D. Nuñez, I. Agudo, and J. Lopez, "A Parametric Family of Attack Models for Proxy Re-Encryption", *28th IEEE Computer Security Foundations Symposium*, CSF'15, pp. 290-301, 2015.
http://doi.org/10.1109/CSF.2015.27
NICS Lab. Publications: https://www.nics.uma.es/publications

A Parametric Family of Attack Models for Proxy Re-Encryption*

David Nuñez, Isaac Agudo, and Javier Lopez

Network, Information and Computer Security Laboratory (NICS Lab)
Universidad de Málaga, Spain
{dnunez, isaac, jlm}@lcc.uma.es

Abstract

Proxy Re-Encryption (PRE) is a type of Public-Key Encryption (PKE) that provides an additional re-encryption functionality. Although PRE is inherently more complex than PKE, attack models for PRE have not been developed further than those inherited from PKE. In this paper we address this gap and define a parametric family of attack models for PRE, based on the availability of both the decryption and re-encryption oracles during the security game. This family enables the definition of fine-grained security notions for PRE, ranging from "plain" IND-CPA to "full" IND-CCA. We analyze some relations among these notions of security, and in particular, the separations, which further support the importance of the re-encryption oracle. The identified separations stem from the study of a new property of PRE, called *privacy of re-encryption keys*, which captures the requirement that re-encryption keys should not be leaked through the re-encryption function. Finally, we show that the scheme by Kirshanova (PKC 2014), which does not satisfy this property, cannot achieve a meaningful security notion for PRE since it is vulnerable to chosen-ciphertext attacks using the re-encryption oracle. This attack emphasizes the fact that PRE schemes that leak re-encryption keys cannot achieve strong security notions.

1 Introduction

Proxy Re-Encryption (PRE) is a type of Public-Key Encryption (PKE) where, in addition to the usual encryption and decryption capabilities, a "re-encryption" functionality allows a proxy to transform ciphertexts under the public key of Alice into ciphertexts decryptable by Bob. In order to do so, the proxy must be in possession of a re-encryption key that makes this process possible. In addition, the proxy cannot learn any information about the encrypted messages, under any of the keys. The notion of proxy re-encryption was introduced in 1998 by Blaze, Bleumer and Strauss [1], and since then multiple schemes have been proposed, based on different constructions and hardness assumptions.

Since PRE schemes are also public-key encryption schemes, it is natural to "reuse" the security notions of PKE. Thus, PRE schemes in the literature aim to achieve IND-CPA [2], IND-CCA1 [3] and IND-CCA2 [4] security. However, the desired security notions are often

*This is the revised version of the paper with the same title that appears in Proceedings of the 28th IEEE Computer Security Foundations Symposium (CSF 2015).

defined in an ad-hoc manner for each scheme, with subtle variations and restrictions, caused by the lack of established definitions of the security notions of PRE, and ultimately, of the attack models, which in the case of PRE are potentially richer than in PKE because of the possibility of re-encryption. While in the PKE setting, the different attack models are determined by the availability of the decryption oracle (i.e., not available in CPA, only before the challenge in CCA1, or throughout the game in CCA2), in the PRE setting it is reasonable to base the attack models on the availability of both the decryption and the re-encryption oracles.

In the context of PKE, there is a wide consensus on the consideration of indistinguishability against adaptive ciphertext attacks (IND-CCA2) as the right notion of security, although weaker notions such as IND-CCA1 may be sufficient in other contexts, such as homomorphic encryption. See [5] for a detailed exposition on the importance of security against chosen ciphertext attacks. The arguments for CCA2-security also hold in the case of proxy re-encryption. Canetti and Hohenberger motivate the necessity of CCA2-security with an example of a decryption oracle in a scenario of encrypted email forwarding, where email user Alice can set the mail server to forward all her encrypted emails to a designated recipient Bob, without giving her secret keys to either the mail server or Bob [4]. In this example, an attacker might gain access to a decryption oracle by producing ciphertexts, emailing them to Alice and then hoping that she responds with "*Did you send the following attachment to me?*", together with the decrypted ciphertext. If instead of this, it is Bob who responds to the attacker attaching the re-encrypted ciphertext, then the attacker would gain access to a re-encryption oracle. The motivation for considering the re-encryption oracle also comes from the study of applications of proxy re-encryption from the literature, where the re-encryption function is deployed as a service, so participants of the system can re-encrypt ciphertexts from one user to another. This service effectively simulates a re-encryption oracle. In fact, expecting the availability of this oracle is a weaker assumption than presuming the availability of a decryption oracle, since the latter requires knowledge of secret keys, and therefore, it probably would be better protected and less likely to be available.

While in the PKE context a *chosen-ciphertext attack* (CCA) means that the adversary is free to choose ciphertexts for querying a decryption oracle, in the PRE context a CCA adversary should also be able to freely re-encrypt ciphertexts. That is, the re-encryption functionality should be captured by any meaningful CCA model for PRE. In fact, the re-encryption oracle alone is able, in some cases, to simulate a decryption oracle, as shown in Section 4.3. This further supports our claims about its relevance and raises questions about what are proper security notions for PRE.

Contributions

In this paper we investigate these issues and present the following results:

- We propose a parametric family of attack models for proxy re-encryption, in terms of the oracles available to the adversary during the security game. This set in turn provides fine-grained security notions for PRE.

- We study some of the relations that arise between these security notions, specifically, for the indistinguishabillity of encryptions goal. The separations we present further support the importance of the re-encryption oracle, since we show that when it leaks re-encryption keys, the scheme cannot achieve a proper security notion for PRE.

- In light of this formalization, we show how a recent "CCA1-secure" proxy re-encryption scheme from PKC 2014 [3] does not actually achieve a meaningful chosen-ciphertext security notion for PRE, since it breaks when one considers an oracle for re-encryption. The methodology described for this attack, based on the leakage of re-encryption keys, can be used for analyzing other schemes, and supports the principle that PRE schemes should not leak re-encryption keys.

Organization

The rest of this paper is organized as follows: In Section 2 we briefly describe the basic concepts and definitions of security for public-key encryption, which serve as a basis for proxy re-encryption. In Section 3 we formalize PRE syntax and introduce a parametric family of attack models, based on the availability of the decryption and re-encryption oracles, which induces a collection of security notions for PRE. In Section 4 we study the relations among these security notions, and provide proofs for some separations that arise when the re-encryption oracle leaks re-encryption keys. In Section 5 we describe a chosen-ciphertext attack against a recent "CCA1-secure" scheme. Finally, Section 6 concludes the paper.

2 Definitions of Security for Public-Key Encryption

In this section we first provide the basic syntax for public-key encryption, and give an overview of their principal definitions of security, which will serve later as the basis for the definitions of security for proxy re-encryption.

2.1 Syntax of PKE schemes

Definition 2.1 (PKE scheme)**.** *A public-key encryption scheme is a tuple of algorithms (*KeyGen*,* Enc*,* Dec*):*

- $\mathsf{KeyGen}(n) \to (pk, sk)$*. On input security parameter* n*, the key generation algorithm* KeyGen *outputs a pair of public and secret keys* (pk, sk)*.*
- $\mathsf{Enc}(pk, m) \to c$*. On input the public key* pk *and a message* $m \in \mathcal{M}$*, the encryption algorithm* Enc *outputs a ciphertext* $c \in \mathcal{C}$*.*
- $\mathsf{Dec}(sk, c) \to m$*. On input the secret key* sk *and a ciphertext* $c \in \mathcal{C}$*, the decryption algorithm* Dec *outputs a message* $m \in \mathcal{M}$ *or the symbol* $\perp$ *indicating* c *is invalid.*

The plaintext and ciphertext spaces are denoted by $\mathcal{M}$ and $\mathcal{C}$, respectively.

2.2 Security notions of PKE schemes

In the context of PKE, security notions are usually defined as the combination of a security goal and an attack model [6]. In this paper we focus on the *indistinguishability of encryptions* (IND) goal, which formalizes the inability of an adversary to distinguish which message a given ciphertext encrypts. A weaker goal is *one-wayness* (OW), which represents the inability of an adversary to extract the underlying plaintext from a given ciphertext. The strongest goal is *non-malleability* (NM), where an adversary should not be able to produce ciphertexts such that,

for a given ciphertext, the plaintexts are meaningfully related. With regard to attack models, three options are usually considered: (i) *chosen-plaintext attack* (CPA), (ii) *non-adaptive chosen-ciphertext attack* (CCA1), and (iii) *adaptive chosen-ciphertext attack* (CCA2). In a CPA model, the only capability of the adversary is to encrypt plaintexts of her choice (although this capability is inherent in a public-key cryptosystem). Under CCA1, the adversary is also given a decryption capability (i.e., a decryption oracle) but only for its use before receiving the challenge ciphertext. Finally, in the CCA2 model, the adversary may use the decryption oracle in any moment, with the only restriction of not asking for the decryption of the challenge ciphertext. Therefore, a security notion can be seen as a tuple *goal-atk*, where *goal* $\in$ {OW, IND, NM}, and *atk* $\in$ {CPA, CCA1, CCA2}.

It can be seen that these attack models are differentiated by the changes on the decryption capabilities of the adversary, which can be modeled through the availability of a *decryption oracle*. Informally, a decryption oracle is a function $\mathcal{O}_{dec}(\cdot)$ that the adversary can query on any ciphertext c (except the challenge ciphertext c^*) and that outputs the decryption of c with the target secret key. No additional oracles are necessary for describing the above notions of security for PKE. As mentioned above, in this paper we focus on the indistinguishability goal, so we are concerned with three possible security notions for PKE. The following definitions, adapted from [6], comprise these security notions in a formal manner.

Definition 2.2. *Let* Π*=(KeyGen, Enc, Dec) be a public-key encryption scheme,* $A = (A_1, A_2)$ *a polynomial-time adversary, and* Ω_1 *and* Ω_2 *the set of available oracles for* A_1 *and* A_2*, respectively. For* $atk \in \{$*CPA, CCA1, CCA2*$\}$*,* $n \in \mathbb{N}$*, and* $\delta \in \{0, 1\}$*, the indistinguishability of encryptions game is defined by the experiment*

$$\begin{array}{l}\text{Experiment } \mathbf{Exp}^{\textit{IND-atk}}_{\Pi,A,\delta}(n)\\ \quad (pk^*, sk^*) \xleftarrow{R} \mathsf{KeyGen}(n); \qquad (m_0, m_1, s) \leftarrow A_1(pk^*);\\ \quad c^* \leftarrow \mathsf{Enc}(pk^*, m_\delta); \qquad d \leftarrow A_2(m_0, m_1, s, c^*);\\ \quad \text{return } d\end{array}$$

where

$$\begin{array}{lll}\text{If } atk = \mathsf{CPA} & \text{then } \Omega_1 = \varnothing & \text{and } \Omega_2 = \varnothing\\ \text{If } atk = \mathsf{CCA1} & \text{then } \Omega_1 = \{\mathcal{O}_{dec}\} & \text{and } \Omega_2 = \varnothing\\ \text{If } atk = \mathsf{CCA2} & \text{then } \Omega_1 = \{\mathcal{O}_{dec}\} & \text{and } \Omega_2 = \{\mathcal{O}_{dec}\}\end{array}$$

Definition 2.3. *Let* Π *be a public-key encryption scheme and* A *a polynomial-time adversary. For* $atk \in \{$*CPA, CCA1, CCA2*$\}$ *and* $n \in \mathbb{N}$*, the advantage of* A *is given by*

$$\mathbf{Adv}^{\textit{IND-atk}}_{\Pi,A}(n) = |\Pr[\mathbf{Exp}^{\textit{IND-atk}}_{\Pi,A,1}(n) = 1] - \Pr[\mathbf{Exp}^{\textit{IND-atk}}_{\Pi,A,0}(n) = 1]|$$

We say that the encryption scheme Π *is IND-atk secure if the advantage* $\mathbf{Adv}^{\textit{IND-atk}}_{\Pi,A}(n)$ *is negligible.*

3 Definitions of Security for Proxy Re-Encryption

The definitions of security in the context of PRE extend those of PKE due to the possibility of re-encrypting ciphertexts. That is, in addition to the KeyGen, Enc and Dec functions, a

PRE scheme defines two more, ReKeyGen and ReEnc, which are associated with this additional capability. This incorporation is what led us to conceive a more comprehensive characterization of the attack models and security notions for PRE. Before proceeding, it is necessary to formalize the syntax for PRE schemes.

3.1 Syntax of PRE schemes

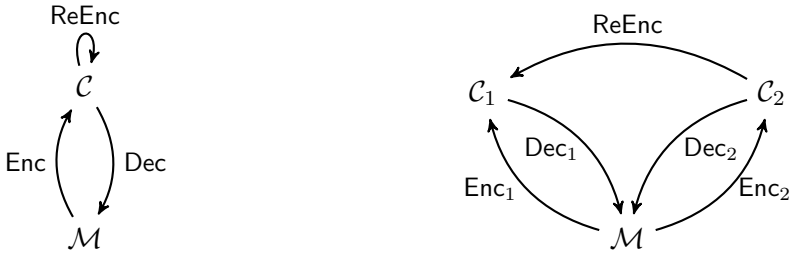
Based on the definitions by Canetti and Hohenberger [4] and Ateniese et al [2], we define the syntax of a proxy re-encryption scheme as follows.

Definition 3.1 (PRE scheme)**.** *A proxy re-encryption scheme is a tuple of algorithms (KeyGen, ReKeyGen, Enc, ReEnc, Dec):*

- $\mathsf{KeyGen}(n) \rightarrow (pk_i, sk_i)$. *On input security parameter n, the key generation algorithm* KeyGen *outputs a pair of public and secret keys (pk_i, sk_i) for user i.*
- $\mathsf{ReKeyGen}(pk_i, sk_i, pk_j, sk_j) \rightarrow rk_{i \rightarrow j}$. *On input the pair of public and secret keys (pk_i, sk_i) for user i and the pair of public and secret keys (pk_j, sk_j) for user j, the re-encryption key generation algorithm* ReKeyGen *outputs a re-encryption key $rk_{i \rightarrow j}$.*
- $\mathsf{Enc}(pk_i, m) \rightarrow c_i$. *On input the public key pk_i and a message $m \in \mathcal{M}$, the encryption algorithm* Enc *outputs a ciphertext $c_i \in \mathcal{C}$.*
- $\mathsf{ReEnc}(rk_{i \rightarrow j}, c_i) \rightarrow c_j$. *On input a re-encryption key $rk_{i \rightarrow j}$ and a ciphertext $c_i \in \mathcal{C}$, the re-encryption algorithm* ReEnc *outputs a second ciphertext $c_j \in \mathcal{C}$ or the symbol $\perp$ indicating c_i is invalid.*
- $\mathsf{Dec}(sk_i, c_i) \rightarrow m$. *On input the secret key sk_i and a ciphertext $c_i \in \mathcal{C}$, the decryption algorithm* Dec *outputs a message $m \in \mathcal{M}$ or the symbol $\perp$ indicating c_i is invalid.*

This definition is oblivious to the specific properties of PRE schemes, such as directionality, number of hops, or interactivity. A PRE scheme is *unidirectional* if the re-encryption keys enable the transformation of ciphertexts only in one direction, from delegator to delegatee, and is *bidirectional* otherwise. We say a PRE scheme is *single-hop* if ciphertexts are re-encryptable just once, while it is *multi-hop* if they are re-encryptable multiple times. Finally, if the secret key sk_j of the user j is not needed in the re-encryption key generation process, then the scheme is *not interactive*, since re-encryption keys for user j can be produced without her participation.

Note that there are more general definitions of the syntax of PRE schemes, such as the one from Ateniese et al. [2], where instead of single encryption and decryption algorithms, there are sets of algorithms $\overrightarrow{\mathsf{Enc}}$ and $\overrightarrow{\mathsf{Dec}}$. In this case, these algorithms are defined over different ciphertext spaces, where the re-encryption function transforms ciphertexts from one space to another, as opposed to the case of a single ciphertext space, where re-encryption maintains the same space. Figure 1 shows the relations among plaintext and ciphertext spaces for different kinds of PRE schemes, where (1a) represents PRE schemes with a single ciphertext space, while (1b) shows the case of two ciphertext spaces. Examples of PRE schemes with a single ciphertext space are [1, 4, 3, 7], while [2, 8, 9] are schemes with two ciphertext spaces. The former are usually associated with multi-hop PRE schemes, although there are some recent single-hop schemes based on lattices that also present this characteristic [3, 7]. The latter are, on the contrary, usually associated with unidirectional schemes, since this way there may be



(a) Single ciphertext space

(b) Two ciphertext spaces

Figure 1: Transformations between plaintext and ciphertext spaces

transformations between ciphertext spaces that are valid only in one direction. In this paper, we restrict ourselves to PRE schemes defined over a single ciphertext space, although our results can be extended to the other case.

3.2 Parametric Family of Chosen-Ciphertext Attack Models

The additional re-encryption capability that is introduced in proxy re-encryption clearly makes the definitions of security more complex. As in the decryption case, the re-encryption capability can be modeled by the access to a re-encryption oracle $\mathcal{O}_{reenc}$. Thus, it is natural to think of a collection of attack models with different access levels to the decryption and re-encryption oracles. In this section we describe a parametric set of chosen-ciphertext attack models, which depends solely on the availability of these oracles for each phase of the security game. This family of attack models in turn implies a collection of security notions. As mentioned before, we will restrict ourselves to the indistinguishability goal.

It is reasonable to argue that meaningful chosen-ciphertext attack models for proxy re-encryption should provide re-encryption capabilities, in the form of a re-encryption oracle. However, access to this oracle is not sufficient guarantee that proper re-encryption capabilities are granted to the adversary. In order to not make the security game trivial for the adversary, it is necessary that the definition of the re-encryption oracle includes some restrictions. At the same time, it is also important that these restrictions are not too strong, since this would achieve weaker notions of security, as pointed out by Canetti and Hohenberger in [4]. In particular, the re-encryption oracle should be able to re-encrypt any ciphertext that is not derived from the challenge ciphertext, between *any* pair of users, including the target user. Otherwise, the security model would be too restrictive, leading to weaker security notions. Therefore, it is paramount that the proposed attack models are accompanied by a reasonable definition of the oracles available, as well as their restrictions.

3.3 Oracles

The following definitions of oracles and their restrictions are adapted from those of Canetti and Hohenberger [4]. A core concept that shapes the restrictions is the notion of *derivatives of the challenge*. As stated before, the security game could be trivial for the adversary if there were no restrictions, since in certain cases a re-encryption oracle can be used to simulate a decryption oracle. For example, consider the case of the challenge ciphertext c^*, which can be potentially re-encrypted from the target public key pk^* to any other one. Canetti and

Hohenberger introduce the notion of derivatives as a means for shaping the oracle restrictions. Informally, the derivatives of the challenge are those pairs (pk, c) that are linked to (pk^*, c^*) through queries to the re-encryption and re-encryption key generation oracles, and which would allow trivial attacks from the adversary.

Definition 3.2 (Derivatives of the challenge)**.** *The set of derivatives of* (pk^*, c^*) *is defined inductively, as follows:*

- (pk^*, c^*) *is a derivative of itself.*
- *If* (pk_j, c_j) *is a derivative of* (pk_i, c_i) *and* (pk_i, c_i) *is a derivative of* (pk^*, c^*)*, then* (pk_j, c_j) *is a derivative of* (pk^*, c^*)*.*
- *If the adversary has issued a re-encryption query* (pk_i, pk_j, c_i) *and obtained a ciphertext* c_j *as response, then* (pk_j, c_j) *is a derivative of* (pk_i, c_i)*.*
- *If the adversary has issued a re-encryption key generation query* (pk_i, pk_j)*, and* $\mathsf{Dec}(pk_j, c_j) \in \{m_0, m_1\}$*, then* (pk_j, c_j) *is a derivative of all pairs* (pk_i, c)*.*

Once this concept has been established, we will describe the oracles involved in the security game. Apart from the decryption and re-encryption oracles, it is necessary to provide additional oracles for dealing with the inherently multi-user nature of proxy re-encryption. These oracles provide the adversary of keys for multiple users, which can be either honest or corrupt depending on whether the adversary is unaware of the corresponding secret key or not.

Let $n \in \mathbb{N}$ be the security parameter for a PRE scheme, and $\mathcal{I}_H$ and $\mathcal{I}_C$ be the sets of indices of honest and corrupt users, respectively. By definition, the target user is deemed honest, so its index $i^* \in \mathcal{I}_H$. The public and private keys of target user are pk_{i^*} and sk_{i^*}, respectively; we will, however, notate them as pk^* and sk^*, for simplicity. We define the following oracles, which will be made available to the adversary during the security game and which can be invoked multiple times in any order:

- Honest key generation $\mathcal{O}_{honest}$: The oracle obtains a new keypair $(pk_i, sk_i) \leftarrow \mathsf{KeyGen}(n)$, adds index i to $\mathcal{I}_H$, and returns the public key pk_i.
- Corrupt key generation $\mathcal{O}_{corrupt}$: The oracle obtains a new keypair $(pk_i, sk_i) \leftarrow \mathsf{KeyGen}(n)$, adds index i to $\mathcal{I}_C$, and returns the pair (pk_i, sk_i).
- Re-encryption key generation $\mathcal{O}_{rkgen}$: On input a pair of public keys (pk_i, pk_j), the oracle returns the re-encryption key $rk_{i\to j} \leftarrow \mathsf{ReKeyGen}(pk_i, sk_i, pk_j, sk_j)$. The adversary is only allowed to make queries where $i \neq j$, and either $i, j \in \mathcal{I}_H$ or $i, j \in \mathcal{I}_C$.
- Re-encryption $\mathcal{O}_{reenc}$: On input (pk_i, pk_j, c), where $i \neq j$ and $i, j \in \mathcal{I}_H \cup \mathcal{I}_C$, the oracle returns the re-encrypted ciphertext $c' \leftarrow \mathsf{ReEnc}(rk_{i\to j}, c)$. The adversary is not allowed to make queries where $j \in \mathcal{I}_C$ and (pk_i, c) is a derivative of (pk^*, c^*).
- Decryption $\mathcal{O}_{dec}$: On input (pk_i, c), where $i \in \mathcal{I}_H \cup \mathcal{I}_C$, the oracle returns $m \leftarrow \mathsf{Dec}(sk_i, c)$. The adversary is not allowed to make queries where (pk_i, c) is a derivative of (pk^*, c^*).

The ability to re-encrypt and decrypt ciphertexts is modeled by $\mathcal{O}_{reenc}$ and $\mathcal{O}_{dec}$. The restrictions on the derivatives of the challenge ciphertext disallow trivial attacks from the adversary, but are flexible enough to support a wide range of queries. For example, it should be possible for the adversary to issue re-encryption queries of the form $(pk^*, pk_j, \hat{c})$, where pk_j is corrupt, but $(pk^*, \hat{c})$ is not a derivative of (pk^*, c^*).

The key generation oracles $\mathcal{O}_{honest}$, $\mathcal{O}_{corrupt}$, and $\mathcal{O}_{rkgen}$ are always available for the adversary,subject to the restrictions above. It can be seen that we assume a *static corruption* model, where the adversary must decide to corrupt a user or not before asking for the generation of the user's keypair [10], hence the distinction between honest and corrupt key generation. In addition, we also put common restrictions regarding how the adversary obtains keys. In particular, we are assuming the *knowledge of secret key* model, where the challenger generates the key material of all users. A stronger model is the *chosen key* model, where the adversary can adaptively choose public keys for malicious users [9].

The restrictions on the re-encryption key generation oracle have a strong influence on the security model; in particular, we disallow the generation of re-encryption keys between honest and corrupt users, as in some cases, this may lead to trivial attacks from the adversary. We stress that the intention of this work is to support definitional unity for PRE by providing a universal framework that captures the essence of CCA-security, thus avoiding the definition of security notions that are particular for each PRE subfamily. Although it would be possible to define stronger notions by removing the restrictions on this oracle, this would render these definitions useless for schemes with useful PRE properties, such as multi-hop or transitiveness, which are of interest for many applications. For instance, Libert and Vergnaud [9] permit all possible re-encryption key generation queries, except those from the target user to a corrupt one. However, this only works when the scheme is single-hop, non-transitive and resistant to collusions.

There are several examples that justify the restrictions for honest-to-corrupt re-encryption keys. For instance, in multi-hop PRE, an adversary could trivially win the game by first re-encrypting the challenge ciphertext to a honest user key, and then using the honest-to-corrupt key to re-encrypt it to a corrupt user key, thus trivially winning the game. The adversary could also achieve a similar result if the scheme is transitive. Another example is found in not collusion-resistant schemes, where the adversary can use a honest-to-corrupt key and a corrupt secret key to extract the honest secret key. With regard to the restrictions to corrupt-to-honest keys, there are two main reasons for that. On the one hand, adversaries in non-interactive PRE schemes do not require the private key of the delegatee for computing re-encryption keys, so the re-encryption key generation oracle would be superfluous. On the other hand, interactive schemes are usually bidirectional, which means that it is possible to compute a honest-to-corrupt key from a corrupt-to-honest key, so the arguments above apply. Therefore, although it is possible to remove these restrictions, it would make the resulting security notions too particular and not relevant for many PRE schemes, which conflicts with our original intention to attain definitions that are universal.

3.4 Attack Models

Inspired by the mnemonic defined in [6] for attack models CCA1 and CCA2, where the number denotes the last adversarial stage during which she has access to a decryption oracle, we analogously define a parametric set of attack models for proxy re-encryption based on the last

adversarial stage during which the adversary has access to the decryption and re-encryption oracles. Thus, our parametric set of attack models is characterized by a pair of indices $i, j \in \{0, 1, 2\}$, so $\mathsf{CCA}_{i,j}$ denotes an attack model where the adversary has a decryption oracle until Phase i, and a re-encryption oracle until Phase j. As extreme cases, we have that a "pure" CPA model is then denoted by $\mathsf{CCA}_{0,0}$, whereas a "pure" CCA model is represented by $\mathsf{CCA}_{2,2}$. There is, however, a range of intermediate attack models, differentiated by the last stage during which the adversary has access to the decryption and re-encryption oracles. In the following, we formalize this concept.

As stated, what actually varies among the proposed attack models is the availability of the decryption and re-encryption oracles. That is, throughout the game the adversary has access to the key generation oracles $\mathcal{O}_{honest}$, $\mathcal{O}_{corrupt}$, and $\mathcal{O}_{rkgen}$. We denote as Ω^{kg} to the set that comprises these oracles. Now, let Ω_1^{cca} and Ω_2^{cca} denote the sets of additional oracles that are available in Phases 1 and 2, respectively; the possible values that these sets can take are therefore $\varnothing, \{\mathcal{O}_{reenc}\}, \{\mathcal{O}_{dec}\}$, and $\{\mathcal{O}_{dec}, \mathcal{O}_{reenc}\}$. Let $\Omega_1 = \Omega_1^{cca} \cup \Omega^{kg}$ and $\Omega_2 = \Omega_2^{cca} \cup \Omega^{kg}$ be the sets of oracles available in Phases 1 and 2, respectively. Since the set Ω^{kg} of key generation oracles is always present, then the sets Ω_1^{cca} and Ω_2^{cca} fully characterize the possible attack models. We can describe the available oracles in a more formal manner as follows. Let $\mathsf{CCA}_{i,j}$ be an attack model for PRE and $k \in \{1, 2\}$, then $\mathcal{O}_{dec} \in \Omega_k^{cca}$, for $i \geq k$, and $\mathcal{O}_{reenc} \in \Omega_k^{cca}$, for $j \geq k$.

Table 1: Parametric Family of Attack Models for PRE

Ω_1^{cca}	Ω_2^{cca}	Attack model
$\varnothing$	$\varnothing$	$\mathsf{CCA}_{0,0} = \mathsf{CPA}$
$\{\mathcal{O}_{reenc}\}$	$\varnothing$	$\mathsf{CCA}_{0,1}$
$\{\mathcal{O}_{reenc}\}$	$\{\mathcal{O}_{reenc}\}$	$\mathsf{CCA}_{0,2}$
$\{\mathcal{O}_{dec}\}$	$\varnothing$	$\mathsf{CCA}_{1,0}$
$\{\mathcal{O}_{dec}, \mathcal{O}_{reenc}\}$	$\varnothing$	$\mathsf{CCA}_{1,1}$
$\{\mathcal{O}_{dec}, \mathcal{O}_{reenc}\}$	$\{\mathcal{O}_{reenc}\}$	$\mathsf{CCA}_{1,2}$
$\{\mathcal{O}_{dec}\}$	$\{\mathcal{O}_{dec}\}$	$\mathsf{CCA}_{2,0}$
$\{\mathcal{O}_{dec}, \mathcal{O}_{reenc}\}$	$\{\mathcal{O}_{dec}\}$	$\mathsf{CCA}_{2,1}$
$\{\mathcal{O}_{dec}, \mathcal{O}_{reenc}\}$	$\{\mathcal{O}_{dec}, \mathcal{O}_{reenc}\}$	$\mathsf{CCA}_{2,2}$

Table 1 describes the possible attack models for proxy re-encryption as a function of the availability of the oracles, characterized by the sets Ω_1^{cca} and Ω_2^{cca}. The different combinations of available oracles produce a parametric set of attack models, with 9 possible choices. Note that we disallow the possibility of having an oracle in Phase 2 but not in Phase 1, as the wording used in the definition of the parametric attack models uses the term "until Phase" for defining oracles' availability and not "in Phase". As a particular case of these attack models we have that when $\Omega_1^{cca} = \Omega_2^{cca} = \varnothing$, the attained model $\mathsf{CCA}_{0,0}$ is indeed CPA, since no decryption and re-encryption oracles are provided.

Not all possible attack models of this set are meaningful in the context of proxy re-encryption. In fact, it seems reasonable to consider the access to a re-encryption oracle to be easier than to a decryption oracle, since usually, re-encryption is performed as a service by an online semi-trusted entity, whereas decryption capabilities are retained by the users. Hence, attack models in which there are more decryption than re-encryption capabilities, and in particular, those where a decryption oracle is provided but not a re-encryption one ($\mathsf{CCA}_{1,0}$ and $\mathsf{CCA}_{2,0}$), do not seem to be appropriate for shaping security in a PRE scenario. This is discussed in more detail in Section 4.4.

We later show general results that separate some of the security notions that arise from these attack models. In particular, we demonstrate how the leakage of re-encryption keys through $\mathcal{O}_{reenc}$ induces a separation between notions of security. In addition, in Section 5 we show a concrete example of a PRE scheme that is proven secure under a $\mathsf{CCA}_{1,0}$ model, but that fails when one considers a $\mathsf{CCA}_{1,1}$ model.

3.5 Security Notions for PRE

In the same fashion as for PKE, security notions for PRE are constructed by combining a security goal (in this case, indistinguishability of encryptions) and the proposed family of attack models. The following definitions are based upon the definition of security for PKE given in Section 2.

Definition 3.3. *Let* Π*=(KeyGen, ReKeyGen, Enc, Dec, ReEnc,) be a PRE scheme,* $A = (A_1, A_2)$ *a polynomial-time adversary, and* Ω_1 *and* Ω_2 *be the set of available oracles for* A_1 *and* A_2*, respectively. For* $i, j \in \{0, 1, 2\}$*,* $\delta \in \{0, 1\}$*, and* $n \in \mathbb{N}$*, the indistinguishability game is defined by the experiment*

$$
\begin{array}{l}
\text{Experiment } \mathbf{Exp}^{\mathsf{IND\text{-}CCA}_{i,j}}_{\Pi,A,\delta}(n) \\
\quad (pk^*, sk^*) \xleftarrow{R} \mathsf{KeyGen}(n); \qquad\qquad (m_0, m_1, s) \leftarrow A_1(pk^*); \\
\quad c^* \leftarrow \mathsf{Enc}(pk^*, m_\delta); \qquad\qquad d \leftarrow A_2(m_0, m_1, s, c^*) \\
\quad \text{return } d
\end{array}
$$

It is required that in the case that $i = 2$ *or* $j = 2$*, the oracle queries from adversary* A_2 *have to satisfy the restrictions about derivatives of the challenge ciphertext* c^**. The sets of available oracles* Ω_1^{cca} *and* Ω_1^{cca} *are defined in accordance to the attack model* $\mathsf{CCA}_{i,j}$*, as shown in Table 1.*

Definition 3.4. *Let* Π *be a proxy re-encryption scheme and* A *a polynomial-time adversary. For* $i, j \in \{0, 1, 2\}$ *and* $n \in \mathbb{N}$*, the advantage of* A *is given by*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}_{i,j}}_{\Pi,A}(n) = |\Pr[\mathbf{Exp}^{\mathsf{IND\text{-}CCA}_{i,j}}_{\Pi,A,1}(n) = 1] - \Pr[\mathbf{Exp}^{\mathsf{IND\text{-}CCA}_{i,j}}_{\Pi,A,0}(n) = 1]|$$

We say that the PRE scheme Π *is* $\mathsf{IND\text{-}CCA}_{i,j}$ *secure if the advantage* $\mathbf{Adv}^{\mathsf{IND\text{-}CCA}_{i,j}}_{\Pi,A}$ *is negligible.*

Note that, in the security game described by the experiment, we are assuming the *selective model*, since the challenger fixes the target public key pk^* at the beginning. This choice is made for the sake of clarity in the proof, and is also followed by other relevant references, such as [2, 9].

Another aspect worth mentioning arises from how we disallow oracle queries that include the challenge ciphertext (or a derivative). As shown in [11] by Bellare et al., there are different ways for formalizing this in the IND-CCA security game for PKE. They identify four styles (SP, SE, BP, and BE), which result from the combination of two orthogonal factors: the first factor is to disallow these queries only in the second phase ("S") or in both phases ("B"); the second factor depends on whether the adversary is penalized a posteriori in case she makes such queries ("P"), or, simply, this possibility is excluded from the experiment ("E"). In the light of this formalization, the limitations posed on the oracle queries from A_2 in the definition of our experiment imply that our security notions fall under the IND-CCA-SE category, as we only exclude adversaries that query derivatives of the challenge ciphertext in phase 2; no restrictions exist on phase 1, as the set of derivatives would be empty in this phase.

Finally, once the security notions have been defined, it is interesting to illustrate them with examples of related schemes. It is reasonable to expect that most of these representatives belong to the central security notions, namely $\mathsf{IND\text{-}CCA}_{0,0}$, $\mathsf{IND\text{-}CCA}_{1,1}$, and $\mathsf{IND\text{-}CCA}_{2,2}$ (see Section 4.4 for a discussion on why we consider these notions as central). For example, the schemes from Blaze et al. [1] and Ateniese et al. [2] are widely-known examples of $\mathsf{IND\text{-}CCA}_{0,0}$ security. Canetti and Hohenberger present in [4] a classic instance of scheme secure under $\mathsf{IND\text{-}CCA}_{2,2}$. Interestingly, and to the best of our knowledge, there are no examples of $\mathsf{IND\text{-}CCA}_{1,1}$-secure schemes in the literature; Kirshanova presented in [3] a scheme which was allegedly secure under this notion, but we show in Section 5 that this scheme is actually $\mathsf{IND\text{-}CCA}_{1,0}$ secure. As for the rest of the notions, we argue in Section 4.4 that they should be considered as degenerate notions due to their asymmetry in the adversarial capabilities. These notions are usually related to attacked schemes: for instance, Koo et al. present in [12] an attack to the scheme from Green and Ateniese [13] that uses the re-encryption oracle during the second phase; thus, assuming no other attack is found, the scheme from Green and Ateniese could be considered as a representative of $\mathsf{IND\text{-}CCA}_{2,1}$ security, as the attack is not valid under this security notion.

4 Relations among PRE security notions

As explained in Section 2, attack models can be combined with security goals to form security notions. We restrict this paper to the indistinguishability goal. Hence, we derive a parametric set of security notions $\mathsf{IND\text{-}CCA}_{i,j}$, for $i, j \in \{0, 1, 2\}$.

In this section we study the relations between these notions, and in particular, the separations that arise when the adversary is able to learn re-encryption keys from a ciphertext and its re-encrypted value. In addition, we study how these relations are influenced when we consider the multihop property. Finally, we discuss which of these security notions should be considered meaningful, which can be useful when addressing the definitions of security for a PRE scheme.

4.1 Implications among PRE security notions

The relations that naturally arise between these security notions can be represented diagrammatically, as depicted in Figure 2. This figure shows the hierarchical relations that are consequence from the trivial implications between the security notions. The following result demonstrates the trivial fact that a PRE scheme that is secure under some security notion remains secure when one lowers the re-encryption capabilities of the adversary. This result is represented in Figure 2 by the horizontal arrows.

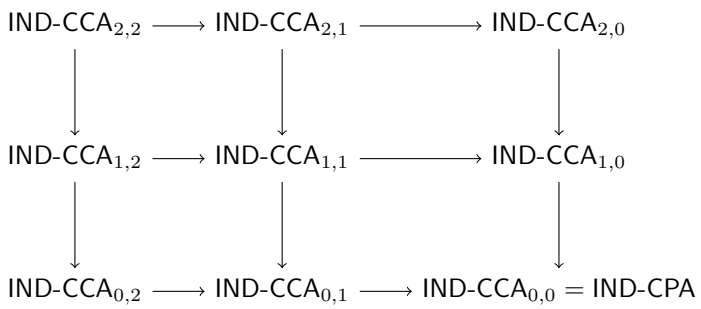


Figure 2: Implications among security notions for PRE

Theorem 4.1 ($\mathsf{IND\text{-}CCA}_{i,j} \Rightarrow \mathsf{IND\text{-}CCA}_{i,j-1}$)**.** *For $i \in \{0,1,2\}$ and $j \in \{1,2\}$, if a PRE scheme is secure in the sense of $\mathit{IND\text{-}CCA}_{i,j}$, then it is also secure in the sense of $\mathit{IND\text{-}CCA}_{i,j-1}$.*

Proof of Theorem 4.1. We prove the contrapositive: if a PRE scheme Π is not secure in the sense of $\mathsf{IND\text{-}CCA}_{i,j-1}$, then it is not secure in the $\mathsf{IND\text{-}CCA}_{i,j}$ sense either. The former means that there is an algorithm B that breaks Π in the $\mathsf{IND\text{-}CCA}_{i,j-1}$ sense with non-negligible advantage ε. From B, we can define an algorithm A for breaking Π in the $\mathsf{IND\text{-}CCA}_{i,j}$ sense and which behaves exactly as B, since the oracles provided to B are also provided to A. Adversary A breaks Π in the $\mathsf{IND\text{-}CCA}_{i,j}$ sense with the same non-negligible advantage ε. □

Similarly, the next result follows from considering the decryption oracle, instead of the re-encryption one. This result is represented in Figure 2 by the vertical arrows.

Theorem 4.2 ($\mathsf{IND\text{-}CCA}_{i,j} \Rightarrow \mathsf{IND\text{-}CCA}_{i,j-1}$)**.** *For $i \in \{1,2\}$ and $j \in \{0,1,2\}$, if a PRE scheme is secure in the sense of $\mathit{IND\text{-}CCA}_{i,j}$, then it is also secure in the sense of $\mathit{IND\text{-}CCA}_{i-1,j}$.*

The proof is analogous to the one for Theorem 4.1.

4.2 Separations among PRE Security Notions

Aside from the relations that arise from the initial – and rather trivial – implications among security notions, we are also interested in the separations that exist between them. Figure 3 shows the separations that we address in this paper; the initial implications shown in Figure 2 are depicted with light dashed arrows for the sake of clarity. Implications of the form $\mathsf{IND\text{-}CCA}_{i,j} \Rightarrow \mathsf{IND\text{-}CCA}_{i',j'}$ mean that each PRE scheme that is secure in the $\mathsf{IND\text{-}CCA}_{i,j}$ sense, it is also secure in the sense of $\mathsf{IND\text{-}CCA}_{i',j'}$. On the contrary, a separation $\mathsf{IND\text{-}CCA}_{i,j} \not\Rightarrow \mathsf{IND\text{-}CCA}_{i',j'}$ means that there is at least a PRE scheme that is secure in the sense of $\mathsf{IND\text{-}CCA}_{i,j}$ but not in the sense of $\mathsf{IND\text{-}CCA}_{i',j'}$.

The separations that we analyze arise from exploiting the vulnerabilities of PRE schemes that do not satisfy the *private re-encryption keys* property. This property was first described in Remark 2.6 of [4], which stated that an adversary should not be able to learn re-encryption keys from a ciphertext and its re-encrypted value. The scheme analyzed in Section 5 is an example of this type of scheme. The following is a more formal and general definition of this property.

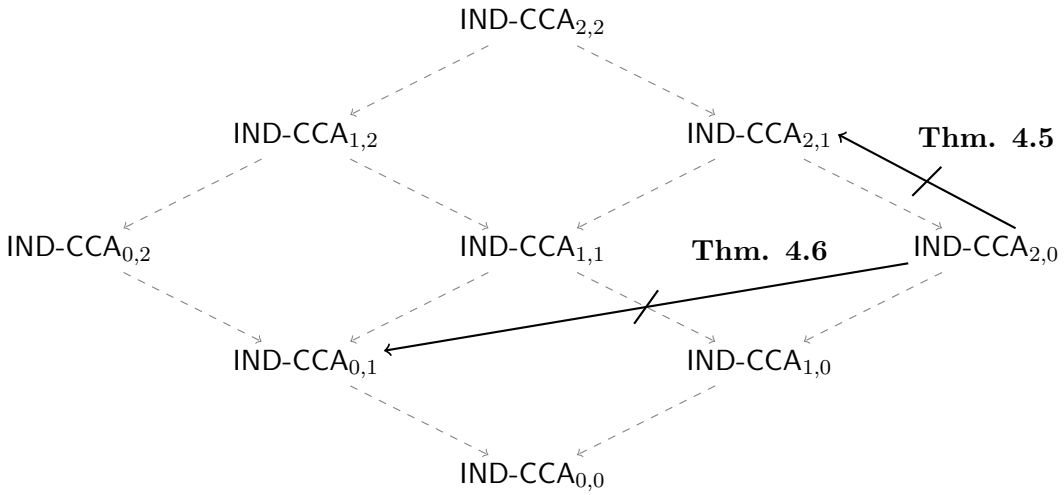


Figure 3: Separations among notions of security.

Definition 4.3 (Private Re-Encryption Keys)**.** *Let* Π *be a proxy re-encryption scheme and* A *a polynomial-time adversary. Let* $\Omega = \{\mathcal{O}_{honest}, \mathcal{O}_{corrupt}, \mathcal{O}_{reenc}\}$ *be the set of available oracles for* A*, and* pk *the public key of a honest user. The scheme* Π *satisfies the* private re-encryption keys *property if the probability that* A *computes a valid* $rk_{pk \to pk'}$*, for some public key* pk' *of her choice, is negligible.*

Note that this definition is oblivious to the nature of the delegatee's public key pk'. If this public key is from a honest user, then this is enough for forbidding the possibility that the scheme achieves $\mathsf{IND\text{-}CCA}_{2,1}$ or $\mathsf{IND\text{-}CCA}_{2,2}$ security, as shown next in Theorem 4.5. A more worrying situation is that pk' belongs to a corrupt user (so A knows sk'), since this completely rules out the chance of the scheme to even be $\mathsf{IND\text{-}CCA}_{0,1}$ secure. This produces a greater separation between notions, as demonstrated in Theorem 4.6.

The original definition of the private re-encryption keys property by Canetti and Hohenberger contains the outline of a plausible attack to schemes that do not fulfill this property. The following result, conveyed by Theorem 4.5, formalizes this attack in order to establish a separation between security notions for PRE. The basic idea behind this attack is that if the attacker can learn $rk_{pk^* \to pk'}$ through queries to $\mathcal{O}_{reenc}$, where pk' belongs to a honest user, then she can win the security game by locally re-encrypting c^* to c', and then querying $\mathcal{O}_{dec}$ with input (pk', c'). Note that this latter query is legal since the re-encryption key has been leaked by queries to $\mathcal{O}_{reenc}$, so it complies with the restrictions for derivatives of the challenge ciphertext. However, it can be seen that this attack strategy only works when the adversary has access to a decryption oracle during the second phase and to a re-encryption oracle in any phase; that is, it can be only of type $\mathsf{CCA}_{2,1}$ or $\mathsf{CCA}_{2,2}$. As a consequence, it is not possible to conduct this attack in any other attack model (i.e., where a re-encryption oracle is not present or where there is no decryption oracle in phase 2)[1]. Therefore, this attack strategy produces the separations expressed by the following theorem.

[1]In principle, this suggests that this attack strategy produces two separations: one from $\mathsf{IND\text{-}CCA}_{1,2}$ to $\mathsf{IND\text{-}CCA}_{2,1}$ (which corresponds to the case when the attack does not work because the decryption oracle is not available in the second phase), and another from $\mathsf{IND\text{-}CCA}_{2,0}$ to $\mathsf{IND\text{-}CCA}_{2,1}$ (the attack does not work be-

```
Algorithm Enc'(pk, x)
  y ← Enc(pk, x)
  return (0, y)

Algorithm Dec'(sk, (y_1, y_2))
  If y_1 = 0 then return Dec(sk, y_2)
  else return ⊥

Algorithm ReEnc'(rk, (y_1, y_2))
  If y_1 = 0 then return (0, ReEnc(rk, y_2))
  else return (1, rk)
```

Figure 4: Modified algorithms in Π'

Theorem 4.5 ($\mathsf{IND\text{-}CCA}_{2,0} \not\Longrightarrow \mathsf{IND\text{-}CCA}_{2,1}$)**.** *If there exists a PRE scheme* Π *that is secure in the sense of* $\mathit{IND\text{-}CCA}_{2,0}$*, then there exists a PRE scheme* Π' *that is secure in the sense of* $\mathit{IND\text{-}CCA}_{2,0}$ *but which is not secure in the sense of* $\mathit{IND\text{-}CCA}_{2,1}$*.*

Proof of Theorem 4.5. First, assume that there exists a PRE scheme $\Pi = (\mathsf{KeyGen}, \mathsf{ReKeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{ReEnc})$ that is secure in the sense of $\mathsf{IND\text{-}CCA}_{1,2}$; otherwise, the theorem is vacuously true. We now define scheme $\Pi' =(\mathsf{KeyGen'}, \mathsf{ReKeyGen'}, \mathsf{Enc'}, \mathsf{Dec'}, \mathsf{ReEnc'})$, where $\mathsf{KeyGen'} = \mathsf{KeyGen}$, $\mathsf{ReKeyGen'} = \mathsf{ReKeyGen}$, and $\mathsf{Enc'}$, $\mathsf{Dec'}$, and $\mathsf{ReEnc'}$) are defined as shown in Figure 4. Informally, Π' simply leaks the re-encryption key during a re-encryption when the first component is different from 0. That is, we are constructing a proxy re-encryption scheme which does not fulfill the property of private re-encryption keys. A re-encryption oracle for this scheme should behave in the same way (i.e., it cannot output a random value instead of the corresponding re-encryption key), since otherwise it could be immediately detected by the adversary by verifying the correctness of the re-encryption.

Claim 4.5.1. Π' *is not secure in the sense of* $\mathit{IND\text{-}CCA}_{2,1}$*.*

Proof of Claim 4.5.1. We define an adversary $A = (A_1, A_2)$ that breaks Π' in the sense of $\mathsf{IND\text{-}CCA}_{2,1}$, with probability 1 and in polynomial time, as follows. As we are working in the selective model, A_1 receives the target public key pk^* at the beginning of the game. In addition, A_1 gets a honest public key pk_h through oracle $\mathcal{O}_{honest}$. According to the definition of the $\mathsf{CCA}_{2,1}$ model, $\mathcal{O}_{reenc}$ is available during the first phase. A_1 queries this oracle with $(pk^*, pk_h, (1, z))$, for a randomly generated ciphertext z. Note that this query is legitimate, since challenge ciphertext has not been generated yet. A_1 receives $rk_{pk^* \to pk_h}$ as part of the output of the re-encryption oracle, includes it in the state s and outputs (m_0, m_1, s). At the guess phase, the challenge ciphertext is $(0, c^*)$, so A_2 receives $(m_0, m_1, s, (0, c^*))$ and extracts $rk_{pk^* \to pk_h}$ from s. Now it computes $\mathsf{ReEnc'}(rk_{pk^* \to pk_h}, (0, c^*)) = (0, c_h)$. Finally, since the decryption oracle $\mathcal{O}_{dec}$ is available during the second phase, according to the definition of the $\mathsf{CCA}_{2,1}$ model, it makes the query $\mathcal{O}_{dec}(pk_h, (0, c_h))$ to recover the message m_δ and determines δ with probability 1. This query is legitimate because $(pk_h, (0, c_h))$ cannot be considered a derivative of the challenge, as the adversary did not obtain the re-encryption key from $\mathcal{O}_{rkgen}$.

cause the re-encryption oracle is never available). In the conference version of this publication [14], we formalized each separation as theorems and provided a proof. However, we noticed some flaws in these proofs. While the fix for the proof of the second separation is immediate, the first one does not seem to be trivial, since it is not possible to generate the re-encryption keys to leak, and therefore, to define the re-encryption oracle in the $\mathsf{CCA}_{1,2}$ attack model. Luckily, the first separation is not relevant with respect to the findings presented in this paper. For this reason, we provide here an amendment for the second separation, expressed by the Theorem 4.5.

Claim 4.5.2. Π' *is secure in the sense of* IND-CCA$_{2,0}$

Proof of Claim 4.5.2. The proof is by contradiction. Suppose that Π' is not secure in the sense of IND-CCA$_{2,0}$. Then, there exists an algorithm B that breaks Π' in the IND-CCA$_{2,0}$ sense with non-negligible advantage ε. From B, we can define an adversary $A = (A_1, A_2)$ that breaks Π in the IND-CCA$_{2,0}$ sense as follows. Algorithm A_1 simply outputs the same results than B_1 for input pk^*, while algorithm A_2 takes input (m_0, m_1, s, c^*) and outputs the result d from calling B_2 with parameters $(m_0, m_1, s, (0, c^*))$. The computations of B are done by A simulating B's decryption oracle $\mathcal{O}^B_{dec}$ using its own oracle $\mathcal{O}^A_{dec}$, as follows: on input $(pk, (c_1, c_2))$ to $\mathcal{O}^B_{dec}$, A verifies that $c_1 = 0$ and outputs $\mathcal{O}^A_{dec}(pk, c_2)$; otherwise, it outputs $\perp$. The simulation is perfect, and A breaks Π in the IND-CCA$_{2,0}$ sense with the same advantage ε.

□

We present now a different attack strategy, which produces a greater separation between PRE security notions. Suppose that the adversary is able to extract $rk_{pk^* \to pk_x}$ through calls to $\mathcal{O}_{reenc}$, where pk_x belongs to a corrupt user. Then, she wins the security game by locally re-encrypting the challenge ciphertext c^* to c_x, which she can decipher using sk_x. The only oracle query to $\mathcal{O}_{reenc}$ is legal since it does not involve a derivative of the challenge ciphertext. It can be seen that this attack strategy can only work when there is a re-encryption oracle available (i.e., any attack model above CCA$_{0,1}$), thus producing a separation between IND-CCA$_{2,0}$ and IND-CCA$_{0,1}$.

Theorem 4.6 (IND-CCA$_{2,0}$ $\not\Longrightarrow$ IND-CCA$_{0,1}$)**.** *If there exists a PRE scheme* Π *that is secure in the sense of* IND-CCA$_{2,0}$*, then there exists a PRE scheme* Π' *that is secure in the sense of* IND-CCA$_{2,0}$ *but which is not secure in the sense of* IND-CCA$_{0,1}$*.*

Proof of Theorem 4.6. The proof in this case is very similar to that of Theorem 4.5, except that we now prove that Π' is not secure in the sense of IND-CCA$_{0,1}$.

Claim 4.6.1. Π' *is not secure in the sense of* IND-CCA$_{0,1}$*.*

Proof of Claim 4.6.1. This proof is similar to the one of Claim 4.5.1. We define an adversary $A = (A_1, A_2)$ that breaks Π' in the sense of IND-CCA$_{0,1}$, with probability 1 and in polynomial time, as follows. Since we are working in the selective model, A_1 receives the target public key pk^* at the beginning of the game. In addition, A_1 gets a pair of corrupt public and private keys (pk_x, sk_x) through oracle $\mathcal{O}_{corrupt}$. Recall that, according to the definition of the CCA$_{0,1}$ attack model, the only additional oracle available in the phase 1 is $\mathcal{O}_{reenc}$. A_1 queries this oracle with $(pk^*, pk_x, (1, z))$, for a randomly generated ciphertext z. A_1 receives $rk_{pk^* \to pk_x}$ as part of the output of the re-encryption oracle, includes it in the state s and outputs (m_0, m_1, s). In phase 2, the challenge ciphertext is $(0, c^*)$, so A_2 receives $(m_0, m_1, s, (0, c^*))$ and extracts $rk_{pk^* \to pk_x}$ from s. Now it computes $\mathsf{ReEnc'}(rk_{pk^* \to pk_x}, (0, c^*)) = (0, c_x)$. Finally, it evaluates $\mathsf{Dec'}(sk_x, (0, c_x))$ to recover the message m_δ and determines δ with probability 1.

□

This theorem entails the following corollary, which formalizes the idea that there are PRE schemes that are secure in the sense of IND-CCA$_{i,0}$, but fail once a re-encryption oracle is introduced.

Corollary 4.7. *If there exists a PRE scheme* Π *that is secure in the sense of* $\mathsf{IND\text{-}CCA}_{i,0}$*, for* $i \in \{0,1,2\}$*, then there exists a PRE scheme* Π' *that is secure in the sense of* $\mathsf{IND\text{-}CCA}_{i,0}$ *but which is not secure in the sense of* $\mathsf{IND\text{-}CCA}_{i,j}$*, for* $j \in \{1,2\}$*.*

The proof of Theorem 4.6 was based on exploiting the violation of the private re-encryption keys property. An interesting side result of this proof is that the violation of this property implies that the scheme cannot satisfy a security notion that considers the re-encryption oracle. This is formalized by the following Theorem.

Theorem 4.8. *Let* Π *be a proxy re-encryption scheme. If* Π *does not satisfy the private re-encryption keys property, then* Π *can only be secure in the sense of* $\mathsf{IND\text{-}CCA}_{i,0}$*, for* $i \in \{0,1,2\}$*.*

Proof of Theorem 4.8. The proof follows a similar strategy to that of Theorem 4.6. If Π does not satisfy the private re-encryption keys property, then it is possible for an adversary to extract $rk_{pk^* \to pk_x}$ through calls to $\mathcal{O}_{reenc}$, where pk_x belongs to a corrupt user. The adversary now wins the security game by locally re-encrypting the challenge ciphertext c^* to c_x, which she can decipher using sk_x. □

The scheme described in Section 5 illustrates this last theorem, as the violation of the privacy of re-encryption keys property can be used to construct a chosen-ciphertext attack using only the re-encryption oracle. We show later that the scheme can be $\mathsf{IND\text{-}CCA}_{1,0}$ secure (i.e., without considering a re-encryption oracle), but not $\mathsf{IND\text{-}CCA}_{1,1}$ secure, as we show how the attacker can use the re-encryption oracle to win the security game. This attack further supports the separation between PKE-based notions (i.e., those that do not consider a re-encryption oracle) and the rest.

4.3 Relations for multihop PRE

In this section we study the effect on the security notions that appears from the following observation: assuming a proper re-encryption oracle (that is, one that enables re-encryption from a honest party to a corrupt one), then the adversary can construct a decryption oracle using the re-encryption one. She solely has to re-encrypt ciphertexts to a corrupt user and decrypt afterwards, since she knows the secret key. More formally, let x be a corrupt user controlled by the adversary, then for all public keys pk_i and all ciphertexts c, where (pk_i, c) is not a derivative of (pk^*, c^*), the decryption oracle can be simulated by the adversary as follows:

$$\mathcal{O}_{dec}(pk_i, c) = \mathsf{Dec}(sk_x, \mathcal{O}_{reenc}(pk_i, pk_x, c)) \tag{1}$$

This equivalence only holds in the multihop case, since when c is already a re-encryption from other ciphertext, then it is not possible to re-encrypt it again; that is, for single-hop PRE schemes this equivalence only holds when the ciphertext has not been re-encrypted (i.e., a second-level ciphertext [2]). On the contrary, multihop PRE schemes permit to re-encrypt ciphertexts indefinitely. Note also that the query $\mathcal{O}_{reenc}(pk_i, pk_x, c)$ is legal as long as (pk_i, c) is not a derivative of (pk^*, c^*). This result allows us to establish additional implications between security notions, for the case of multihop PRE.

In the $\mathsf{IND\text{-}CCA}_{0,1}$ scenario, the re-encryption oracle is only provided in Phase 1 and no decryption oracle is available. For a multihop scheme, it is easy to check that Equation 1 describes a valid simulation of the decryption oracle during Phase 1. Since the challenge ciphertext

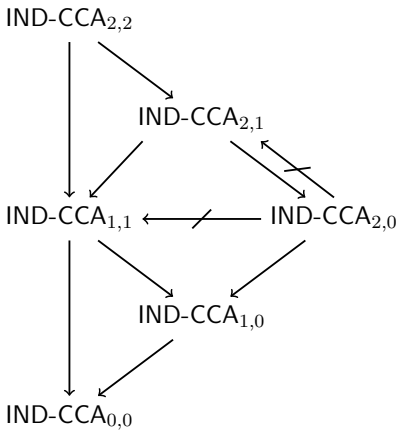


Figure 5: Relations among security notions for multihop PRE

has not yet been generated, neither the re-encryption nor decryption oracles have any kind of restriction on the input ciphertext. Thus, if a re-encryption oracle for Phase 1 is provided, the decryption oracle becomes redundant, since it can be easily simulated by the adversary. Therefore, $\mathsf{IND\text{-}CCA}_{0,1} \Rightarrow \mathsf{IND\text{-}CCA}_{1,1}$.

Now we assume that the re-encryption oracle is also available in Phase 2. In this case, there is a restriction on the input ciphertexts, as described in Section 3, so it cannot be a derivative of the challenge ciphertext. Note that the very same restriction is applicable to the decryption oracle during the same phase. As in the previous case, Equation 1 describes a valid simulation of the decryption oracle, this time for both phases. Similarly, if a re-encryption oracle is provided until Phase 2, then the decryption oracle becomes redundant, as it can be simulated by the adversary using the re-encryption oracle. Therefore, $\mathsf{IND\text{-}CCA}_{0,2} \Rightarrow \mathsf{IND\text{-}CCA}_{2,2}$.

These two results, together with the fact that $\mathsf{IND\text{-}CCA}_{1,1} \Rightarrow \mathsf{IND\text{-}CCA}_{0,1}$ and $\mathsf{IND\text{-}CCA}_{2,2} \Rightarrow \mathsf{IND\text{-}CCA}_{0,2}$, imply that $\mathsf{IND\text{-}CCA}_{1,1} \equiv \mathsf{IND\text{-}CCA}_{0,1}$ and $\mathsf{IND\text{-}CCA}_{2,2} \equiv \mathsf{IND\text{-}CCA}_{0,2}$. Therefore, the previous set of relations between security notions can be simplified for the multihop case, as shown in Figure 5. For clarity, the notions of $\mathsf{IND\text{-}CCA}_{0,2}$ and $\mathsf{IND\text{-}CCA}_{1,2}$ have been assimilated into $\mathsf{IND\text{-}CCA}_{2,2}$, and $\mathsf{IND\text{-}CCA}_{0,1}$ into $\mathsf{IND\text{-}CCA}_{1,1}$.

These results strengthen the idea that the re-encryption oracle is somewhat more powerful than the decryption oracle, and that in some cases, renders it completely redundant.

4.4 Discussion

From the parametric family of attack models for PRE we propose, we have developed in this paper a set of security notions and showed the relations among them. However, that does not necessarily mean that all the notions are equally meaningful for the context of PRE. Recall that one of the main characteristics of PRE is that re-encryption capabilities are granted to some semi-trusted entity that acts as *proxy*, transforming ciphertexts from one public key to another. In this case, it seems reasonable to think of the proxy as a re-encryption oracle, and since it is a semi-trusted entity, its availability for an adversary may be easier than in the case of the decryption oracle. In fact, there are several examples of devised applications of PRE where re-encryption is performed as a service by an online semi-trusted entity, whereas decryption

capabilities are retained by the users [2, 15].

Hence, attack models in which the decryption capabilities are greater than for re-encryption, and particularly, those where a decryption oracle is provided but not a re-encryption one ($\mathsf{CCA}_{2,0}$ and $\mathsf{CCA}_{1,0}$), do not seem to be appropriate for shaping security in scenarios where proxy re-encryption is applied. The scheme we analyze in Section 5 is an example of construction that is only secure under this kind of attack model. In our opinion, it seems difficult to strongly justify any asymmetry between the adversarial capabilities concerning decryption and re-encryption. For that reason, we consider that the representative attack models for PRE are $\mathsf{CCA}_{2,2}$, $\mathsf{CCA}_{1,1}$, and $\mathsf{CCA}_{0,0}$ (which can be denoted as CPA). The rest of attack models (and therefore, security notions) can be considered as the formalization of transitional models, but that fail to properly capture adversarial capabilities of meaningful scenarios. Nevertheless, these transitional models can be of use when reasoning about security notions, e.g., as intermediate milestones during the design of PRE schemes [16].

In the next section, we describe an example of a recent PRE scheme, which is allegedly "CCA1-secure". However, our analysis shows that, according to our definitions of attack models for PRE, this scheme is in fact $\mathsf{IND\text{-}CCA}_{1,0}$ secure, since it does not consider a re-encryption oracle. Furthermore, it cannot achieve a better security notion since it leaks the re-encryption keys when the re-encryption oracle is introduced. This impossibility is a consequence of Theorem 4.6.

5 Security Analysis of a Recent "CCA1-secure" PRE scheme

The scheme analyzed in this section was proposed by Kirshanova at PKC 2014 [3]. Although, the scheme is said to be "CCA1-secure", its security proof does not provide meaningful re-encryption capabilities to the adversary. An analysis of the security proof reveals that this scheme is actually secure under $\mathsf{IND\text{-}CCA}_{1,0}$, because the decryption oracle is available until the challenge, but it lacks of a re-encryption oracle. As discussed in Section 4.4, we do not consider this as a meaningful notion for PRE because of the asymmetry between the adversarial capabilities concerning decryption and re-encryption. In fact, $\mathsf{IND\text{-}CCA}_{1,0}$ is, together with $\mathsf{IND\text{-}CCA}_{0,1}$, the closest notion to $\mathsf{IND\text{-}CCA}_{0,0} \equiv \mathsf{IND\text{-}CPA}$. In this section we describe an attack to this scheme under a more significant notion, namely $\mathsf{IND\text{-}CCA}_{1,1}$. That is, this scheme cannot achieve $\mathsf{IND\text{-}CCA}_{1,1}$ in its current form. The main idea behind this attack is that failure to fulfill the private re-encryption keys property leads to insecure schemes when one considers the re-encryption oracle. The proposed attack is applicable even when a re-encryption oracle is provided only before the challenge phase. In addition, such oracle must not have strong restrictions, since this would imply an overly weak security model, as discussed in Section 3. In particular, it should be possible to ask for the re-encryption of ciphertexts that are unrelated to c^*, from the target user to a corrupt one.

5.1 Description of the scheme

First we will describe the scheme from Kirshanova. This is one of the first PRE schemes based on the hardness of lattice problems, specifically, the Learning With Errors (LWE) [17] and Short Integer Solution (SIS) [18] problems. Some details, such as a comprehensive characterization of the random distributions used and the validity checks in the decryption algorithm, are omitted for the sake of clarity. We refer the reader to [3] for a complete description of the scheme, and to [19] for more details on the original PKE scheme upon Kirshanova's scheme is based.

Let n be the security parameter. The modulus $q = poly(n)$ is a large prime power and $k = \log q$. Set $\bar{m} = O(nk)$ and $m = \bar{m} + 2nk$. Let D_R be a random distribution over $\mathbb{Z}^{\bar{m} \times nk}$ that samples the secret keys R and D_e a random distribution over $\mathbb{Z}^m$ that samples error vectors e; details about these distributions are omitted. Invert_O and Sample_O are two algorithms for inverting the LWE-function and sampling preimages from the SIS-function, respectively. Let $G \in \mathbb{Z}_q^{n \times nk}$ be a specially-constructed public matrix that makes these algorithms efficient. The scheme is as follows:

- $\mathsf{KeyGen}(n)$: choose $A_0 \leftarrow \mathbb{Z}_q^{n \times \bar{m}}$, $R_1, R_2 \leftarrow D_R$ and an invertible matrix $H \leftarrow \mathbb{Z}_q^{nk \times nk}$. Next, define $A_1 = -A_0 R_1 \in \mathbb{Z}_q^{n \times nk}$ and $A_2 = -A_0 R_2 \in \mathbb{Z}_q^{n \times nk}$, and compose the matrix $A = [A_0 | A_1 | A_2] \in \mathbb{Z}_q^{n \times m}$. The public key is the pair $pk = (A, H)$, while the secret key is matrix $sk = [R_1 | R_2] \in \mathbb{Z}^{\bar{m} \times 2nk}$.

- $\mathsf{Enc}(pk = ([A_0 | A_1 | A_2], H), m \in \{0,1\}^{nk})$: choose a non-zero invertible matrix H_u, and a vector $s \leftarrow \mathbb{Z}_q^n$. Set $A_u = [A_0 | A_1 + HG | A_2 + H_u G]$. Sample error vector $e \leftarrow D_e$. Compute $b^t = 2(s^t A_u \mod q) + e^t + (0, 0, enc(m))^t \mod 2q$, where the first zero vector has dimension $\bar{m}$, the second has dimension nk and enc is an encoding function. Output the ciphertext $c = (H, b) \in \mathbb{Z}_q^{n \times n} \times \mathbb{Z}_{2q}^m$.

- $\mathsf{Dec}(pk = ([A_0 | A_1 | A_2], H), sk = [R_1 | R_2], c = (H_u, b))$: Using matrix H_u compute $A_u = [A_0 | A_1 + HG | A_2 + H_u G]$. With the secret key call algorithm $\mathsf{Invert}_O([R_1 | R_2], A_u, b \mod q, H_u)$. As output we receive two vectors $z \in \mathbb{Z}_q^n$ and $e \in \mathbb{Z}_q^m$ that satisfy $b^t = z^t A + e^t \mod q$. Let $v = b - e \mod 2q$. Compute

$$v^t \begin{bmatrix} R_1 & R_2 \\ I & 0 \\ 0 & I \end{bmatrix} \mod 2q$$

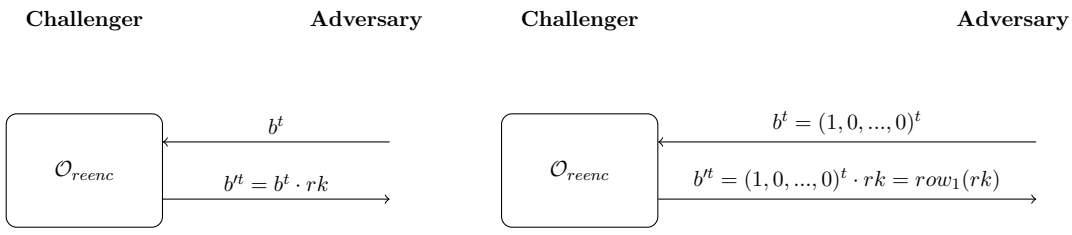
  and apply enc^{-1} to the last nk coordinates.

- $\mathsf{ReKeyGen}(pk = ([A_0 | A_1 | A_2], H), sk = [R_1 | R_2], pk' = ([A'_0 | A'_1 | A'_2], H'))$: Let $Y = [A'_0 | A'_1 + H'G | A'_2 - A_2]$ and y_i be the i-th column of Y. Execute $\mathsf{Sample}_O(y_i, [A_0 | A_1], R_1, H)$ for each column vector y_i and concatenate the column vector outputs to form matrix X. It can be seen that this matrix satisfies that $[A_0 | A_1] X = Y$. Parse matrix X as $[X_0 | X_1 | X_2]$, where the block $X_0 \in \mathbb{Z}^{(\bar{m}+nk) \times \bar{m}}$ is the output corresponding to the first part of Y, $X_1 \in \mathbb{Z}^{(\bar{m}+nk) \times nk}$ to the second one, and $X_2 \in \mathbb{Z}^{(\bar{m}+nk) \times nk}$ to the last one [2]. Finally, output the re-encryption key $rk_{pk \to pk'}$:

$$rk_{pk \to pk'} = \begin{bmatrix} X_0 & X_1 & X_2 \\ 0 & 0 & I \end{bmatrix}$$

- $\mathsf{ReEnc}(rk_{pk \to pk'}, c = (H_u, b))$: to change the underlying public key in the ciphertext component b, compute $b'^t = b^t \cdot rk_{pk \to pk'}$. Finally, output $c' = (H_u, b')$.

[2] In the original scheme, matrix X is further decomposed into smaller blocks (e.g., X_0 is decomposed in X_{00} and X_{01}), but this is not necessary for our analysis.



(a) The re-encryption oracle (b) Using a unit vector as input

Figure 6: Leaking re-encryption keys

5.2 Attack under the IND-CCA$_{1,1}$ setting

In this section we describe an attack to Kirshanova's scheme under the IND-CCA$_{1,1}$ setting. Once we give the attacker the capability of using a re-encryption oracle, and given the fact that the re-encryption function in the Kirshanova PRE scheme does not check for the validity of its inputs, then the attacker can query the re-encryption oracle with a series of deceptive ciphertexts, which ultimately leak the re-encryption key from the target user to a corrupt one. Note that these queries are completely legal, as the input ciphertexts are not related to the challenge ciphertext (in fact, when the re-encryption key is leaked the challenge ciphertext has not been produced yet). Once the attacker asks for the challenge ciphertext, she can re-encrypt it for a corrupt user under her control, and distinguish the original challenge message. This attack constitutes an instantiation of one of the generic attacks used for separating security notions described in Section 4, namely the one used in the proof of Theorem 4.6.

Assuming a IND-CCA$_{1,1}$ setting, then we can construct a CCA$_{1,1}$ attack (i.e., only consulting the decryption and re-encryption oracles before the challenge) as follows. First, as we are working in the selective model, the challenger gives the target public key pk^* to the attacker at the beginning of the game. In addition, the attacker asks for a pair of corrupt public and private keys (pk_x, sk_x) to oracle $\mathcal{O}_{corrupt}$. Next, recall that the re-encryption function simply multiplies the original ciphertext by the re-encryption key, as shown in Equation 2 and depicted by Figure 6a, without performing any kind of prior or subsequent checks:

$$[b_0'^t|b_1'^t|b_2'^t] = [b_0^t|b_1^t|b_2^t] \cdot \begin{bmatrix} X_0 & X_1 & X_2 \\ 0 & 0 & I \end{bmatrix} = [b_0^t|b_1^t] \cdot [X_0|X_1|X_2] + [0|0|b_2^t] \tag{2}$$

The attacker can take advantage from the behavior of the re-encryption oracle and ask for the re-encryption of specially-crafted "trap" ciphertexts $b_i^t = [b_{i,0}^t|b_{i,1}^t|b_{i,2}^t]$, so that $[b_{i,0}^t|b_{i,1}^t] = \hat{u}_i$, where $\hat{u}_i \in \mathbb{Z}_{2q}^{1\times \bar{m}+nk}$ is the i-th row unit vector, for $1 \leq i \leq \bar{m}+nk$, and $b_{i,2}^t$ is chosen randomly in $\mathbb{Z}_{2q}^{1\times nk}$. Next, the attacker proceeds with $\bar{m}+nk$ queries to $O_{reenc}(pk^*, pk_x, (H_i, b_i))$, for random H_i, to obtain re-encrypted ciphertexts (H_i, b_i'). It can be seen that, after removing the term $[0|0|b_{i,2}^t]$, the vector b_i' from the re-encrypted ciphertexts will have the form $b_i' = \hat{u} \cdot [X_0|X_1|X_2]$, which corresponds to the i-th row of the re-encryption key. A simplified illustration of this stage of the attack is shown in Figure 6b. Finally, the attacker simply stacks the results together to obtain the matrix X of the re-encryption key $rk_{pk^*\to pk_x}$. Note that, originally, re-encryption keys are defined over $\mathbb{Z}$, but with these queries she actually obtains its representation in $\mathbb{Z}_{2q}$; nonetheless, the obtained re-encryption key is equivalent for making computations, and thus, equally valid.

The knowledge of this re-encryption key gives the attacker enough power to decrypt the challenge ciphertext during the second phase of the game, and hence distinguish the original message. Note that the queries to the re-encryption oracle can be carried out before the challenge ciphertext is generated, so it is complies with the $\mathsf{IND\text{-}CCA}_{1,1}$ security notion (i.e., it cannot be considered an adaptive attack).

This attack can be refined as follows. Assume that the re-encryption oracle could "detect" the aforementioned queries as trap ciphertexts and return a random output (althought that could mean producing invalid re-encryptions for valid ciphertexts that happen to have the same form). In this case, the attacker may be unable to detect the situation, since she does not know the underlying message of the trap ciphertexts (i.e., she does not know whether or not the trap ciphertexts are valid encryptions under target public key pk^*).

We now describe another way of extracting the sought-after re-encryption key, by concealing these trap vectors in the following way. The attacker produces m ciphertexts $(H_i, p_i^t) = \mathsf{Enc}(pk^*, m_i)$, for random $m_i \in \{0,1\}^{nk}$, and constructs a matrix $P \in \mathbb{Z}_{2q}^{m \times m}$, so that each p_i^t is the i-th row of P. Note that, by the hardness assumption of the encryption scheme, the distribution of vectors p_i^t is within negligible statistical distance from the uniform distribution, as described by the LWE hardness assumption. Hence, with overwhelming probability, P is an invertible matrix, and $P^{-1}P = I$. In the case that P were not invertible, the attacker could simply "resample" more ciphertexts and produce a new P.

Next, the attacker makes m queries $\mathcal{O}_{reenc}(pk^*, pk_x, (H_i, p_i^t))$, with $1 \leq i \leq m$. By the definition of the re-encryption function, the outcome of each query contains the vector $p_i'^t = p_i^t \cdot rk_{pk^* \to pk_x}$. Let $P' \in \mathbb{Z}_{2q}^{m \times m}$, so that each $p_i'^t$ is the i-th row of P'. Hence, it can be seen that $P' = P \cdot rk_{pk^* \to pk_x}$. Finally, she simply computes $rk_{pk^* \to pk_x} = P^{-1} \cdot P' = P^{-1} \cdot P \cdot rk_{pk^* \to pk_x}$. Using this re-encryption key, the attacker decrypts the challenge ciphertexts as described before.

5.3 Analysis of the attack

There are two main reasons that explain why this attack is possible. The first is related to the construction itself, whereas the second concerns the underlying security model.

Firstly, the scheme leaks the re-encryption key through queries to the re-encryption oracle. That is, it fails to satisfy the private re-encryption keys property. This characteristic is what makes possible for the attacker to learn the re-encryption key from the target user to one under her control, thus winning the security game. This is a consequence of how the re-encryption function is constructed, which is simply the multiplication of a ciphertext by the re-encryption key. Thus, when the input ciphertexts are carefully chosen, it is possible to recover the desired re-encryption key. A possible solution to thwart this attack consists on adding a small noise after the multiplication. Another issue with the re-encryption function is that it is performed without any kind of validity check on the input. This poses an interesting problem since this validity check should be performed without the proxy seeing the original plaintexts. The usual solution to this problem is to make the ciphertexts *publicly verifiable* [8]. A consequence of this property is that it prevents the proxy from acting as an oracle [20], which is exactly the weak point of the analyzed scheme.

Secondly, the adversarial model used in this scheme (i.e., $\mathsf{CCA}_{1,0}$) does not provide access to a re-encryption oracle during the security game, so the scheme is oblivious to the fact that it is insecure when one considers access to this oracle, as in $\mathsf{CCA}_{1,1}$. As discussed in Section 4.4, the only attack models deemed meaningful for PRE are $\mathsf{CCA}_{2,2}$, $\mathsf{CCA}_{1,1}$, and $\mathsf{CCA}_{0,0}$, as there

should be very strong reasons for justifying any asymmetry between the adversarial capabilities concerning decryption and re-encryption. In addition, the re-encryption capabilities should be wide enough to allow the adversary to ask for the re-encryption of ciphertexts that are unrelated to the challenge ciphertext, for any pair of users.

6 Conclusions

In this paper we have studied the notions of security for proxy re-encryption by identifying a parametric family of attack models that not only considers the availability of the decryption oracle (as in PKE), but also the re-encryption oracle. Although this seems to be a rather natural step, to the best of our knowledge, it has not been explicitly considered before. This parametric family of attack models for PRE allows us to define a collection of security notions, whose relations we analyze. This type of study is necessary to achieve a better understanding of the definitions of security upon which design proxy re-encryption schemes. As stated by Bellare, Hofheinz and Kiltz in [11], "*Cryptography is founded on definitions (...) In order to have firm foundations – in particular a unique interpretation and common understanding of results – it is important to have definitional unity, meaning that different definitions intending or claiming to represent the same notion should really do so*". The work on this paper is inspired by this principle.

Part of this paper has been concerned with studying separations between PRE notions of security. In particular, we have established separations between security notions by exploiting the private re-encryption keys property, or more accurately, the failure to fulfill this property. The consequence of these separations is that schemes that leak re-encryption keys through queries to the re-encryption oracle cannot achieve strong security notions. These results strengthen the idea that meaningful chosen-ciphertext attacks for PRE should consider both oracles (decryption and re-encryption).

In addition, these separation results have been illustrated with an example of a recent PRE scheme from PKC 2014 [3] that is said to be "CCA1-secure", but that fails to achieve a meaningful notion of security for PRE, since we construct a chosen-ciphertext attack based on queries to the re-encryption oracle.

Acknowledgments

We thank Santiago Zanella Béguelin and the anonymous reviewers for their insightful comments and suggestions. This work was partly supported by the Junta de Andalucía through the project FISICCO (P11-TIC-07223). The first author has been funded by a FPI fellowship from the Junta de Andalucía as part of the project PISCIS (P10-TIC-06334).

References

[1] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. *Advances in Cryptology—EUROCRYPT'98*, pages 127–144, 1998.

[2] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security*, 9(1):1–30, 2006.

[3] Elena Kirshanova. Proxy re-encryption from lattices. In *Public-Key Cryptography–PKC 2014*, pages 77–94. Springer, 2014.

[4] R. Canetti and S. Hohenberger. Chosen-ciphertext secure proxy re-encryption. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 185–194. ACM, 2007.

[5] Victor Shoup. *Why chosen ciphertext security matters*. IBM TJ Watson Research Center, 1998.

[6] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology—CRYPTO'98*, pages 26–45. Springer, 1998.

[7] David Nuñez, Isaac Agudo, and Javier Lopez. NTRUReEncrypt: An efficient proxy re-encryption scheme based on NTRU. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, ASIA CCS '15, pages 179–189, New York, NY, USA, 2015. ACM.

[8] Jian Weng, Robert H Deng, Shengli Liu, and Kefei Chen. Chosen-ciphertext secure bidirectional proxy re-encryption schemes without pairings. *Information Sciences*, 180(24):5077–5089, 2010.

[9] B. Libert and D. Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. *Information Theory, IEEE Transactions on*, 57(3):1786–1802, 2011.

[10] G. Ateniese, K. Benson, and S. Hohenberger. Key-private proxy re-encryption. *Topics in Cryptology–CT-RSA 2009*, pages 279–294, 2009.

[11] Mihir Bellare, Dennis Hofheinz, and Eike Kiltz. Subtleties in the definition of IND-CCA: When and how should challenge decryption be disallowed? *Journal of Cryptology*, 28(1):29–48, 2015.

[12] Woo Kwon Koo, Jung Yeon Hwang, and Dong Hoon Lee. Security vulnerability in a non-interactive ID-based proxy re-encryption scheme. *Information Processing Letters*, 109(23):1260–1262, 2009.

[13] M. Green and G. Ateniese. Identity-based proxy re-encryption. In *Applied Cryptography and Network Security*, pages 288–306. Springer, 2007.

[14] David Nuñez, Isaac Agudo, and Javier Lopez. A parametric family of attack models for proxy re-encryption. In *Proceedings of the 28th IEEE Computer Security Foundations Symposium*, CSF'15, pages 290–301. IEEE Computer Society, 2015.

[15] 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*, pages 69–78. ACM, 2006.

[16] David Nuñez, Isaac Agudo, and Javier Lopez. On the application of generic CCA-secure transformations to proxy re-encryption. *Security and Communication Networks*, 2016.

[17] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[18] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108. ACM, 1996.

[19] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology–EUROCRYPT 2012*, pages 700–718. Springer, 2012.

[20] Jun Shao, Peng Liu, and Jian Weng. CCA-Secure PRE scheme without public verifiability. *IACR Cryptology ePrint Archive*, 2010:357, 2010.