, given a BDH instance (g^a, g^b, g^c), if we embed the BDH problem by setting $h_1 = g^a, pk_{i',2} = g^b$ and $r^* = c$, then we can solve BDH problem by computing $e(h_1, pk_{i',2})^{r^*} = c_1/Y^{r^*}$. However, under our DBDH assumption, it is not an easy task to check whether c_1 has the right form. Fortunately, if $\mathcal{A}$ only makes q_{hkg} times honest key generation queries, q_2 times decryption and proof queries. Then, $\mathcal{C}$ can randomly choose another two integers $k_1^* \leftarrow \{1, \ldots, q_{hkg}\}, k_2^* \leftarrow \{1, \ldots, q_2\}$, and hopes that $\mathbf{E}_3$ will happen for the first time at the k_2^*-th decryption and proof query with the public key $pk_{i'}$ output by the k_1^*-th honest key generation query. It is easy to check that $\mathcal{C}$ will guess (k^*, k_1^*, k_2^*) right with non-negligible probability (i.e., at least $1/q_{hkg}^2 q_2$). To simulate the attack environment for $\mathcal{A}$, $\mathcal{C}$ sets $param = (g, h_1 = g^a, h_2 = g^{x_h}, u = h_1 g^{y_u}, v = h_1^{x_v} g^{y_v}, w = h_1^{x_w} g^{y_w}, Y = e(h_1, h_2)$, where $x_h, x_v, x_w, y_u, y_v, y_w \leftarrow \mathbb{Z}_p$. $\mathcal{C}$ also honestly generates public keys $pk_i = (pk_{i,1} = h_1^{\alpha_i}, pk_{i,2} = g^{\beta_i})$ for $i \neq \{i^*, i'\}$, and embeds BDH problem by setting $pk_{i^*} = (pk_{i^*,1} = g^{\alpha_{i^*}}, pk_{i^*,2} = g^{\beta_{i^*}})$, $pk_{i'} = (pk_{i',1} = h_1^{\alpha_{i'}}, pk_{i',2} = (g^b)^{\beta_{i'}})$, and $C^* = (s^*, c_0^* = g^c, c_1^* = (g^c)^{\alpha_{i^*}}, c_2^* = H_1(Z)||H_2(Z) \oplus m_b, c_3^* = (g^c)^{t^* y_u + s^* y_v + y_w})$ where $Z = Y^c = e(g^a, g^c)^{x_h}$, $t^* = H_0(c_0^*, c_2^*)$ and $s^* = -\frac{t^* + x_w}{x_v}$. Obviously, C^* is valid ciphertext under pk_{i^*} with randomness c. Moreover, with the knowledge of $(x_h, x_v, x_w, y_u, y_v, y_w)$ and $\{\alpha_i, \beta_i\}$, $\mathcal{C}$ almost can perfectly answer all queries if he guesses (k^*, k_1^*, k_2^*) right. Note that if $\mathbf{E}_3$ happens for the first time at the k_{2^*}-th decryption and proof query with public key $pk_{i'}$ output by the k_{1^*}-th honest key generation query and ciphertext $C' = (s^*, c_0^*, c_1, c_2^*, c_3^*)$, we have $c_1 = Y^c \cdot e(h_1, pk_{i',2})^c = e(g^a, g^c)^{x_h} \cdot e(g^a, g^b)^{c\beta_{i'}}$, thus $\mathcal{C}$ can solve BDH problem by computing $e(g,g)^{abc} = (c_1/e(g^a, g^c)^{x_h})^{1/\beta_{i'}}$. Thus, under our DBDH assumption, $\Pr[\mathbf{E}_3]$ is negligible. We defer a detailed proof to a full version.

In all, the probability that all the events do not happen is at least $1/q_{hkg} - negl(k)$, thus the advantage that $\mathcal{B}$ solves DBDH problem is at least $1/q_{hkg} \cdot \epsilon - negl(k)$, which completes the proof.

Lemma 4. *Assume DBDH problem is hard, $\mathcal{H}_0$ is a family of collision-resistant hash functions, $\mathcal{H}_1$ is a family of collision-resistant and one-way hash functions, and $\mathcal{H}_2$ is a family of universal hash functions, our single-hop unidirectional PRENO scheme is CCA secure at the* 1*st level.*

Note that in the proof of Lemma 3, $\mathcal{B}$ can always generate re-encryption keys from honest generated public keys except the challenge one to malicious public keys, and that the $1st$ level ciphertexts output by the re-encryption algorithm and the encryption algorithm have the same distribution. Thus, to prove this Lemma, $\mathcal{B}$ can keep an extra $2nd$ level "challenge" public key in the mind (not provide to $\mathcal{A}$), and generates the challenge public key as for the honest ones in the proof of Lemma 3, and normally generates all other public keys (i.e., by first choosing the secret keys). Thus, the proof of Lemma 3 can be adapted to this lemma. Actually, one can construct an algorithm $\mathcal{B}$ that sets the system parameters as in the proof of Lemma 3, Then, $\mathcal{B}$ honestly generates all the public-secret key pairs except the challenge one $pk^* = (pk_{i^*,1} = h_1^{\alpha_{i^*}}, pk_{i^*,2} = h_2^{-1} g^{\beta_{i^*}})$. Finally, $\mathcal{B}$ sets the challenge ciphertext $C^* = (s^*, c_0^* = g^c, c_1^* = e(h_1, g^c)^{\beta_{i^*}},\ c_2^* = H_1(Z)||H_2(Z) \oplus m_b, c_3^* = (g^c)^{t^* y_u + s^* y_v + y_w})$ where $t^* = H_0(c_0^*, c_2^*)$ and $s^* = -\frac{t^* + x_w}{x_v}$. Thus, all $\mathcal{A}$'s queries can be perfectly simulated by $\mathcal{B}$ as in the proof of Lemma 3, we omit the details.