Appears in *Advances in Cryptology – Crypto 99 Proceedings*, Lecture Notes in Computer Science Vol. ??, ed., Springer-Verlag, 1999.

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

Mihir Bellare* Amit Sahai†

February 1999

Abstract

We prove the equivalence of two definitions of non-malleable encryption appearing in the literature— the original one of Dolev, Dwork and Naor and the later one of Bellare, Desai, Pointcheval and Rogaway. 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, California 92093, USA. E-Mail: mihir@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/mihir. Supported in part by NSF CAREER Award CCR-9624439 and a 1996 Packard Foundation Fellowship in Science and Engineering.

† MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139, USA. E-Mail: amits@theory.lcs.mit.edu. Supported by a DOD/NDSEG Graduate Fellowship and partially by DARPA grant DABT-96-C-0018.

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. Goldwasser and Micali's notion of indistinguishability [8] forms the accepted formalization of this goal. A second goal, introduced by Dolev, Dwork and Naor [5], 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 [11, 12].

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]) or electronic payment (cf. [13]). The interest is witnessed by attention to classification of the notions of encryption [2] and new efficient constructions of non-malleable schemes [3, 4].

In our discussions below, we begin for simplicity by focusing on the case where the notions are considered under chosen-plaintext attacks. We will discuss the extensions to stronger attacks later.

1.1 Themes in foundations of encryption

EQUIVALENCES. We said above that indistinguishability was the "accepted" formalization of the privacy goal. This agreement is due in large part to a body of work that has established that numerous other formalizations put forth to capture privacy are actually equivalent to indistinguishability. In particular this is true of semantic security [8] and for a notion of privacy based on computational entropy [14, 10]. These foundational results have since been refined and extended to other settings [7]. These equivalences are a cornerstone of our understating of privacy, providing evidence that we have in fact found the "right" formalization.

CHARACTERIZATIONS. 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. For this reason, it is somewhat difficult to use in 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 a characterization that is simple and easy to work with, a role which for privacy is played by the notion of indistinguishability.

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 different formalizations to capture the notion of non-malleable encryption. The first is the original definition of Dolev, Dwork and Naor [5], which we call SNM (simulation based non-malleability). A second, somewhat simpler formalization was introduced by Bellare, Desai, Pointcheval and Rogaway [2], and we call it CNM (comparison based non-malleability). A priori, at least, the two seem to have important differences.

Second, there is no simple 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 [5]), 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 formalizations. (Formal definitions of both notions can be found in Section 3.) 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 $\mathbf{y}$. 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 SNM notion 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 CNM formalization, there is no simulator: it is required instead 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 SNM notion does not allow the simulator access to the decryption oracle even when the adversary gets it, while the CNM notion allows the adversary access to the decryption oracle in both games. These differences would raise a priori doubts about whether the two notions could be equivalent. In particular, SNM would appear to be stronger since it asks for simulation even without access to decryption ability.

1.3 The equivalence

In this paper we show that the above two notions of non-malleability are in fact equivalent. (This holds under all three kinds of attack mentioned above.) In other words, if a particular encryption scheme meets the SNM notion of security, then it also meets the CNM notion, and vice versa.

Thus, we are saying that two formalizations of non-malleability, that were originally proposed with somewhat different intuitions behind them, 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 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 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. This represents a tradeoff between goals and attacks.

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 chosen-ciphertext attack in which 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. We call this type of attack a "parallel chosen-ciphertext attack". 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.

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.

Recall that non-malleability at first glance is capturing a very different sort of notion than indistinguishability: the ability to modify a ciphertext in some way so as to create plaintext dependencies. This can be done even if one doesn't "know" the plaintexts involved, so it seems different from privacy. Our characterization brings out the fact that the difference is not so great: the increased severity of the goal of non-malleability can be compensated for by a strong attack under indistinguishability.

1.5 Extensions and discussion

We have focused above mostly on non-malleability under chosen-plaintext attack. Let us briefly discuss the extensions of the results to the two kinds of chosen-ciphertext attack. They are referred to in [2] as CCA1 and CCA2, and correspond to lunch-time [11] and adaptive [12] attacks.

As mentioned above, our equivalence result extends to the stronger attack forms. However, for the case of the strongest type of attack, namely CCA2, the result is not really a novelty, because it can be derived as a consequence of the results of [2] and [6]. (Specifically each of these showed that their notion of non-malleability under CCA2 is equivalent to indistinguishability under CCA2, so this makes the two notions of non-malleability under CCA2 equivalent to each other.) Thus the interest of our results is largely for the case of chosen-plaintext attack and CCA1.

A similar situation holds with regard to the characterization. It is simple to provide an appropriate extension of indistinguishability under parallel attack to the CCA1 and CCA2 settings, and we can show the characterization holds. But adding a parallel attack to CCA2 leaves the latter unchanged, and our results just collapses to the already know equivalence between non-malleability and indistinguishability in the CCA2 case.

1.6 Relations among notions of security

Our new characterization of non-malleability as indistinguishability under a parallel chosen-ciphertext attack simplifies the discussion of relationships among the notions of security studied in [2, 6]. Our characterization shows that non-malleability with respect to chosen-plaintext attack, lunchtime chosen-ciphertext attack, or adaptive chosen-ciphertext attack is equivalent to indistinguishability with respect to a parallel chosen-ciphertext attack (denoted PA0), a lunchtime attack followed by a parallel attack (denoted PA1), or an adaptive chosen ciphertext attack (denote CCA2), respectively. Thus, by our equivalence, the six standard notions of security are equivalent to indistinguishability with respect to five different types of attack, denoted $\mathrm{CPA}, \mathrm{PA0}, \mathrm{CCA1}, \mathrm{PA1}$, and CCA2.

Now, to show that these notions are all distinct, it suffices to show for each pair of notions that an encryption scheme secure against the weaker form of attack can be modified to fail against the stronger attack, but still be secure against the weaker form of attack. Using our new characterization one can in fact separate all these notions by simply following the following guidelines: To make a system insecure under a lunchtime attack, one simply needs to fix a particular ciphertext which decrypts to the secret key. In order to require that a non-parallel attack be used to do this, one may add a level of indirection by having a particular ciphertext that decrypts to another randomly chosen ciphertext, whose decryption is the secret key. To modify a system that is secure against lunchtime attacks in order to make it fail against an adaptive parallel attack, one simply establishes a rule that a ciphertext can be modified in a canonical manner to produce a new ciphertext that decrypts to the same value. Again, to require that a non-parallel attack be used, one may again add a level of indirection, by having the modified ciphertext decrypt to another ciphertext, which decrypts to the original message. These intuitions, which are very natural given the attacks, can readily be made into proofs which are simpler than those given by [2] to accomplish this goal.

Note also that each of [2] and [6] established relations using their own definitions of non-malleability. Their results are unified by our results showing the two notions of non-malleability are the same.

Above, we only consider parallel chosen-ciphertext attacks in the adaptive stage of the attack, *i.e.* when the adversary has seen the challenge ciphertext. However, it is quite natural to consider a parallel attack as also possible in the first stage of the attack, before the challenge ciphertext is known. This leads naturally to nine different notions of security against chosen-ciphertext attacks. It could be interesting to investigate these attacks and any relations which may exist among them.

1.7 Related work

Halevi and Krawczyk introduce a weak version of chosen-ciphertext attack which they call a one-ciphertext *verification* attack [9]. 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. No equivalences are proved in [9].

2 Preliminaries

Experiments. We use standard notations and conventions for writing probabilistic algorithms and experiments. If A is a probabilistic 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 $y \leftarrow A(x_1, x_2, \ldots)$ denote the experiment of picking r at random and letting y be $A(x_1, x_2, \ldots; r)$. If S is a finite set then $x \leftarrow S$ is the operation of picking an element uniformly from S. If α is neither an algorithm nor a set then $x \leftarrow \alpha$ is a simple assignment statement. We say that y *can be output by* $A(x_1, x_2, \ldots)$ if there is some r such that $A(x_1, x_2, \ldots; r) = y$. Also, to clarify complicated probabilistic statements, we define random variables as experiments.

Syntax and conventions. The syntax of an encryption scheme specifies what kinds of algorithms make it up. Formally, 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*, is a probabilistic algorithm that 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*, is a probabilistic algorithm that 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 (pk, sk) which can be output by $\mathcal{K}(1^k)$, for all $x \in \{0,1\}^*$, and for 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.

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 $k \geq k_c$.

Notation. 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: **for** $1 \leq i \leq |\mathbf{y}|$ **do** $\mathbf{x}[i] \leftarrow \mathcal{D}_{sk}(\mathbf{y}[i])$.

We will consider relations of arity t where t will be polynomial in the security parameter k. Rather than writing $R(x_1, \ldots, x_t)$ we write $R(x, \mathbf{x})$, meaning the first argument is special and the rest are bunched into a vector $\mathbf{x}$ with $|\mathbf{x}| = t - 1$.

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. The question is how exactly one measures the advantage of the adversary. This turns out to need care. One possible formalization is that of [5, 6], which is based on the idea of simulation; it asks that for every adversary there exists a certain type of "simulator" that does just as well as the adversary but *without* being given y. In another, somewhat simpler formalization introduced in [2], there is no simulator; security is defined via an experiment involving the adversary alone. We begin below by presenting these two notions.

3.1 Definition of SNM

In this subsection we describe the definition of non-malleable encryption of [5, 6], which we call SNM for "simulation based non-malleability." The SNM formulation fixes a polynomial time computable relation R, which is viewed as taking four arguments, $R(x, \mathbf{x}, M, s_1)$, with $\mathbf{x}$ being a vector with an arbitrary number of components.

The adversary $A = (A_1, A_2)$ runs in two stages. The first stage of the adversary, namely A_1, computes (the encoding of) a distribution M on messages 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 also $y \notin \mathbf{y}$ and $\perp \notin \mathbf{x}$, where $\mathbf{x} = \mathcal{D}_{sk}(\mathbf{y})$.

The requirement for security is that for any polynomial time adversary A and any polynomial time relation R there exists a polynomial time algorithm $S = (S_1, S_2)$, 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 $R(x, \mathbf{x}, M, s_1)$ and $\perp \notin \mathbf{x}$.

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.

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, and let $S = (S_1, S_2)$ be an algorithm (the "simulator"). For atk $\in$ {cpa, cca1, cca2} and $k \in \mathsf{N}$ define

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

where

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

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 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 A that runs in time $p(k)$ and outputs a valid message space M sampleable in time $p(k)$, there exists a polynomial-time algorithm $S = (S_1, S_2)$ such that $\mathsf{Adv}^{\mathtt{snm-atk}}_{A,S,\Pi}(R, k)$ is negligible. It is understood above that in the case of CCA2, A_2 is not allowed to ask its oracle for the decryption of the challenge ciphertext y. ▌

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 is important; it stems from the fact that encryption is not intended to conceal the length of the plaintext.

3.2 Definition of CNM

We now recall the definition of non-malleable encryption of [2], which we call CNM for "comparison based non-malleability." Let $A = (A_1, A_2)$ be an adversary. The first stage of the adversary, namely A_1, is given the public key pk, and outputs a description of a valid message space, described by a sampling algorithm M. The second stage of the adversary, namely A_2, receives an encryption y of a random message x drawn from M. The adversary then outputs a (description of a) relation R and a vector $\mathbf{y}$. We insist that no component of $\mathbf{y}$ be equal to y. The adversary hopes that $R(x, \mathbf{x})$ holds, where $\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$. An adversary (A_1, A_2) is *successful* if it can do this with a probability significantly more than that with which $R(x, \mathbf{x})$ holds if she 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.

Definition 3.2 [CNM-CPA, CNM-CCA1, CNM-CCA2] Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $A = (A_1, A_2)$ be an adversary. For atk $\in \{\text{cpa}, \text{cca1}, \text{cca2}\}$ and $k \in \mathsf{N}$ define

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

where

$\mathsf{Expt}^{\mathtt{cnm-atk}}_{A,\Pi}(k)$	$\widetilde{\mathsf{Expt}}^{\mathtt{cnm-atk}}_{A,\Pi}(k)$
$(pk, sk) \leftarrow \mathcal{K}(1^k)$	$(pk, sk) \leftarrow \mathcal{K}(1^k)$
$(M, s) \leftarrow A_1^{\mathcal{O}_1}(pk)$	$(M, s) \leftarrow A_1^{\mathcal{O}_1}(pk)$
$x \leftarrow M$	$x, \tilde{x} \leftarrow M$
$y \leftarrow \mathcal{E}_{pk}(x)$	$\tilde{y} \leftarrow \mathcal{E}_{pk}(\tilde{x})$
$(R, \mathbf{y}) \leftarrow A_2^{\mathcal{O}_2}(s, y)$	$(R, \tilde{\mathbf{y}}) \leftarrow A_2^{\mathcal{O}_2}(s, \tilde{y})$
$\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$	$\tilde{\mathbf{x}} \leftarrow \mathcal{D}_{sk}(\tilde{\mathbf{y}})$
`return` 1 iff $(y \notin \mathbf{y}) \wedge R(x, \mathbf{x})$	`return` 1 iff $(\tilde{y} \notin \tilde{\mathbf{y}}) \wedge R(x, \tilde{\mathbf{x}})$

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 say that Π is secure in the sense of CNM-ATK if for every polynomial $p(k)$: if A runs in time $p(k)$, outputs a (valid) message space M sampleable in time $p(k)$, and outputs a relation R computable in time $p(k)$, then $\mathsf{Adv}^{\mathtt{cnm-atk}}_{A,\Pi}(\cdot)$ is negligible. It is understood above that in the case of CCA2, A_2 is not allowed to ask its oracle for the decryption of the challenge ciphertext y. ▌

We declare the adversary unsuccessful when some ciphertext $\mathbf{y}[i]$ does not have a valid decryption (that is, $\perp \in \mathbf{x}$), because in this case, the receiver is simply going to reject the adversary's message anyway.

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 New notion: IND under parallel attack

We present a new notion of security for a public key encryption scheme: indistinguishability under a parallel chosen-ciphertext attack.

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 show that this notion is equivalent to both forms of non-malleability given above.

In this 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. We do this by considering the second stage A_2 of the adversary as made up of two sub-stages, $A_{2,q}$ and $A_{2,g}$, the "q" standing for "query" and the "g" for "guess". The first stage outputs the parallel query; the second is given the response, and completes the guessing work of the second stage. More precisely, view the adversary $A = (A_1, A_2)$ where $A_2 = (A_{2,q}, A_{2,g})$ as a pair. Algorithm A_1 is run on input the public key, pk. At the end of A_1's execution she outputs a triple (x_0, x_1, s_1), the first two components being messages which we insist be *of the same length*, and the last being state information (possibly including pk) which she wants to preserve. 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. It is the job of the pair $(A_{2,q}, A_{2,g})$ to try to determine if y was selected as the encryption of x_0 or x_1, *i.e.* try to guess b. To make this determination, first run $A_{2,q}$ on x_0, x_1, s_1, y and let it output a parallel query $\mathbf{c}$ and state information s_2. (The latter will include y, s_1 if necessary.) Then run $A_{2,g}$ on input $\mathcal{D}_{sk}(\mathbf{c}), s_2$ and get a guess g. The adversary wins if $g = b$.

We can add this parallel attack to any of the previous attacks CPA, CCA1, CCA2, yielding respectively the attacks PA0, PA1, PA2. 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 = PA2. For concision and clarity we simultaneously define indistinguishability with respect to PA0, PA1, and PA2. The only difference lies in whether or not A_1 and A_2 are given decryption oracles. We let the string atk be instantiated by any of the formal symbols pa0, pa1, pa2, while ATK is then the corresponding formal symbol from PA0, PA1, PA2.

Definition 4.1 [IND-PA0, IND-PA1, IND-PA2] Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $A = (A_1, A_2)$ be an adversary. For atk $\in \{\text{pa0}, \text{pa1}, \text{pa2}\}$ and $k \in \mathsf{N}$, let

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

where

$$
\begin{array}{l}
\mathsf{Expt}^{\text{ind-atk}}_{A,\Pi}(k) \\
\quad (pk, sk) \leftarrow \mathcal{K}(1^k) \\
\quad (x_0, x_1, s_1) \leftarrow A_1^{\mathcal{O}_1}(pk) \\
\quad b \leftarrow \{0,1\} \\
\quad y \leftarrow \mathcal{E}_{pk}(x_b) \\
\quad (\mathbf{c}, s_2) \leftarrow A_{2,q}^{\mathcal{O}_2}(x_0, x_1, s_1, y) \\
\quad \mathbf{p} \leftarrow \mathcal{D}_{sk}(\mathbf{c}) \\
\quad g \leftarrow A_{2,g}^{\mathcal{O}_2}(\mathbf{p}, s_2) \\
\quad \texttt{return } 1 \text{ iff } (y \notin \mathbf{c}) \wedge (g = b)
\end{array}
$$

and

$$
\begin{array}{llll}
\text{If atk} = \text{pa0} & \text{then} \quad \mathcal{O}_1(\cdot) = \varepsilon & \text{and} & \mathcal{O}_2(\cdot) = \varepsilon \\
\text{If atk} = \text{pa1} & \text{then} \quad \mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot) & \text{and} & \mathcal{O}_2(\cdot) = \varepsilon \\
\text{If atk} = \text{pa2} & \text{then} \quad \mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot) & \text{and} & \mathcal{O}_2(\cdot) = \mathcal{D}_{sk}(\cdot)
\end{array}
$$

We say that Π is secure in the sense of IND-ATK if A being polynomial-time implies that $\mathsf{Adv}^{\text{ind-atk}}_{A,\Pi}(\cdot)$ is negligible. We insist, above, that A_1 outputs x_0, x_1 with $|x_0| = |x_1|$. In the case of PA2, we further insist that A_2 does not ask its oracle to decrypt y. ▮

5 Results

Here we establish the equivalence of all three notions discussed. This is established by the following sequence of propositions.

Proposition 5.1 [CNM-ATK $\Rightarrow$ SNM-ATK] *For any* ATK $\in \{\text{CPA}, \text{CCA1}, \text{CCA2}\}$, *if encryption scheme* Π *is secure in the sense of* CNM-ATK *then* Π *is secure in the sense of* SNM-ATK.

Proposition 5.2 [SNM-ATK $\Rightarrow$ IND-PXX] *For any* ATK $\in \{\text{CPA}, \text{CCA1}, \text{CCA2}\}$, *if encryption scheme* Π *is secure in the sense of* SNM-ATK *then* Π *is secure in the sense of* IND-PXX, *where*

If ATK = CPA *then* PXX = PA0
If ATK = CCA1 *then* PXX = PA1
If ATK = CCA2 *then* PXX = PA2

Proposition 5.3 [IND-PXX $\Rightarrow$ CNM-ATK] *For any* PXX $\in \{\text{PA0}, \text{PA1}, \text{PA2}\}$, *if encryption scheme* Π *is secure in the sense of* IND-PXX *then* Π *is secure in the sense of* CNM-ATK, *where*

If PXX = PA0 *then* ATK = CPA
If PXX = PA1 *then* ATK = CCA1
If PXX = PA2 *then* ATK = CCA2

We now turn to the proofs.

5.1 CNM **implies** SNM

How does SNM compare with CNM? Let us first consider the question under CPA. It is easy to see that CNM-CPA $\Rightarrow$ 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, A'_1, is identical to A_1. 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-CPA is actually weaker, corresponding to the ability to simulate by simulators which are also given the decryption oracle.

However, this appearance is false. In fact, CNM-CPA 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-CPA definition, it is possible to design a simulator according to the SNM definition.)

Proof of Proposition 5.1: Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the given encryption scheme. Let R and $A = (A_1, A_2)$ 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 of which it knows the corresponding decryption key:

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

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 $\mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi}(R, k)$ 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 $B = (B_1, B_2)$ attacking Π in the sense of CNM-ATK.

Algorithm $B_1^{\mathcal{O}_1}(pk)$	Algorithm $B_2^{\mathcal{O}_2}((M, s_1, s_2), y)$
$(M, s_1, s_2) \leftarrow A_1^{\mathcal{O}_1}(pk)$	Define R' by $R'(a, \mathbf{b}) = 1$ iff $R(a, \mathbf{b}, M, s_1) = 1$
return $(M, (M, s_1, s_2))$	$\mathbf{y} \leftarrow A_2^{\mathcal{O}_2}(s_2, y)$
	return $(R', \mathbf{y})$

It is clear from the definition of B that $\mathsf{Expt}^{\text{cnm-atk}}_{B,\Pi}(k)$ is precisely the same as $\mathsf{Expt}^{\text{snm-atk}}_{A,\Pi}(R, k)$. Now, let us expand the definition of $\mathsf{Expt}^{\text{snm-atk}}_{S,\Pi}(R, k)$, substituting in the definition of S given above. Once we eliminate lines that do not affect the outcome of the experiment, this yields:

$\mathsf{Expt}^{\text{snm-atk}}_{S,\Pi}(R,k)$
$\quad (pk', sk') \leftarrow \mathcal{K}(1^k)$
$\quad (M, s_1, s_2) \leftarrow A_1^{\tilde{\mathcal{O}}_1}(pk')$
$\quad x, \tilde{x} \leftarrow M$
$\quad \tilde{y} \leftarrow \mathcal{E}_{pk'}(\tilde{x})$
$\quad \tilde{\mathbf{y}} \leftarrow A_2^{\tilde{\mathcal{O}}_2}(s_2, \tilde{y})$
$\quad \tilde{\mathbf{x}} \leftarrow \mathcal{D}_{sk'}(\tilde{\mathbf{y}})$
$\qquad$ `return` 1 iff $(\tilde{y} \notin \tilde{\mathbf{y}}) \wedge R(x, \tilde{\mathbf{x}}, M, s_1)$

Thus, glancing at the definition of B, we see that this experiment is precisely the same as $\widetilde{\mathsf{Expt}}^{\text{cnm-atk}}_{B,\Pi}(k)$ with pk and sk replaced by the pk' and sk' chosen by the simulator. Hence, $\mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi}(R,k) = \mathsf{Adv}^{\text{cnm-atk}}_{B,\Pi}(k)$. But the latter is negligible since Π is secure in the sense of CNM-ATK, so the former is negligible too. ▮

5.2 SNM-ATK ⇒ IND-PXX

Proof of Proposition 5.2: We already know that SNM-CCA2 and IND-CCA2 are equivalent [6]. But IND-PA2 and IND-CCA2 are obviously identical since in both cases, a chosen-ciphertext attack is allowed in the second stage, and this subsumes a parallel attack. Thus we need prove the proposition only for the cases of ATK = CPA and ATK = 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-PXX sense. Let $B = (B_1, B_2)$ be an IND-PXX adversary attacking Π. We want to show that $\mathsf{Adv}^{\text{ind-pxx}}_{B,\Pi}(\cdot)$ 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 B has as an IND-PXX adversary using a parallel attack. What allows us to do this is to pick the relation R to capture the success condition of B's parallel attack. Adversaries A and B have access to an oracle $\mathcal{O}_1$ in their first stage (but we can assume that oracle in their second stage $\mathcal{O}_2 = \varepsilon$), with this oracle being instantiated according to the attack ATK as per the definitions.

To get some intuition it is best to think of ATK = CPA, meaning A is allowed only a chosen-plaintext attack. However, B has (limited) access to a decryption oracle; it is allowed the parallel query. How then can A "simulate" B? 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 B will make. Now, B 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" B'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 B.

Notice that for this to work it is crucial that B's query is a parallel one. If B 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 B, 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. Given the IND-PXX adversary $B = (B_1, B_{2,q}, B_{2,g})$, we want to define the SNM-ATK adversary $A = (A_1, A_2)$. The first stage A_1 is:

Algorithm $A_1^{\mathcal{O}_1}(pk)$
  $(m_0, m_1, t_1) \leftarrow B_1^{\mathcal{O}_1}(pk)$
  Let M be a canonical encoding of the uniform distribution over $\{m_0, m_1\}$
    along with an encoding of the ordered pair (m_0, m_1)
  Let $s_1 = \varepsilon$ and $s_2 = (m_0, m_1, t_1, pk)$
  return (M, s_1, s_2)

The second stage A_2 is:

Algorithm $A_2(s_2)$ where $s_2 = (m_0, m_1, t_1, pk)$
  $(\mathbf{c}, t_2) \leftarrow B_{2,q}(m_0, m_1, t_1, y)$
  Choose random coins σ for $B_{2,g}$
  $e_1 \leftarrow \mathcal{E}_{pk}(t_2)$; $e_2 \leftarrow \mathcal{E}_{pk}(\sigma)$
  Let $\mathbf{y} = (e_1, e_2, \mathbf{c}[1], \ldots, \mathbf{c}[|\mathbf{c}|])$
  return y

Notice above that A_2 picks coins σ for $B_{2,g}$. We can think of each stage of B as picking its own coins afresh, since any information needing to be communicated from stage to stage is passed along in the state information. Now, here is the relation R.

Relation $R(x, \mathbf{x}, M, s_1)$
  Let t_2 and σ be the first two components of $\mathbf{x}$
  Let the remaining components form the vector $\mathbf{p}$
  If (M is not a valid canonical encoding of an ordered pair of strings (m_0, m_1)
    and the uniform distribution over $\{m_0, m_1\}$)
    then return 0
  Let b be such that $x = m_b$
  return 1 iff $B_{2,g}(\mathbf{p}, t_2; \sigma) = b$

Notice that R is polynomial time computable.

If one expands the definition of $\mathsf{Expt}^{\text{snm-atk}}_{A,\Pi}(R, k)$ using the definitions of R and A above, by eliminating unnecessary lines we see that the experiment is the same as $\mathsf{Expt}^{\text{ind-atk}}_{B,\Pi}(k)$ up to negligible factors.

To conclude the proof, we need only show that the probability that any simulator S will succeed in attacking R as defined above in the experiment $\mathsf{Expt}^{\text{snm-atk}}_{S,\Pi}(R, k)$ is at most $\frac{1}{2}$. By construction, in order to satisfy R the first stage of S must output a distribution M that is uniform on two messages m_0 and m_1. Suppose S does so with probability $q \leq 1$. Now let p_b be the probability that S create output so as to cause $B_{2,g}$ to output b. Since the behavior of S is independent of the chosen $x \in M$, $p_0 + p_1 \leq 1$. Hence, also by independence, the probability that S will succeed in causing R to accept is bounded by $q(p_0 \cdot \Pr[x = m_0] + p_1 \cdot \Pr[x = m_1]) \leq q(p_0 \cdot \frac{1}{2} + p_1 \cdot \frac{1}{2}) \leq \frac{1}{2}$.

Thus, we have that

$$\begin{aligned}
\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) &= \Pr[\mathsf{Expt}^{\text{ind-atk}}_{B,\Pi}(k) = 1] - \frac{1}{2} \\
&\leq \Pr[\mathsf{Expt}^{\text{snm-atk}}_{A,\Pi}(R,k) = 1] - \frac{1}{2} + \epsilon(k) \\
&\leq \Pr[\mathsf{Expt}^{\text{snm-atk}}_{A,\Pi}(R,k) = 1] - \Pr[\mathsf{Expt}^{\text{snm-atk}}_{S,\Pi}(R,k) = 1] + \epsilon(k) \\
&\leq \mathsf{Adv}^{\text{snm-atk}}_{A,\Pi}(R,k) + \epsilon(k)
\end{aligned}$$

where $\epsilon(k)$ is a negligible function. Since by assumption $\mathsf{Adv}^{\text{snm-atk}}_{A,\Pi}(R,k)$ is negligible, this completes the proof. ▌

5.3 IND-PXX $\Rightarrow$ CNM-ATK

Proof of Proposition 5.3: We are assuming that Π is secure in the IND-PXX sense. We will show it is also secure in the CNM-ATK sense.

Let $B = (B_1, B_2)$ be an CNM-ATK adversary attacking Π. We want to show that $\mathsf{Adv}^{\text{cnm-atk}}_{B,\Pi}(\cdot)$ is negligible. To this end, we describe an IND-PXX adversary $A = (A_1, A_2)$ attacking Π. We wish to show that A will have the same advantage as an IND-PXX adversary as B has as an CNM-ATK adversary. The definition of a parallel attack was chosen to make this proof easy, and the intuition will be simple: since A has access to a parallel decryption oracle in the second stage, she can decrypt the ciphertexts that B outputs, and check herself to see if B's relation holds.

Given the CNM-ATK adversary $B = (B_1, B_2)$, we define the IND-PXX adversary $A = (A_1, A_{2,q}, A_{2,g})$ as follows:

Algorithm $A_1^{\mathcal{O}_1}(pk)$
  $(M, t) \leftarrow B_1^{\mathcal{O}_1}(pk)$
  $x_0, x_1 \leftarrow M$; $s_1 \leftarrow (M, t)$
  return (x_0, x_1, s_1)

Algorithm $A_{2,q}^{\mathcal{O}_2}(x_0, x_1, s_1, y)$ where $s_1 = (M, t)$
  $(R, \mathbf{c}) \leftarrow B_2^{\mathcal{O}_2}(M, t, y)$
  $s_2 \leftarrow (R, x_0, x_1, \mathbf{c}, y)$
  return $(\mathbf{c}, s_2)$

Algorithm $A_{2,g}^{\mathcal{O}_2}(\mathbf{p}, s_2)$ where $s_2 = (R, x_0, x_1, \mathbf{c}, y)$
  if $(y \notin \mathbf{c}) \wedge R(x_0, \mathbf{p})$
    then $g \leftarrow 0$
    else $g \leftarrow \{0, 1\}$
  return g

A straightforward calculation establishes that the advantage of the IND-PXX adversary given above will be negligibly close to the advantage of the CNM-ATK adversary, completing the proof. ▌

Acknowledgments

Mihir Bellare was supported in part by NSF CAREER Award CCR-9624439 and a 1996 Packard Foundation Fellowship in Science and Engineering. Amit Sahai was supported by a DOD/NDSEG Graduate Fellowship and partially by DARPA grant DABT-96-C-0018.

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 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 *Proceedings*, 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 *Proceedings*, Lecture Notes in Computer Science Vol. 950, A. De Santis ed., Springer-Verlag, 1994.

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

[5] D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography. *Proceedings of the* 23rd *Annual Symposium on Theory of Computing*, ACM, 1991. Also *Technical Report CS95-27, Weizmann Institute of Science*, 1995.

[6] D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography. Manuscript, 1998.

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

[8] S. Goldwasser and S. Micali, Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

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

[10] S. Micali, C. Rackoff and R. Sloan, The notion of security for probabilistic cryptosystems. *SIAM J. of Computing*, April 1988.

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

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

[13] SETCo (Secure Electronic Transaction LLC), The SET standard — book 3 — formal protocol definitions (version 1.0). May 31, 1997. Available from `http://www.setco.org/`

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