On the Complete Non-Malleability of the Fujisaki-Okamoto Transform*

Daniele Friolo[†1], Matteo Salvino[‡2], and Daniele Venturi[§1]

[1]Department of Computer Science, Sapienza University of Rome, Rome, Italy
[2]Research Institute CODE, Universität der Bundeswehr München, Munich, Germany

Abstract

The Fujisaki-Okamoto (FO) transform (CRYPTO 1999 and JoC 2013) turns any weakly (i.e., IND-CPA) secure public-key encryption (PKE) scheme into a strongly (i.e., IND-CCA) secure key encapsulation method (KEM) in the random oracle model (ROM). Recently, the FO transform re-gained momentum as part of CRISTAL-Kyber, selected by the NIST as the PKE winner of the post-quantum cryptography standardization project.

Following Fischlin (ICALP 2005), we study the *complete non-malleability* of KEMs obtained via the FO transform. Intuitively, a KEM is completely non-malleable if no adversary can maul a given public key and ciphertext into a new public key and ciphertext encapsulating a related key for the underlying blockcipher.

On the negative side, we find that KEMs derived via FO are *not* completely non-malleable in general. On the positive side, we show that complete non-malleability holds in the ROM by assuming the underlying PKE scheme meets an additional property, or by a slight tweak of the transformation.

*This is the full version of an accepted paper of ACNS '23.

[†]friolo@di.uniroma1.it

[‡]matteo.salvino@unibw.de —The work was carried out whilst the author was a student at Sapienza University of Rome, Rome, IT.

[§]venturi@di.uniroma1.it

Contents

1 Introduction

Public-key encryption (PKE) allows Alice to encrypt a message under a Bob's public key, so that Bob can decrypt the ciphertext using the corresponding secret key. Several security notions for PKE have been proposed in the literature. The most basic one, namely *indistinguishability against chosen-plaintext attacks* (IND-CPA) requires that an adversary, given the public key, cannot distinguish between the encryption of two messages.

Non-malleability. As noted for the first time by Dolev *et al.* [9],IND-CPA appears to be insufficient for many applications. Consider for instance the setting of private auctions. Here, a bidder can sample its own pair of public/secret keys, encrypt the bid b using the public key, and send the encryption together with the public key to the auctioneer. After all the participants have sent their bid, the auctioneer can declare the winner by asking each party to reveal the secret key (or the bid itself, along with the random coins used for encryption). A malicious user, given a ciphertext c containing the bid of another party, can try to construct a ciphertext c' that, when decrypted, leads to a bid b' such that $b' > b$.

In light of such malleability attacks, stronger security notions for PKE schemes have been introduced. These include the notions of *non-malleability under chosen-plaintext* and *chosen-ciphertext attacks* [9, 3, 6, 18] (NM-CPA and NM-CCA), and *indistinguishability under chosen-ciphertext attacks* (IND-CCA). All of these notions imply that the attacker, given the public key and a target ciphertext, is unable to craft a mauled ciphertext whose underlying plaintext is related to the one contained in the target ciphertext.

Complete non-malleability. In 2005, Fischlin [12] noted that non-malleability might be still insufficient for some applications. In fact, the above notions do not account for the possibility that the attacker may try to maul the public key as well. For instance, consider again the setting of private auctions. A malicious user, knowing a ciphertext c and the public key pk, may try to craft a public key pk' and a ciphertext c' which encrypt a bid $b' > b$. To capture these attacks, Fischlin introduced *complete non-malleability*, which rules out such adversaries.

As noted by Fischlin himself, completely non-malleable PKE has several useful applications, including key-agreement protocols with security against unknown key attacks, and signature schemes with security against strong unforgeability attacks (as needed, e.g., in e-cash systems).

Known constructions. Fischlin [12] showed that a simple variant of RSA-OAEP is completely non-malleable in the random oracle model (ROM).[1] Ventre and Visconti [22] later gave two constructions of completely non-malleable PKE in the common reference string (CRS) model, based on non-interactive zero-knowledge (NIZK) proofs for all of NP.

Subsequent work provided more efficient constructions of completely non-malleable PKE without random oracles, using both pairing-based assumptions [7, 15] and lattice-based assumptions [20, 21].

1.1 Our Contributions

In practice, due to its computational overhead, PKE is never used to encrypt long messages. Rather, as it happens in many real-world protocols (including TLS), the parties use public-key techniques in order to establish a common secret key for a blockcipher, which can be later used in order to encrypt any subsequent communication of arbitrary length. In the literature this paradigm is also known as the *key/data encapsulation method* (KEM/DEM), or simply hybrid encryption. This motivates our main question:

Can we get efficient *constructions of* completely non-malleable *PKE via the KEM/DEM paradigm?*

[1] He also proves that the original version of RSA-OAEP, as well as the Cramer-Shoup PKE [8], is not completely non-malleable.

Our main contribution is a positive answer to the above question. Namely, we put forward natural notions of complete non-malleability for KEMs and show that these notions are sufficient to imply completely non-malleable PKE with small ciphertext rate. Furthermore, we show that an already existing, widely-used, KEM meets our notions. We elaborate on these contributions below.

Definitions. A *key encapsulation method* (KEM) is made of two algorithms: An encapsulation algorithm that, given the public key pk, outputs a ciphertext c encapsulating a secret key K; and a decapsulation algorithm that, given the secret key sk corresponding to pk, allows to recover K. Similarly to PKE, several non-malleability properties for KEMs have been introduced.

In Section 3, we put forward three indistinguishability-based variants of completely non-malleable KEMs (dubbed NM-CPA*, NM-CCA1* and NM-CCA2*), capturing different flavors of chosen-plaintext and chosen-ciphertext attacks. In Section 3.1, we define the corresponding simulation-based variants (dubbed SNM-CPA*, SNM-CCA1* and SNM-CCA2*), and show the equivalence between the NM-ATK* and SNM-ATK* notions for ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$. More specifically, we show that for NM-CPA* and SNM-CPA* the equivalence holds for so-called *complete relations*, while for NM-CCA1* and SNM-CCA1*, and for NM-CCA2* and SNM-CCA2*, the equivalence holds for a restricted set of relations called *lacking relations* (see Section 3.1 for details). These findings are in line with the work by Ventre and Visconti [22], who showed analogous results for completely non-malleable PKE.

Analysis of Fujiaski-Okamoto. As our main contribution, we analyze the complete non-malleability of the Fujiaski-Okamoto (FO) transform [13]. Recall that the FO transform turns any IND-CPA secure PKE into an IND-CCA secure KEM in the ROM, without affecting the ciphertext size, and at the cost of a very small extra computation effort w.r.t. the underlying PKE scheme (when the RO is replaced with a real-world hash function like SHA-256). Recently, the FO transform re-gained momentum as part of CRISTAL-Kyber, selected by the NIST as the PKE winner of the post-quantum cryptography standardization project [17]. In this light, we believe that investigating further security properties of the FO transform is a very natural research question.

Our analysis follows the modular analysis of the FO transform due to Hofheinz *et al.* [14]. Here, one interprets the FO transform as a sequence of two transformations T and U:

- The transformation T starts with any IND-CPA PKE. The encryption algorithm runs the encryption algorithm of the underlying PKE scheme but sets its randomness to $\mathsf{G}(m)$, where G is a RO. The decryption algorithm runs the decryption algorithm of the underlying PKE scheme, and returns $\perp$ if the decrypted message m' is $\perp$ or if the encryption of m' with randomness $\mathsf{G}(m')$ does not equal the ciphertext.

- The transformation U takes a PKE scheme satisfying different flavours of one-wayness (which are achieved by the transformation T), and outputs an IND-CCA secure KEM. This transformation essentially comes in 2 variants.[2] U_m calculates the encapsulated key K by randomly choosing a message m from the message space of the underlying PKE scheme, encrypting m under pk, and then computing K as $\mathsf{H}(m)$, where H is a RO. U instead computes the key as $\mathsf{H}(m, c)$.

First, in Section 4.1, we show a concrete attack against the transformation U_m that works even when considering the weakest flavour of complete non-malleability (i.e., NM-CPA*). We take the El-Gamal PKE scheme as the base PKE scheme to be transformed by T and U into a KEM scheme. In particular, we prove that an adversary can appropriately maul the public key pk and the ciphertext c encrypting m, and come up with a ciphertext c' encrypting the same message m under a different public key pk'. Since the encapsulated key computed by U_m is $\mathsf{H}(m)$, the key encapsulated by c and c' will be the same. Thus, complete non-malleability is trivially broken.

Second, in Section 4.2, we show that the transformation U is not completely non-malleable. To see this, it suffices to take a contrived PKE scheme in which we add a dummy bit to the public key of a PKE scheme

[2] Each of U and U_m also comes in 2 variants, but the difference is irrelevant for what follows.

satisfying the one-wayness properties required by the transformation U. This additional bit is completely ignored by the encryption algorithm and does not effect one-wayness. However, one can trivially break complete non-malleability by flipping the last bit of the public key. On the positive side, we show that U does achieve complete non-malleability assuming the underlying PKE scheme meets a natural public-key uniqueness property, where the latter essentially means that an adversary cannot come up with different public keys $\mathsf{pk}, \mathsf{pk}'$ for which there exist a message m and a ciphertext c such that c is a valid encryption of m under both pk and pk'.

Indeed, we point out that uniqueness seems to be a standard property to achieve non-malleability, e.g. quasi-unique responses for Fiat-Shamir signatures [12] and [11].

Finally, in Section 4.3, we show how to tweak the transform U in order to obtain complete non-malleability without requiring public-key uniqueness. For this, it suffices to compute the key K as $\mathsf{H}(m, c, \mathsf{pk})$. This way, even if the attacker can break public-key uniqueness, the random oracle will ensure that the two encapsulated keys are independent. We notice that a similar technique was already used in [10], where regular CCA security of the FO transform in the multi-user setting is achieved by adding just a fraction of the public key with high min-entropy as an input to the hash function. However, it is not clear whether complete non-malleability of FO is achievable with such a slight modification.

Relation with completely non-malleable PKE. In Section 5, we show that by combining a completely non-malleable KEM with a non-malleable secret-key encryption (SKE) scheme we obtain a completely non-malleable PKE using the KEM/DEM paradigm. Furthermore, we observe that one can always obtain a completely non-malleable KEM by encrypting a random secret key via a completely non-malleable PKE.

1.2 Related Work

Nagao *et al.* in [16] analyze standard non-malleability in the context of key encapsulation. In particular, they consider different flavours of non-malleable KEM, such as NM-CPA, NM-CCA1 and NM-CCA2.

Ventre and Visconti [22] note that the stronger CCA2 notion of complete non-malleability is not strictly necessary for some of the applications proposed by Fischlin [12]. Hence, they put forward weaker flavours of complete non-malleability, and establish the relations between the standard comparison-based notions NM-ATK* and their simulation-based counterparts SNM-ATK* for ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$. They also give two constructions of completely non-malleable NM-CCA2* secure PKE: one in the CRS model using NIZK proofs for all of NP, and one in the plain model using interactive encryption.

Barbosa and Farshim [1] consider an equivalent indistinguishability-based notion of complete non-malleability based on so-called strong CCA security, in which the (strong) decryption oracle provides decryptions under arbitrarily chosen public keys. Duman *et al.* [10] analyze CCA security of the FO transform in the multi-user setting.

2 Preliminaries

In this section we introduce some basic notation and recall a few standard definitions that will be used later to prove some of our results.

2.1 Notation

We use calligraphic letters to denote sets, such as $\mathcal{X}$, and lower-case letters for variables, such as x. We use $x \leftarrow\!\!\$\ \mathcal{X}$ to indicate that x is picked uniformly at random from $\mathcal{X}$. A similar notation is used in the presence of a randomized or probabilistic algorithm A. Indeed, $x \leftarrow\!\!\$\ \mathsf{A}(\cdot)$ means that x is the output of the randomized algorithm A. Alternatively, when a random coin r is given as an input of A, we equivalently write $x := \mathsf{A}(\cdot; r)$. All the algorithms we will consider are PPT (Probabilistic Polynomial Time), i.e. for any input $x \in \{0,1\}^*$ and random coin r, $\mathsf{A}(x; r)$ terminates in at most polynomial many steps, in the size of its inputs. When an algorithm A has access to a set of oracles $\{\mathsf{O}_1, \ldots, \mathsf{O}_n\}$, we use the notation $\mathsf{A}^{\mathsf{O}_1, \ldots, \mathsf{O}_n}$, to indicate

that A can interact in a black-box manner with oracles $\mathsf{O}_1, \ldots, \mathsf{O}_n$ during its computation. We denote with $\lambda \in \mathbb{N}$ the security parameter and we will assume that all the algorithms we will consider take λ as an input. A function $\nu : \mathbb{N} \to [0,1]$ is negligible if for every polynomial $p(n)$ $\exists N \in \mathbb{N}$ s.t. $\forall n_0 \geq N, \nu(n_0) < \frac{1}{p(n_0)}$. We denote with $\mathsf{negl}(\lambda)$ any function that is negligible in λ. Given two random variables X and Y, we denote $X \approx_c Y$ when X and Y are computationally indistinguishable, and with $X \equiv Y$ when X and Y are identically distributed.

2.2 Public-Key Encryption

A public-key encryption (PKE) scheme Π consists of three algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, together with a message space $\mathcal{M}$ (which we assume to be efficiently recognizable) where:

- The key generation algorithm Gen takes as input 1^λ and outputs a public-private key pair $(\mathsf{pk}, \mathsf{sk})$.
- The encryption algorithm Enc takes as inputs a public key pk and a message $m \in \mathcal{M}$, and outputs an encryption c of the message m under pk.
- The deterministic decryption algorithm Dec takes as inputs a decryption key sk and a ciphertext c, and outputs either a message $m \in \mathcal{M}$, or $\perp$ (denoting failure).

Next, we define both correctness and security of PKE as needed for our purposes. Some of the definitions below are taken verbatim from [14].

Definition 1 (γ**-uniformity**). *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with message space $\mathcal{M}$.Given $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$, a message $m \in \mathcal{M}$ and a ciphertext c we define the γ-uniformity function as follows*

$$\gamma(m, c) = \Pr\left[c = \mathsf{Enc}(\mathsf{pk}, m)\right],$$

where the probability is taken over the choice of the random coins used for encrypting m under pk.

We say that Π is γ-uniform if, for any $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda)$, any message $m \in \mathcal{M}$, and any ciphertext $c \in \{0,1\}^*$, it holds that $\gamma(m, c) \leq \gamma$.

Definition 2 (**correctness**). *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with message space $\mathcal{M}$. A PKE scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is correct if $\forall m \in \mathcal{M}, \forall (pk, sk) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m\right] = 1.$$

2.2.1 OW-PCA and OW-PCVA.

We recall the definitions of one-wayness under plaintext checking attacks (OW-PCA) and one-wayness under plaintext and validity checking attacks (OW-PCVA).

Definition 3 (OW-ATK). *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme with message space $\mathcal{M}$. We define the following $\mathsf{PKE}^{ow\text{-}atk}$ games for $atk \in \{pca, pcva\}$*

Experiment $\mathsf{PKE}^{ow\text{-}atk}_{\Pi,\mathsf{A}}(\lambda)$	***Oracle*** $\mathsf{PCO}(\mathsf{sk}^*, m, c)$
$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$	**return** 1 *iff*
$m^* \leftarrow\!\!\$\ \mathcal{M}$	$(\mathsf{Dec}(\mathsf{sk}^*, c) = m) \wedge (m \neq \perp)$
$c^* \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}^*, m^*)$	
$m' \leftarrow\!\!\$\ \mathsf{A}^{\mathsf{O}_1}(\mathsf{pk}, c^*)$	***Oracle*** $\mathsf{CVO}^{(c^*)}(\mathsf{sk}^*, c)$
return $\mathsf{PCO}(\mathsf{sk}^*, m', c^*)$	$m := \mathsf{Dec}(\mathsf{sk}^*, c)$
	return 1 *iff* $m \neq \perp$

In the experiment above,

$$\begin{aligned} &\textit{if } atk = pca \textit{ then } \mathsf{O}_1 = \mathsf{PCO}(\mathsf{sk}^*, \cdot, \cdot), \\ &\textit{if } atk = pcva \textit{ then } \mathsf{O}_1 = \mathsf{PCO}(\mathsf{sk}^*, \cdot, \cdot), \mathsf{CVO}^{(c^*)}(\mathsf{sk}^*; \cdot), \end{aligned}$$

where $\mathsf{CVO}^{(c^*)}(\mathsf{sk}, \cdot)$ *means that* A *is allowed to query the* CVO *algorithm for any ciphertext distinct from the challenge ciphertext* c^**. We say that* Π *is OW-ATK secure if for all PPT* A*,*

$$\Pr\left[\mathsf{PKE}^{ow\text{-}atk}_{\Pi,\mathsf{A}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda).$$

2.2.2 Complete Non-Malleability.

Finally, we recall the indistinguishability-based security definition for completely non-malleable PKE as defined by Ventre and Visconti [22].

Definition 4 (NM-CPA*, NM-CCA1*, NM-CCA2*). *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme, an let* $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ *be a PPT adversary. For* $atk \in \{cpa, cca1, cca2\}$ *and* $\lambda \in \mathbb{N}$ *let*

$$\mathsf{PKE}^{nm\text{-}atk*}_{\Pi,\mathsf{A}}(\lambda) \approx_c \mathsf{PKE}^{nm\text{-}atk*}_{\Pi,\mathsf{A},\$}(\lambda),$$

where the experiments $\mathsf{PKE}^{nm\text{-}atk*}_{\Pi,\mathsf{A}}(\lambda)$ *and* $\mathsf{PKE}^{nm\text{-}atk*}_{\Pi,\mathsf{A},\$}(\lambda)$ *are defined as follows:*

Experiment $\mathsf{PKE}^{nm\text{-}atk*}_{\Pi,\mathsf{A}}(\lambda)$	***Experiment*** $\mathsf{PKE}^{nm\text{-}atk*}_{\Pi,\mathsf{A},\$}(\lambda)$
$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$	$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$
$(\mathcal{M}, s) \leftarrow\!\!\$\ \mathsf{A}_1^{\mathsf{O}_1}(\mathsf{pk})$	$(\mathcal{M}, s) \leftarrow\!\!\$\ \mathsf{A}_1^{\mathsf{O}_1}(\mathsf{pk})$
$m^* \leftarrow\!\!\$\ \mathcal{M}$	$m^*, \tilde{m} \leftarrow\!\!\$\ \mathcal{M}$
$c^* \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}^*, m^*)$	$c^* \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}^*, m^*)$
$(\mathsf{pk}, R, c) \leftarrow\!\!\$\ \mathsf{A}_2^{\mathsf{O}_2}(\mathcal{M}, \mathsf{pk}^*, s, c^*)$	$(\mathsf{pk}, R, c) \leftarrow\!\!\$\ \mathsf{A}_2^{\mathsf{O}_2}(\mathcal{M}, \mathsf{pk}^*, s, c^*)$
return 1 *iff* $\exists(m, r)$ *s.t.*	**return** 1 *iff* $\exists(m, r)$ *s.t.*
$(c = \mathsf{Enc}(\mathsf{pk}, m; r)) \wedge$	$(c = \mathsf{Enc}(\mathsf{pk}, m; r)) \wedge$
$(c \neq c^* \vee \mathsf{pk} \neq \mathsf{pk}^*) \wedge$	$(c \neq c^* \vee \mathsf{pk} \neq \mathsf{pk}^*) \wedge$
$(m \neq \perp) \wedge R(m, m^*, \mathsf{pk}, \mathsf{pk}^*, c)$	$(m \neq \perp) \wedge R(m, \tilde{m}, \mathsf{pk}, \mathsf{pk}^*, c)$

In the experiments above

$$\begin{aligned} &\textit{if } atk = cpa \textit{ then } \mathsf{O}_1 = \epsilon && \textit{and } \mathsf{O}_2 = \epsilon, \\ &\textit{if } atk = cca1 \textit{ then } \mathsf{O}_1 = \mathsf{Dec}(\mathsf{sk}^*, \cdot) && \textit{and } \mathsf{O}_2 = \epsilon, \\ &\textit{if } atk = cca2 \textit{ then } \mathsf{O}_1 = \mathsf{Dec}(\mathsf{sk}^*, \cdot) && \textit{and } \mathsf{O}_2 = \mathsf{Dec}^{(c^*)}(\mathsf{sk}^*, \cdot), \end{aligned}$$

where $\mathsf{Dec}^{(c)}(\mathsf{sk}, \cdot)$ *means that* A *is allowed to query* Dec *oracle for any ciphertext distinct from the challenge ciphertext* c^**.*

2.3 Secret-Key Encryption

A *secret-key encryption* scheme (SKE) consists of a triple of algorithm $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ together with a message space $\mathcal{M}$ and a key space $\mathcal{K}$, in which:

- The key generation algorithm Gen takes as input 1^λ and outputs a secret key $\mathsf{K} \in \mathcal{K}$.

- The encryption algorithm Enc takes as input the secret key $\mathsf{K} \in \mathcal{K}$ and a message $m \in \mathcal{M}$, and outputs a ciphertext c.
- The decryption algorithm Dec takes as input the secret key $\mathsf{K} \in \mathcal{K}$ and a ciphertext c, and outputs a message $m \in \mathcal{M}$, or $\perp$ denoting failure.

Let us consider the flavour of correctness needed for our scopes.

Definition 5 (ϵ-correctness)**.** *A SKE scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* ϵ*-correct if*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, c) \neq m \mid \mathsf{K} \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda); c \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{K}, m)\right] \leq \epsilon.$$

To make our security proofs go through, we use the non-malleability security definition NM-ATK with ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$ introduced by Bellare et al. [2]. As highligthed in [4, 5] the original definitions were introduced the asymmetric setting [6, 9, 19] but can be "lifted" to the symmetric setting using the encryption oracle based template of [2]. Hence, by leveraging the results of Bellare et al [6], NM-ATK for ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$ is equivalent to the indistinguishability-based counterpart IND-ATK [6, 4, 5].

Definition 6 (**NM-CPA, NM-CCA1, NM-CCA2**)**.** *Given a set of relations* $\mathcal{R}$*, a SKE scheme* Π *is NM-ATK secure with respect to any relation* $R \in \mathcal{R}$*, if for any NM-ATK adversary* $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$*,*

$$\mathsf{SKE}^{nm\text{-}atk}_{\Pi,\mathsf{A}}(\lambda) \approx_c \mathsf{SKE}^{nm\text{-}atk}_{\Pi,\mathsf{A},\$}(\lambda),$$

where the experiments are defined as follows

Experiment $\mathsf{SKE}^{nm\text{-}atk}_{\Pi,\mathsf{A}}(\lambda)$	***Experiment*** $\mathsf{SKE}^{nm\text{-}atk}_{\Pi,\mathsf{A},\$}(\lambda)$
$\mathsf{K}^* \leftarrow\!\!\$\ \mathcal{K}$	$\mathsf{K}^* \leftarrow\!\!\$\ \mathcal{K}$
$(\mathcal{M}, s) \leftarrow\!\!\$\ \mathsf{A}_0^{\mathsf{O}_1}(1^\lambda)$	$(\mathcal{M}, s) \leftarrow\!\!\$\ \mathsf{A}_0^{\mathsf{O}_1}(1^\lambda)$
$m^* \leftarrow\!\!\$\ \mathcal{M}$	$m^*, \tilde{m} \leftarrow\!\!\$\ \mathcal{M}$
$c^* \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{K}^*, m^*)$	$c^* \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{K}^*, m^*)$
$(R, c') \leftarrow\!\!\$\ \mathsf{A}_1^{\mathsf{O}_2}(\mathcal{M}, s, c^*)$	$(R, c') \leftarrow\!\!\$\ \mathsf{A}_1^{\mathsf{O}_2}(\mathcal{M}, s, c^*)$
$m' := \mathsf{Dec}(\mathsf{K}^*, c')$	$m' := \mathsf{Dec}(\mathsf{K}^*, c')$
return 1 *iff*	**return** 1 *iff*
$(m' \neq \perp) \wedge (c' \neq c^*) \wedge R(m^*, m')$	$(m' \neq \perp) \wedge (c' \neq c^*) \wedge R(\tilde{m}, m')$

In the experiments above

$$\begin{aligned}
&\textit{if } atk = cpa \textit{ then } \mathsf{O}_1 = \mathsf{Enc}(\mathsf{K}^*, \cdot) && \textit{and } \mathsf{O}_2 = \mathsf{Enc}(\mathsf{K}^*, \cdot),\\
&\textit{if } atk = cca1 \textit{ then } \mathsf{O}_1 = \mathsf{Enc}(\mathsf{K}^*, \cdot), \mathsf{Dec}(\mathsf{K}^*, \cdot) && \textit{and } \mathsf{O}_2 = \mathsf{Enc}(\mathsf{K}^*, \cdot),\\
&\textit{if } atk = cca2 \textit{ then } \mathsf{O}_1 = \mathsf{Enc}(\mathsf{K}^*, \cdot), \mathsf{Dec}(\mathsf{K}^*, \cdot) && \textit{and } \mathsf{O}_2 = \mathsf{Enc}(\mathsf{K}^*, \cdot), \mathsf{Dec}^{(c^*)}(\mathsf{K}^*, \cdot),
\end{aligned}$$

where $\mathsf{Dec}^{(c^*)}(\mathsf{K}^*, \cdot)$ *means that* A *is allowed to query the* Dec *oracle for any ciphertext* c *distinct from the challenge ciphertext* c^**.*

3 Completely Non-Malleable KEMs

In this section, we formally introduce key encapsulation methods (KEMs). Then, following the work on completely non-malleable PKE schemes of Fischlin [12], and then Ventre and Visconti [22], we extend the notions of complete non-malleability of PKEs to the context of KEMs. We follow a similar blueprint of [22] (that, in turn, bases its definitions on [3]) by introducing three indistinguishability-based security notions for completely non-malleable KEMs dubbed NM-CPA*, NM-CCA1* and NM-CCA2*. We further introduce,

in Section 3.1, three simulation-based notions dubbed SNM-CPA*, SNM-CCA1* and SNM-CCA2* and investigate the relationship between indistuinghishability-based and the simulation-based notions.

A KEM scheme consists of a triple of algorithms $\Pi = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$, together with a key space $\mathcal{K}$,in which:

- The key generation algorithm Gen takes as input 1^λ and outputs a public-private key pair $(\mathsf{pk}, \mathsf{sk})$.
- The encapsulation algorithm Encaps takes as input a public key pk, and output a ciphertext c as well as a key K.
- The decapsulation algorithm Decaps takes as input a private key sk and a ciphertext c and returns a key $\mathsf{K} \in \mathcal{K}$ or $\perp$ (denoting failure).

First of all, we start by considering the flavour of correctness needed for our scopes.

Definition 7 (ϵ-correctness)**.** *A KEM scheme* $\Pi = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ *is* ϵ*-correct if*

$$\Pr\left[\mathsf{Decaps}(\mathsf{sk}, c) \neq \mathsf{K} \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda); (c, \mathsf{K}) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk})\right] \leq \epsilon.$$

Before diving into the definitions complete non-malleability for KEMs, we introduce the notion of complete relation for KEMs, firstly defined by Fischlin [12] in the setting of completely non-malleable PKE. A complete relation R in the KEM setting is a probabilistic algorithm taking as inputs two public keys pk and pk', two encapsulation keys K and K^*, and a ciphertext c'. It outputs 1 if the relation is satisfied, and 0 otherwise. We will refer as $\mathcal{R}$ to be the set of complete relations.

In the indistinguishability-based notion of complete non-malleability, we ask the adversary to distinguish between two experiments. In both of them, we let the adversary learn the challenge public key, ciphertext and encapsulated key, and then make the adversary output a new public key pk', a relation R and a new ciphertext c'. If there exists a key $\mathsf{K}' \neq \mathsf{K}$ and randomness r such that c' and K' can be obtained by running the encapsulation algorithm with $\mathsf{pk}' \neq \mathsf{pk}^*$ and randomness r, then the experiment will output 1. In the left-side experiment the adversary will receive the key K^* encapsulated in c^*, while in the right-side experiment the key K^* is sampled randomly and hence totally unrelated from the key encapsulated in the received ciphertext. Note that the adversary may come up with a triple (pk', R, c') such that there exists a key encapsulated in c' satisfying the relation $R(\mathsf{K}', \mathsf{K}^*, \mathsf{pk}', \mathsf{pk}^*, c')$, but the adversary may have negligible advantage in distinguishing whether the key was encapsulated in c^* or it was randomly chosen.

Definition 8 (**NM-CPA*, NM-CCA1*, NM-CCA2***)**.** *Given a set of relations* $\mathcal{R}$*, a key-encapsulation mechanism* $\Pi = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ *is NM-ATK* secure with respect to any relation* $R \in \mathcal{R}$*, if for any NM-ATK* adversary* $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ *for* $atk \in \{cpa, cca1, cca2\}$ *and for all* $\lambda \in \mathbb{N}$*,*

$$\mathsf{KEM}^{nm\text{-}atk*}_{\Pi,\mathsf{A}}(\lambda) \approx_c \mathsf{KEM}^{nm\text{-}atk*}_{\Pi,\mathsf{A},\$}(\lambda),$$

where the experiments are defined as follows:

Experiment $\mathsf{KEM}^{nm\text{-}atk*}_{\Pi,\mathsf{A}}(\lambda)$

$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$

$\mathsf{st} \leftarrow\!\!\$\ \mathsf{A}_1^{\mathcal{O}_1}(\mathsf{pk}^*)$

$(c^*, \mathsf{K}^*) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk}^*)$

$(\mathsf{pk}', R, c') \leftarrow\!\!\$\ \mathsf{A}_2^{\mathcal{O}_2}(\mathsf{pk}^*, c^*, \mathsf{K}^*, \mathsf{st})$

return 1 *if* $\exists(\mathsf{K}', r)$ *such that*

$((c', \mathsf{K}') = \mathsf{Encaps}(\mathsf{pk}'; r)) \wedge$

$(\mathsf{pk}' \neq \mathsf{pk}^* \vee \mathsf{K}' \neq \mathsf{K}^*) \wedge (\mathsf{K}' \neq \perp)$

$\wedge\ R(\mathsf{K}', \mathsf{K}^*, \mathsf{pk}', \mathsf{pk}^*, c')$

Experiment $\mathsf{KEM}^{nm\text{-}atk*}_{\Pi,\mathsf{A},\$}(\lambda)$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$

$\mathsf{st} \leftarrow\!\!\$\ \mathsf{A}_1^{\mathcal{O}_1}(\mathsf{pk}^*)$

$\mathsf{K}^* \leftarrow\!\!\$\ \{0,1\}^\lambda$

$(\hat{c}, \widehat{\mathsf{K}}) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk}^*)$

$(\mathsf{pk}', R, c') \leftarrow\!\!\$\ \mathsf{A}_2^{\mathcal{O}_2}(\mathsf{pk}^*, \hat{c}, \mathsf{K}^*, \mathsf{st})$

return 1 *if* $\exists(\mathsf{K}', r)$ *such that*

$((c', \mathsf{K}') = \mathsf{Encaps}(\mathsf{pk}'; r)) \wedge$

$(\mathsf{pk}' \neq \mathsf{pk}^* \vee \mathsf{K}' \neq \mathsf{K}^*) \wedge (\mathsf{K}' \neq \perp)$

$\wedge\ R(\mathsf{K}', \mathsf{K}^*, \mathsf{pk}', \mathsf{pk}^*, c')$

In the experiments above,

$$\begin{aligned} &\textit{if } atk = cpa \textit{ then } \mathsf{O}_1 = \epsilon && \textit{and } \mathsf{O}_2 = \epsilon, \\ &\textit{if } atk = cca1 \textit{ then } \mathsf{O}_1 = \mathsf{Decaps}(\mathsf{sk}^*, \cdot) && \textit{and } \mathsf{O}_2 = \epsilon, \\ &\textit{if } atk = cca2 \textit{ then } \mathsf{O}_1 = \mathsf{Decaps}(\mathsf{sk}^*, \cdot) && \textit{and } \mathsf{O}_2 = \mathsf{Decaps}^{(c^*)}(\mathsf{sk}^*, \cdot), \end{aligned}$$

where $\mathsf{Decaps}^{(c^*)}(\mathsf{sk}, \cdot)$ *means that* A *is allowed to query* Decaps *algorithm for any ciphertext distinct from the challenge ciphertext* c^**.*

In [12], Fischlin showed that there exist encryption and signatures schemes that are not NM-CCA2* secure, even though they are NM-CCA2 secure, i.e. the CCA secure in standard non-malleability notion of PKE (see [6] for the formal definition). As we show in the following theorem, this holds also in the case of KEMs. Let $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$.

Theorem 1. *Assume that there exists a NM-ATK secure KEM* $\Pi = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$*; then there exists a NM-ATK secure KEM* $\Pi' = (\mathsf{Gen}', \mathsf{Encaps}', \mathsf{Decaps}')$ *which is not NM-ATK* secure.*

Proof. The intuition behind the proof is to define the scheme Π' in such a way that the adversary can leverage its structure to succeed in the NM-ATK* experiment. In particular, we define Π' in the following way:

Algorithm $\mathsf{Gen}'(1^\lambda)$	**Algorithm** $\mathsf{Encaps}'(\mathsf{pk}')$	**Algorithm** $\mathsf{Decaps}'(\mathsf{sk}, c)$
$(\mathsf{pk}, \mathsf{sk}) = \mathsf{Gen}(1^\lambda)$	Parse pk' as $\mathsf{pk}' := \mathsf{pk}\|\|b$	$\mathsf{K} := \mathsf{Decaps}'(\mathsf{sk}, c)$
$b \leftarrow\!\!\$\ \{0,1\}$	$(c, \mathsf{K}) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk})$	
$\mathsf{pk}' := \mathsf{pk}\|\|b$		
return $(\mathsf{pk}', \mathsf{sk})$	**return** (c, K)	**return** K

We can clearly see that Π' is not NM-ATK* secure. Indeed, an efficient adversary A receiving a public key pk, the challenge ciphertext c and an encapsulation key K (either the real or the fake one), just have to flip the last bit of pk denoted $\mathsf{pk}' := \mathsf{pk}||\bar{b}$, a relation R and the challenge ciphertext c. In this case, A will always succeeds in breaking NM-ATK* security of Π'.

However, Π' is still NM-ATK secure. Indeed, this is true because the adversary has to break NM-ATK security of Π' under the key pk, but for how $\Pi^{'}$ is defined, this is equivalent to break NM-ATK security of Π. If an adversary is able to break NM-ATK security of Π', then he can also break NM-ATK security of Π, and this represent a contradiction to our assumption that Π is NM-ATK secure. □

3.1 Relation between NM-ATK* and SNM-ATK*

We introduce the simulation-based variant of NM-ATK* security, dubbed SNM-ATK*. The left-side of the simulation-based experiment is identical to the left-side of the indistinguishability-based notion (Definition 8), whereas in the right-side experiment the challenge key K^* is sampled at random, and the adversary is replaced with a simulator knowing only public information, i.e. the challenge public key pk^* .

Definition 9 (SNM-CPA*, SNM-CCA1*, SNM-CCA2*). *Given a set of relations* $\mathcal{R}$*, a key-encapsulation mechanism* $\Pi = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ *is secure in the sense of SNM-ATK* with respect to any relation* $R \in \mathcal{R}$*, if for any SNM-ATK* adversary* $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$*, for* $atk \in \{cpa, cca1, cca2\}$*, there exists a simulator* S *such that*

$$\mathsf{KEM}^{snm\text{-}atk*}_{\Pi,\mathsf{A}}(\lambda) \approx_c \mathsf{KEM}^{snm\text{-}atk*}_{\Pi,\mathsf{S}}(\lambda),$$

where the experiments $\mathsf{KEM}^{snm\text{-}atk*}_{\Pi,\mathsf{A}}(\lambda)$ *and* $\mathsf{KEM}^{snm\text{-}atk*}_{\Pi,\mathsf{S}}(\lambda)$ *are defined as follows:*

Experiment $\mathsf{KEM}^{snm\text{-}atk*}_{\Pi,\mathsf{A}}(\lambda)$	***Experiment*** $\mathsf{KEM}^{snm\text{-}atk*}_{\Pi,\mathsf{S}}(\lambda)$
$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$	$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$
$\mathsf{st} \leftarrow\!\!\$\ \mathsf{A}_1^{\mathcal{O}_1}(\mathsf{pk}^*)$	$\mathsf{K}^* \leftarrow\!\!\$\ \{0,1\}^\lambda$
$(c^*, \mathsf{K}^*) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk}^*)$	
$(\mathsf{pk}', R, c') \leftarrow\!\!\$\ \mathsf{A}_2^{\mathsf{O}_2}(\mathsf{pk}^*, c^*, \mathsf{K}^*, \mathsf{st})$	$(\mathsf{pk}', R, c') \leftarrow\!\!\$\ \mathsf{S}(\mathsf{pk}^*)$
return 1 *if* $\exists(\mathsf{K}', r)$ *such that*	**return** 1 *if* $\exists(\mathsf{K}', r)$ *such that*
$((c', \mathsf{K}') = \mathsf{Encaps}(\mathsf{pk}'; r)) \wedge$	$((c', \mathsf{K}') = \mathsf{Encaps}(\mathsf{pk}'; r)) \wedge$
$(\mathsf{pk}' \neq \mathsf{pk} \vee \mathsf{K}' \neq \mathsf{K}^*) \wedge (\mathsf{K}' \neq \perp)$	$(\mathsf{pk}' \neq \mathsf{pk} \vee \mathsf{K}' \neq \mathsf{K}^*) \wedge (\mathsf{K}' \neq \perp)$
$\wedge\ R(\mathsf{K}', \mathsf{K}^*, \mathsf{pk}', \mathsf{pk}^*, c')$	$\wedge\ R(\mathsf{K}', \mathsf{K}^*, \mathsf{pk}', \mathsf{pk}^*, c')$

In the experiments above,

$$\begin{aligned}
&\textit{if } atk = cpa \textit{ then } \mathsf{O}_1 = \epsilon && \textit{and } \mathsf{O}_2 = \epsilon,\\
&\textit{if } atk = cca1 \textit{ then } \mathsf{O}_1 = \mathsf{Decaps}(\mathsf{sk}, \cdot) && \textit{and } \mathsf{O}_2 = \epsilon,\\
&\textit{if } atk = cca2 \textit{ then } \mathsf{O}_1 = \mathsf{Decaps}(\mathsf{sk}, \cdot) && \textit{and } \mathsf{O}_2 = \mathsf{Decaps}^{(c^*)}(\mathsf{sk}, \cdot),
\end{aligned}$$

where $\mathsf{Decaps}^{(c^*)}(\mathsf{sk}, \cdot)$ *means that* A *is allowed to query* Decaps *algorithm for any ciphertext distinct from the challenge ciphertext* c^*.

We will now show the relationship between the indistinguisability-based definitions and simulation-based ones.

In particular, we will show that our indistinguisability-based definition of NM-CPA* security implies the corresponding simulation-based definition, and then argue about the restrictions we need to prove equivalence between NM-CCA1*/NM-CCA2* and the simulation-based counterpart. Conversely, we show that the simulation-based definition NM-ATK* implies the indistinguishability-based definition SNM-ATK* for ATK $\in \{\text{CPA}, \text{CCA1}, \text{CCA2}\}$.

Theorem 2 (NM-CPA* $\implies$ SNM-CPA*). *If a KEM* $\Pi = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ *is NM-CPA* secure (Definition 8), then* Π *is SNM-CPA* secure (Definition 9).*

Proof. We follow a similar approach of Ventre and Visconti [22] to prove that a if PKE is NM-CPA* , then it is also secure for SNM-CPA*. The intuition behind the proof is that, given an adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$, the simulator S simply runs A. In particular, we define S as follows:

Simulator $\mathsf{S}(\mathsf{pk})$:

$\mathsf{st} \leftarrow\!\!\$\ \mathsf{A}_1(\mathsf{pk})$

$(\hat{c}, \widehat{\mathsf{K}}) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk})$

$(\mathsf{pk}', R, c') \leftarrow\!\!\$\ \mathsf{A}_2(\mathsf{pk}, \hat{c}, \widehat{\mathsf{K}}, \mathsf{st})$

return (pk', R, c')

The simulator can directly run the adversary A because A doesn't have access to any oracle, which in turn means that S doesn't need to know the private key corresponding to the challenge public key pk. In order to show that Π is secure in the sense of SNM-CPA*, we first note that the experiment $\mathsf{KEM}^{nm\text{-}cpa*}_{\Pi,\mathsf{A}}(\lambda)$ is indeed equivalent to $\mathsf{KEM}^{snm\text{-}cpa*}_{\Pi,\mathsf{A}}(\lambda)$, and thus,

$$\mathsf{KEM}^{nm\text{-}cpa*}_{\Pi,\mathsf{A}}(\lambda) \equiv \mathsf{KEM}^{snm\text{-}cpa*}_{\Pi,\mathsf{A}}(\lambda).$$

Now, by replacing the simulator in $\mathsf{KEM}^{snm\text{-}cpa*}_{\Pi,\mathsf{S}}$ with the simulator given above, we get that the resulting experiment is equivalent to that of $\mathsf{KEM}^{nm\text{-}cpa*}_{\Pi,\mathsf{A},\$}$. Thus, we get that

$$\mathsf{KEM}^{nm\text{-}cpa*}_{\Pi,\mathsf{A},\$}(\lambda) \equiv \mathsf{KEM}^{snm\text{-}cpa*}_{\Pi,\mathsf{S}}(\lambda).$$

□

Extending to NM-CCA1* and NM-CCA2* security. Unfortunately, the direct approach used to prove Theorem 3 cannot be used to show that the game-based definitions of NM-CCA1* and NM-CCA2* imply the corresponding simulation-based counterpart. A NM-ATK* adversary A, for $ATK \in \{CCA1, CCA2\}$, has access to the decapsulation oracle, and therefore the simulator S in order to perfectly simulate the NM-ATK* experiment, must be able to simulate the queries to such oracle. However, S doesn't have access to any oracle, so the only way for the simulator to simulate such queries is to know the secret key corresponding to the given public key pk. Naturally, this cannot be a realistic assumption, and therefore we let S to generate its own pair of public-private keys $(\hat{\mathsf{pk}}, \hat{\mathsf{sk}})$, and use $\hat{\mathsf{sk}}$ to answer the queries to the decapsulation oracle made by A. This assumption has an effect on the set of relations that can consider. Since the simulator sends its own public key $\hat{\mathsf{pk}}$ to the adversary ignoring the challenge key pk^*, the relations under consideration must be independent of pk^* as well, otherwise A has no way to succeed in the SNM-ATK* experiment without knowing the challenge public key pk^*. As introduced by Ventre and Visconti in [22], we will consider such a special type of relations, called *lacking* relations. More formally, a relation R is lacking if it is a complete relation that ignores the input of the challenge public key pk^*, i.e. R is lacking if and only if, for all $\mathsf{K}^*, \mathsf{K}' \in \mathcal{K}$, all ciphertext c' produced by the Encaps algorithm, all public keys $\mathsf{pk}^*, \mathsf{pk}'$ produced by the key generation algorithm, $R(\mathsf{K}', \mathsf{K}^*, \mathsf{pk}', \cdot, c') = R(\mathsf{K}', \mathsf{K}^*, \mathsf{pk}', \mathsf{pk}^*, c')$.

Theorem 3 (NM-ATK* $\implies$ SNM-ATK*). *Let $\mathcal{R}$ denote the set of lacking relations, and $ATK \in \{CCA1, CCA2\}$. If a KEM scheme $\Pi = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ is NM-ATK* secure (Definition 8) with respect to $\mathcal{R}$ then Π is SNM-ATK* secure with respect to $\mathcal{R}$ (Definition 9).*

Due to its similarity to the proof of Theorem 2, we have provided the sketch of the proof.

Proof. [Sketch] When the simulator S receives the challenge public key pk, generates a new pair of keys and runs an adversary A in the simulation-based experiment, on input the new public key. S computes the encapsulation using the new public key and gives the resulting ciphertext and key to A as the challenge ciphertext and key. Next, the simulator uses the corresponding encapsulation key to answer the queries to the decapsulation oracle made by A. At the end, when the adversary returns a public key pk', a relation R, and a ciphertext c', the simulator just returns (pk', R, c'). Since A is a NM-ATK* adversary, the final ciphertext c' is obtained using a public key that is different from the given one. Furthermore, since we are considering lacking relations, it is straightforward to see that S succeeds whenever A does. □

We will now show that the converse implication holds for $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$.

Theorem 4 (SNM-ATK* $\implies$ NM-ATK*). *If a KEM scheme $\Pi = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ is SNM-ATK* secure (Definition 9) with respect to a set of relations $\mathcal{R}$ then Π is NM-ATK* secure with respect to $\mathcal{R}$ (Definition 8).*

Proof. The converse implication follows directly. Let us consider an adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ of the NM-ATK* experiment as follows.

Adversary $\mathsf{A}_1^{\mathsf{O}_1}(\mathsf{pk}^*)$:	**Adversary** $\mathsf{A}_2^{\mathsf{O}_2}(\mathsf{pk}^*, c^*, \mathsf{K}^*, \mathsf{st})$:
return $\mathsf{st} := \epsilon$	$(\mathsf{pk}', R, c') \leftarrow\!\!\$\ \mathsf{S}(\mathsf{pk}^*)$
	return (pk', R, c')

In order to show that Π is secure in the sense of NM-CPA*, we first note that the experiment $\mathsf{KEM}^{snm\text{-}atk*}_{\Pi,\mathsf{A}}(\lambda)$ is indeed equivalent to $\mathsf{KEM}^{nm\text{-}atk*}_{\Pi,\mathsf{A}}(\lambda)$, and thus,

$$\mathsf{KEM}^{nm\text{-}atk*}_{\Pi,\mathsf{A}}(\lambda) \equiv \mathsf{KEM}^{snm\text{-}atk*}_{\Pi,\mathsf{A}}(\lambda).$$

Now, by replacing the adversary A in $\mathsf{KEM}^{nm\text{-}atk*}_{\Pi,\mathsf{A},\$}(\lambda)$ with the one described above, it straightforward to see that

$$\mathsf{KEM}^{nm\text{-}atk*}_{\Pi,\mathsf{A},\$}(\lambda) \equiv \mathsf{KEM}^{snm\text{-}atk*}_{\Pi,\mathsf{S}}(\lambda).$$

□

4 Analysis of Fujisaki-Okamoto transforms

In the following, we analyze complete non-malleability of the FO transforms. To do that, we will consider the modular treatment of the FO transforms pursued by [14]. Each FO transform is an application of two transformations, namely T and U. T takes as input an IND-CPA/OW-CPA secure PKE scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and a random oracle G, and outputs a deterministic OW-PCVA PKE scheme $\Pi_1 = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ (when Π is OW-CPA, it should also satisfy γ-uniformity for a sufficiently large γ). The encryption algorithm of the transformed PKE runs the encryption algorithm of the underlying IND-CPA secure PKE scheme Π but uses $\mathsf{G}(m)$ as its randomness, i.e. $\mathsf{Enc}_1(\mathsf{pk}, m) := \mathsf{Enc}(\mathsf{pk}, m; \mathsf{G}(m))$. The decryption algorithm runs the decryption algorithm of the underlying PKE scheme, i.e. $m' := \mathsf{Dec}(\mathsf{sk}, c)$, and returns $\perp$ if $m' = \perp$ or the re-encryption of m under public-key pk and randomness $\mathsf{G}(m)$ does not match with c, i.e. if $\mathsf{Enc}(\mathsf{pk}, m'; \mathsf{G}(m')) \neq c$.

Given a PKE scheme satisfying OW-PCVA or some different flavour of one-wayness (depending on the trasformation we are going to use) and a random oracle H, four variant of the transformation U can be used to produce an IND-CCA2 KEM scheme. In fig Fig. 1 we recall the algorithms needed to instantiate the the $\mathsf{U}^{\perp}$ and $\mathsf{U}^{\not\perp}$ transformations. The algorithms for the $\mathsf{U}^{\perp}_m$ and $\mathsf{U}^{\not\perp}_m$ transformations are the same of $\mathsf{U}^{\perp}$ and $\mathsf{U}^{\not\perp}$ respectively, except that the encapsulation algorithm computes the key K as $\mathsf{H}(m)$, and the decapsulation algorithm outputs $\mathsf{K} := \mathsf{H}(m)$ when $m \neq \perp$.

Algorithm $\mathsf{Gen}^{\not\perp}(1^\lambda)$
$(\mathsf{pk}', \mathsf{sk}') \leftarrow\!\!\$\ \mathsf{Gen}_1(1^\lambda)$
$s \leftarrow\!\!\$\ \mathcal{M}$
$\mathsf{sk} := (\mathsf{sk}', s)$
return $(\mathsf{pk}', \mathsf{sk})$

Algorithm $\mathsf{Decaps}^{\perp}(\mathsf{sk}, c)$
$m' := \mathsf{Dec}_1(\mathsf{sk}, c)$
if $m' = \perp$
  return $\perp$
return $\mathsf{K} := \mathsf{H}(m', c)$

Algorithm $\mathsf{Encaps}(\mathsf{pk})$
$m \leftarrow\!\!\$\ \mathcal{M}$
$c \leftarrow\!\!\$\ \mathsf{Enc}_1(\mathsf{pk}, m)$
$\mathsf{K} := \mathsf{H}(m, c)$
return (K, c)

Algorithm $\mathsf{Decaps}^{\not\perp}(\mathsf{sk}, c)$
Parse $\mathsf{sk} = (\mathsf{sk}', s)$
$m' := \mathsf{Dec}_1(\mathsf{sk}', c)$
if $m' = \perp$
  return $\mathsf{K} := \mathsf{H}(m', s)$
return $\mathsf{K} := \mathsf{H}(m', c)$

Figure 1: Algorithms needed by the transformations $\mathsf{U}^{\perp}[\Pi_1, \mathsf{H}] = (\mathsf{Gen}_1, \mathsf{Encaps}, \mathsf{Decaps}^{\perp})$ and $\mathsf{U}^{\not\perp}[\Pi_1, \mathsf{H}] = (\mathsf{Gen}^{\perp}, \mathsf{Encaps}, \mathsf{Decaps}^{\not\perp})$.

In Section 4.1, we will analyze the $\mathsf{U}^{\perp}_m/\mathsf{U}^{\not\perp}_m$ transformations and show that they are not completely non-malleable by giving an attack that can be performed when the underlying OW-PCVA PKE scheme is obtained by applying the transformation T to a widely known IND-CPA PKE scheme.

Then, in Section 4.2, we will analyze the $\mathsf{U}^{\perp}/\mathsf{U}^{\not\perp}$ transformations and show that they are not completely non-malleable by constructing a contrived OW-PCVA PKE scheme that, when given as input to $\mathsf{U}^{\perp}/\mathsf{U}^{\not\perp}$, leads to a KEM whose NM-ATK* security can be easily broken. Then, we show that it suffices to assume a very natural property of the underlying OW-PCVA PKE scheme, named public-key uniqueness, to achieve complete non-malleability.

Finally, in Section 4.3, we show that it is possible to achieve a completely non-malleable KEM without assuming public-key uniqueness of the underlying PKE scheme with a little tweak to $\mathsf{U}^{\perp}/\mathsf{U}^{\not\perp}$.

4.1 Analysis of the $\mathsf{U}_m^{\perp/\not\perp}$ transformations

In the following, we will show that $\mathsf{U}_m^{\perp}$ and $\mathsf{U}_m^{\not\perp}$ transformations lead to a KEM that is not completely non-malleable. In particular, we show a concrete attack that can be carried out to both $\tilde{\Pi}_m^{\perp} := \mathsf{U}_m^{\perp}[\mathsf{T}[\Pi, \mathsf{G}], \mathsf{H}]$ and $\tilde{\Pi}_m^{\not\perp} := \mathsf{U}_m^{\not\perp}[\mathsf{T}[\Pi, \mathsf{G}], \mathsf{H}]$ even against the weaker notion of NM-CPA*. Let Π be the El-Gamal encryption scheme.

When running the experiment $\mathsf{KEM}^{nm\text{-}cpa*}_{\tilde{\Pi}_m^{\perp},\mathsf{A}}(\lambda)$ with $\tilde{\Pi}_m^{\perp}$ (resp. $\tilde{\Pi}_m^{\not\perp}$), an efficient adversary A receives as input a public key $\mathsf{pk}^* = (\mathsf{params}, h)$, the challenge ciphertext c^*, and an encapsulation key K^*. The challenge ciphertext c^* is computed as $(c_1^* = g^r, c_2^* = h^r \cdot m^*)$ with $r = \mathsf{G}(m^*)$, $\mathsf{params} = (\mathbb{G}, g, q)$, where g is the generator of a cyclic group $\mathbb{G}$ of order q, $h = g^x$ for $x \leftarrow\!\!\$\ \mathbb{Z}_q$, and m^* is a randomly sampled message from $\mathbb{G}$. The challenge key K^* is either the output of the encapsulation algorithm on input m^* (i.e., $\mathsf{K}^* = \mathsf{H}(m^*)$), or it is sampled from the uniform distribution over the key space $\mathcal{K}$.

Now, A can craft a new public key $\mathsf{pk}' = (\mathsf{params}, h \cdot g^{x'})$, for $x' \leftarrow\!\!\$\ \mathbb{Z}_q$, and a ciphertext $c' = (c_1^*, c_2^* \cdot {c_1^*}^{x'})$. It is straightforward to see that, since $\mathsf{Enc}(\mathsf{pk}', m') = (g^r, g^{(x+x')r} m^*) = c'$, there exist a key K' such that $(c', \mathsf{K}') = \mathsf{Encaps}(\mathsf{pk}')$, where $c' = \mathsf{Enc}(\mathsf{pk}', m^*)$. Finally, since $\mathsf{K}' := \mathsf{H}(m')$ with $m' = m^*$, then $\mathsf{K}' = \mathsf{K}^*$.

Now, A can define the relation between pk^* and pk' as follows:

$$R_{\mathsf{pk}}(\mathsf{pk}^* = (\mathsf{params}, h), \mathsf{pk}' = (\mathsf{params}, h')) = 1 \text{ iff } h' = h \cdot g^{x'}, x' \in \mathbb{Z}_p.$$

Moreover, A can define the relation between K^* and K' as the identity relation, i.e. $R_{\mathsf{K}}(\mathsf{K}^*, \mathsf{K}') = 1$ iff $\mathsf{K}^* = \mathsf{K}'$. As shown above, A can come up with a $\mathsf{pk}' \neq \mathsf{pk}$ satisfying the relation $R(\mathsf{K}^*, \mathsf{K}', \mathsf{pk}^*, \mathsf{pk}, c') = R_{\mathsf{K}}(\mathsf{K}^*, \mathsf{K}') \wedge R_{\mathsf{pk}}(\mathsf{pk}^*, \mathsf{pk}')$. Such relation, when the ciphertext c' is computed as above (i.e. $c' = (c_1^*, c_2^* \cdot {c_1^*}^{x'})$), leads the $\mathsf{KEM}^{nm\text{-}cpa*}_{\tilde{\Pi}_m^{\perp},\mathsf{A}}(\lambda)$ experiment output 1 and the $\mathsf{KEM}^{nm\text{-}cpa*}_{\tilde{\Pi}_m^{\perp},\mathsf{A},\$}(\lambda)$ output 0 with non-negligible probability.

In the following, we will show that the $\mathsf{U}^{\perp/\not\perp}$ transformations can be turned into a completely non-malleable KEM without much effort.

4.2 Analysis of the $\mathsf{U}^{\perp/\not\perp}$ transformations

The $\mathsf{U}^{\perp}$ and $\mathsf{U}^{\not\perp}$ transformations do not naturally satisfy complete non-malleability. Indeed, given an OW-PCVA PKE scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, we can construct a contrived OW-PCVA PKE scheme $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ that is trivially insecure against the NM-CPA* experiment. The scheme Π' follows:

Algorithm $\mathsf{Gen}'(1^\lambda)$	**Algorithm** $\mathsf{Enc}'(\mathsf{pk}', m)$	**Algorithm** $\mathsf{Dec}'(\mathsf{sk}, m)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$	Parse $\mathsf{pk}' = (\mathsf{pk}, b)$	$m := \mathsf{Dec}(\mathsf{sk}, m)$
$b \leftarrow\!\!\$\ \{0,1\}$	$c \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, m)$	**return** m
$\mathsf{pk}' := \mathsf{pk} \| b$	**return** c	
return $(\mathsf{pk}', \mathsf{sk})$		

It is straightforward to see that the adversary of NM-CPA*, when run with $\tilde{\Pi}^{\perp} := \mathsf{U}^{\perp}[\Pi', \mathsf{H}]$, distinguishes between the two experiments with non-negligible probability. Indeed, an adversary A can come up with a public key pk' different of the public key generated by $\tilde{\Pi}^{\perp}$ that leads to the same ciphertext c^* encapsulating K^* as computed in the $\mathsf{KEM}^{nm\text{-}cpa*}_{\tilde{\Pi}^{\perp},\mathsf{A}}$ experiment. To be precise, the adversary A might define a relation $R(\mathsf{K}^*, \mathsf{K}', \mathsf{pk}^*, \mathsf{pk}, \cdot) = R_{\mathsf{K}}(\mathsf{K}^*, \mathsf{K}') \wedge R_{\mathsf{pk}}(\mathsf{pk}^*, \mathsf{pk}')$, where $R_{\mathsf{K}}(\mathsf{K}^*, \mathsf{K}')$ is the identity function and $R_{\mathsf{pk}}(\mathsf{pk}^*, \mathsf{pk}') = 1$ iff, given that pk^* can be parsed as (pk, b) with $b \in \{0,1\}$, then pk' equals $(\mathsf{pk}, 1-b)$. If c^* is the encryption of a message m^* under pk^*, then it trivially encrypts m^* under pk'. Hence, the key $\mathsf{K}^* := \mathsf{H}(m^*, c^*)$ will be the same.

An idea to avoid such an artificial attack, is to restrict the set of admitted PKE schemes to the ones for which such attack is not possible to carry out in the first place. As we will show, this suffice to guarantee complete non-malleability of $\tilde{\Pi}^{\perp}$. To do that, we introduce a natural property called public-key uniqueness, informally stating that it is infeasible for an adversary to come up with two different public keys leading to the same ciphertext when encrypting the same message. For example, in El-Gamal, the encryption of the same message with two different public keys will always lead to a different ciphertext for any possible random coins and any possible message. This property, which we formalize below, is indeed achieved by most of the known PKE schemes.

Definition 10 (Public-Key Uniqueness)**.** *A public-key encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space* $\mathcal{M}$ *is public-key unique if, for all unbounded/PPT adversary* A*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr\left[\exists r, r', m : \begin{array}{c}\mathsf{Enc}(\mathsf{pk}, m; r) = \mathsf{Enc}(\mathsf{pk}', m; r') \\ \wedge\ (\mathsf{pk} \neq \mathsf{pk}')\end{array} \;\middle|\; (\mathsf{pk}, \mathsf{pk}') \leftarrow\!\!\$\ \mathsf{A}(1^{\lambda})\right] \leq \mathsf{negl}(\lambda).$$

We say that Π *is perfect public-key unique if do not exist two public keys* pk *and* pk' *with* $\mathsf{pk}' \neq \mathsf{pk}$ *such that* $\mathsf{Enc}(\mathsf{pk}, m; r) = \mathsf{Enc}(\mathsf{pk}', m; r')$ *for any* $m \in \mathcal{M}$ *and any randomness* r, r'*.*

In the following, we show that the KEM $\tilde{\Pi}^{\perp} = \mathsf{U}^{\perp}[\Pi_1, \mathsf{H}]$ is completely non-malleable when the PKE Π_1 is OW-PCVA and satisfies perfect public-key uniqueness. Note that our result trivially extends when Π_1 satisfies public-key uniqueness against PPT or unbounded adversaries.

Theorem 5 (Π_1 pk-unique OW-PCVA $\overset{ROM}{\Longrightarrow}$ $\tilde{\Pi}^{\perp}$ NM-CCA2*)**.** *Assuming the existence of a random oracle* H*, if* Π_1 *is a correct OW-PCVA secure PKE (Definition 3) satisfying perfect public-key uniqueness (Definition 10), then* $\tilde{\Pi}^{\perp}$ *defined as above is a correct NM-CCA2* secure KEM (Definition 8).*

Proof. Correctness of $\tilde{\Pi}^{\perp}$ trivially follows from the correctness of Π_1. The idea behind the proof is to simulate the decapsulation oracle without using the secret key. We can do that by answering the decapsulation queries with a random key, and next simulate the random oracle H by using the plaintext checking oracle $\mathsf{PCO}(\mathsf{sk}^*, \cdot, \cdot)$, provided by the OW-PCVA game. Furthermore, we will use the ciphertext validity oracle $\mathsf{CVO}^{(c^*)}(\mathsf{sk}^*, \cdot)$ in order to reject decapsulation queries for invalid ciphertexts. Before proceeding with the proof we need to make the following observations. Since a NM-CCA2* adversary against $\tilde{\Pi}^{\perp}$ is allowed to choose both the public key and the ciphertext, the decapsulation oracle can receive ciphertexts encrypted using a public key $\mathsf{pk}' \neq \mathsf{pk}^*$. However, since the decapsulation oracle does not know the secret key associated to pk', we cannot require to it to check the validity of such ciphertexts. In other words, the decapsulation oracle is only required to check the validity of ciphertexts encrypted using the challenge public key pk^*. A similar observation shall be done for the oracle H. Due to the fact that also H can receive ciphertexts encrypted using any public key distinct from pk^*, we should be able to check the validity of such ciphertexts. In other words, given a message m and a ciphertext c, we should allow H to check whether $m = \mathsf{Dec}_1(\mathsf{sk}', c)$. However, since H doesn't know the secret key associated with pk', this desired behavior cannot be achieved. For this reason, the same approach used for $\mathsf{Decaps}_{\perp}^{(c^*)}(\mathsf{sk}^*, \cdot)$ applies. Let B be a PPT adversary that breaks NM-CCA2* security of $\tilde{\Pi}^{\perp}$ with non-negligible probability issuing a polynomially bounded number of queries to $\mathsf{Decaps}^{\perp}$ and H. The sequence of games we are going to consider is described in Fig. 2.

Lemma 1. $\mathsf{KEM}^{nm\text{-}cca2*}_{\tilde{\Pi}^{\perp},\mathsf{B}}(\lambda) \equiv \mathbf{G}_0^{\mathsf{B}}(\lambda)$.

Proof. Let us start by noticing that in game $\mathbf{G}_0$ the challenger takes a uniform message m^*, computes $c^* \leftarrow\!\!\$\ \mathsf{Enc}_1(\mathsf{pk}^*, m^*)$, $\mathsf{K}^* := \mathsf{H}(m^*, c^*)$ and outputs $(\mathsf{pk}^*, c^*, \mathsf{K}^*)$. This game coincides exactly with the left experiment of the NM-CCA2* definition. Thus, $\mathsf{KEM}^{nm\text{-}cca2*}_{\tilde{\Pi}^{\perp},\mathsf{B}}(\lambda) \equiv \mathbf{G}_0^{\mathsf{B}}(\lambda)$. □

Lemma 2. $\mathbf{G}_0^{\mathsf{B}}(\lambda) \equiv \mathbf{G}_1^{\mathsf{B}}(\lambda)$.

Games $\mathbf{G}_0^{\mathsf{B}}(\lambda) - \mathbf{G}_3^{\mathsf{B}}(\lambda)$

$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}_1(1^\lambda)$

$\mathsf{st} \leftarrow\!\!\$\ \mathsf{B}_1^{\mathsf{Decaps}_\perp(\mathsf{sk}^*,\cdot),\mathsf{H}(\cdot,\cdot)}(\mathsf{pk}^*)$

$m^* \leftarrow\!\!\$\ \mathcal{M}$

$c^* \leftarrow\!\!\$\ \mathsf{Enc}_1(\mathsf{pk}, m^*)$

$\mathsf{K}^* := \mathsf{H}(m^*, c^*)$ $\boxed{\#\mathbf{G}_0, \mathbf{G}_1}$

$\mathsf{K}^* \leftarrow\!\!\$\ \{0,1\}^\lambda$ $\boxed{\#\mathbf{G}_2, \mathbf{G}_3}$

$(\mathsf{pk}', R, c') \leftarrow\!\!\$\ \mathsf{B}_2^{\mathsf{Decaps}_\perp^{(c^*)}(\mathsf{sk}^*,\cdot),\mathsf{H}(\cdot,\cdot)}(\mathsf{pk}^*, c^*, \mathsf{K}^*, \mathsf{st})$

return 1 iff $\exists(\mathsf{K}', r)$ such that

$(c', \mathsf{K}') := \mathsf{Encaps}(\mathsf{pk}'; r) \wedge (\mathsf{K}' \neq \perp)$

$\wedge\ (\mathsf{pk}' \neq \mathsf{pk}^* \vee \mathsf{K}' \neq \mathsf{K}^*)$

$\wedge\ R(\mathsf{K}', \mathsf{K}^*, \mathsf{pk}', \mathsf{pk}^*, c')$

Oracle $\mathsf{Decaps}_\perp^{(c^*)}(\mathsf{sk}^*, c)$ $\#\mathbf{G}_0, \mathbf{G}_3$

$m' := \mathsf{Dec}_1(\mathsf{sk}^*, c)$

if $m' = \perp$

  return $\perp$

return $\mathsf{K} := \mathsf{H}(m', c)$

Oracle $\mathsf{H}(m, c)$

if $\exists \mathsf{K}$ s.t. $(m, c, \mathsf{K}) \in \mathcal{L}_H$

  return K

$\mathsf{K} \leftarrow\!\!\$\ \mathcal{K}$

if $\mathsf{Dec}_1(\mathsf{sk}^*, c) = m$ $\boxed{\#\mathbf{G}_1, \mathbf{G}_2}$

  if $c = c^*$ $\boxed{\#\mathbf{G}_2}$

    return $\perp$ $\boxed{\#\mathbf{G}_2}$

  if $\exists \mathsf{K}'$ s.t. $(c, \mathsf{K}') \in \mathcal{L}_D$ $\boxed{\#\mathbf{G}_1}$

    $\mathsf{K} := \mathsf{K}'$ $\boxed{\#\mathbf{G}_1}$

  else $\boxed{\#\mathbf{G}_1}$

    $\mathcal{L}_D := \mathcal{L}_D \cup \{(c, \mathsf{K})\}$ $\boxed{\#\mathbf{G}_1}$

$\mathcal{L}_H := \mathcal{L}_H \cup \{(m, c, \mathsf{K})\}$

return K

Oracle $\mathsf{Decaps}_\perp^{(c^*)}(\mathsf{sk}^*, c)$ $\#\mathbf{G}_1, \mathbf{G}_2$

if $\exists \mathsf{K}$ s.t. $(c, \mathsf{K}) \in \mathcal{L}_D$

  return K

$\mathsf{K} \leftarrow\!\!\$\ \mathcal{K}$

$m' := \mathsf{Dec}_1(\mathsf{sk}^*, c)$

if $m' = \perp$

  $\mathsf{K} := \perp$

$\mathcal{L}_D := \mathcal{L}_D \cup \{(c, \mathsf{K})\}$

$\mathcal{L}_H := \mathcal{L}_H \cup \{(m', c, \mathsf{K})\}$

return K

Figure 2: Sequence of games needed to prove Theorem 5 and the consequential oracle modifications.

Proof. Differently from $\mathbf{G}_0$, in game $\mathbf{G}_1$ we have modified the oracles $\mathsf{Decaps}_{\perp}^{(c^*)}$ and H in order to avoid the usage of the secret key. In particular, $\mathbf{G}_1$ defines two sets $\mathcal{L}_H$ and $\mathcal{L}_D$, where $\mathcal{L}_H$ contains all entries of the form (m, c, K) when either $\mathsf{Decaps}_{\perp}^{(c^*)}$ is queried about a ciphertext c or H was queried about (m, c), and $\mathcal{L}_D$ contains all the entries (c, K) when either $\mathsf{Decaps}_{\perp}^{(c^*)}$ is queried about c or H is queried about (m, c) for $m = \mathsf{Dec}_1(\mathsf{sk}', c)$. Now, we want to show that the view of B in $\mathbf{G}_0$ and $\mathbf{G}_1$ is distributed exactly in the same manner. For this purpose, let us consider a ciphertext c' and a message $m' = \mathsf{Dec}_1(\mathsf{sk}', c')$ for which B has never been queried $\mathsf{Decaps}_{\perp}^{(c^*)}$:

- **Case** $m' = \perp$: in game $\mathbf{G}_0$, $\mathsf{Dec}_1(\mathsf{sk}', c')$ will return $\perp$ to indicate that c' is a malformed ciphertext, which is exactly the behavior the $\mathsf{Decaps}_{\perp}^{(c^*)}$ has in game $\mathbf{G}_1$. Regarding the behavior of H, we can see that in both games H returns a randomly chosen key.
- **Case** $m' \neq \perp$: in game $\mathbf{G}_0$, $\mathsf{Decaps}_{\perp}^{(c^*)}$ returns $K := \mathsf{H}(m', c')$ which is either a fresh key randomly chosen from the key space if (m', c') has never been queried to H, or taken from $\mathcal{L}_H$ if (m', c') was already stored in $\mathcal{L}_H$. In game $\mathbf{G}_1$ we need to consider two sub-cases:
  - B first queries H about (m', c') and then queries $\mathsf{Decaps}_{\perp}^{(c^*)}$ about c': In this case, H returns a key K which is either a fresh key randomly chosen from the key space, or it is already stored in $\mathcal{L}_H$. Since $\mathsf{Decaps}_{\perp}^{(c^*)}$ has not been queried about c' yet, H will add (c, K) to $\mathcal{L}_D$. Next, when $\mathsf{Decaps}_{\perp}^{(c^*)}$ will be queried about c', it will return a key K stored in $\mathcal{L}_D$ that coincides with the key stored in $\mathcal{L}_H$, as in game $\mathbf{G}_0$.
  - $\mathcal{B}$ first queries $\mathsf{Decaps}_{\perp}^{(c^*)}$ about c' and then queries H about (m', c'): In this case, $\mathsf{Decaps}_{\perp}^{(c^*)}$ returns a randomly chosen K, which is added to both $\mathcal{L}_D$ and $\mathcal{L}_H$. Subsequently, when H is queried about (m', c'), the oracle will return the key K stored in $\mathcal{L}_H$ that coincides with the key K stored in $\mathcal{L}_D$. This ensures that $\mathsf{Decaps}_{\perp}^{(c^*)}(c') = \mathsf{H}(m', c') = \mathsf{K}$, as in game $\mathbf{G}_0$.

Therefore, we have that $\mathbf{G}_1^{\mathsf{B}}(\lambda) \equiv \mathbf{G}_0^{\mathsf{B}}(\lambda)$ □

Lemma 3. $\mathbf{G}_1^{\mathsf{B}}(\lambda) \approx_c \mathbf{G}_2^{\mathsf{B}}(\lambda)$.

Proof. In game $\mathbf{G}_2$ we make the following two modifications. First, the challenger takes a uniformly sampled key rather than the real key output by the oracle H, and second we make the oracle H output $\perp$ when queried about (m^*, c^*). By denoting the latter event QUERY, we notice that $\mathbf{G}_1$ and $\mathbf{G}_2$ are identically distributed conditioned to the event QUERY not happening. Thus, the only hope for the adversary to distinguish between the $\mathbf{G}_1$ and $\mathbf{G}_2$ is to trigger the QUERY event in $\mathbf{G}_2$. It is easy to see that when B queries H in game $\mathbf{G}_1$ about (m^*, c^*), H will return the challenge encapsulation key K^*. Instead, in game $\mathbf{G}_2$, the game returns $\perp$ when B queries H about (m^*, c^*). Let us now assume that QUERY is not triggered. Since K^* is either an output of H or a randomly chosen value, the adversary can only try to distinguish by guessing the plaintext m^* of c^*, calculate $\mathsf{K}' := \mathsf{H}(m^*, c^*)$ and then check whether K' is equal to K^*. However, this coincides with the QUERY event. Alternatively, the adversary might try to distinguish by making $\mathbf{G}_1$ always output 1, i.e. B may try to come up with a tuple (pk', R, c') for which the relation $R(\mathsf{K}', \mathsf{K}^*, \mathsf{pk}', \mathsf{pk}^*, c')$ holds for a key K^* produced by the encapsulation algorithm under pk' (as in $\mathbf{G}_1$), but does not hold when such key is randomly chosen (as in $\mathbf{G}_2$). Note that H is a random oracle, so the set of possible relations is restricted to the ones having $\mathsf{K}' = \mathsf{K}^* = \mathsf{H}(m^*, c^*)$, otherwise even a single bit different than (m^*, c^*) in the input of H leads to an independent output. For such a relation the adversary may try to find a public key pk' for which the encapsulation of K^* under pk' leads to c^*. However, for the perfect public-key uniqueness property of Π_1, such public key does not exist. Since K' can be computed only by giving as input to H a pair (m', c') in which either $m' \neq m^*$ or $c' \neq c^*$, such a key is independent from the challenge key. Hence, the distribution of the adversary's view in both games is identical given that QUERY does not happen.
Now, to estimate $\Pr[\mathsf{QUERY}]$ we construct an efficient adversary A breaking OW-PCVA of Π_1 when QUERY occurs. In particular, we define A in Fig. 3. Notice that A perfectly simulates $\mathbf{G}_1$. Indeed, the occurrence

Adversary $\mathsf{A}^{\mathsf{PCO}(\mathsf{sk}^*,\cdot,\cdot),\mathsf{CVO}^{(c^*)}(\mathsf{sk}^*,\cdot)}(\mathsf{pk}, c^*)$

$\mathsf{K}^* \leftarrow\!\!\$\ \mathcal{K}$

$(\mathsf{pk}', R, c') \leftarrow\!\!\$\ \mathsf{B}_2^{\mathsf{Decaps}_\perp^{(c^*)}(\cdot),\mathsf{H}(\cdot,\cdot)}(\mathsf{pk}^*, c^*, \mathsf{K}^*)$
if $\exists (m, c^*) \in \mathcal{L}_H$ s.t. $\mathsf{PCO}(\mathsf{sk}^*, m, c^*) = 1$
  return m
return $\perp$

Oracle $\mathsf{Decaps}_\perp^{(c^*)}(\mathsf{sk}, c)$

if $\exists c$ s.t. $(c, \mathsf{K}) \in \mathcal{L}_D$
  return K
if $\mathsf{CVO}^{(c^*)}(\mathsf{sk}^*, c) = 0$
  return $\perp$
$\mathsf{K} \leftarrow\!\!\$\ \mathcal{K}$
$\mathcal{L}_D := \mathcal{L}_D \cup \{(c, \mathsf{K})\}$
return K

Oracle $\mathsf{H}(m, c)$

if $\exists \mathsf{K}$ s.t. $(m, c, \mathsf{K}) \in \mathcal{L}_H$
  return K
$\mathsf{K} \leftarrow\!\!\$\ \mathcal{K}$
if $\mathsf{PCO}(\mathsf{sk}^*, m, c) = 1$
  if $\exists \mathsf{K}'$ s.t. $(c, \mathsf{K}') \in \mathcal{L}_D$
    $\mathsf{K} := \mathsf{K}'$
  else
    $\mathcal{L}_D := \mathcal{L}_D \cup \{(c, \mathsf{K})\}$
$\mathcal{L}_H := \mathcal{L}_H \cup \{(m, c)\}$
  return K

Figure 3: Adversary A breaking security of the underlying OW-PCVA PKE.

of QUERY implies that B has queried H about (m, c), in which $(m, c) \in \mathcal{L}_H$ for $m = m^*$ and $c = c^*$. A will return $m = m^*$ and win the OW-PCVA experiment. Since such condition coincides with the QUERY event, we get that the probability of B of triggering QUERY coincides with the probability of A in winning the OW-PCVA experiment, i.e.

$$\Pr[\mathsf{QUERY}] = \Pr\left[\mathsf{PKE}^{ow\text{-}pcva}_{\Pi_1,\mathsf{A}}(\lambda)\right].$$

□

Lemma 4. $\mathbf{G}_2^{\mathsf{B}}(\lambda) \approx_c \mathbf{G}_3^{\mathsf{B}}(\lambda)$.

Proof. Notice that game $\mathbf{G}_3$ is identically distributed to $\mathbf{G}_0$, except that the challenge encapsulation key is randomly chosen from the key space. We will show that the view of B in $\mathbf{G}_2$ and $\mathbf{G}_3$ is identically distributed, under the condition that the QUERY event does not occur.

Let us fix a ciphertext c' and a message $m' = \mathsf{Dec}_2(\mathsf{sk}^*, c')$. We consider two cases:

- **Case** $m' \notin \mathcal{M}$: in game $\mathbf{G}_3$, when $\mathsf{Decaps}_\perp^{(c^*)}$ is queried about c', it will return $\perp$, which is exactly what $\mathsf{Decaps}_\perp^{(c^*)}$ returns in $\mathbf{G}_2$.
- **Case** $m' \in \mathcal{M}$. Here, we need to consider two sub-cases:
  - B first queries $\mathsf{Decaps}_\perp^{(c^*)}$ and then H: assume that neither $\mathsf{Decaps}_\perp^{(c^*)}$ nor H have been queried before about c' and (m', c') respectively. In $\mathbf{G}_2$, when B queries $\mathsf{Decaps}_\perp^{(c^*)}$, $\mathsf{Decaps}_\perp^{(c^*)}$ will return a uniform key K, add an entry of the form (c', K) to $\mathcal{L}_D$ and an entry of the form (m', c', K) to $\mathcal{L}_H$. Next, when B queries H about (m', c', pk'), H will return the same key returned by $\mathsf{Decaps}_\perp^{(c^*)}$ since $(m', c', \mathsf{K}) \in \mathcal{L}_H$. In $\mathbf{G}_3$, $\mathsf{Decaps}_\perp^{(c^*)}$ will return a uniform key $\mathsf{K} = \mathsf{H}(m', c')$. Next, when B queries H about (m', c'), H will return the same key K. Therefore, in both games we have $\mathsf{Decaps}_\perp^{(c^*)} = \mathsf{K} = \mathsf{H}(m', c')$.
  - B first queries H and then $\mathsf{Decaps}_\perp^{(c^*)}$: As before, let us assume that neither $\mathsf{Decaps}_\perp^{(c^*)}$ nor H have been queried before about c' and (m', c') respectively. In $\mathbf{G}_2$, when B queries H, H will return

a uniform key K and add an entry of the form (c', K) to $\mathcal{L}_D$. Next, when $\mathsf{Decaps}_{\perp}^{(c^*)}$ is queried about c', $\mathsf{Decaps}_{\perp}^{(c^*)}$ will return the same key returned by H, since $(c', \mathsf{K}) \in \mathcal{L}_D$. In game $\mathbf{G}_3$ when B queries H, H will return a uniform key K and add an entry of the form (m', c', K) to $\mathcal{L}_H$. Next, when $\mathsf{Decaps}_{\perp}^{(c^*)}$ is queried about c', $\mathsf{Decaps}_{\perp}^{(c^*)}$ will return the same key K, due to the fact that $(m', c', \mathsf{K}) \in \mathcal{L}_H$. Thus, in both games we have $\mathsf{H}(m', c') = \mathsf{K} = \mathsf{Decaps}_{\perp}^{(c^*)}$.

Since the only hope for the adversary is to trigger the QUERY event in $\mathbf{G}_3$, the probability for B to distinguish between $\mathbf{G}_2$ and $\mathbf{G}_3$ is bounded by the probability of winning the OW-PCVA game of the underlying PKE scheme. □

Lemma 5. $\mathbf{G}_3^{\mathsf{B}}(\lambda) \equiv \mathsf{KEM}_{\tilde{\Pi}_1, \mathsf{B}, \$}^{nm\text{-}cca2*}(\lambda)$.

Proof. Since $\mathbf{G}_3$ is similar to $\mathbf{G}_0$, with the only difference that the encapsulation key K^* is uniform and independent from the one obtained by querying H, for the same considerations that we did for $\mathbf{G}_0$, it holds that the two distributions are identically distributed. □

Combining the above lemmas, we get that $\mathsf{KEM}_{\tilde{\Pi}^{\perp}, \mathsf{B}}^{nm\text{-}cca2*}(\lambda) \equiv \mathbf{G}_0^{\mathsf{B}}(\lambda) \equiv \mathbf{G}_1^{\mathsf{B}}(\lambda) \approx_c \mathbf{G}_2^{\mathsf{B}}(\lambda) \approx_c \mathbf{G}_3^{\mathsf{B}}(\lambda) \equiv \mathsf{KEM}_{\tilde{\Pi}_1^{\perp}, \mathsf{B}, \$}^{nm\text{-}cca2*}(\lambda)$, thus

$$\mathsf{KEM}_{\tilde{\Pi}_1^{\perp}, \mathsf{B}}^{nm\text{-}cca2*}(\lambda) \approx_c \mathsf{KEM}_{\tilde{\Pi}_1^{\perp}, \mathsf{B}, \$}^{nm\text{-}cca2*}(\lambda).$$

□

4.3 Modified transformation $\hat{\mathsf{U}}^{\perp}$

In the following, we leverage the idea of prefix hashing introduced by Duman et al. [10], to construct a completely non-malleable KEM without requiring public-key uniqueness of the underlying PKE. In particular, our $\hat{\mathsf{U}}^{\perp}$ is identical to $\mathsf{U}^{\perp}$, except that now the encapsulation algorithm gives as input to the random oracle H also the public key pk together with the message m and the ciphertext c from the underlying PKE scheme, i.e. $\mathsf{K} := \mathsf{H}(m, c, \mathsf{pk})$. Note that now the decapsulation oracle must take as input also the challenge public key together with the challenge secret key in order to recompute $\mathsf{H}(m, c, \mathsf{pk})$. The theorem below states that $\tilde{\Pi}_1^{\perp} := \hat{\mathsf{U}}^{\perp}[\Pi_1, \mathsf{H}]$ is completely non-malleable. Since the techniques used to prove the theorem below are similar to the ones used to prove Theorem 5, we will only highlight the changes in the sequence of games w.r.t. the proof of Theorem 5, and the changes needed to prove some lemma when required.

Theorem 6 (Π_1 OW-PCVA $\overset{ROM}{\Longrightarrow}$ $\tilde{\Pi}_1^{\perp}$ NM-CCA2*)**.** *Assuming the existence of a random oracle* H*, if* Π_1 *is a correct OW-PCVA secure PKE (Definition 3), then* $\tilde{\Pi}_1^{\perp}$ *defined above is a correct NM-ATK* secure KEM (Definition 8).*

Proof. The sequence of games and the consequential differences in the oracles are described in Fig. 4.

Lemma 6. $\mathsf{KEM}_{\tilde{\Pi}_1^{\perp}, \mathsf{B}}^{nm\text{-}cca2*}(\lambda) \equiv \mathbf{G}_0^{\mathsf{B}}(\lambda)$.

Proof. The proof is identical to the one of Lemma 1. □

Lemma 7. $\mathbf{G}_0^{\mathsf{B}}(\lambda) \equiv \mathbf{G}_1^{\mathsf{B}}(\lambda)$.

Proof. The proof is identical to the one of Lemma 2. □

Lemma 8. $\mathbf{G}_1^{\mathsf{B}}(\lambda) \approx_c \mathbf{G}_2^{\mathsf{B}}(\lambda)$.

Games $\mathbf{G}_0^{\mathsf{B}}(\lambda) - \mathbf{G}_3^{\mathsf{B}}(\lambda)$

$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}_1(1^\lambda)$

$\mathsf{st} \leftarrow\!\!\$\ \mathsf{B}_1^{\mathsf{Decaps}_\perp^{(c^*)}(\mathsf{sk}^*,\cdot),\mathsf{H}(\cdot,\cdot,\cdot)}(\mathsf{pk}^*)$

$m^* \leftarrow\!\!\$\ \mathcal{M}$

$c^* \leftarrow\!\!\$\ \mathsf{Enc}_1(\mathsf{pk}, m^*)$

$\mathsf{K}^* := \mathsf{H}(m^*, c^*, \mathsf{pk}^*)$ $\boxed{\#\mathbf{G}_0, \mathbf{G}_1}$

$\mathsf{K}^* \leftarrow\!\!\$\ \{0,1\}^\lambda$ $\boxed{\#\mathbf{G}_2, \mathbf{G}_3}$

$(\mathsf{pk}', R, c') \leftarrow\!\!\$\ \mathsf{B}_2^{\mathsf{Decaps}_\perp^{(c^*)}(\mathsf{sk}^*,\cdot),\mathsf{H}(\cdot,\cdot,\cdot)}(\mathsf{pk}^*, c^*, \mathsf{K}^*, \mathsf{st})$

return 1 iff $\exists(\mathsf{K}', r)$ such that

$(c', \mathsf{K}') := \mathsf{Encaps}(\mathsf{pk}'; r) \wedge (\mathsf{K}' \neq \perp)$

$\wedge\ (\mathsf{pk}' \neq \mathsf{pk}^* \vee \mathsf{K}' \neq \mathsf{K}^*)$

$\wedge\ R(\mathsf{K}', \mathsf{K}^*, \mathsf{pk}', \mathsf{pk}^*)$

Oracle $\mathsf{Decaps}_\perp^{(c^*)}(\mathsf{sk}^*, \mathsf{pk}^*, c)$ $\#\mathbf{G}_0, \mathbf{G}_3$

$m := \mathsf{Dec}_1(\mathsf{sk}^*, c)$

if $m' = \perp$

  return $\perp$

return $\mathsf{K} := \mathsf{H}(m', c, \mathsf{pk}^*)$

Oracle $\mathsf{H}(m, c, \mathsf{pk})$

if $\exists \mathsf{K}$ s.t. $(m, c, \mathsf{pk}, \mathsf{K}) \in \mathcal{L}_H$

  return K

$\mathsf{K} \leftarrow\!\!\$\ \mathcal{K}$

if $\mathsf{Dec}_1(\mathsf{sk}^*, c) = m$ $\boxed{\#\mathbf{G}_1, \mathbf{G}_2}$

  if $c = c^*$

    return $\perp$ $\boxed{\#\mathbf{G}_2}$

  if $\exists \mathsf{K}'$ s.t. $(c, \mathsf{K}') \in \mathcal{L}_D$ $\boxed{\#\mathbf{G}_1}$

    $\mathsf{K} := \mathsf{K}'$ $\boxed{\#\mathbf{G}_1}$

  else $\boxed{\#\mathbf{G}_1}$

    $\mathcal{L}_D := \mathcal{L}_D \cup \{(c, \mathsf{K})\}$ $\boxed{\#\mathbf{G}_1}$

$\mathcal{L}_H := \mathcal{L}_H \cup \{(m, c, \mathsf{pk}, \mathsf{K})\}$

return K

Oracle $\mathsf{Decaps}_\perp^{(c^*)}(\mathsf{sk}^*, \mathsf{pk}^*, c)$ $\#\mathbf{G}_1, \mathbf{G}_2$

if $\exists \mathsf{K}$ s.t. $(c, \mathsf{K}) \in \mathcal{L}_D$

  return K

$\mathsf{K} \leftarrow\!\!\$\ \mathcal{K}$

$m' := \mathsf{Dec}_1(\mathsf{sk}^*, c)$

if $m' = \perp$

  $\mathsf{K} := \perp$

$\mathcal{L}_D := \mathcal{L}_D \cup \{(c, \mathsf{K})\}$

$\mathcal{L}_H := \mathcal{L}_H \cup \{(m', c, \mathsf{pk}^*, \mathsf{K})\}$

return K

Figure 4: Sequence of games needed to prove Theorem 6 and the consequential modifications of the oracles.

Proof. The proof is identical to Lemma 3, except that now we do not use public key uniqueness of the underlying PKE scheme to argue that the $\mathbf{G}_1$ and $\mathbf{G}_2$ are identically distributed conditioned to the fact that the event QUERY does not happen. In this case, the adversary can try to distinguish between $\mathbf{G}_1$ and $\mathbf{G}_2$ by guessing the plaintext m^* of c^*, calculate $\mathsf{K}' := \mathsf{H}(m^*, c^*, \mathsf{pk}^*)$ and then check whether K' is equal to K^* or not. However, this coincides with the QUERY event. Alternatively, the adversary might try to distinguish making $\mathbf{G}_1$ always output 1, i.e. B tries to come up with a tuple (pk', R, c') for which the relation $R(\mathsf{K}', \mathsf{K}^*, \mathsf{pk}', \mathsf{pk}^*, c')$ holds for a key K^* encapsulated by c^* under pk' (as in $\mathbf{G}_1$), but does not hold when such key is randomly chosen (as in $\mathbf{G}_2$). However, for the random oracle assumption, since pk' is part of the input of H, the key $\mathsf{K}' := \mathsf{H}(m^*, c^*, \mathsf{pk}')$ will be independent from $\mathsf{K}^* := \mathsf{H}(m^*, c^*, \mathsf{pk}^*)$. Thus, the distribution of $\mathbf{G}_1$ and $\mathbf{G}_2$ is identical when QUERY is not triggered.

As in Lemma 3, to estimate $\Pr[\mathsf{QUERY}]$ we construct an efficient adversary A breaking OW-PCVA of PKE_1 when QUERY occurs. We define A in Fig. 5. Notice that A perfectly simulates $\mathbf{G}_1$. Indeed, the

Adversary $\mathsf{A}^{\mathsf{PCO}(\mathsf{sk}^*,\cdot,\cdot),\mathsf{CVO}^{(c^*)}(\mathsf{sk}^*,\cdot)}(\mathsf{pk}^*, c^*)$

$\mathsf{K}^* \leftarrow\!\!\$\ \mathcal{K}$

$(\mathsf{pk}', R, c') \leftarrow\!\!\$\ \mathsf{B}_2^{\mathsf{Decaps}_{\perp}^{(c^*)}(\cdot),\mathsf{H}(\cdot,\cdot,\cdot)}(\mathsf{pk}^*, c^*, \mathsf{K}^*)$

if $\exists (m, c^*, \mathsf{pk}^*) \in \mathcal{L}_H$ s.t. $\mathsf{PCO}(\mathsf{sk}^*, m, c^*) = 1$

  return m

return $\perp$

Oracle $\mathsf{Decaps}_{\perp}^{(c^*)}(c)$

if $\exists (c, \mathsf{K}) \in \mathcal{L}_D$

  return K

if $\mathsf{CVO}^{(c^*)}(\mathsf{sk}^*, c) = 0$

  return $\perp$

$\mathsf{K} \leftarrow\!\!\$\ \mathcal{K}$

$\mathcal{L}_D := \mathcal{L}_D \cup \{(c, \mathsf{K})\}$

return K

Oracle $\mathsf{H}(m, c, \mathsf{pk})$

if $\exists \mathsf{K}$ s.t. $(m, c, \mathsf{pk}, \mathsf{K}) \in \mathcal{L}_H$

  return K

$\mathsf{K} \leftarrow\!\!\$\ \mathcal{K}$

if $\mathsf{PCO}(\mathsf{sk}^*, m, c) = 1$

  if $\exists \mathsf{K}'$ s.t. $(c, \mathsf{K}') \in \mathcal{L}_D$

    $\mathsf{K} := \mathsf{K}'$

  else

    $\mathcal{L}_D := \mathcal{L}_D \cup \{(c, \mathsf{K})\}$

$\mathcal{L}_H := \mathcal{L}_H \cup \{(m, c, \mathsf{pk}, \mathsf{K})\}$

return K

Figure 5: Adversary A breaking security of the underlying OW-PCVA PKE.

occurrence of QUERY implies that B has queried H about (m, c, pk), in which $(m, c, \mathsf{pk}) \in \mathcal{L}_H$ for $m = m^*$, $c = c^*$ and $\mathsf{pk} = \mathsf{pk}^*$. A then returns $m = m^*$. Since such event coincides with QUERY, we get that the probability of B of triggering QUERY coincides with the probability of A in winning the OW-PCVA experiment, i.e.

$$\Pr[\mathsf{QUERY}] = \Pr\left[\mathsf{PKE}_{\Pi_1,\mathsf{A}}^{ow\text{-}pcva}(\lambda)\right] \leq \mathsf{negl}(\lambda).$$

□

Lemma 9. $\mathbf{G}_2^{\mathsf{B}}(\lambda) \approx_c \mathbf{G}_3^{\mathsf{B}}(\lambda)$.

Proof. The proof is identical to the one of Lemma 4. □

Lemma 10. $\mathbf{G}_3^{\mathsf{B}}(\lambda) \equiv \mathsf{KEM}_{\bar{\Pi}_1,\mathsf{B},\S}^{nm\text{-}cca2*}(\lambda)$.

Proof. Since $\mathbf{G}_3$ is similar to $\mathbf{G}_0$, with the only difference that the encapsulation key K^* is uniform it is independent from the one obtained by querying H, for the same considerations that we did for $\mathbf{G}_0$, it holds that the two distributions are identically distributed. □

Combining the above lemmas, we get that $\mathsf{KEM}^{nm\text{-}cca2*}_{\tilde{\Pi}_1^{\perp},\mathsf{B}}(\lambda) \equiv \mathbf{G}^{\mathsf{B}}_0(\lambda) \equiv \mathbf{G}^{\mathsf{B}}_1(\lambda) \approx_c \mathbf{G}^{\mathsf{B}}_2(\lambda) \approx_c \mathbf{G}^{\mathsf{B}}_3(\lambda) \equiv \mathsf{KEM}^{nm\text{-}cca2*}_{\tilde{\Pi}_1^{\perp},\mathsf{B},\$}(\lambda)$, thus

$$\mathsf{KEM}^{nm\text{-}cca2*}_{\tilde{\Pi}_1^{\perp},\mathsf{B}}(\lambda) \approx_c \mathsf{KEM}^{nm\text{-}cca2*}_{\tilde{\Pi}_1^{\perp},\mathsf{B},\$}(\lambda).$$

□

5 Relation with Completely Non-Malleable PKE

In the following, we will show which kind of relationships exist between a NM-ATK* PKE schemes and NM-ATK* KEM schemes. We will proceed by first showing that a NM-ATK* PKE scheme is intrinsically a NM-ATK* KEM scheme, and next we will introduce a construction to prove that a NM-ATK* KEM can be used together with a NM-ATK SKE scheme to construct a NM-ATK* PKE scheme by using the KEM/DEM paradigm.

5.1 NM-ATK* PKE $\implies$ NM-ATK* KEM

Wlog, we can assume that $\mathcal{M} = \mathcal{K}$. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme, and $\Pi' = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ be a KEM scheme defined as follows:

Algorithm $\mathsf{Encaps}(\mathsf{pk})$	**Algorithm** $\mathsf{Decaps}(\mathsf{sk}, c)$
$\mathsf{K} \leftarrow\!\!\$\ \mathcal{K}$	$\mathsf{K} := \mathsf{Dec}(\mathsf{sk}, c)$
$c \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, \mathsf{K})$	
return (c, K)	**return** K

Theorem 7 (NM-ATK* PKE $\implies$ NM-ATK* KEM). *If Π is a NM-ATK* secure PKE (Definition 4) with respect to a set of relations $\mathcal{R}$, then Π' is a NM-ATK* secure KEM (Definition 8) with respect to $\mathcal{R}$.*

Proof. Let us assume that there exists a PPT adversary A that breaks NM-ATK* security of Π' with non-negligible probability, then we can build an efficient distinguisher D that breaks NM-ATK* security of Π. The intuition behind the proof is that D will choose a distribution over the message space in such a way that only two messages, say K and K', can be sampled by the the challenger playing NM-ATK* of the PKE scheme Π. In particular, D does the following:

1. Takes as input a public key pk and chooses a message distribution $\mathcal{M}$ from which only two messages can be chosen, which we call K and K'.
2. Takes as input a challenge ciphertext c which is either an encryption of K under pk or an encryption of K' under pk.
3. Run $\mathsf{A}(\mathsf{pk}, c, \mathsf{K})$. When A asks a decapsulation-oracle query for a ciphertext $\hat{c}$ do the following:
   (a) Query the decryption oracle $\mathsf{Dec}(\mathsf{sk}^*, \cdot)$ about $\hat{c}$ to obtain a key $\hat{\mathsf{K}}$.
   (b) Return $\hat{\mathsf{K}}$ to A.
4. When A returns (pk', R, c'), return (pk', R, c') as well.

Let us analyze the behavior of D. First, notice that, independently from c, D will always send the tuple $(\mathsf{pk}, c, \mathsf{K})$ to A. Recall also that the message space of c is restricted to messages K and K'. In particular:

- when c is an encryption of K under pk, D the view of A when run as a subroutine of D is identically distributed to its view in $\mathsf{KEM}^{nm\text{-}atk*}_{\Pi',\mathsf{A}}(\lambda)$.

- When c is an encryption of K' under pk, since K' is uniform and independent from K, the view of A when run as a subroutine of D is distributed identically to its view in $\mathsf{KEM}^{nm\text{-}atk*}_{\Pi',\mathsf{A},\$}(\lambda)$.

To summarize, the probability that D distinguishes between $\mathsf{PKE}^{nm\text{-}atk*}_{\Pi,\mathsf{D}}(\lambda)$ and $\mathsf{PKE}^{nm\text{-}atk*}_{\Pi,\mathsf{D},\$}(\lambda)$ is the same of A distinguishing between $\mathsf{KEM}^{nm\text{-}atk*}_{\Pi',\mathsf{A},\$}(\lambda)$ and $\mathsf{KEM}^{nm\text{-}atk*}_{\Pi',\mathsf{A},\$}(\lambda)$, that we assumed to be non-negligible. This leads to a contradiction due to the NM-ATK* security of Π. □

5.2 NM-ATK* KEM + NM-ATK SKE $\Longrightarrow$ NM-ATK* PKE

It is well known that a KEM scheme alone doesn't allow to build a PKE scheme, due to the fact that an encapsulation algorithm can be instantiated by encrypting a uniformly chosen message. To solve this issue, a secret-key encryption scheme can be used along with a key-encapsulation mechanism.

Let $\Pi^{kem} = (\mathsf{Gen}^{kem}, \mathsf{Encaps}, \mathsf{Decaps})$ be a NM-ATK* KEM with key space $\mathcal{K}$, and $\Pi^{ske} = (\mathsf{Gen}^{ske}, \mathsf{Enc}, \mathsf{Dec})$ be a NM-ATK SKE scheme with message space $\mathcal{M}$ and the same key space $\mathcal{K}$ of Π^{kem}, we can construct a hybrid PKE scheme $\Pi^{hy} = (\mathsf{Gen}^{hy}, \mathsf{Enc}^{hy}, \mathsf{Dec}^{hy})$ as defined below.

Algorithm $\mathsf{Gen}^{hy}(1^\lambda)$	**Algorithm** $\mathsf{Enc}^{hy}(\mathsf{pk}, m)$	**Algorithm** $\mathsf{Dec}^{hy}(\mathsf{sk}, c)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}^{kem}(1^\lambda)$	$(c, \mathsf{K}) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk})$	$\mathsf{K} := \mathsf{Decaps}(\mathsf{sk}, c)$
return $(\mathsf{pk}, \mathsf{sk})$	$c' \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{K}, m)$	**return** $m := \mathsf{Dec}(\mathsf{K}, c)$
	return (c, c')	

We prove that if Π^{kem} is a completely non-malleable KEM and Π^{ske} is a non-malleable SKE, then Π^{hy} is a completely non-malleable PKE.

Theorem 8. *If Π^{kem} is an NM-ATK* secure KEM (Definition 8) with respect to a set of relations $\mathcal{R}$ and Π^{ske} is an NM-ATK secure SKE (Definition 6) with respect to a set of relations $\mathcal{R}' \subseteq \mathcal{R}$, then the scheme Π^{hy} described above is a NM-ATK* secure PKE scheme (Definition 4) with respect to $\mathcal{R}$.*

Proof. Correctness of the obtained PKE follows from the ϵ-correctness of the underlying KEM and SKE schemes. The idea behind the proof is that, given the challenge ciphertext $c^* = (c_1^*, c_2^*)$, we can use NM-ATK* security of Π^{kem} to decouple the key encapsulated in c_1^* from the key used in c_2^* to encrypt the message with the underlying SKE scheme Π^{ske}. At this point, since the encapsulated key is randomly chosen and independent from the encryption key, it is not possible to for A to distinguish between a correct encryption (i.e., where the encapsulated key and the encryption key are the same) and an encryption where the key encapsulated in c^* is randomly chosen and independent from the one used to encrypt c_2^*. This holds even if A is allowed to maul pk^* into some related public key pk'. The next step is to use the NM-ATK security of Π^{ske} to decouple m^* from the relation R, i.e. given a ciphertext c_2^* encrypting m^*, it is infeasible for an adversary to distinguish between the experiment where the relation R was checked by using either m^* or $\tilde{m}$. Finally, NM-ATK* security of Π^{kem} can be used to re-join together the key encapsulated in c_1^* with the key used to encrypt m in c_2^*. Let $\mathsf{A} = \mathsf{A}^{hy}$, the sequence of games is described in Fig. 6. The part of the proof required for a specific flavour of NM-ATK* will be highligthed with a tag [NM-ATK*].

Lemma 11. $\mathbf{G}_0(\lambda) \approx_c \mathbf{G}_1(\lambda)$.

Proof. Let us assume that A^{hy} can distinguish between $\mathbf{G}_0$ and $\mathbf{G}_1$ with non-negligible probability. We can construct an adversary A^{nm} breaking NM-ATK* security of Π^{kem}. A^{nm} behaves as follows:

1. Take as input a public key pk^*, a ciphertext c and a key $\hat{\mathsf{K}}$, where either $\hat{\mathsf{K}} = \mathsf{K}^*$ (the key encapsulated in c) or $\hat{K} \leftarrow\!\!\$\ \mathcal{K}$.

Game $\mathbf{G}_0^{\mathsf{A}}(\lambda)$

$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$

$(\mathcal{M}, s) \leftarrow\!\!\$\ \mathsf{A}_1^{\mathsf{O}_1}(\mathsf{pk})$

$m^* \leftarrow\!\!\$\ \mathcal{M}$

$(c_1^*, \mathsf{K}^*) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk}^*)$

$c_2^* \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{K}^*, m^*)$

$(\mathsf{pk}, R, c) \leftarrow\!\!\$\ \mathsf{A}_2^{\mathsf{O}_2}(\mathcal{M}, \mathsf{pk}^*, s, c^*)$

return 1 iff $\exists (m, r)$ s.t.

$(c = \mathsf{Enc}^{hy}(\mathsf{pk}, m; r)) \wedge$

$(c \neq c^* \vee \mathsf{pk}' \neq \mathsf{pk}) \wedge$

$(m \neq \bot) \wedge R(m, m^*, \mathsf{pk}, \mathsf{pk}^*, c)$

Game $\mathbf{G}_1^{\mathsf{A}}(\lambda)$

$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$

$(\mathcal{M}, s) \leftarrow\!\!\$\ \mathsf{A}_1^{\mathsf{O}_1(\cdot)}(\mathsf{pk})$

$m^* \leftarrow\!\!\$\ \mathcal{M}$

$\hat{\mathsf{K}} \leftarrow\!\!\$\ \mathcal{K}$

$(c_1^*, \mathsf{K}^*) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk}^*)$

$c_2^* \leftarrow\!\!\$\ \mathsf{Enc}(\hat{\mathsf{K}}, m^*)$

$(\mathsf{pk}, R, c) \leftarrow\!\!\$\ \mathsf{A}_2^{\mathsf{O}_2(\cdot)}(\mathcal{M}, \mathsf{pk}^*, s, c^*)$

return 1 iff $\exists (m, r)$ s.t.

$(c = \mathsf{Enc}^{hy}(\mathsf{pk}, m; r)) \wedge$

$(c \neq c^* \vee \mathsf{pk}' \neq \mathsf{pk}) \wedge$

$(m \neq \bot) \wedge R(m, m^*, \mathsf{pk}, \mathsf{pk}^*, c)$

Game $\mathbf{G}_2^{\mathsf{A}}(\lambda)$

$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$

$(\mathcal{M}, s) \leftarrow\!\!\$\ \mathsf{A}_1^{\mathsf{O}_1}(\mathsf{pk})$

$m^*, \tilde{m} \leftarrow\!\!\$\ \mathcal{M}$

$\hat{\mathsf{K}} \leftarrow\!\!\$\ \mathcal{K}$

$(c_1^*, \mathsf{K}^*) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk}^*)$

$c_2^* \leftarrow\!\!\$\ \mathsf{Enc}(\hat{\mathsf{K}}, m^*)$

$(\mathsf{pk}, R, c) \leftarrow\!\!\$\ \mathsf{A}_2^{\mathsf{O}_2}(\mathcal{M}, \mathsf{pk}^*, s, c^*)$

return 1 iff $\exists (m, r)$ s.t.

$(c = \mathsf{Enc}^{hy}(\mathsf{pk}, m; r)) \wedge$

$(c \neq c^* \vee \mathsf{pk}' \neq \mathsf{pk}) \wedge$

$(m \neq \bot) \wedge R(m, \tilde{m}, \mathsf{pk}, \mathsf{pk}^*, c)$

Game $\mathbf{G}_3^{\mathsf{A}}(\lambda)$

$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$

$(\mathcal{M}, s) \leftarrow\!\!\$\ \mathsf{A}_1^{\mathsf{O}_1(\cdot)}(\mathsf{pk})$

$m^*, \tilde{m} \leftarrow\!\!\$\ \mathcal{M}$

$(c_1^*, \mathsf{K}^*) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk}^*)$

$c_2^* \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{K}^*, m^*)$

$(\mathsf{pk}, R, c) \leftarrow\!\!\$\ \mathsf{A}_2^{\mathsf{O}_2(\cdot)}(\mathcal{M}, \mathsf{pk}^*, s, c^*)$

return 1 iff $\exists (m, r)$ s.t.

$(c = \mathsf{Enc}^{hy}(\mathsf{pk}, m; r)) \wedge$

$(c \neq c^* \vee \mathsf{pk}' \neq \mathsf{pk}) \wedge$

$(m \neq \bot) \wedge R(m, \tilde{m}, \mathsf{pk}, \mathsf{pk}^*, c)$

Figure 6: Sequence of games needed to prove Theorem 8.

2. Run $\mathsf{A}^{hy}(\mathsf{pk}^*)$.
   [NM-CCA*1/NM-CCA2*] When A^{hy} asks a decryption-oracle query about a ciphertext (c_1, c_2) query $\mathsf{Decaps}(\mathsf{sk}^*, \cdot)$ about c_1 to obtain a key K' and return $m := \mathsf{Dec}(\mathsf{K}', c_2)$.
   When A^{hy} outputs a message distribution $\mathcal{M}$, take a uniform message $m \leftarrow\!\!\$\ \mathcal{M}$, compute $c \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{K}, m)$ and return $c^* = (c, c')$ to A^{hy}.

3. [NM-CCA2*] When A^{hy} asks a decryption-oracle query about a ciphertext (c_1, c_2) do the following:

   - if $c_1 = c$ and $c_2 = c'$ then return $m := \perp$ (i.e the query is not admissible).
   - if $c_1 = c$ and $c_2 \neq c'$ then return $m := \mathsf{Dec}(\hat{\mathsf{K}}, c_2)$.
   - else, query $\mathsf{Decaps}^{(c^*)}(\mathsf{sk}^*, \cdot)$ about c_1 to obtain a key K' and return $m := \mathsf{Dec}(\mathsf{K}', c_2)$.

4. When A^{hy} outputs $(\mathsf{pk}, R, (c_1, c_2))$, output (pk, R', c_1), where $R'(\cdot, \cdot, \mathsf{pk}, \mathsf{pk}^*, c) = R(\cdot, \cdot, \mathsf{pk}, \mathsf{pk}^*, c)$.

Notice that, since the only difference between $\mathbf{G}_0$ and $\mathbf{G}_1$ is that in $\mathbf{G}_1$ the key is chosen at random, the only hope for he adversary A^{hyb} to distinguish between the two hybrids is by finding a relation holding between pk, pk^*, and c that is satisfied in $\mathbf{G}_0$ but not in $\mathbf{G}_1$ (or vice-versa). Hence, $R'(\cdot, \cdot, \mathsf{pk}, \mathsf{pk}^*, c) = R(\cdot, \cdot, \mathsf{pk}, \mathsf{pk}^*, c)$ is indeed a suitable relation for A^{nm}. When $\hat{\mathsf{K}}$ taken as input by A^{nm} is K^*, then A^{nm} perfectly simulates $\mathbf{G}_0$. When $\hat{\mathsf{K}}$ taken as input by A^{nm} is randomly chosen, A^{nm} perfectly simulates $\mathbf{G}_1$. If A^{hy} distinguishes between $\mathbf{G}_0$ and $\mathbf{G}_1$ with non-negligible probability, then A^{nm} breaks NM-ATK* security of the underlying KEM scheme with non-negligible probability. This leads to a contradiction. □

Lemma 12. $\mathbf{G}_1(\lambda) \approx_c \mathbf{G}_2(\lambda)$.

Proof. Let us assume that A^{hy} can distinguish between $\mathbf{G}_1$ and $\mathbf{G}_2$ with non-negligible probability, we construct an adversary A^{atk} breaking NM-ATK security of Π^{ske}. A^{atk} behaves as follows:

1. Receive as input a key K.

2. Generate a pair $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\!\$\ \mathsf{Gen}^{kem}(1^\lambda)$.

3. Run $\mathsf{A}^{hy}(\mathsf{pk}^*)$.
   [NM-CCA*1/NM-CCA2*] When A^{hy} asks a decryption-oracle query about a ciphertext (c_1, c_2), query the decryption oracle $\mathsf{Dec}(\hat{\mathsf{K}}, \cdot)$ of the NM-CCA experiment about c_2 to obtain a message m.

4. When A^{hy} outputs a message distribution $\mathcal{M}$, output $\mathcal{M}$ to the challenger.

5. When receiving a ciphertext c' from the challenger, compute $(c_1^*, \mathsf{K}^*) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk}^*)$, and output (c_1^*, c') to A^{hy}.

6. [NM-CCA2*] When A^{hy} asks a decryption-oracle query about a ciphertext (c_1, c_2) do the following:

   - if $c_1 = c_1^*$ and $c_2 = c'$ then return $m := \perp$ (i.e the query is not admissible).
   - else, query $\mathsf{Dec}^{(c')}(\hat{\mathsf{K}}, \cdot)$ about c_2 to obtain $m := \mathsf{Dec}(\mathsf{K}', c_2)$. Then, outputs m to A^{hyb}.

7. When A^{hy} outputs $(\mathsf{pk}, R, (c_1, c_2))$, A^{atk} output (R', c_2) to the challenger, where $R'(m_0, m_1) = R(m_0, m_1, \cdot, \cdot, \cdot)$. The challenger either checks $R'(m, m^*) = 1$ where $m^* := \mathsf{Dec}(\hat{\mathsf{K}}, c')$ if A^{atk} is in $\mathsf{SKE}^{nm\text{-}atk}_{\Pi^{ske}, \mathsf{A}^{atk}}$ or checks if $R'(m, \tilde{m}) = 1$ if $\tilde{m}$ is an randomly chosen message independent from c^*. Note that the only difference between $\mathbf{G}_1$ and $\mathbf{G}_2$ is that the game checks that m^* given as an input to R is encrypted in c_2^*, whereas in $\mathbf{G}_2$ the relation R takes as an input a random $\tilde{m}$. Thus, the only hope for A^{hyb} in distinguishing between the two hybrids is by finding a relation holding between m and m^* but not between m and $\tilde{m}$ (or vince-versa). Thus, we are allowed to cast $R'(m_0, m_1)$ as $R(m_0, m_1, \cdot, \cdot, \cdot)$. When the relation R takes m^* in input, then A^{atk} perfectly simulates $\mathbf{G}_1$. When the relation R takes a random $\tilde{m}$ in input, A^{atk} perfectly simulates $\mathbf{G}_2$.

If A^{hy} distinguishes between $\mathbf{G}_1$ and $\mathbf{G}_2$ with non-negligible probability, then A^{atk} breaks NM-ATK security of the underlying SKE scheme with non-negligible probability. This leads to a contradiction. □

Lemma 13. $\mathbf{G}_2^{\mathsf{A}}(\lambda) \approx_c \mathbf{G}_3^{\mathsf{A}}(\lambda)$

Proof. Let us assume that A^{hy} can distinguish between $\mathbf{G}_2$ and $\mathbf{G}_3$ with non-negligible probability, we construct an adversary A^{nm} breaking NM-ATK* security of Π^{kem}. A^{nm} behaves as follows:

1. Takes as input a public key pk^*, a ciphertext c and a key $\hat{\mathsf{K}}$, where either $\hat{\mathsf{K}} = \mathsf{K}^*$ (the key encapsulated in c) or $\hat{K} \leftarrow\!\!\$\ \mathcal{K}$.
2. Run $\mathsf{A}^{hy}(\mathsf{pk}^*)$.
   [NM-CCA*1/NM-CCA2*] When A^{hy} asks a decryption-oracle query about a ciphertext (c_1, c_2) query $\mathsf{Decaps}(\mathsf{sk}^*, \cdot)$ about c_1 to obtain a key K' and return $m := \mathsf{Dec}(\mathsf{K}', c_2)$.
3. When A^{hy} outputs a message distribution $\mathcal{M}$, take two uniform messages $m, \tilde{m} \leftarrow\!\!\$\ \mathcal{M}$, compute $c \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{K}, m)$ and return $c^* = (c, c')$ to A^{hy}.
4. [NM-CCA2*] When A^{hy} asks a decryption-oracle query about a ciphertext (c_1, c_2) do the following:
   - if $c_1 = c$ and $c_2 = c'$ then return $m := \perp$ (i.e the query is not admissible).
   - if $c_1 = c$ and $c_2 \neq c'$ then return $m := \mathsf{Dec}(\hat{\mathsf{K}}, c_2)$.
   - else, query $\mathsf{Decaps}^{(c^*)}(\mathsf{sk}^*, \cdot)$ about c_1 to obtain a key K' and return $m := \mathsf{Dec}(\mathsf{K}', c_2)$.
5. When A^{hy} outputs $(\mathsf{pk}, R, (c_1, c_2))$, output (pk, R', c_1), where $R'(\cdot, \cdot, \mathsf{pk}, \mathsf{pk}^*, c) = R(\cdot, \cdot, \mathsf{pk}, \mathsf{pk}^*, c)$.

Notice that, since the only difference between $\mathbf{G}_2$ and $\mathbf{G}_3$ is that in $\mathbf{G}_2$ the key is chosen at random, the only hope for he adversary A^{hyb} to distinguish between the two hybrids is by finding a relation holding between pk, pk^*, and c that is satisfied in $\mathbf{G}_2$ but not in $\mathbf{G}_3$ (or vice-versa). Hence, R' is a suitable relation for A^{nm}. When the kehy $\hat{\mathsf{K}}$ taken as input by A^{nm} is randomly chosen, it perfectly simulates $\mathbf{G}_2$. When the key $\hat{\mathsf{K}}$ taken as input by A^{nm} is K^*, then A^{nm} perfectly simulates $\mathbf{G}_3$. If A^{hy} distinguishes between $\mathbf{G}_2$ and $\mathbf{G}_3$ with non-negligible probability, then A^{nm} breaks NM-ATK* security of the underlying KEM scheme with non-negligible probability. This leads to a contradiction. □

It is easy to see that $\mathbf{G}_0^{\mathsf{A}}(\lambda) \equiv \mathsf{PKE}^{nm\text{-}atk*}_{\Pi^{hy},\mathsf{A}}(\lambda)$ and that $\mathbf{G}_3^{\mathsf{A}}(\lambda) \equiv \mathsf{PKE}^{nm\text{-}atk*}_{\Pi^{hy},\mathsf{A},\$}(\lambda)$. by combining the above lemmas, $\mathsf{PKE}^{nm\text{-}atk*}_{\Pi^{hy},\mathsf{A}}(\lambda) \equiv \mathbf{G}_0^{\mathsf{A}}(\lambda) \approx_c \mathbf{G}_1^{\mathsf{A}}(\lambda) \approx_c \mathbf{G}_2^{\mathsf{A}}(\lambda) \approx_c \mathbf{G}_3^{\mathsf{A}}(\lambda) \equiv \mathsf{PKE}^{nm\text{-}atk*}_{\Pi^{hy},\mathsf{A},\$}(\lambda)$. Hence, $\mathsf{PKE}^{nm\text{-}atk*}_{\Pi^{hy},\mathsf{A}}(\lambda) \approx_c \mathsf{PKE}^{nm\text{-}atk*}_{\Pi^{hy},\mathsf{A},\$}(\lambda)$. □

References

[1] Barbosa, M., Farshim, P.: Relations among notions of complete non-malleability: Indistinguishability characterisation and efficient construction without random oracles. In: Steinfeld, R., Hawkes, P. (eds.) ACISP (2010)

[2] Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: 38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997. pp. 394–403. IEEE Computer Society (1997)

[3] Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: Krawczyk, H. (ed.) Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1462, pp. 26–45. Springer (1998)

[4] Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1976, pp. 531–545. Springer (2000)

[5] Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. J. Cryptol. **21**(4), 469–491 (2008)

[6] Bellare, M., Sahai, A.: Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In: Wiener, M.J. (ed.) CRYPTO. vol. 1666, pp. 519–536. Springer (1999)

[7] Boldyreva, A., Fischlin, M.: Analysis of random oracle instantiation scenarios for OAEP and other practical schemes. In: Shoup, V. (ed.) Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3621, pp. 412–429. Springer (2005)

[8] Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1462, pp. 13–25. Springer (1998)

[9] Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (extended abstract). In: Koutsougeras, C., Vitter, J.S. (eds.) Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA. pp. 542–552. ACM (1991)

[10] Duman, J., Hövelmanns, K., Kiltz, E., Lyubashevsky, V., Seiler, G.: Faster lattice-based kems via a generic fujisaki-okamoto transform using prefix hashing. In: Kim, Y., Kim, J., Vigna, G., Shi, E. (eds.) CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021. pp. 2722–2737. ACM (2021)

[11] Faust, S., Kohlweiss, M., Marson, G.A., Venturi, D.: On the non-malleability of the fiat-shamir transform. In: Galbraith, S.D., Nandi, M. (eds.) Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7668, pp. 60–79. Springer (2012)

[12] Fischlin, M.: Completely non-malleable schemes. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3580, pp. 779–790. Springer (2005)

[13] Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1666, pp. 537–554. Springer (1999)

[14] Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the fujisaki-okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10677, pp. 341–371. Springer (2017)

[15] Libert, B., Yung, M.: Efficient completely non-malleable public key encryption. In: Abramsky, S., Gavoille, C., Kirchner, C., auf der Heide, F.M., Spirakis, P.G. (eds.) Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I. Lecture Notes in Computer Science, vol. 6198, pp. 127–139. Springer (2010)

[16] Nagao, W., Manabe, Y., Okamoto, T.: On the equivalence of several security notions of key encapsulation mechanism. IACR Cryptol. ePrint Arch. p. 268 (2006), http://eprint.iacr.org/2006/268

[17] NIST: Nist announces first four quantum-resistant cryptographic algorithms (2022), https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms

[18] Pass, R., Shelat, A., Vaikuntanathan, V.: Relations among notions of non-malleability for encryption. In: Kurosawa, K. (ed.) ASIACRYPT. vol. 4833, pp. 519–535 (2007)

[19] Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings. Lecture Notes in Computer Science, vol. 576, pp. 433–444. Springer (1991)

[20] Sepahi, R., Steinfeld, R., Pieprzyk, J.: Lattice-based completely non-malleable PKE in the standard model (poster). In: Parampalli, U., Hawkes, P. (eds.) Information Security and Privacy - 16th Australasian Conference, ACISP 2011, Melbourne, Australia, July 11-13, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6812, pp. 407–411. Springer (2011)

[21] Sepahi, R., Steinfeld, R., Pieprzyk, J.: Lattice-based completely non-malleable public-key encryption in the standard model. Des. Codes Cryptogr. **71**(2), 293–313 (2014)

[22] Ventre, C., Visconti, I.: Completely non-malleable encryption revisited. In: Cramer, R. (ed.) Public Key Cryptography - PKC 2008, 11th International Workshop on Practice and Theory in Public-Key Cryptography, Barcelona, Spain, March 9-12, 2008. Proceedings. Lecture Notes in Computer Science, vol. 4939, pp. 65–84. Springer (2008)