A preliminary version of this paper appears in *Advances in Cryptology – CRYPTO '99*, Lecture Notes in Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999. This revised version corrects some mistakes from the preliminary version.

Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization

Mihir Bellare* Amit Sahai†

July 2006

Abstract

We prove the equivalence of two definitions of non-malleable encryption, one based on the simulation approach of Dolev, Dwork and Naor [6] and the other based on the comparison approach of Bellare, Desai, Pointcheval and Rogaway [2]. Our definitions are slightly stronger than the original ones. The equivalence relies on a new characterization of non-malleable encryption in terms of the standard notion of indistinguishability of Goldwasser and Micali. We show that non-malleability is equivalent to indistinguishability under a "parallel chosen ciphertext attack," this being a new kind of chosen ciphertext attack we introduce, in which the adversary's decryption queries are not allowed to depend on answers to previous queries, but must be made all at once. This characterization simplifies both the notion of non-malleable encryption and its usage, and enables one to see more easily how it compares with other notions of encryption. The results here apply to non-malleable encryption under any form of attack, whether chosen-plaintext, chosen-ciphertext, or adaptive chosen-ciphertext.

Keywords: Asymmetric encryption, Non-malleability, Indistinguishability, equivalence between notions, semantic security.

---

*Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0404, USA. E-Mail: `mihir@cs.ucsd.edu`. URL: `http://www-cse.ucsd.edu/users/mihir`. Supported in part by NSF CNS-0524765.

†Department of Computer Science, 3731E Boelter Hall, University of California at Los Angeles, Los Angeles, CA 90095, USA. E-Mail: `sahai@cs.ucla.edu`. URL: `http://www.cs.ucla.edu/~sahai`. Supported in part by an Alfred P. Sloan Foundation Research Fellowship, grants from the NSF Cybertrust and ITR programs including CNS-0456717 and CCF-0205594, and a generous equipment grant from Intel.

Contents

1 Introduction

Public-key encryption has several goals in terms of protecting the data that is encrypted. The most basic is *privacy*, where the goal is to ensure that an attacker does not learn any useful information about the data from the ciphertext. Indistinguishability and semantic security [9] are formalizations of this goal. A second goal, introduced by Dolev, Dwork and Naor [6], is non-malleability, which, roughly, requires that an attacker given a challenge ciphertext be unable to modify it into another, different ciphertext in such a way that the plaintexts underlying the two ciphertexts are "meaningfully related" to each other. Both these goals can be considered under attacks of increasing severity: chosen-plaintext attacks, and two kinds of chosen ciphertext attacks [15, 16].

Recent uses of public-key encryption have seen a growing need for, and hence attention to, stronger than basic forms of security, like non-malleability. This kind of security is important when encryption is used as a primitive in the design of higher level protocols, for example for key distribution (cf. [1]). The interest is witnessed by attention to classification of the notions of encryption [2, 6] and efficient constructions of non-malleable schemes [3, 5].

1.1 Themes in foundations of encryption

Our confidence that we have the "right" formalizations of privacy is due in part to results which show that several definitions, based on different approaches and intuition, are the equivalent in the sense that a scheme meets one definition if and only if it meets the others. In particular, definitions based on indistinguishability, semantic security and computational entropy have been shown to be equivalent [9, 17, 14]. These foundational results have since been refined and extended to other settings [8]. These equivalences are a cornerstone of our understating of privacy.

Semantic security captures in perhaps the most direct way one's intuition of a good notion of privacy. (Roughly, it says that "whatever can be efficiently computed about a message given the ciphertext can be computed without the ciphertext"). But it is a relatively complex and subtle notion to formalize, and hard to use to analyze applications of encryption. Indistinguishability has the opposite attributes. The formalization is simple, appealing and easy to use. (It says that if we take two equal-length messages m_0, m_1, an adversary given an encryption of a random one of them cannot tell which it was with a probability significantly better than that of guessing). It is by far the first choice when analyzing the security of an encryption based application. But it is less clear (by just a direct examination of the definition) that it really captures an intuitively strong notion of privacy. However, we know it does, because it is in fact equivalent to semantic security. Accordingly, we can view indistinguishability as a "characterization" of semantic security, a simple, easy to use notion backed by the fact of being equivalent to the more naturally intuitive one.

Thus, beyond equivalences between notions, one also seeks characterizations that are simple and easy to work with.

1.2 Questions for non-malleability

The foundations of non-malleable encryption are currently not as well laid as those of privacy, for several reasons.

First, there are in the literature two formalizations. The first is the original one of Dolev, Dwork and Naor [6], which we call simulation based non-malleability (SNM). A second approach was introduced by Bellare, Desai, Pointcheval and Rogaway [2], and we call it comparison based non-malleability (CNM). A priori, at least, the two seem to have important differences. Second, there is no simple and easy-to-use characterization of non-malleable encryption akin to indistinguishability for privacy. Rather, the current formalizations of non-malleability follow the definitional paradigm of semantic security and in particular both existing formulations are quite complex (even though that of [2] is somewhat simpler than that of [6]), and subtle at the level of details. A consequence is that non-malleability can be hard to use. The applicability of non-malleability would be increased by having some simple characterization of the notion.

Although not required for the statement of our results, it may be instructive to try to convey some rough idea of the existing definitional approaches. The definitions involve considering some relation R between plaintexts, having an adversary output a distribution on some set of messages, and then setting up a challenge-response game. The adversary is given as input a ciphertext y of a plaintext x drawn from the message distribution, and must produce a vector of ciphertexts $\mathbf{y}$, none of whose components is y. If $\mathbf{x}$ is the plaintext vector corresponding to $\mathbf{y}$, security requires, roughly, that the adversary's ability to make $R(x, \mathbf{x})$ true in this game is not much more than her ability to make it true had she had to produce $\mathbf{y}$ without being given y at all, namely given no information about x other than its distribution. The two known definitions differ in how exactly they measure the success in the last part of the game. The simulation-based notion, as the name indicates, is based on the simulation paradigm: a scheme is secure if for any adversary there exists a simulator which does almost as well without any information about the challenge ciphertext given to the adversary. In the comparison-based formalization, there is no simulator. Instead, it is required that the success probability of the adversary under the "real" challenge and a "fake" challenge be about the same. Besides the fundamental difference of one being simulation based and the other not, the first notion does not allow the simulator access to the decryption oracle even when the adversary gets it —meaning when chosen-ciphertext attacks are being considered— while, in this case, the second notion allows the adversary the same access to the decryption oracle in both games.

1.3 The equivalence

In this paper we formalize a definition of simulation-based non-malleability and one of comparison-based non-malleability and show that they are equivalent. The equivalence holds for the three major classes of attacks usually considered in the literature, namely chosen-plaintext (CPA), type-1 chosen ciphertext [2], also called lunch-time attack [15], and type-2 chosen ciphertext [2], also called adaptive chosen-ciphertext attack [16, 6]. That is, for ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$ we have notions SNM-ATK and CNM-ATK, and show they are equivalent. In other words, an encryption scheme meets the SNM-ATK notion of security if and only if it meets the CNM-ATK notion of security.

Our definitions are slightly stronger than the original ones of [6, 2] in one respect. Namely, [2] declare an adversary unsuccessful if its output $\mathbf{y}$ contains an invalid ciphertext. Dolev, Dwork and Naor [6] express a similar intent, although, as discussed in Section 6, there is some ambiguity about exactly how they intend this to be formalized. We, instead, leave it up to the relation to decide whether the adversary wins or loses even when $\perp \in \mathbf{y}$, meaning the relation is defined even on vectors $\mathbf{x}$ containing $\perp$. Lindell [13] points to an advantage of the stronger SNM-ATK, CNM-ATK notions we have introduced, namely that they imply non-malleability of the encryption of multiple ciphertexts. (Interestingly, the proof of this relies on the characterization we discuss next.) Another advantage of our notions, of course, is the equivalence itself. We note that also in (past) work on privacy, it has not been uncommon to slightly modify existing definitions to establish equivalences, so that such endeavors and results also serve to refine our definitions.

Thus, we are saying that two formalizations of non-malleability that are underlain by somewhat different intuitions are, in fact, capturing the same underlying notion. Like the equivalences amongst notions of privacy, this result serves to strengthen our conviction that this single, unified notion of non-malleability is in fact the appropriate one.

1.4 An indistinguishability-based characterization

Perhaps even more interesting than the above-mentioned equivalence is a result used to prove it. This is a new characterization of non-malleability that we feel simplifies the notion, makes it easier to use in applications, and increases our understanding of it and its relation to the more classic notions. Roughly speaking, we show that non-malleability is actually just a form of indistinguishability, but under a certain special type of chosen-ciphertext attack that we introduce and call a *parallel chosen-ciphertext attack.* Thus, what appears to be a different adversarial goal (namely, the ability to modify a ciphertext in such a way as to create relations between the underlying plaintexts) corresponds actually to the standard goal of privacy, as long as we add power to the attack model.

Our characterization dispenses with the relation R and the message space: it is just about a game involving two messages. To illustrate, consider non-malleability under chosen-plaintext attack. Our characterization says this is equivalent to indistinguishability under a parallel chosen-ciphertext attack. In this attack, the adversary gets to ask a single vector query of the decryption oracle. This means it specifies a sequence $\mathbf{c}[1], \ldots, \mathbf{c}[n]$ of ciphertexts, and obtains the corresponding plaintexts $\mathbf{p}[1], \ldots, \mathbf{p}[n]$ from the oracle. But the choice of $\mathbf{c}[2]$ is not allowed to depend on the answer to $\mathbf{c}[1]$, and so on. (So we can think of all the queries as made in parallel, hence the name. Perhaps a better name would have been non-adaptive queries, but the term non-adaptive is already in use in another way in this area and was best avoided.) This query is allowed to be a function of the challenge ciphertext. In more detail the game is that we take two equal-length messages m_0, m_1, give the adversary a ciphertext y of a random one of them, and now allow it a single parallel vector decryption oracle query, the only constraint on which is that the query not contain y in any component. The adversary wins if it can then say which of the two messages m_0, m_1 had been encrypted to produce the challenge y, with a probability significantly better than that of guessing.

Thus, as mentioned above, our notion makes no mention of a relation R or a probability space on messages, let alone of a simulator. Instead, it follows an entirely standard paradigm, the only twist being the nature of the attack model, or alternatively, what is given to the distinguisher. We defer giving more intuition about our indistinguishability-based notion until after we give a formal definition (see the end of Section 4).

A special case that might be worth noting is that when the relation R is binary, the parallel attack need contain just one ciphertext. In general, the number of parallel queries needed is one less than the arity of R.

The characterization holds for all three forms of attack ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$. We introduce parallel chosen-ciphertext attacks PCAX $\in \{\mathrm{PCA0}, \mathrm{PCA1}, \mathrm{PCA2}\}$ corresponding to each of these, and show that CNM-ATK and SNM-ATK are equivalent to IND-PCAX.

1.5 Discussion, extensions and other relations

Six notions of encryption, namely non-malleability and indistinguishability, each under the three forms of attack, are related in [2, 6] by showing either an implication or a separation between each pair of notions. The same picture emerges even though the two works use different formalizations of non-malleability. Given that our notions of non-malleability are slightly different from the ones used in either of these papers, we revisit this question. However, as indicated in

Figure 1, we again show that the same relations continue to hold, meaning the picture is the same as before modulo the substitution of the new notions of non-malleability. In particular, $\mathrm{CNM\text{-}CCA2}, \mathrm{SNM\text{-}CCA2}$ and IND-PCA2 are all equivalent to the standard IND-CCA2. We note that our characterization in terms of indistinguishability under parallel attack helps to provide simpler proofs of these implications and separations than given in [2, 6] because it reduces us to considering relations between five notions of indistinguishability, namely indistinguishability under the attacks $\mathrm{CPA}, \mathrm{PCA0}, \mathrm{CCA1}, \mathrm{PCA1}$ and $\mathrm{CCA2} = \mathrm{PCA2}$. See Section 6 for more details.

The proceedings version of our paper [4] had claimed that our results held for the original definitions of non-malleability of [6, 2]. (Actually, the formal definitions in [4] had been of the same stronger SNM-ATK and CNM-ATK notions defined here, and the theorems and proofs correctly established the same relations as here, but the Introduction, and discussion surrounding the definitions, had given the impression that we were talking about the original notions.) This discrepancy was pointed out to us by Lindell [13] and is rectified in the current version. Section 6 discusses in more detail the relations between the old and new notions.

1.6 Related work

Halevi and Krawczyk introduce a weak version of chosen-ciphertext attack which they call a one-ciphertext *verification* attack [10]. This is not the same as a parallel attack. In their attack, the adversary generates a single plaintext along with a candidate ciphertext, and is allowed to ask a verification query, namely whether or not the pair is valid. In our notion, the adversary has more power: it can access the decryption oracle.

Katz and Yung [12] provide relations among notions of security for symmetric (i.e. shared key) encryption schemes. In this context they mention that the stronger form of non-malleability (considered here) in which we do not impose an automatic fail on the adversary depending on whether the ciphertext vector contains an invalid ciphertext, may be more appropriate for some applications, and its use would simplify their proofs.

2 Preliminaries

Notation. Let $\mathsf{N} = \{1, 2, 3, \ldots\}$ denote the set of positive integers. Unless otherwise indicated, an algorithm is randomized. If A is an algorithm, then $A(x_1, x_2, \ldots; r)$ is the result of running A on inputs $x_1, x_2, \ldots$ and coins r. We let $y \stackrel{\$}{\leftarrow} A(x_1, x_2, \ldots)$ denote the experiment of picking r at random and letting $y = A(x_1, x_2, \ldots; r)$. We say that y *can be output by* $A(x_1, x_2, \ldots)$ if there is some r such that $A(x_1, x_2, \ldots; r) = y$. We denote by "$X \Rightarrow x$" the event that algorithm X has output x. An experiment is an algorithm. When say that a tuple $A = (A_1, A_2, \ldots)$ of algorithms is polynomial time we mean that each component algorithm is polynomial time. If S is a finite set then $x \stackrel{\$}{\leftarrow} S$ is the operation of picking an element uniformly at random from S.

Encryption schemes. An asymmetric encryption scheme is given by a triple of algorithms, $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, where

- $\mathcal{K}$, the *key generation algorithm* takes a security parameter $k \in \mathsf{N}$ (provided in unary) and returns a pair (pk, sk) of matching public and secret keys.
- $\mathcal{E}$, the *encryption algorithm*, takes a public key pk and a message $x \in \{0, 1\}^*$ to produce a ciphertext y.
- $\mathcal{D}$, the *decryption algorithm*, is a deterministic algorithm which takes a secret key sk and ciphertext y to produce either a message $x \in \{0, 1\}^*$ or a special symbol $\perp$ to indicate that the

ciphertext was invalid.

We require that for all positive integers k, all (pk, sk) which can be output by $\mathcal{K}(1^k)$, all $x \in \{0,1\}^*$, and all y that can be output by $\mathcal{E}_{pk}(x)$, we have that $\mathcal{D}_{sk}(y) = x$. We also require that $\mathcal{K}$, $\mathcal{E}$ and $\mathcal{D}$ can be computed in polynomial time. As the notation indicates, the keys are indicated as subscripts to the algorithms.

FURTHER NOTATION AND TERMINOLOGY. We will need to discuss vectors of plaintexts or ciphertexts. A vector is denoted in boldface, as in $\mathbf{x}$. We denote by $|\mathbf{x}|$ the number of components in $\mathbf{x}$, and by $\mathbf{x}[i]$ the i-th component, so that $\mathbf{x} = (\mathbf{x}[1], \ldots, \mathbf{x}[|\mathbf{x}|])$. We extend the set membership notation to vectors, writing $x \in \mathbf{x}$ or $x \notin \mathbf{x}$ to mean, respectively, that x is in or is not in the set $\{\, \mathbf{x}[i] \;:\; 1 \leq i \leq |\mathbf{x}| \,\}$. It will be convenient to extend the decryption notation to vectors with the understanding that operations are performed component-wise. Thus $\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$ is shorthand for the following:

$$\text{For } i = 1, \ldots, |\mathbf{y}| \text{ do } \mathbf{x}[i] \leftarrow \mathcal{D}_{sk}(\mathbf{y}[i]) \,.$$

Recall that a function $\epsilon : \mathsf{N} \to \mathbf{R}$ is *negligible* if for every constant $c \geq 0$ there exists an integer k_c such that $\epsilon(k) \leq k^{-c}$ for all $k \geq k_c$.

3 Two definitions of non-malleable encryption

In the setting of non-malleable encryption, the goal of an adversary, given a ciphertext y, is not (as with indistinguishability) to learn something about its plaintext x, but rather to output a vector $\mathbf{y}$ of ciphertexts whose decryption $\mathbf{x}$ is "meaningfully related" to x, meaning that $R(x, \mathbf{x})$ holds for some relation R. There are several approaches to formalizing security. One approach is that of [6], which asks that for every adversary there exists an appropriate "simulator" that does just as well as the adversary but *without* being given y. Another, somewhat simpler approach is that of [2], where there is no simulator: security is defined by comparing the success probability of the adversary given y to the success of the same adversary given the encryption of a message unrelated to x. We begin by presenting below a formal definition corresponding to each of these two approaches.

3.1 Definition of SNM

In this subsection we describe a definition of non-malleable encryption based on the approach of [6]. We call it SNM for "simulation based non-malleability."

Our SNM formulation fixes a polynomial time computable relation R, which is viewed as taking four arguments, $x, \mathbf{x}, M, s_1$, with $\mathbf{x}$ being a vector with an arbitrary number of components, each component drawn from $\{0,1\}^* \cup \{\perp\}$. The relation returns `true` or `false`. Given any such input, the relation returns either true or false.

The adversary $A = (A_1, A_2)$ runs in two stages. The first stage of the adversary, namely A_1, is given the public key pk and computes (the encoding of) a distribution M on messages (strings) and also some state information: a string s_1 to pass to the relation R, and a string s_2 to pass on to A_2. (At A_1's discretion, either of these might include M and pk.) We call M the message space. It must be *valid*, which means that all strings having non-zero probability under M are of the same length.

The second stage of the adversary, namely A_2, receives s_2 and the encryption y of a random message x drawn from M. Algorithm A_2 then outputs a vector of ciphertexts $\mathbf{y}$. We say that A is successful if $R(x, \mathbf{x}, M, s_1)$ holds, and require that $y \notin \mathbf{y}$.

The requirement for security is that for any polynomial time adversary A and any polynomial time relation R there exists a polynomial time $S = (S_1, S_2)$, the simulator, that, without being given y, has about the same success probability as A. The experiment here is that S_1 is first run on pk to produce M, s_1, s_2, then x is selected from M, then S_2 is run on s_2 (but no encryption of x) to produce $\mathbf{y}$. Success means $\mathbf{x} = \mathcal{D}_{sk}(\mathbf{y})$ satisfies $R(x, \mathbf{x}, M, s_1)$. Again, M is required to be valid.

For CCA2 both A_1 and A_2 get the decryption oracle, but A_2 is not allowed to call it on the challenge ciphertext y; for CCA1 just A_1 gets the decryption oracle; and for CPA neither A_1 nor A_2 get it. However, a key feature of the SNM definition is that *no matter what the attack, the simulator does not get a decryption oracle, even though the adversary may get one.*

We now provide a formal definition for simulation-based non-malleability. When we say $\mathcal{O}_i = \varepsilon$, where $i \in \{1, 2\}$, we mean $\mathcal{O}_i$ is the function which, on any input, returns the empty string, ε. We let the string atk be instantiated by any of the formal symbols cpa, cca1, cca2, while ATK is then the corresponding formal symbol from CPA, CCA1, CCA2.

Definition 3.1 [SNM-CPA, SNM-CCA1, SNM-CCA2] Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme, let R be a relation, let $A = (A_1, A_2)$ be an adversary consisting of a pair of algorithms, and let $S = (S_1, S_2)$ be a pair of algorithms that we call the simulator. For atk $\in$ {cpa, cca1, cca2} and $k \in \mathsf{N}$ define

$$\mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi,R}(k) \stackrel{\text{def}}{=} \Pr[\,\mathsf{Expt}^{\text{snm-atk}}_{A,\Pi,R}(k) \Rightarrow 1\,] - \Pr[\,\mathsf{Expt}^{\text{snm-atk}}_{S,\Pi,R}(k) \Rightarrow 1\,]\ ,$$

where

$\mathsf{Expt}^{\text{snm-atk}}_{A,\Pi,R}(k)$	$\mathsf{Expt}^{\text{snm-atk}}_{S,\Pi,R}(k)$
$(pk, sk) \stackrel{\$}{\leftarrow} \mathcal{K}(1^k)$	$(pk, sk) \stackrel{\$}{\leftarrow} \mathcal{K}(1^k)$
$(M, s_1, s_2) \stackrel{\$}{\leftarrow} A_1^{\mathcal{O}_1}(pk)$	$(M, s_1, s_2) \stackrel{\$}{\leftarrow} S_1(pk)$
$x \stackrel{\$}{\leftarrow} M$; $y \stackrel{\$}{\leftarrow} \mathcal{E}_{pk}(x)$	$x \stackrel{\$}{\leftarrow} M$
$\mathbf{y} \stackrel{\$}{\leftarrow} A_2^{\mathcal{O}_2}(s_2, y)$	$\mathbf{y} \stackrel{\$}{\leftarrow} S_2(s_2)$
$\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$	$\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$
If $R(x, \mathbf{x}, M, s_1)$ then return 1	If $R(x, \mathbf{x}, M, s_1)$ then return 1
Else return 0	Else return 0

and

If atk = cpa	then	$\mathcal{O}_1(\cdot) = \varepsilon$	and	$\mathcal{O}_2(\cdot) = \varepsilon$
If atk = cca1	then	$\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$	and	$\mathcal{O}_2(\cdot) = \varepsilon$
If atk = cca2	then	$\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$	and	$\mathcal{O}_2(\cdot) = \mathcal{D}_{sk}(\cdot)$

We only consider adversaries A that are *legitimate* in the sense that with probability one the following are true in the first experiment: message space M is valid, $y \notin \mathbf{y}$, and, in the case of CCA2, adversary A_2 does not query its decryption oracle with the challenge ciphertext y. We only consider simulators that are legitimate in the sense that message space M in the second experiment is valid. We say that Π is secure in the sense of SNM-ATK if for every polynomial $p(k)$, every R computable in time $p(k)$, every (legitimate) A that runs in time $p(k)$ and outputs a message space M sampleable in time $p(k)$, there exists a (legitimate) polynomial-time simulator $S = (S_1, S_2)$ such that $\mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi,R}(\cdot)$ is negligible. ▌

The condition that $y \notin \mathbf{y}$ is made in order to not give the adversary credit for the trivial and unavoidable action of copying the challenge ciphertext. The requirement that M is valid stems from the fact that encryption is not intended to conceal the length of the plaintext.

3.2 Definition of CNM

In this subsection we describe a definition of non-malleable encryption based on the approach of [2]. We call it CNM for "comparison based non-malleability."

The adversary $C = (C_1, C_2)$ runs in two stages. The first stage of the adversary, namely C_1, is given the public key pk, and outputs a description of a valid message space, described by a sampling algorithm M, as well as state information s to pass on to C_2. The second stage of the adversary, namely C_2, receives s and an encryption y of a random message x drawn from M. It then outputs a (description of a) relation R together with a vector $\mathbf{y}$, where R is viewed as taking two arguments, $x, \mathbf{x}$, with $\mathbf{x}$ being a vector with an arbitrary number of components, each component drawn from $\{0,1\}^* \cup \{\perp\}$. The relation returns `true` or `false`. We consider the probability that $R(x, \mathbf{x})$ holds, where $\mathbf{x} = \mathcal{D}_{sk}(\mathbf{y})$ and it is required that $y \notin \mathbf{y}$. Alongside, we consider the probability that $R(x, \mathbf{x})$ holds, again requiring $y \notin \mathbf{y}$, if C_2 had been given as input not an encryption of x but rather an encryption of some $\tilde{x}$ also chosen uniformly from M, independently of x. The advantage of the adversary is the difference between these two probabilities.

We now provide the formal definition. Below we let the string atk be instantiated by any of the formal symbols cpa, cca1, cca2, while ATK is then the corresponding formal symbol from CPA, CCA1, CCA2.

Definition 3.2 [CNM-CPA, CNM-CCA1, CNM-CCA2] Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $C = (C_1, C_2)$ be an adversary consisting of a pair of algorithms. For atk $\in \{\text{cpa}, \text{cca1}, \text{cca2}\}$ and $k \in \mathsf{N}$ define

$$\mathsf{Adv}^{\text{cnm-atk}}_{C,\Pi}(k) \stackrel{\text{def}}{=} \Pr[\,\mathsf{Expt}^{\text{cnm-atk-1}}_{C,\Pi}(k) \Rightarrow 1\,] - \Pr[\,\mathsf{Expt}^{\text{cnm-atk-0}}_{C,\Pi}(k) \Rightarrow 1\,]\ ,$$

where

$\mathsf{Expt}^{\text{cnm-atk-1}}_{C,\Pi}(k)$	$\mathsf{Expt}^{\text{cnm-atk-0}}_{C,\Pi}(k)$
$(pk, sk) \stackrel{\$}{\leftarrow} \mathcal{K}(1^k)$	$(pk, sk) \stackrel{\$}{\leftarrow} \mathcal{K}(1^k)$
$(M, s) \stackrel{\$}{\leftarrow} C_1^{\mathcal{O}_1}(pk)$	$(M, s) \stackrel{\$}{\leftarrow} C_1^{\mathcal{O}_1}(pk)$
$x \stackrel{\$}{\leftarrow} M$	$x \stackrel{\$}{\leftarrow} M\ ;\ \tilde{x} \stackrel{\$}{\leftarrow} M$
$y \stackrel{\$}{\leftarrow} \mathcal{E}_{pk}(x)$	$\tilde{y} \stackrel{\$}{\leftarrow} \mathcal{E}_{pk}(\tilde{x})$
$(R, \mathbf{y}) \stackrel{\$}{\leftarrow} C_2^{\mathcal{O}_2}(s, y)$	$(R, \tilde{\mathbf{y}}) \stackrel{\$}{\leftarrow} C_2^{\mathcal{O}_2}(s, \tilde{y})$
$\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$	$\tilde{\mathbf{x}} \leftarrow \mathcal{D}_{sk}(\tilde{\mathbf{y}})$
If $R(x, \mathbf{x})$ then return 1 else return 0	If $R(x, \tilde{\mathbf{x}})$ then return 1 else return 0

and

If atk = cpa	then	$\mathcal{O}_1(\cdot) = \varepsilon$	and	$\mathcal{O}_2(\cdot) = \varepsilon$
If atk = cca1	then	$\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$	and	$\mathcal{O}_2(\cdot) = \varepsilon$
If atk = cca2	then	$\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$	and	$\mathcal{O}_2(\cdot) = \mathcal{D}_{sk}(\cdot)$

We only consider adversaries C that are *legitimate* in the sense that with probability one the following are true in both experiments: message space M is valid, $y \notin \mathbf{y}$, and, in the case of CCA2, adversary C_2 does not query its decryption oracle with the challenge ciphertext y. We say that Π is secure in the sense of CNM-ATK if for every polynomial $p(k)$, every (legitimate) C that runs in time $p(k)$, outputs a message space M sampleable in time $p(k)$, and outputs a relation R computable in time $p(k)$, it is the case that $\mathsf{Adv}^{\text{cnm-atk}}_{C,\Pi}(\cdot)$ is negligible. ▌

The major difference between SNM and CNM is that the former asks for a simulator and the latter does not, but some more minor differences exist too. For example in SNM the relation R is fixed beforehand, while in CNM it is generated dynamically by the adversary in its second stage.

4 Indistinguishability under parallel chosen-ciphertext attack

We present a new notion of security for a public key encryption scheme that we call indistinguishability under a parallel chosen-ciphertext attack. It is in the style of the classical definition of indistinguishability of encryptions from [9, 14]. Here, malleability is not evident in any explicit way; there is no relation R, and the adversary does not output ciphertexts, but rather tries to predict information about the plaintext. Nonetheless we will show that this notion is equivalent to both forms of non-malleability given above.

In the attack, the adversary is allowed to query the decryption oracle a polynomial number of times, but the different queries made are not allowed to depend on each other. A simple way to visualize this is to imagine the adversary making the queries "in parallel," as a vector $\mathbf{c}$ where $\mathbf{c}[1], \ldots, \mathbf{c}[n]$ are ciphertexts, for $n = |\mathbf{c}|$. The oracle replies with $\mathcal{D}_{sk}(\mathbf{c}) = (\mathcal{D}_{sk}(\mathbf{c}[1]), \ldots, \mathcal{D}_{sk}(\mathbf{c}[n]))$, the vector of the corresponding plaintexts. Only one of these parallel queries is allowed, and it is always in the second stage, meaning can be a function of the challenge ciphertext.

It is convenient to make the parallel query quite explicit in the formalization. The adversary $I = (I_1, I_2, I_3)$ runs in three stages. The first stage of the adversary, namely I_1, is given the public key pk and computes a pair x_0, x_1 of messages (strings), required to be of the same length, and also some state information s_1 to pass on to the second stage. A random one of x_0 and x_1 is now selected, say x_b. A "challenge" y is determined by encrypting x_b under pk. The second stage of the adversary, namely I_2, receives s_1 and y, and outputs a parallel query $\mathbf{c}$ and state information s_2. We require that $y \notin \mathbf{c}$. The reply $\mathbf{p} = \mathcal{D}_{sk}(\mathbf{c})$ to this query, a vector with entries in $\{0,1\}^* \cup \{\perp\}$, is now computed. The third stage of the adversary, namely I_3, receives $\mathbf{p}$ and s_2, and outputs a bit g that is its "guess" as to the value of the challenge bit b. The adversary wins if $g = b$. Its advantage is the excess over one-half of the probability that it wins, scaled up to be a number between zero and one.

We can add this parallel attack to any of the previous attacks CPA, CCA1, CCA2, yielding respectively the attacks PCA0, PCA1, PCA2. Note that since in CCA2, the second stage of the adversary can already do *adaptive* chosen ciphertext attacks, giving it the ability to perform a parallel attack yields no additional power, so in fact CCA2 = PCA2. For concision and clarity we simultaneously define indistinguishability with respect to PCA0, PCA1, and PCA2. The only difference lies in whether or not I_1 and I_2 are given decryption oracles. We let the string atk be instantiated by any of the formal symbols pca0, pca1, pca2, while ATK is then the corresponding formal symbol from PCA0, PCA1, PCA2.

Definition 4.1 [IND-PA0, IND-PA1, IND-PA2] Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $I = (I_1, I_2, I_3)$ be an adversary consisting of a triple of algorithms. For atk $\in$ {pca0, pca1, pca2} and $k \in \mathsf{N}$, let

$$\mathsf{Adv}_{I,\Pi}^{\text{ind-atk}}(k) \stackrel{\text{def}}{=} 2 \cdot \Pr[\,\mathsf{Expt}_{I,\Pi}^{\text{ind-atk-b}}(k) \Rightarrow b\,] - 1$$

where the probability is over a random choice of b from $\{0,1\}$ and over the coins of the following experiment defined for each $b \in \{0,1\}$:

$\mathsf{Expt}_{I,\Pi}^{\text{ind-atk-b}}(k)$
 $(pk, sk) \stackrel{\$}{\leftarrow} \mathcal{K}(1^k)$
 $(x_0, x_1, s_1) \stackrel{\$}{\leftarrow} I_1^{\mathcal{O}_1}(pk)$; $y \stackrel{\$}{\leftarrow} \mathcal{E}_{pk}(x_b)$
 $(\mathbf{c}, s_2) \stackrel{\$}{\leftarrow} I_2^{\mathcal{O}_2}(x_0, x_1, s_1, y)$; $\mathbf{p} \leftarrow \mathcal{D}_{sk}(\mathbf{c})$
 $g \stackrel{\$}{\leftarrow} I_3^{\mathcal{O}_2}(\mathbf{p}, s_2)$
 Return g

and

$$\begin{array}{llll} \text{If atk} = \text{pca0} & \text{then} & \mathcal{O}_1(\cdot) = \varepsilon & \text{and} \quad \mathcal{O}_2(\cdot) = \varepsilon \\ \text{If atk} = \text{pca1} & \text{then} & \mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot) & \text{and} \quad \mathcal{O}_2(\cdot) = \varepsilon \\ \text{If atk} = \text{pca2} & \text{then} & \mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot) & \text{and} \quad \mathcal{O}_2(\cdot) = \mathcal{D}_{sk}(\cdot) \end{array}$$

We only consider adversaries I that are *legitimate* in the sense that for each $b \in \{0,1\}$, with probability one the following is true in the above experiment: $|x_0| = |x_1|$, $y \notin \mathbf{c}$ and, in the case of CCA2, adversary I_2 does not query its decryption oracle with the challenge ciphertext y. We say that Π is secure in the sense of IND-ATK if $\mathsf{Adv}^{\text{ind-atk}}_{I,\Pi}(\cdot)$ is negligible for every (legitimate) polynomial-time adversary I. ▮

Note that IND-PCA2 is equivalent to IND-CCA2. Indeed, in the CCA2 case, the parallel query is redundant since the adversary already has a decryption oracle in the second stage.

It is convenient for some of our proofs to exploit the following alternative formulation of the advantage:

Lemma 4.2 Let Π be an encryption scheme and let $I = (I_1, I_2, I_3)$ be an adversary. Then for atk $\in \{\text{pca0}, \text{pca1}, \text{pca2}\}$ and all $k \in \mathsf{N}$ we have:

$$\mathsf{Adv}^{\text{ind-atk}}_{I,\Pi}(k) \quad = \quad \Pr[\,\mathsf{Expt}^{\text{ind-atk-1}}_{A,\Pi}(k) \Rightarrow 1\,] - \Pr[\,\mathsf{Expt}^{\text{ind-atk-0}}_{A,\Pi}(k) \Rightarrow 1\,]\;. \quad ▮$$

The proof is a standard conditioning argument, but we would like to give the details for completeness.

Proof of Lemma 4.2: Taking the probability over a random choice of $b \in \{0,1\}$ and the coins of the experiments involved, we have

$$\begin{aligned} &\frac{1}{2} + \frac{1}{2} \cdot \mathsf{Adv}^{\text{ind-atk}}_{I,\Pi}(k) \\ &\quad = \Pr[\,\mathsf{Expt}^{\text{ind-atk-b}}_{I,\Pi}(k) \Rightarrow b\,] \\ &\quad = \Pr\Big[\,\mathsf{Expt}^{\text{ind-atk-b}}_{I,\Pi}(k) \Rightarrow b \;\mid\; b = 1\Big] \cdot \Pr[\,b = 1\,] + \Pr\Big[\,\mathsf{Expt}^{\text{ind-atk-b}}_{I,\Pi}(k) \Rightarrow b \;\mid\; b = 0\Big] \cdot \Pr[\,b = 0\,] \\ &\quad = \Pr[\,\mathsf{Expt}^{\text{ind-atk-1}}_{I,\Pi}(k) \Rightarrow 1\,] \cdot \frac{1}{2} + \Pr[\,\mathsf{Expt}^{\text{ind-atk-0}}_{I,\Pi}(k) \Rightarrow 0\,] \cdot \frac{1}{2} \\ &\quad = \Pr[\,\mathsf{Expt}^{\text{ind-atk-1}}_{I,\Pi}(k) \Rightarrow 1\,] \cdot \frac{1}{2} + \Big(1 - \Pr[\,\mathsf{Expt}^{\text{ind-atk-0}}_{I,\Pi}(k) \Rightarrow 1\,]\Big) \cdot \frac{1}{2} \\ &\quad = \Pr[\,\mathsf{Expt}^{\text{ind-atk-1}}_{I,\Pi}(k) \Rightarrow 1\,] \cdot \frac{1}{2} - \Pr[\,\mathsf{Expt}^{\text{ind-atk-0}}_{I,\Pi}(k) \Rightarrow 1\,] \cdot \frac{1}{2} + \frac{1}{2}\;. \end{aligned}$$

Rearranging terms completes the proof of the lemma. ▮

Some Intuition. One reason the indistinguishability-based formulation of privacy is so-named is that it is equivalent to saying that the output of an adversary when given an encryption of m_0 is computationally indistinguishable from the output of the adversary when given an encryption of m_1, where m_0, m_1 are messages it itself previously produced. Our new notion of indistinguishability under parallel chosen-ciphertext attack extends this in the following way. After an adversary chooses m_0, m_1, allow it to output two things, namely an arbitrary string, and a set of ciphertexts that we shall call the *adversarial ciphertexts*. Our definition requires that the adversary's output, *together with the decryptions of the adversarial ciphertexts*, be computationally indistinguishable in the two cases. (Namely, when it got an encryption of m_0 as input and when it got an encryption of m_1

as input.) In other words, when an encryption of m_0 given to the adversary is replaced with an encryption of m_1, even the *contents* of encrypted messages that the adversary sends can't change in any computationally noticeable way. This might make the relation to non-malleability that we establish more intuitive.

5 Results

We show that the three notions defined above are equivalent.

5.1 Result statements

The following sequence of theorems establishes the equivalence of all three notions discussed above.

Theorem 5.1 [CNM-ATK $\rightarrow$ SNM-ATK] For any ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$, if encryption scheme Π is secure in the sense of CNM-ATK then Π is secure in the sense of SNM-ATK. ▌

The proof of the above is in Section 5.2.

Theorem 5.2 [SNM-ATK $\rightarrow$ IND-PCAX] For any ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$, if encryption scheme Π is secure in the sense of SNM-ATK then Π is secure in the sense of IND-PCAX, where

If ATK = CPA then PCAX = PCA0
If ATK = CCA1 then PCAX = PCA1
If ATK = CCA2 then PCAX = PCA2 ▌

The proof of the above is in Section 5.3.

Theorem 5.3 [IND-PCAX $\rightarrow$ CNM-ATK] For any PCAX $\in \{\mathrm{PCA0}, \mathrm{PCA1}, \mathrm{PCA2}\}$, if encryption scheme Π is secure in the sense of IND-PCAX then Π is secure in the sense of CNM-ATK, where

If PCAX = PCA0 then ATK = CPA
If PCAX = PCA1 then ATK = CCA1
If PCAX = PCA2 then ATK = CCA2 ▌

The proof of the above is in Section 5.4.

5.2 Proof of Theorem 5.1: CNM $\rightarrow$ SNM

Let us first consider the case that ATK = CPA. Namely we claim that CNM-CPA implies SNM-CPA. Intuitively, the CNM-CPA definition can be viewed as requiring that for every adversary A there exist a specific type of simulator, which we can call a "canonical simulator," $A' = (A_1, A_2')$. The first stage, as the notation indicates, is identical to that of A. The second simulator stage A_2' simply chooses a random message from the message space M that was output by A_1, and runs the adversary's second stage A_2 on the encryption of that message. Since A does not have a decryption oracle, A' can indeed do this. With some additional appropriate tailoring we can construct a simulator that meets the conditions of the definition of SNM-CPA.

Let us try to extend this line of thought to CCA1 and CCA2. If we wish to continue to think in terms of the canonical simulator, the difficulty is that this "simulator" would, in running A, now need access to a decryption oracle, which is not allowed under SNM. Thus, it might appear that CNM-ATK is actually weaker than SNM-ATK for ATK $\neq$ CPA, corresponding to the ability to simulate by simulators which are also given the decryption oracle.

However, this appearance is false. In fact, CNM-ATK is not weaker; rather, CNM-ATK implies SNM-ATK for all three types of attacks ATK, including CCA1 and CCA2. (In other words, if a scheme meets the CNM-ATK definition, it is possible to design a simulator according to the SNM-ATK definition.)

Proof of Theorem 5.1: Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the given encryption scheme. Let relation R and adversary $A = (A_1, A_2)$ be given. To show the scheme is secure in the sense of SNM-ATK we need to construct a simulator $S = (S_1, S_2)$. The idea is that S will run A on a newly chosen public key whose corresponding decryption key it knows:

Algorithm $S_1(pk)$	Algorithm $S_2(\tilde{s}_2)$ where $\tilde{s}_2 = (M, s_2, pk, pk', sk')$
$(pk', sk') \xleftarrow{\$} \mathcal{K}(1^k)$	$\tilde{x} \xleftarrow{\$} M$; $\tilde{y} \xleftarrow{\$} \mathcal{E}_{pk'}(\tilde{x})$
$(M, s_1, s_2) \xleftarrow{\$} A_1^{\tilde{\mathcal{O}}_1}(pk')$	$\tilde{\mathbf{y}} \xleftarrow{\$} A_2^{\tilde{\mathcal{O}}_2}(s_2, \tilde{y})$; $\tilde{\mathbf{x}} \leftarrow \mathcal{D}_{sk'}(\tilde{\mathbf{y}})$; $\mathbf{y} \xleftarrow{\$} \mathcal{E}_{pk}(\tilde{\mathbf{x}})$
$\tilde{s}_2 \leftarrow (M, s_2, pk, pk', sk')$	Return $\mathbf{y}$
Return $(M, s_1, \tilde{s}_2)$	

where

If atk = cpa	then	$\tilde{\mathcal{O}}_1(\cdot) = \varepsilon$	and	$\tilde{\mathcal{O}}_2(\cdot) = \varepsilon$
If atk = cca1	then	$\tilde{\mathcal{O}}_1(\cdot) = \mathcal{D}_{sk'}(\cdot)$	and	$\tilde{\mathcal{O}}_2(\cdot) = \varepsilon$
If atk = cca2	then	$\tilde{\mathcal{O}}_1(\cdot) = \mathcal{D}_{sk'}(\cdot)$	and	$\tilde{\mathcal{O}}_2(\cdot) = \mathcal{D}'_{sk'}(\cdot)$

A key point is that the simulator, being in possession of sk', can indeed run A with the stated oracles. (That's how it avoids needing access to the "real" oracles $\mathcal{O}_1, \mathcal{O}_2$ that are provided to A and might depend on sk.) Now we want to show that $\mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi,R}(\cdot)$ is negligible. We will do this using the assumption that Π is secure in the sense of CNM-ATK. To that end, we consider the following adversary $C = (C_1, C_2)$ attacking Π in the sense of CNM-ATK:

Algorithm $C_1^{\mathcal{O}_1}(pk)$	Algorithm $C_2^{\mathcal{O}_2}((M, s_1, s_2), y)$
$(M, s_1, s_2) \xleftarrow{\$} A_1^{\mathcal{O}_1}(pk)$	Define R' by $R'(a, \mathbf{b}) = 1$ iff $R(a, \mathbf{b}, M, s_1) = 1$
Return $(M, (M, s_1, s_2))$	$\mathbf{y} \xleftarrow{\$} A_2^{\mathcal{O}_2}(s_2, y)$
	Return $(R', \mathbf{y})$

It is clear from the definition of C that

$$\Pr[\,\mathsf{Expt}^{\text{cnm-atk-1}}_{C,\Pi}(k) \Rightarrow 1\,] \;=\; \Pr[\,\mathsf{Expt}^{\text{snm-atk}}_{A,\Pi,R}(k) \Rightarrow 1\,]$$

for all $k \in \mathsf{N}$. Now, let us expand the definition of $\mathsf{Expt}^{\text{snm-atk}}_{S,\Pi,R}(k)$, substituting in the definition of S given above. We get the code on the left below:

01.	$(pk, sk) \xleftarrow{\$} \mathcal{K}(1^k)$	
02.	$(pk', sk') \xleftarrow{\$} \mathcal{K}(1^k)$	$(pk', sk') \xleftarrow{\$} \mathcal{K}(1^k)$
03.	$(M, s_1, s_2) \xleftarrow{\$} A_1^{\tilde{\mathcal{O}}_1}(pk')$	$(M, s_1, s_2) \xleftarrow{\$} A_1^{\tilde{\mathcal{O}}_1}(pk')$
04.	$s_2' \leftarrow (M, s_2, pk, pk', sk')$	
05.	$x \xleftarrow{\$} M$; $\tilde{x} \xleftarrow{\$} M$	$x \xleftarrow{\$} M$; $\tilde{x} \xleftarrow{\$} M$
06.	$\tilde{y} \xleftarrow{\$} \mathcal{E}_{pk'}(\tilde{x})$	$\tilde{y} \xleftarrow{\$} \mathcal{E}_{pk'}(\tilde{x})$
07.	$\tilde{\mathbf{y}} \xleftarrow{\$} A_2^{\tilde{\mathcal{O}}_2}(s_2, \tilde{y})$	$\tilde{\mathbf{y}} \xleftarrow{\$} A_2^{\tilde{\mathcal{O}}_2}(s_2, \tilde{y})$
08.	$\tilde{\mathbf{x}} \leftarrow \mathcal{D}_{sk'}(\tilde{\mathbf{y}})$	$\tilde{\mathbf{x}} \leftarrow \mathcal{D}_{sk'}(\tilde{\mathbf{y}})$
09.	$\mathbf{y} \xleftarrow{\$} \mathcal{E}_{pk}(\tilde{\mathbf{x}})$	
10.	$\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$	
11.	If $R(x, \mathbf{x}, M, s_1)$ then return 1	If $R(x, \tilde{\mathbf{x}}, M, s_1)$ then return 1
12.	Else return 0	Else return 0

Examining the code on the left we notice that $\mathbf{x} = \tilde{\mathbf{x}}$. This means that we can substitute $\tilde{\mathbf{x}}$ for $\mathbf{x}$ in line 11. This means $\mathbf{x}, \mathbf{y}$ are not used in determining what the code returns, and thus lines 09, 10 can be dropped. Line 04 can also be dropped, because s_2' is never referred to. Since this means that pk, sk are no longer referred to, line 01 can be dropped as well. The resulting code is on the right above. We see that this code is equivalent to that of $\mathsf{Expt}^{\text{cnm-atk-0}}_{C,\Pi}(k)$, so that

$$\Pr[\,\mathsf{Expt}^{\text{cnm-atk-0}}_{C,\Pi}(k) \Rightarrow 1\,] \;=\; \Pr[\,\mathsf{Expt}^{\text{snm-atk}}_{S,\Pi,R}(k) \Rightarrow 1\,]$$

for all $k \in \mathsf{N}$. Thus for all $k \in \mathsf{N}$ we have

$$\mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi,R}(k) \;=\; \mathsf{Adv}^{\text{cnm-atk}}_{C,\Pi}(k)\,.$$

But Π is assumed secure in the sense of CNM-ATK, so $\mathsf{Adv}^{\text{cnm-atk}}_{C,\Pi}(\cdot)$ is negligible. The above implies that $\mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi,R}(\cdot)$ is negligible too. So Π is secure in the sense of SNM-ATK. ▮

5.3 Proof of Theorem 5.2: SNM-ATK → IND-PCAX

The case that ATK = CCA2 is easy since, as we have already noted, IND-PCA2 = IND-CCA2, and thus our claim is simply that SNM-CCA2 → IND-CCA2, which can be shown just as in [6]. Let us discuss the interesting case, namely when ATK $\in$ {CPA, CCA1}.

We are assuming that encryption scheme Π is secure in the SNM-ATK sense. We will show it is also secure in the IND-PCAX sense. Let $I = (I_1, I_2, I_3)$ be an IND-PCAX adversary attacking Π. We want to show that $\mathsf{Adv}^{\text{ind-atk}}_{I,\Pi}(\cdot)$ is negligible. To this end, we describe a relation R and an SNM-ATK adversary $A = (A_1, A_2)$ attacking Π using R. We wish to show that A will have the same advantage attacking Π using R as I has as an IND-PCAX adversary using a parallel attack. What allows us to do this is to pick the relation R to capture the success condition of I's parallel attack.

To get some intuition it is best to think of ATK = CPA, meaning A is allowed only a chosen-plaintext attack. However, I has (limited) access to a decryption oracle; it is allowed the parallel query. How then can A "simulate" I? The key observation is that the non-malleability goal involves an "implicit" ciphertext attack on the part of the adversary, even under CPA. This arises from the ciphertext vector $\mathbf{y}$ that such an adversary outputs. It gets decrypted, and the results are fed into the relation R. Thus, the idea of our proof is to make A output, as its final response, the parallel query that I will make. Now, I would expect to get back the response and compute with it, which A can't do; once it has output its final ciphertext, it stops, and the relation R gets evaluated on the corresponding plaintext. So we define R in such a way that it "completes" I's computation.

A useful way to think about this is as if A were trying to "communicate" with R, passing it the information that R needs to execute I.

Notice that for this to work it is crucial that I's query is a parallel one. If I were making the usual adaptive queries, A could not output a single ciphertext vector, because it would have to know the decryption of the first ciphertext query before it would even know the ciphertext which is the second query. Yet, for the non-malleability game, A must output a single vector.

This is the rough idea. There are a couple of subtleties. R needs to know several pieces of information that depend on the computation of some stages of I, such as coin tosses. A must communicate them to R. The only mechanism that A has to communicate with R is via the ciphertext vector $\mathbf{y}$ that A outputs, whose decryption is fed to R. So any information that A wants to pass along, it encrypts and puts in this vector.

Now let us provide a more complete description.

Proof of Theorem 5.2: Suppose we are given IND-PCAX adversary $I = (I_1, I_2, I_3)$. We assume that the two messages m_0, m_1 output by I_1 are always distinct. This is wlog because we can always modify I to make this true without decreasing its advantage, as follows: any time I_1 outputs equal messages, we have it instead output some unequal ones, say 0 and 1, and have I_3 return a random bit as its guess.

The first case in the proof is that ATK $\in$ {CPA, CCA1}. We define SNM-ATK adversary $A = (A_1, A_2)$ as follows:

Algorithm $A_1^{\mathcal{O}_1}(pk)$	Algorithm $A_2(s_2, y)$ where $s_2 = (m_0, m_1, t_1, pk)$
$(m_0, m_1, t_1) \xleftarrow{\$} I_1^{\mathcal{O}_1}(pk)$	$(\mathbf{c}, t_2) \xleftarrow{\$} I_2(m_0, m_1, t_1, y)$
$M \leftarrow \{m_0, m_1\}$	Choose random coins σ for I_3
$s_1 \leftarrow (m_0, m_1)$	$e_1 \xleftarrow{\$} \mathcal{E}_{pk}(t_2)$; $e_2 \xleftarrow{\$} \mathcal{E}_{pk}(\sigma)$
$s_2 \leftarrow (m_0, m_1, t_1, pk)$	$\mathbf{y} \leftarrow (e_1, e_2, \mathbf{c}[1], \ldots, \mathbf{c}[\lvert\mathbf{c}\rvert])$
Return (M, s_1, s_2)	Return $\mathbf{y}$

Above, by $M \leftarrow \{m_0, m_1\}$ we mean that M is (a canonical encoding of) the message space that puts a uniform distribution on the set $\{m_0, m_1\}$. Notice above that A_2 picks coins σ for I_3. We can think of each stage of I as picking its own coins afresh, since any information needing to be communicated from stage to stage is passed along in the state information.

Before making any claims about security, we need to address a technical point. We need A to be legitimate, which means we require that $y \notin \{e_1, e_2\}$. To ensure this, we can modify I if necessary to ensure that $t_2 \notin \{m_0, m_1\}$ and $\sigma \notin \{m_0, m_1\}$ —and hence that $y \notin \{e_1, e_2\}$— without affecting its advantage. We could do this, for example, by ensuring that $|t_2| > l$ and $|\sigma| > l$ where $l = |m_0| = |m_1|$. In the first case, just append some extra bits to t_2 and ask the algorithm I_3 that uses t_2 to ignore these bits. In the second case, increase the length of the random tape of I_3. (The algorithm can use the appropriate prefix.)

Now, let us specify the relation R:

Relation $R(x, \mathbf{x}, M, s_1)$
    If s_1 is not a pair of distinct strings then return `false`
    Let m_0, m_1 be such that $s_1 = (m_0, m_1)$
    If $|\mathbf{x}| < 2$ then return `false`
    $t_2 \leftarrow \mathbf{x}[1]$; $\sigma \leftarrow \mathbf{x}[2]$; $\mathbf{p} \leftarrow (\mathbf{x}[3], \ldots, \mathbf{x}[|\mathbf{x}|])$

If $M \neq \{m_0, m_1\}$ then return false
If $x = m_0$ then $b \leftarrow 0$ else $b \leftarrow 1$
$g \leftarrow I_3(\mathbf{p}, t_2; \sigma)$
If $g = b$ then return true else return false

The 5th line above tests that M is a canonical encoding of the message space that puts a uniform distribution on $\{m_0, m_1\}$, where m_0, m_1 are the distinct messages specified by s_1. Notice that R is polynomial time computable. Also notice we use the assumption that ATK $\neq$ CCA2, for otherwise I_3 needs a decryption oracle and R could not execute it.

We claim that

$$\Pr[\,\mathsf{Expt}^{\text{ind-atk-b}}_{I,\Pi}(k) \Rightarrow b\,] \;=\; \Pr[\,\mathsf{Expt}^{\text{snm-atk}}_{A,\Pi,R}(k) \Rightarrow 1\,] \tag{1}$$

for all $k \in \mathsf{N}$, where the probability is over a random choice of b from $\{0, 1\}$ and the coins of the experiments. This can be seen by examining the experiments in question and using the definitions of R and A above. Notice that we use here the fact that the messages m_0, m_1 are always distinct, which tells us that the bit b computed by the relation identifies the challenge message.

The assumption that Π is secure in the sense of SNM-ATK tell us that there is a polynomial-time simulator $S = (S_1, S_2)$ such that $\mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi,R}(\cdot)$ is negligible. We claim that

$$\Pr[\,\mathsf{Expt}^{\text{snm-atk}}_{S,\Pi,R}(k) \Rightarrow 1\,] \;\leq\; \frac{1}{2} \tag{2}$$

for all $k \in \mathsf{N}$. This is justified as follows. By construction, in order to satisfy R, the first stage S_1 of S must set its output state s_1 to be a pair of distinct strings which we now denote (m_0, m_1). Also, it must set M to the uniform distribution on $\{m_0, m_1\}$, and, to be valid, must make the lengths of m_0 and m_1 the same. Let p be the probability with which all this happens. Let $x = m_c$ be the random message chosen in $\mathsf{Expt}^{\text{snm-atk}}_{S,\Pi,R}(k)$. So c is a random bit. Then the bit b computed by R equals c. On the other hand, S_2 gets no information about x, and thus b. So the ciphertext vector $\mathbf{y}$ that S_2 outputs, and hence the inputs $\mathbf{p}, t_2$ and coins σ on which R runs I_3, are independent of b, and hence so is the bit g output by I_3. Thus the probability that $b = g$ is $p/2 \leq 1/2$.

Now, for any $k \in \mathsf{N}$ we have the following, where the probability is over a random choice of b from $\{0, 1\}$ and the coins of the experiments:

$$\begin{aligned}
\mathsf{Adv}^{\text{ind-atk}}_{I,\Pi}(k) &= 2 \cdot \Pr[\,\mathsf{Expt}^{\text{ind-atk-b}}_{I,\Pi}(k) \Rightarrow b\,] - 1 \\
&= 2 \cdot \Pr[\,\mathsf{Expt}^{\text{snm-atk}}_{A,\Pi,R}(k) \Rightarrow 1\,] - 1 \\
&\leq 2 \cdot \Pr[\,\mathsf{Expt}^{\text{snm-atk}}_{A,\Pi,R}(k) \Rightarrow 1\,] - 2 \cdot \Pr[\,\mathsf{Expt}^{\text{snm-atk}}_{S,\Pi,R}(k) \Rightarrow 1\,] \\
&= 2 \cdot \mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi,R}(k)\;.
\end{aligned}$$

The first equation above is by Definition 4.1. The second uses Equation (1). In the next step we used Equation (2), and lastly Definition 3.1. Finally, since $\mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi,R}(\cdot)$ is negligible, the proof is complete for the case that ATK $\in \{\text{CPA}, \text{CCA1}\}$.

We now consider the case that ATK = CCA2. With $I = (I_1, I_2, I_3)$ as above, we construct $A = (A_1, A_2)$ as follows:

Algorithm $A_1^{\mathcal{O}_1}(pk)$	Algorithm $A_2^{\mathcal{O}_2}(s_2, y)$ where $s_2 = (m_0, m_1, t_1, pk)$
$(m_0, m_1, t_1) \xleftarrow{\$} I_1^{\mathcal{O}_1}(pk)$	$(\mathbf{c}, t_2) \xleftarrow{\$} I_2^{\mathcal{O}_2}(m_0, m_1, t_1, y)$
$M \leftarrow \{m_0, m_1\}$	$\mathbf{p} \leftarrow \mathcal{D}_{sk}(\mathbf{c})$
$s_1 \leftarrow (m_0, m_1)$	$g \xleftarrow{\$} I_3^{\mathcal{O}_2}(\mathbf{p}, t_2)$
$s_2 \leftarrow (m_0, m_1, t_1, pk)$	If $g = 0$ then $\mathbf{y}[1] \xleftarrow{\$} \mathcal{E}_{pk}(0m_0)$ else $\mathbf{y}[1] \xleftarrow{\$} \mathcal{E}_{pk}(0m_1)$
Return (M, s_1, s_2)	Return $\mathbf{y}$

The second line of I_2 is implemented via calls to the decryption oracle $\mathcal{O}_2$. We are denoting by $0a$ the string obtained by pre-pending a 0 bit to a string a. Since $|m_0| = |m_1|$ we have $\{m_0, m_1\} \cap \{0m_0, 0m_1\} = \emptyset$, which implies that A is legitimate. Now the relation R is

`Relation` $R(x, \mathbf{x}, M, s_1)$
  If s_1 is not a pair of distinct strings then return `false`
  Let m_0, m_1 be such that $s_1 = (m_0, m_1)$
  If $M \neq \{m_0, m_1\}$ then return `false`
  If $|\mathbf{x}| \neq 1$ then return `false`
  If $\mathbf{x}[1] = 0x$ then return `true` else return `false`

Notice that R is polynomial time computable. The assumption that Π is secure in the sense of SNM-ATK tell us that there is a polynomial-time simulator $S = (S_1, S_2)$ such that $\mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi,R}(\cdot)$ is negligible. Equations (1) and (2) are true just as above, and hence the proof is concluded in the same way. ▌

5.4 Proof of Theorem 5.3: IND-PCAX $\rightarrow$ CNM-ATK

We are assuming that Π is secure in the IND-PCAX sense. We will show it is also secure in the CNM-ATK sense.

Let $C = (C_1, C_2)$ be an CNM-ATK adversary attacking Π. We will present an IND-PCAX adversary $I = (I_1, I_2, I_3)$ attacking Π whose advantage is at least that of C. The intuition is simple: since I has access to a parallel decryption oracle in the second stage, she can decrypt the ciphertexts that C outputs, and check herself to see if C's relation holds. The construction and analysis follow.

Proof of Theorem 5.3: Given the CNM-ATK adversary $C = (C_1, C_2)$, we define the IND-PCAX adversary $I = (I_1, I_2, I_3)$ as follows:

Algorithm $I_1^{\mathcal{O}_1}(pk)$	Algorithm $I_2^{\mathcal{O}_2}(m_0, m_1, t, y)$	Algorithm $I_3^{\mathcal{O}_2}(\mathbf{p}, (R, m_1))$
$(M, t) \xleftarrow{\$} C_1^{\mathcal{O}_1}(pk)$	$(R, \mathbf{c}) \xleftarrow{\$} C_2^{\mathcal{O}_2}(t, y)$	If $R(m_1, \mathbf{p})$ then $g \leftarrow 1$
$m_0 \xleftarrow{\$} M$; $m_1 \xleftarrow{\$} M$	Return $(\mathbf{c}, (R, m_1))$	Else $g \leftarrow 0$
Return (m_0, m_1, t)		Return g

By examination of the experiments involved one can check that

$$\Pr[\,\mathsf{Expt}^{\text{ind-atk-1}}_{I,\Pi}(k) \Rightarrow 1\,] \;=\; \Pr[\,\mathsf{Expt}^{\text{cnm-atk-1}}_{C,\Pi}(k) \Rightarrow 1\,] \qquad (3)$$

$$\Pr[\,\mathsf{Expt}^{\text{ind-atk-0}}_{I,\Pi}(k) \Rightarrow 1\,] \;=\; \Pr[\,\mathsf{Expt}^{\text{cnm-atk-0}}_{C,\Pi}(k) \Rightarrow 1\,] \qquad (4)$$

for all $k \in \mathsf{N}$. Subtracting, and using Lemma 4.2, we get

$$\mathsf{Adv}^{\text{ind-atk}}_{I,\Pi}(k) \;=\; \mathsf{Adv}^{\text{cnm-atk}}_{C,\Pi}(k)$$

for all $k \in \mathsf{N}$. But $\mathsf{Adv}^{\text{ind-atk}}_{I,\Pi}(\cdot)$ is negligible by assumption, hence so is $\mathsf{Adv}^{\text{cnm-atk}}_{C,\Pi}(\cdot)$. ▮

6 Discussion, Relations and Extensions

We discuss the relations of our definitions to other ones. Let us begin with some definitions.

The SNM-ATK* and CNM-ATK* notions. In our SNM-ATK and CNM-ATK definitions, we assume $R(x, \mathbf{x}, M, s_1)$ and $R(x, \mathbf{x})$, respectively, are defined also when $\perp \in \mathbf{x}$, so that the adversary may be successful even when $\perp \in \mathcal{D}_{sk}(\mathbf{y})$. In contrast, in the corresponding original definitions of [6] and [2], the situation is different. In [2], the adversary is considered unsuccessful if it outputs $\mathbf{y}$ such that $\perp \in \mathcal{D}_{sk}(\mathbf{y})$, and in [6], there is some ambiguity about this case. To better discuss these variants, let us provide some formal definitions. Let SNM-ATK* be defined by replacing "$R(x, \mathbf{x}, M, s_1)$" by "$R(x, \mathbf{x}, M, s_1)$ and $\perp \notin \mathbf{x}$" in the next-to-last lines of both experiments in Definition 3.1. Similarly, let CNM-ATK* be defined by replacing "$R(x, \mathbf{x})$" by "$R(x, \mathbf{x})$ and $\perp \notin \mathbf{x}$" in the last lines of both experiments in Definition 3.2.

The intuition behind the SNM-ATK*, CNM-ATK* definitions is that the receiver will not take any action on an invalid ciphertext. The SNM-ATK, CNM-ATK definitions, however, reflect the view that a receiver might take some action even on invalid ciphertexts, or that a sender may learn that it sent an invalid ciphertext due to the receiver's inaction. Thus it is best left to the relation to determine whether the adversary wins or loses even when $\perp \in \mathbf{y}$.

Relation to original notions. CNM-ATK* is exactly the notion of [2]. Regarding whether SNM-ATK* is the notion of [6], there is some ambiguity. Lindell [13] has pointed out that one could interpret [6] as simply not allowing the adversary to output $\mathbf{y}$ such that $\perp \in \mathcal{D}_{sk}(\mathbf{y})$. In our language, consider the notion of a legitimate adversary being further constrained to require that it never output $\mathbf{y}$ with $\perp \in \mathcal{D}_{sk}(\mathbf{y})$. Then this *exclusive* version of the definition quantifies only over adversaries that are legitimate in this sense. Lindell [13] shows that the DDN-Lite encryption scheme [7] is secure under the (CPA case of the) exclusive definition. He also provides an attack showing this scheme is not secure in the sense of SNM-CPA, thereby separating these notions. It is however not known whether or not this scheme is secure in the sense of SNM-CPA* or CNM-CPA*.

To us, the merit of the exclusive formalization is questionable. The legitimacy conditions we ourselves have imposed on the adversary are for convenience and simplicity only. It would be equivalent to have the experiments return 0 if they are not met, because given an illegitimate adversary, it is possible to construct a legitimate adversary with about the same running time and the same or greater advantage. This does not appear to be true for the exclusive definition. Intuitively, our view is that an adversary should itself be able to check whether or not it is behaving legitimately. This is true under our notions of legitimacy but does not seem true under that of the exclusive definition.

We would like to add that we were not able to fully understand the intent of [6] at several points. In fact we believe that one of the contributions of our paper is to pin down formal and unambiguous definitions of simulation-based non-malleable encryption. But SNM-ATK* and SNM-ATK represent our best guesses as to what [6] meant in some details and thus might differ from the intent of [6] in more ways than we have discussed.

In any case, since we believe that the SNM-ATK and SNM-ATK* notions appropriately capture the issue of whether or not the adversary is successful when $\perp \in \mathcal{D}_{sk}(\mathbf{y})$ in the simulation-based setting, and since this issue is interesting, we propose now to further consider the SNM-ATK* and CNM-ATK* notions, the merits of our notions relative to these, and how all these notions relate.

Non-malleability for multiple ciphertexts. Lindell [13] points to an advantage of the

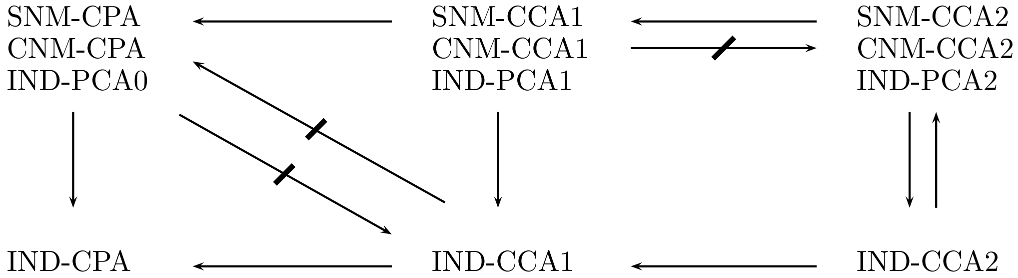

Figure 1: Relations between notions of encryption. An arrow A $\rightarrow$ B is an implication, meaning every scheme meeting notion A also meets notion B, while a barred arrow A $\not\rightarrow B$ is a separation, meaning that (under the assumption that there is a scheme meeting notion A) there is a scheme meeting notion A but not notion B. Only a minimal set of arrows and barred arrows is shown; others can be inferred. The picture is complete in the sense that it implies either an implication or a separation between any pair of notions.

---

stronger SNM-ATK, CNM-ATK notions we have introduced, namely that they imply non-malleability of the encryption of multiple ciphertexts. The notion being considered here is an extension of the usual non-malleability framework in which the message-space M returns a vector of messages rather than a single one. A corresponding vector of challenge ciphertexts is then generated for the adversary, based on which it computes $\mathbf{y}$ as before. Interestingly, the easy way to see that the new notions imply non-malleability of the encryption of multiple ciphertexts is to use our characterization. One observes that our results extend to the multiple encryption setting, and then observes that security of the extended IND-ATK notion is implied by security of the original one. This again points to the value of indistinguishability-based characterizations in reasoning about non-malleability.

We do not know whether SNM-ATK* or CNM-ATK* imply corresponding non-malleability for multiple ciphertexts. However, Lindell [13] shows that the exclusive version of the (CPA case of the) definition of [6] does not imply (the corresponding version of) non-malleability for multiple ciphertexts. He obtains this result by observing that the DDN-Lite is not non-malleable (in this sense) for multiple ciphertexts. (But, as we already noted, he has shown it meets the exclusive version of the [6] definition.)

Relations among notions of encryption. Bellare, Desai, Pointcheval and Rogaway [2] consider six notions of encryption, namely $\text{CNM-ATK}^*, \text{IND-ATK}$ for $\text{ATK} \in \{\text{CPA}, \text{CCA1}, \text{CCA2}\}$, and relate them, showing either an implication or a separation between each pair. Dolev, Dwork and Naor [6] do the same for their three notions of non-malleability and the notions IND-ATK for $\text{ATK} \in \{\text{CPA}, \text{CCA1}, \text{CCA2}\}$. (And the relation pictures are the same in both cases, meaning up to the interchange of CNM-ATK* with the corresponding notion of [6].) Given that our notions of non-malleability are different from the ones used in either of these papers, we revisit this question. As indicated in Figure 1, however, we show that the same relations continue to hold, meaning the picture is the same as before modulo the substitution of the new notions of non-malleability. However, we can use our characterization in terms of indistinguishability under parallel attack to give proofs that are somewhat simpler than those of [2, 6]. Let us now provide some details.

By our results, we can work with the IND-PCAX notions, avoiding the need to reason directly about non-malleability. This means we are considering relations between five notions of indistinguishability, namely indistinguishability under the attacks $\text{CPA}, \text{PCA0}, \text{CCA1}, \text{PCA1}$ and $\text{CCA2} = \text{PCA2}$. This makes all the implications in Figure 1 obvious, for each implication A $\rightarrow$ B is simply of IND with the attack for A being at least as strong as that for B.

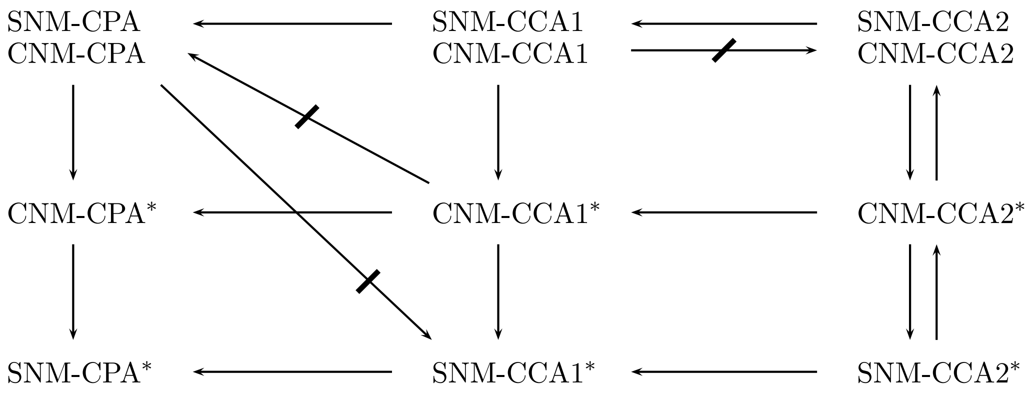

Figure 2: Relations between notions of non-malleable encryption. An arrow A $\rightarrow$ B is an implication, meaning every scheme meeting notion A also meets notion B, while a barred arrow A $\not\rightarrow$ B is a separation, meaning that (under the assumption that there is a scheme meeting notion A) there is a scheme meeting notion A but not notion B. Not all possible relations are resolved; several remain open questions.

Now we turn to the separations. We know that IND-CCA1 $\not\rightarrow$ CNM-CPA* [2] and IND-PCA0 = CNM-CPA $\rightarrow$ CNM-CPA* (Figure 2), whence IND-CCA1 $\not\rightarrow$ IND-PCA0. The other separations need to be revisited.

To establish a separation A $\not\rightarrow$ B we need to take a scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ secure in the sense of A and modify it to a new scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ secure in the sense of A but not B. To show IND-PCA0 $\not\rightarrow$ IND-CCA1, the construction given in [2] to show CNM-CPA* $\not\rightarrow$ IND-CCA1 continues to work but the proof is a little easier with IND-PCA0. Similarly, to show IND-PCA1 $\not\rightarrow$ IND-PCA2 = IND-CCA, the construction given in [2] to show CNM-CCA1 $\not\rightarrow$ IND-CCA continues to work but the proof is a little easier with IND-PCA1.

Relations between non-malleability notions. The next question we ask is how the variants of non-malleability we have just defined relate to our definitions and to each other. In particular, is SNM-ATK* equivalent to SNM-ATK? Is CNM-ATK* equivalent to CNM-ATK? Are SNM-ATK*, CNM-ATK* equivalent? Figure 2 summarizes whatever we know about this, and we now discuss it.

Figure 2 resolves all relations between the first and second row notions, and also all relations between the first and third row notions. This can be seen by following arrows. For example, CNM-CPA $\not\rightarrow$ CNM-CCA1* because otherwise we would get CNM-CPA $\rightarrow$ SNM-CCA1*, contradicting a shown separation. What is left open is relations between some second and third row notions. For example, are SNM-ATK* and CNM-ATK* equivalent? This comes down to asking whether or not SNM-CPA* implies CNM-CPA* and whether or not SNM-CCA1* implies CNM-CCA1*. Now let us discuss how the shown relations are obtained.

The proof of Theorem 5.1 extends to show that CNM-ATK* $\rightarrow$ SNM-ATK* for all ATK $\in$ $\{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$, justifying the corresponding implications in Figure 2. The other implications are either trivial, results in this paper, or from [6, 2]. The CNM-CCA1 $\not\rightarrow$ CNM-CCA2 separation is inherited from Figure 1.

Herranz, Hofheinz and Kiltz showed that CNM-CPA* $\not\rightarrow$ CNM-CPA. Very roughly, the idea is to modify a CNM-CPA* scheme so that it remains CNM-CPA* but there are certain special ciphertexts such that the decryption of some random half of them XOR to the secret key, while the other half decrypt to $\perp$. Their constructed scheme is however not CNM-CCA1 so this does not show that CNM-CCA1* $\not\to$ CNM-CCA1. An extension by Bellare shows not only this but something stronger, namely that CNM-CCA1 does not even imply CNM-CPA, and this is the separation shown in Figure 2 since it implies the two others we have just discussed. The extension uses the same secret sharing idea except that the XOR of the decryptions of certain related ciphertexts is the message underlying the "main" ciphertext rather than the secret key. To make this work, the coins for the secret sharing are obtained by applying a PRF to a part of the ciphertext. Both the original result and the extension appear in [11]. Note that DDN-Lite does not serve to show that CNM-CPA* $\not\to$ CNM-CPA or SNM-CPA* $\not\to$ SNM-CPA, because, although it is not CNM-CPA [13], it is not known to be SNM-CPA*.

There remains only to justify the CNM-CPA $\not\to$ SNM-CCA1* separation shown in Figure 2. From Figure 1 we know that CNM-CPA $\not\to$ IND-CCA1. But SNM-CCA1* $\to$ IND-CCA1, so it must be that CNM-CPA $\not\to$ SNM-CCA1*.

Attempting to extend our results. There are some tempting ways to attempt to extend our proofs to show that SNM-ATK* and CNM-ATK* are equivalent for all ATK. It is worth discussing these to see where they fail. In the following, ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}\}$.

The proof of Theorem 5.2, showing that SNM-ATK $\to$ IND-PCAX, does not directly extend to show that SNM-ATK* $\to$ IND-PCAX. The reason is that an IND-PCAX adversary can submit a parallel decryption query $\mathbf{c}$ such that $\perp \in \mathbf{c}$ and utilize the response in its third stage. We need the relation R to execute this third stage, and would not be able to do this if we have to give up when $\mathbf{c}$ contains $\perp$. There is an obvious way to attempt to fix this, however. Namely, modify the definition of IND-PCAX to IND-PCAX*, where the latter replaces the last line in Definition 4.1 with: "If $\perp \notin \mathbf{p}$ then return g else return $1-b$." In other words, here too, if the adversary makes a parallel decryption query containing an invalid ciphertext, it automatically loses. Now, our proof shows that SNM-ATK* $\to$ IND-PCAX*. So the final question is whether our proof of Theorem 5.3, showing that IND-PCAX $\to$ CNM-ATK, extends to show that IND-PCAX* $\to$ CNM-ATK*. The obvious modification to make is that, in the definition of I_3, replace "If $R(m_1, \mathbf{p})$ then $g \leftarrow 1$" with "If $R(m_1, \mathbf{p})$ and $\perp \notin \mathbf{p}$ then $g \leftarrow 1$". But now, although Equation (3) remains true, Equation (4) may not be true.

Acknowledgments

We thank Yehuda Lindell for his communications [13] and also for many subsequent clarifying discussions. We thank Herranz, Hofheinz and Kiltz [11] for permission to discuss their results.

References

[1] M. Bellare, R. Canetti and H. Krawczyk, A modular approach to the design and analysis of authentication and key exchange protocols. *Proceedings of the 30th Annual Symposium on the Theory of Computing*, ACM, 1998.

[2] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, Relations among notions of security for public-key encryption schemes. *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[3] M. Bellare and P. Rogaway, Optimal asymmetric encryption – How to encrypt with RSA. *Advances in Cryptology – EUROCRYPT '94*, Lecture Notes in Computer Science Vol. 950, A. De Santis ed., Springer-Verlag, 1994.

[4] M. Bellare and A. Sahai, Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. *Advances in Cryptology – CRYPTO '99*, Lecture Notes in Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999. Preliminary version of our paper.

[5] R. Cramer and V. Shoup, A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[6] D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography. *SIAM Journal of Computing*, Vol. 30, No. 2, 2000, pp. 391–437.

[7] C. Dwork, The Non-malleability lectures. Course Notes for CS 359, Stanford University, Spring 1999. Available from author.

[8] O. Goldreich, A uniform complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, Vol. 6, 1993, pp. 21–53.

[9] S. Goldwasser and S. Micali, Probabilistic encryption. *Journal of Computer and System Sciences*, Vol. 28, No. 2, 1984, pp. 270–299.

[10] S. Halevi and H. Krawczyk, Public-key cryptography and password protocols. *Proceedings of the 5th Annual Conference on Computer and Communications Security*, ACM, 1998.

[11] J. Herranz, D. Hofheinz and E. Kiltz, KEM/DEM: Necessary and sufficient conditions for secure hybrid encryption, Manuscript in preparation, June 2006.

[12] J. Katz and M. Yung, Characterization of security notions for probabilistic private-key encryption. *Journal of Cryptology*, Vol. 19, No. 1, 2006, pp. 67–96.

[13] Y. Lindell. Private communication, May 2003.

[14] S. Micali, C. Rackoff and R. Sloan, The notion of security for probabilistic cryptosystems. *SIAM J. of Computing*, Vol. 17, No. 2, April 1988, pp. 412–426.

[15] M. Naor and M. Yung, Public-key cryptosystems provably secure against chosen ciphertext attacks. *Proceedings of the 22nd Annual Symposium on the Theory of Computing*, ACM, 1990.

[16] C. Rackoff and D. Simon, Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. *Advances in Cryptology – CRYPTO '91*, Lecture Notes in Computer Science Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.

[17] A. Yao, Theory and applications of trapdoor functions. *Proceedings of the 23rd Symposium on Foundations of Computer Science*, IEEE, 1982.