# Relations Among Notions of Security for Public-Key Encryption Schemes

Mihir Bellare[1], Anand Desai[1], David Pointcheval[2], and Phillip Rogaway[3]

[1] Dept. of Computer Science & Engineering, University of California at San Diego,
9500 Gilman Drive, La Jolla, CA 92093, USA.
{mihir,adesai}@cs.ucsd.edu
URL: http://www-cse.ucsd.edu/users/{mihir,adesai}

[2] Laboratoire d'Informatique de l'École Normale Supérieure, 75005 Paris, France,
and GREYC, Dépt d'Informatique, Université de Caen, 14032 Caen Cedex, France.
david.pointcheval@ens.fr
URL: http://www.dmi.ens.fr/~pointche/

[3] Dept. of Computer Science, Engineering II Bldg., University of California at Davis,
Davis, CA 95616, USA.
rogaway@cs.ucdavis.edu
URL: http://www.cs.ucdavis.edu/~rogaway/

**Abstract.** We compare the relative strengths of popular notions of security for public key encryption schemes. We consider the goals of privacy and non-malleability, each under chosen plaintext attack and two kinds of chosen ciphertext attack. For each of the resulting pairs of definitions we prove either an implication (every scheme meeting one notion must meet the other) or a separation (there is a scheme meeting one notion but not the other, assuming the first notion can be met at all). We similarly treat plaintext awareness, a notion of security in the random oracle model. An additional contribution of this paper is a new definition of non-malleability which we believe is simpler than the previous one.

**Keywords:** Asymmetric encryption, Chosen ciphertext security, Non-malleability, Rackoff-Simon attack, Plaintext awareness, Relations among definitions.

## 1 Introduction

In this paper we compare the relative strengths of various notions of security for public key encryption. We want to understand which definitions of security imply which others. We start by sorting out some of the notions we will consider.

### 1.1 Notions of Encryption Scheme Security

A convenient way to organize definitions of secure encryption is by considering separately the various possible *goals* and the various possible *attack models*, and then obtain each definition as a pairing of a particular goal and a particular attack model. This viewpoint was suggested to us by Moni Naor [22].

H. Krawczyk (Ed.): CRYPTO'98, LNCS 1462, pp. 26–46, 1998.

We consider two different goals: *indistinguishability of encryptions*, due to Goldwasser and Micali [17], and *non-malleability*, due to Dolev, Dwork and Naor [11]. Indistinguishability (IND) formalizes an adversary's inability to learn any information about the plaintext $x$ underlying a challenge ciphertext $y$, capturing a strong notion of privacy. Non-malleability (NM) formalizes an adversary's inability, given a challenge ciphertext $y$, to output a different ciphertext $y'$ such that the plaintexts $x, x'$ underlying these two ciphertexts are "meaningfully related". (For example, $x' = x + 1$.) It captures a sense in which ciphertexts can be tamper-proof.

Along the other axis we consider three different attacks. In order of increasing strength these are *chosen plaintext attack* (CPA), *non-adaptive chosen ciphertext attack* (CCA1), and *adaptive chosen ciphertext attack* (CCA2). Under CPA the adversary can obtain ciphertexts of plaintexts of her choice. In the public key setting, giving the adversary the public key suffices to capture these attacks. Under CCA1, formalized by Naor and Yung [23], the adversary gets, in addition to the public key, access to an oracle for the decryption function. The adversary may use this decryption function only for the period of time preceding her being given the challenge ciphertext $y$. (The term non-adaptive refers to the fact that queries to the decryption oracle cannot depend on the challenge $y$. Colloquially this attack has also been called a "lunchtime," "lunch-break," or "midnight" attack.) Under CCA2, due to Rackoff and Simon [24], the adversary again gets (in addition to the public key) access to an oracle for the decryption function, but this time she may use this decryption function even on ciphertexts chosen after obtaining the challenge ciphertext $y$, the only restriction being that the adversary may not ask for the decryption of $y$ itself. (The attack is called adaptive because queries to the decryption oracle can depend on the challenge $y$.) As a mnemonic for the abbreviations CCA1 / CCA2, just remember that the bigger number goes with the stronger attack.

One can "mix-and-match" the goals $\{\text{IND}, \text{NM}\}$ and attacks $\{\text{CPA}, \text{CCA1}, \text{CCA2}\}$ in any combination, giving rise to six notions of security:

$$\text{IND-CPA, IND-CCA1, IND-CCA2,} \quad \text{NM-CPA, NM-CCA1, NM-CCA2} \ .$$

Most are familiar (although under different names). IND-CPA is the notion of [17];[1] IND-CCA1 is the notion of [23]; IND-CCA2 is the notion of [24]; NM-CPA, NM-CCA1 and NM-CCA2 are from [11,12,13].

### 1.2 Implications and Separations

In this paper we work out the relations between the above six notions. For each pair of notions $\mathbf{A}, \mathbf{B} \in \{$IND-CPA, IND-CCA1, IND-CCA2, NM-CPA, NM-CCA1, NM-CCA2$\}$, we show one of the following:

- $\mathbf{A} \Rightarrow \mathbf{B}$: A proof that if $\Pi$ is any encryption scheme meeting notion of security $\mathbf{A}$ then $\Pi$ also meets notion of security $\mathbf{B}$.

[1] Goldwasser and Micali referred to IND-CPA as polynomial security, and also showed this was equivalent to another notion, semantic security.

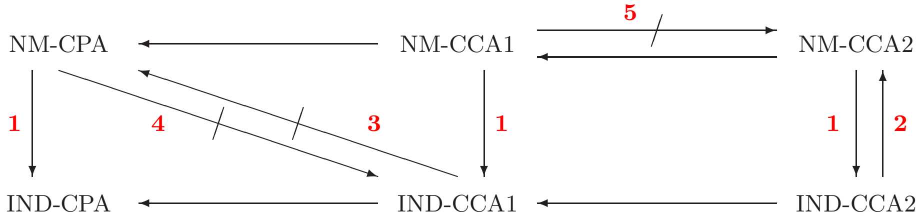

**Fig. 1.** *An arrow is an implication, and in the directed graph given by the arrows, there is a path from* **A** *to* **B** *if and only* $\mathbf{A} \Rightarrow \mathbf{B}$*. The hatched arrows represent separations we actually prove; all others follow automatically. The number on an arrow or hatched arrow refers to the theorem in this paper which establishes this relationship.*

- $\mathbf{A} \not\Rightarrow \mathbf{B}$: A construction of an encryption scheme $\Pi$ that provably meets notion of security **A** but provably does *not* meet notion of security **B**.[2]

We call a result of the first type an *implication*, and a result of the second type a *separation*. For each pair of notions we provide one or the other, so that no relation remains open.

These results are represented diagrammatically in Figure 1. The (unhatched) arrows represent implications that are proven or trivial, and the hatched arrows represent explicitly proven separations. Specifically, the non-trivial implication is that IND-CCA2 implies NM-CCA2, and the separations shown are that IND-CCA1 does not imply NM-CPA; nor does NM-CPA imply IND-CCA1; nor does NM-CCA1 imply NM-CCA2.

Figure 1 represents a complete picture of relations in the following sense. View the picture as a graph, the edges being those given by the (unhatched) arrows. (So there are eight edges.) We claim that for any pair of notions **A**, **B**, it is the case that **A** implies **B** if and only if there is a path from **A** to **B** in the graph. The "if" part of this claim is of course clear from the definition of implication. The "only if" part of this claim can be verified for any pair of notions by utilizing the hatched and unhatched arrows. For example, we claim that IND-CCA1 does not imply IND-CCA2. For if we had that IND-CCA1 implies IND-CCA2 then this, coupled with NM-CCA1 implying IND-CCA1 and IND-CCA2 implying NM-CCA2, would give NM-CCA1 implying NM-CCA2, which we know to be false.

That IND-CCA2 implies all of the other notions helps bolster the view that adaptive CCA is the "right" version of CCA on which to focus. (IND-CCA2 has already proven to be a better tool for protocol design.) We thus suggest that, in the future, "CCA" should be understood to mean adaptive CCA.

[2] This will be done under the assumption that there exists *some* scheme meeting notion **A**, since otherwise the question is vacuous. This (minimal) assumption is the only one made.

### 1.3 Plaintext Awareness

Another adversarial goal we will consider is *plaintext awareness* (PA), first defined by Bellare and Rogaway [4]. PA formalizes an adversary's inability to create a ciphertext $y$ without "knowing" its underlying plaintext $x$. (In the case that the adversary creates an "invalid" ciphertext what she should know is that the ciphertext is invalid.)

So far, plaintext awareness has only been defined in the random oracle (RO) model. Recall that in the RO model one embellishes the customary model of computation by providing all parties (good and bad alike) with a random function $H$ from strings to strings. See [3] for a description of the random oracle model and a discussion of its use.

The six notions of security we have described can be easily "lifted" to the RO model, giving six corresponding definitions. Once one makes such definitional analogs it is easily verified that all of the implications and separations mentioned in Section 1.2 and indicated in Figure 1 also hold in the RO setting. For example, the RO version of IND-CCA2 implies the RO version of NM-CCA2.

Since PA has only been defined in the RO model it only makes sense to compare PA with other RO notions. Our results in this vein are as follows. Theorem 6 shows that PA (together with the RO version of IND-CPA) implies the RO version of IND-CCA2. In the other direction, Theorem 7 shows that the RO version of IND-CCA2 does not imply PA.

### 1.4 Definitional Contributions

Beyond the implications and separations we have described, we have two definitional contributions: a new definition of non-malleability, and a refinement to the definition of plaintext awareness.

The original definition of non-malleability [11,12,13] is in terms of simulation, requiring, for every adversary, the existence of some appropriate simulator. We believe our formulation is simpler. It is defined via an experiment involving only the adversary; there is no simulator. Nonetheless, it does not lose strength: Theorem 8 (due to [5]) says that our definition implies that of [12,13] under any form of attack. The definitions are not known to be equivalent because the other direction is open. See Appendix A.

We stress that the results in this paper are not affected by the definitional change; they hold under either definition. We view the new definition as an additional, orthogonal contribution which could simplify the task of working with non-malleability. We also note that our definitional idea lifts to other settings, like defining semantic security [17] against chosen ciphertext attacks. (Semantic security seems not to have been defined against CCA.)

With regard to plaintext awareness, we make a small but important refinement to the definition of [4]. The change allows us to substantiate their claim that plaintext awareness implies chosen ciphertext security and non-malleability, by giving us that PA (plus IND-CPA) implies the RO versions of IND-CCA2 and NM-CCA2. Our refinement is to endow the adversary with an encryption oracle, the queries to which are not given to the extractor. See Section 4.

### 1.5 Motivation

In recent years there has been an increasing role played by public key encryption schemes which meet notions of security beyond IND-CPA. We are realizing that one of their most important uses is as tools for designing higher level protocols. For example, encryption schemes meeting IND-CCA2 appear to be the right tools in the design of authenticated key exchange protocols in the public-key setting [1]. As another example, the designers of SET (Secure Electronic Transactions) selected an encryption scheme which achieves more than IND-CPA [25]. This was necessary, insofar as the SET protocols would be *wrong* if instantiated by a primitive which achieves *only* IND-CPA security. Because encryption schemes which achieve more than IND-CPA make for easier-to-use (or harder-to-misuse) tools, emerging standards rightly favor them.

We comment that if one takes the CCA models "too literally" the attacks we describe seem rather artificial. Take adaptive CCA, for example. How could an adversary have access to a decryption oracle, yet be forbidden to use it on the one point she really cares about? Either she has the oracle and can use it as she likes, or she does not have it at all. Yet, in fact, just such a setting effectively arises when encryption is used in session key exchange protocols. In general, one should not view the definitional scenarios we consider too literally, but rather understand that these are the right notions for schemes to meet when these schemes are to become generally-useful tools in the design of high level protocols.

### 1.6 Related Work and Discussion

The most recent version of the work of Dolev, Dwork and Naor (the manuscript [13]) has, independently of our work, considered the question of relations between notions of encryption, and contains (currently in Remark 3.6) various claims that overlap to some extent with ours. (Public versions of their work, namely the 1991 proceedings version [11] and the 1995 technical report [12], do not contain these claims.)

It is not the purpose of this paper to discuss specific schemes designed for meeting any of the notions of security described in this paper. Nonetheless, as a snapshot of the state of the art, we attempt to summarize what is known about meeting "beyond IND-CPA" notions of security. Schemes proven secure under standard assumptions include that of [23], which meets IND-CCA1, that of [11], which meets IND-CCA2, and the much more efficient recent scheme of Cramer and Shoup [8], which also meets IND-CCA2. Next are the schemes proven secure in a random oracle model; here we have those of [3,4], which meet PA and are as efficient as schemes in current standards. Then there are schemes without proofs, such as those of [9,26]. Finally, there are schemes for non-standard models, like [15,24].

It follows from our results that the above mentioned scheme of [8], shown to meet IND-CCA2, also meets NM-CCA2, and in particular is non-malleable under all three forms of attack.

Bleichenbacher [6] has recently shown that a popular encryption scheme, RSA PKCS #1, does not achieve IND-CCA1.

We comment that non-malleability is a general notion that applies to primitives other than encryption [11]. Our discussion is limited to its use in asymmetric encryption. Similarly, chosen ciphertext attack applies to both the symmetric and asymmetric settings, but this work is specific to the latter.

Due to space limitations, we have omitted various parts of this paper. A full version of the paper is available [2].

## 2 Definitions of Security

This section provides formal definitions for the six notions of security of an asymmetric (ie., public key) encryption scheme discussed in Section 1.1. Plaintext awareness will be described in Section 4. We begin by describing the *syntax* of an encryption scheme, divorcing syntax from the notions of security.

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 $\alpha$ 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 $A(x_1, x_2, \ldots ; r) = y$.

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 $k \geq k_c$.

### 2.1 Framework

The formalizations that follow have a common framework that it may help to see at a high level first. In formalizing both indistinguishability and non-malleability we regard an adversary $A$ as a pair of probabilistic algorithms, $A = (A_1, A_2)$. (We will say that $A$ is polynomial time if both $A_1$ and $A_2$ are.) This corresponds to $A$ running in two "stages." The exact purpose of each stage depends on the particular adversarial goal, but for both goals the basic idea is that in the first stage the adversary, given the public key, seeks and outputs some "test instance," and in the second stage the adversary is issued a challenge ciphertext $y$ generated as a probabilistic function of the test instance, in a manner depending on the goal. (In addition $A_1$ can output some state information $s$ that will be passed to $A_2$.) Adversary $A$ is successful if she passes the challenge, with what "passes" means again depending on the goal.

We consider three types of attacks under this setup.

In a *chosen-plaintext attack* (CPA) the adversary can encrypt plaintexts of her choosing. Of course a CPA is unavoidable in the public-key setting: knowing the public key, an adversary can, on her own, compute a ciphertext for any plaintext she desires. So in formalizing definitions of security under CPA we "do nothing" beyond giving the adversary access to the public key; that's already enough to make a CPA implicit.

In a *non-adaptive chosen ciphertext attack* (CCA1) we give $A_1$ (the public key and) access to a decryption oracle, but we do not allow $A_2$ access to a decryption oracle. This is sometimes called a non-adaptive chosen ciphertext attack, in that the decryption oracle is used to generate the test instance, but taken away before the challenge appears.

In an *adaptive chosen ciphertext attack* (CCA2) we continue to give $A_1$ (the public key and) access to a decryption oracle, but also give $A_2$ access to the same decryption oracle, with the only restriction that she cannot query the oracle on the challenge ciphertext $y$. This is an extremely strong attack model.

As a mnemonic, the number $i$ in CCA$i$ can be regarded as the number of adversarial stages during which she has access to a decryption oracle. Additionally, the bigger number corresponds to the stronger (and chronologically later) formalization.

By the way: we do not bother to explicitly give $A_2$ the public key, because $A_1$ has the option of including it in $s$.

### 2.2 Indistinguishability of Encryptions

The classical goal of secure encryption is to preserve the privacy of messages: an adversary should not be able to learn from a ciphertext information about its plaintext beyond the length of that plaintext. We define a version of this notion, indistinguishability of encryptions (IND), following [17,21], through a simple experiment. Algorithm $A_1$ is run on input the public key, $pk$. At the end of $A_1$'s execution she outputs a triple $(x_0, x_1, s)$, 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 $A_2$'s job to try to determine if $y$ was selected as the encryption of $x_0$ or $x_1$, namely to determine the bit $b$. To make this determination $A_2$ is given the saved state $s$ and the challenge ciphertext $y$.

For concision and clarity we simultaneously define indistinguishability with respect to CPA, CCA1, and CCA2. 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 $\mathrm{cpa}, \mathrm{cca1}, \mathrm{cca2}$, while ATK is then the corresponding formal symbol from $\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}$. 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, $\varepsilon$.

**Definition 1.** [IND-CPA, IND-CCA1, IND-CCA2] Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $A = (A_1, A_2)$ be an adversary. For $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca1}, \mathrm{cca2}\}$ and $k \in \mathsf{N}$ let $\mathsf{Adv}^{\text{ind-atk}}_{A,\Pi}(k) \stackrel{\text{def}}{=}$

$$2 \cdot \Pr\Big[(pk, sk) \leftarrow \mathcal{K}(1^k) \ ; \ (x_0, x_1, s) \leftarrow A_1^{\mathcal{O}_1}(pk) \ ; \ b \leftarrow \{0,1\} \ ; \ y \leftarrow \mathcal{E}_{pk}(x_b) : \\ A_2^{\mathcal{O}_2}(x_0, x_1, s, y) = b\Big] - 1$$

where

$$\begin{array}{l} \text{If atk} = \text{cpa then } \mathcal{O}_1(\cdot) = \varepsilon \quad\quad\ \text{ and } \mathcal{O}_2(\cdot) = \varepsilon \\ \text{If atk} = \text{cca1 then } \mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot) \text{ and } \mathcal{O}_2(\cdot) = \varepsilon \\ \text{If atk} = \text{cca2 then } \mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot) \text{ and } \mathcal{O}_2(\cdot) = \mathcal{D}_{sk}(\cdot) \end{array}$$

We insist, above, that $A_1$ outputs $x_0, x_1$ with $|x_0| = |x_1|$. In the case of CCA2, we further insist that $A_2$ does not ask its oracle to decrypt $y$. We say that $\Pi$ 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. □

### 2.3 Non-Malleability

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 $\{\mathbf{x}[i] : 1 \le i \le |\mathbf{x}|\}$. It will be convenient to extend the decryption notation to vectors with the understanding that operations are performed componentwise. Thus $\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$ is shorthand for the following: **for** $1 \le i \le |\mathbf{y}|$ **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 $|\mathbf{x}| = t-1$.

Idea. The notion of non-malleability was introduced in [11], with refinements in [12,13]. The goal of the adversary, given a ciphertext $y$, is not (as with indistinguishability) to learn something about its plaintext $x$, but only 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 [11,12,13], 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 $y$. Here, we introduce a novel formalization which seems to us to be simpler. Our formalization does not ask for a simulator, but just considers an experiment involving the adversary. It turns out that our notion implies DDN's, but the converse is not known. See Appendix A for a brief comparison.

Our formalization. Let $A = (A_1, A_2)$ be an adversary. In the first stage of the adversary's attack, $A_1$, given the public key $pk$, outputs a description of a message space, described by a sampling algorithm $M$. The message space must be *valid*, which means that it gives non-zero probability only to strings of some one particular length. In the second stage of the adversary's attack, $A_2$ receives an encryption $y$ of a random message, say $x$, drawn from $M$. The adversary then outputs a (description of a) relation $R$ and a vector $\mathbf{y}$ (no component of which is $y$). She 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 she can do this with a probability significantly more than that with which $R(\tilde{x}, \mathbf{x})$ holds for some random hidden $\tilde{x} \leftarrow M$.

**Definition 2.** [NM-CPA, NM-CCA1, NM-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 $\in$ {cpa, cca1, cca2} and $k \in \mathsf{N}$ define

$$\mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(k) \overset{\text{def}}{=} \left| \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) - \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) \right|$$

where $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) \overset{\text{def}}{=}$

$$\Pr\Big[(pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (M, s) \leftarrow A_1^{\mathcal{O}_1}(pk)\ ;\ x \leftarrow M\ ;\ y \leftarrow \mathcal{E}_{pk}(x)\ ;$$
$$(R, \mathbf{y}) \leftarrow A_2^{\mathcal{O}_2}(M, s, y)\ ;\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) :\ y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(x, \mathbf{x})\Big]$$

and $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) \overset{\text{def}}{=}$

$$\Pr\Big[(pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (M, s) \leftarrow A_1^{\mathcal{O}_1}(pk)\ ;\ x, \tilde{x} \leftarrow M\ ;\ y \leftarrow \mathcal{E}_{pk}(x)\ ;$$
$$(R, \mathbf{y}) \leftarrow A_2^{\mathcal{O}_2}(M, s, y)\ ;\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) :\ y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(\tilde{x}, \mathbf{x})\Big]$$

where

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 insist, above, that $M$ is valid: $|x| = |x'|$ for any $x, x'$ that are given non-zero probability in the message space $M$. We say that $\Pi$ is secure in the sense of NM-ATK if for every polynomial $p(k)$: if $A$ runs in time $p(k)$, outputs a (valid) message space $M$ samplable in time $p(k)$, and outputs a relation $R$ computable in time $p(k)$, then $\mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(\cdot)$ 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. Otherwise, she could output the equality relation $R$, where $R(a, b)$ holds iff $a = b$, and output

$\mathbf{y} = (y)$, and be successful with probability one. We also 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 requirement that $M$ is valid is important; it stems from the fact that encryption is not intended to conceal the length of the plaintext.

One might want to strengthen the notion to require that the adversary's advantage remains small even in the presence a priori information about the message $x$; such incorporation of message "history" was made in Goldreich's formalizations of semantic security [14] and the definition of non-malleability in [12,13]. For simplicity we have omitted histories, but note that the above definition can be easily enhanced to take histories into account, and we explain how in [2].

## 3 Relating IND and NM

We state more precisely the results summarized in Figure 1 and provide proofs. As mentioned before, we summarize only the main relations (the ones that require proof); all other relations follow as corollaries.

### 3.1 Results

The first result, that non-malleability implies indistinguishability under any type of attack, was of course established by [11] in the context of their definition of non-malleability, but since we have a new definition of non-malleability, we need to re-establish it. The (simple) proof of the following is in [2].

**Theorem 1.** [NM-ATK ⇒ IND-ATK] *If encryption scheme $\Pi$ is secure in the sense of* NM-ATK *then $\Pi$ is secure in the sense of* IND-ATK *for any attack* $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$.

*Remark 1.* Recall that the relation $R$ in Definition 2 was allowed to have any polynomially bounded arity. However, the above theorem holds even under a weaker notion of NM-ATK in which the relation $R$ is restricted to have arity two.

The proof of the following is in Section 3.2.

**Theorem 2.** [IND-CCA2 ⇒ NM-CCA2] *If encryption scheme $\Pi$ is secure in the sense of* IND-CCA2 *then $\Pi$ is secure in the sense of* NM-CCA2.

*Remark 2.* Theorem 2 coupled with Theorem 1 and Remark 1 says that in the case of CCA2 attacks, it suffices to consider binary relations, meaning the notion of NM-CCA2 restricted to binary relations is equivalent to the general one.

Now we turn to separations. Adaptive chosen ciphertext security implies non-malleability according to Theorem 2. In contrast, the following says that non-adaptive chosen ciphertext security does *not* imply non-malleability. The proof is in Section 3.3.

**Theorem 3.** [IND-CCA1$\not\Rightarrow$NM-CPA] *If there exists an encryption scheme $\Pi$ which is secure in the sense of* IND-CCA1*, then there exists an encryption scheme $\Pi'$ which is secure in the sense of* IND-CCA1 *but which is not secure in the sense of* NM-CPA*.*

Now one can ask whether non-malleability implies chosen ciphertext security. The following says it does not even imply the non-adaptive form of the latter. (As a corollary, it certainly does not imply the adaptive form.) The proof is in Section 3.4.

**Theorem 4.** [NM-CPA$\not\Rightarrow$IND-CCA1] *If there exists an encryption scheme $\Pi$ which is secure in the sense of* NM-CPA*, then there exists an encryption scheme $\Pi'$ which is secure in the sense of* NM-CPA *but which is not secure in the sense of* IND-CCA1*.*

Now the only relation that does not immediately follow from the above results or by a trivial reduction is that the version of non-malleability allowing CCA1 does not imply the version that allows CCA2. See Section 3.5 for the proof of the following.

**Theorem 5.** [NM-CCA1$\not\Rightarrow$NM-CCA2] *If there exists an encryption scheme $\Pi$ which is secure in the sense of* NM-CCA1*, then there exists an encryption scheme $\Pi'$ which is secure in the sense of* NM-CCA1 *but which is not secure in the sense of* NM-CCA2*.*

### 3.2 Proof of Theorem 2

We are assuming that encryption scheme $\Pi$ is secure in the IND-CCA2 sense. We show it is also secure in the NM-CCA2 sense. The intuition is simple: since the adversary has access to the decryption oracle, she can decrypt the ciphertexts she would output, and so the ability to output ciphertexts is not likely to add power.

For the proof, let $B = (B_1, B_2)$ be an NM-CCA2 adversary attacking $\Pi$. We must show that $\mathsf{Adv}^{\text{nm-cca2}}_{B,\Pi}(k)$ is negligible. To this end, we describe an IND-CCA2 adversary $A = (A_1, A_2)$ attacking $\Pi$.

```
Algorithm $A_1^{\mathcal{D}_{sk}}(pk)$
    $(M, s) \leftarrow B_1^{\mathcal{D}_{sk}}(pk)$
    $x_0 \leftarrow M$ ; $x_1 \leftarrow M$
    $s' \leftarrow (M, s)$
    return $(x_0, x_1, s')$
```

```
Algorithm $A_2^{\mathcal{D}_{sk}}(x_0, x_1, s', y)$ where $s' = (M, s)$
    $(R, \mathbf{y}) \leftarrow B_2^{\mathcal{D}_{sk}}(M, s, y)$ ; $\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$
    if $(y \notin \mathbf{y} \wedge \perp \notin \mathbf{x} \wedge R(x_0, \mathbf{x}))$ then $d \leftarrow 0$
        else $d \leftarrow \{0,1\}$
    return $d$
```

Notice $A$ is polynomial time under the assumption that the running time of $B$, the time to compute $R$, and the time to sample from $M$ are all bounded by a fixed polynomial in $k$. The advantage of $A$ is given by $\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) = p_k(0) - p_k(1)$ where for $b \in \{0,1\}$ we let $p_k(b) \overset{\text{def}}{=}$

$$\Pr\Big[\,(pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (x_0, x_1, s') \leftarrow A_1^{\mathcal{D}_{sk}}(pk)\ ;\ y \leftarrow \mathcal{E}_{pk}(x_b) :$$
$$A_2^{\mathcal{D}_{sk}}(x_0, x_1, s', y) = 0\,\Big]\,.$$

Also for $b \in \{0,1\}$ we let $p'_k(b) \stackrel{\text{def}}{=}$

$$\Pr\Big[\ (pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (M, s) \leftarrow B_1^{\mathcal{D}_{sk}}(pk)\ ;\ x_0, x_1 \leftarrow M\ ;\ y \leftarrow \mathcal{E}_{pk}(x_b)\ ;$$
$$(R, \mathbf{y}) \leftarrow B_2^{\mathcal{D}_{sk}}(M, s, y)\ ;\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) :\ y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(x_0, \mathbf{x})\ \Big].$$

Now observe that $A_2$ may return 0 either when $\mathbf{x}$ is $R$-related to $x_0$ or as a result of the coin flip. Continuing with the advantage then,

$$\mathsf{Adv}_{A,\Pi}^{\text{ind-cca2}}(k) = p_k(0) - p_k(1) \ = \ \frac{1}{2}\cdot[1+p'_k(0)] - \frac{1}{2}\cdot[1+p'_k(1)] = \frac{1}{2}\cdot[p'_k(0) - p'_k(1)]$$

We now observe that the experiment of $B_2$ being given a ciphertext of $x_1$ and $R$-relating $\mathbf{x}$ to $x_0$, is exactly that defining $\mathsf{Succ}_{B,\Pi,\$}^{\text{nm-cca2}}(k)$. On the other hand, in case it is $x_0$, we are looking at the experiment defining $\mathsf{Succ}_{B,\Pi}^{\text{nm-cca2}}(k)$. So

$$\mathsf{Adv}_{B,\Pi}^{\text{nm-cca2}}(k) \ = \ p'_k(0) - p'_k(1) \ = \ 2 \cdot \mathsf{Adv}_{A,\Pi}^{\text{ind-cca2}}(k)\ .$$

But we know that $\mathsf{Adv}_{A,\Pi}^{\text{ind-cca2}}(k)$ is negligible because $\Pi$ is secure in the sense of IND-CCA2. It follows that $\mathsf{Adv}_{B,\Pi}^{\text{nm-cca2}}(k)$ is negligible, as desired.

### 3.3 Proof of Theorem 3

Assume there exists some IND-CCA1 secure encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, since otherwise the theorem is vacuously true. We now modify $\Pi$ to a new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ which is also IND-CCA1 secure but not secure in the NM-CPA sense. This will prove the theorem.

The new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ is defined as follows. Here $\overline{x}$ denotes the bitwise complement of string $x$, namely the string obtained by flipping each bit of $x$.

```
Algorithm $\mathcal{K}'(1^k)$
    $(pk, sk) \leftarrow \mathcal{K}(1^k)$
    return $(pk, sk)$

Algorithm $\mathcal{E}'_{pk}(x)$
    $y_1 \leftarrow \mathcal{E}_{pk}(x)$ ; $y_2 \leftarrow \mathcal{E}_{pk}(\overline{x})$
    return $y_1 \| y_2$

Algorithm $\mathcal{D}'_{sk}(y_1 \| y_2)$
    return $\mathcal{D}_{sk}(y_1)$
```

In other words, a ciphertext in the new scheme is a pair $y_1 \,\|\, y_2$ consisting of the encryption of the message and its complement. In decrypting, the second component is ignored. In [2] we establish that $\Pi'$ is not secure in the sense of NM-CPA sense, while it is secure in the sense of IND-CCA1.

### 3.4 Proof of Theorem 4

Let's first back up a bit and provide some intuition about why the theorem might be true and how we can prove it.

INTUITION AND FIRST ATTEMPTS. At first glance, one might think NM-CPA *does* imply IND-CCA1 (or even IND-CCA2), for the following reason. Suppose an adversary has a decryption oracle, and is asked to tell whether a given ciphertext $y$ is the encryption of $x_0$ or $x_1$, where $x_0, x_1$ are messages she has chosen earlier. She is not allowed to call the decryption oracle on $y$. It seems then the only strategy she could have is to modify $y$ to some related $y'$, call the decryption oracle on $y'$, and use the answer to somehow help her determine whether the decryption of $y$ was $x_0$ or $x_1$. But if the scheme is non-malleable, creating a $y'$ meaningfully related to $y$ is not possible, so the scheme must be chosen-ciphertext secure.

The reasoning above is fallacious. The flaw is in thinking that to tell whether $y$ is an encryption of $x_0$ or $x_1$, one must obtain a decryption of a ciphertext $y'$ related to the challenge ciphertext $y$. In fact, what can happen is that there are certain strings whose decryption yields information about the secret key itself, yet the scheme remains non-malleable.

The approach to prove the theorem is to modify a NM-CPA scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ to a new scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ which is also NM-CPA but can be broken under a non-adaptive chosen ciphertext attack. (We can assume a NM-CPA scheme exists since otherwise there is nothing to prove.) A first attempt to implement the above idea (of having the decryption of certain strings carry information about the secret key) is straightforward. Fix some ciphertext $u$ not in the range of $\mathcal{E}$ and define $\mathcal{D}'_{sk}(u) = sk$ to return the secret key whenever it is given this special ciphertext. In all other aspects, the new scheme is the same as the old one. It is quite easy to see that this scheme falls to a (non-adaptive) chosen ciphertext attack, because the adversary need only make query $u$ of its decryption oracle to recover the entire secret key. The problem is that it is not so easy to tell whether this scheme remains non-malleable. (Actually, we don't know whether it is or not, but we certainly don't have a proof that it is.)

As this example indicates, it is easy to patch $\Pi$ so that it can be broken in the sense of IND-CCA1; what we need is that it also be easy to prove that it remains NM-CPA secure. The idea of our construction below is to use a level of indirection: $sk$ is returned by $\mathcal{D}'$ in response to a query $v$ which is itself a random string that can only be obtained by querying $\mathcal{D}'$ at some other known point $u$. Intuitively, this scheme will be NM-CPA secure since $v$ will remain unknown to the adversary.

Our construction. Given a non-malleable encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ we define a new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ as follows. Here $b$ is a bit.

```
Algorithm $\mathcal{K}'(1^k)$
    $(pk, sk) \leftarrow \mathcal{K}(1^k)$
    $u, v \leftarrow \{0,1\}^k$
    $pk' \leftarrow pk \,\|\, u$
    $sk' \leftarrow sk \,\|\, u \,\|\, v$
    return $(pk', sk')$
```

```
Algorithm $\mathcal{E}'_{pk \,\|\, u}(x)$
    $y \leftarrow \mathcal{E}_{pk}(x)$
    return $0 \,\|\, y$
```

```
Algorithm $\mathcal{D}'_{sk \,\|\, u \,\|\, v}(b \,\|\, y)$
    if $b = 0$ then return $\mathcal{D}_{sk}(y)$
    else if $y = u$ then return $v$
         else if $y = v$ return $sk$
              else return $\perp$
```

Analysis. The proof of Theorem 4 is completed by establishing that $\Pi'$ is vulnerable to a IND-CCA1 attack but remains NM-CPA secure. The proofs of these claims can be found in [2].

### 3.5 Proof of Theorem 5

The approach, as before, is to take a NM-CCA1 secure encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and modify it to a new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ which is also NM-CCA1 secure, but can be broken in the NM-CCA2 sense.

Intuition. Notice that the construction of Section 3.4 will no longer work, because the scheme constructed there, not being secure in the sense of IND-CCA1, will certainly not be secure in the sense of NM-CCA1, for the same reason: the adversary can obtain the decryption key in the first stage using a couple of decryption queries. Our task this time is more complex. We want queries made in the second stage, after the challenge is received, to be important, meaning they can be used to break the scheme, yet, somehow, queries made in the first stage cannot be used to break the scheme. This means we can no longer rely on a simplistic approach of revealing the secret key in response to certain queries. Instead, the "breaking" queries in the second stage must be a function of the challenge ciphertext, and cannot be made in advance of seeing this ciphertext. We implement this idea by a "tagging" mechanism. The decryption function is capable of tagging a ciphertext so as to be able to "recognize" it in a subsequent query, and reveal in that stage information related specifically to the ciphertext, but not directly to the secret key. The tagging is implemented via pseudorandom function families.

Our construction. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the given NM-CCA1 secure encryption scheme. Fix a family $F = \{ F^k : k \geq 1 \}$ of pseudorandom functions as per [18]. (Notice that this is not an extra assumption. We know that the existence of even a IND-CPA secure encryption scheme implies the existence of a one-way function [20] which in turn implies the existence of a family of pseudorandom functions [19,18].) Here each $F^k = \{ F_K : K \in \{0,1\}^k \}$ is a finite collection in which each key $K \in \{0,1\}^k$ indexes a particular function $F_K\colon \{0,1\}^k \to \{0,1\}^k$. We define the new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ as follows. Recall that $\varepsilon$ is the empty string.

```
Algorithm $\mathcal{K}'(1^k)$
    $(pk, sk) \leftarrow \mathcal{K}(1^k)$
    $K \leftarrow \{0,1\}^k$
    $sk' \leftarrow sk \,\|\, K$
    return $(pk, sk')$
```

```
Algorithm $\mathcal{E}'_{pk}(x)$
    $y \leftarrow \mathcal{E}_{pk}(x)$
    return $0 \,\|\, y \,\|\, \varepsilon$
```

```
Algorithm $\mathcal{D}'_{sk \,\|\, K}(b \,\|\, y \,\|\, z)$ where $b$ is a bit
    if $(b = 0) \wedge (z = \varepsilon)$ then return $\mathcal{D}_{sk}(y)$
    else if $(b = 1) \wedge (z = \varepsilon)$ then return $F_K(y)$
        else if $(b = 1) \wedge (z = F_K(y))$ return $\mathcal{D}_{sk}(y)$
            else return $\perp$
```

Analysis. The proof of Theorem 5 is completed by establishing that $\Pi'$ is vulnerable to a NM-CCA2 attack but remains NM-CCA1 secure. Formal proofs of these two claims can be found in [2]. Let us sketch the intuition here.

The first is easy to see. In stage 2, given challenge ciphertext $0\|y\,\|\,\varepsilon$, the adversary would like to get back $\mathcal{D}_{sk'}(0\|y\,\|\,\varepsilon) = \mathcal{D}_{sk}(y)$, but is not allowed to query its oracle at $0\|y\,\|\,\varepsilon$. However, she can query $1\|y\|\varepsilon$ to get $F_K(y)$ and then query $1\|y\|F_K(y)$ to get back the decryption of $y$ under $sk$. At that point she can easily win.

The key point for the second claim is that to defeat the scheme, the adversary must obtain $F_K(y)$ where $0 \| y \| \varepsilon$ is the challenge. However, to do this she requires the decryption oracle. This is easy for an NM-CCA2 adversary but not for an NM-CCA1 adversary, which has a decryption oracle available only in the first stage, when $y$ is not yet known. Once $y$ is provided (in the second stage) the possibility of computing $F_K(y)$ is small because the decryption oracle is no longer available to give it for free, and the pseudorandomness of $F$ makes it hard to compute on one's own.

## 4 Results on PA

In this section we define plaintext awareness and prove that it implies the random oracle version of IND-CCA2, but is not implied by it.

Throughout this section we shall be working exclusively in the RO model. As such, all notions of security defined earlier refer, in this section, to their RO counterparts. These are obtained in a simple manner. To modify Definitions 1 and 2, begin the specified experiment (the experiment which defines advantage) by choosing a random function $H$ from the set of all functions from strings to infinite strings. Then provide an $H$-oracle to $A_1$ and $A_2$, and allow that $\mathcal{E}_{pk}$ and $\mathcal{D}_{sk}$ may depend on $H$ (which we write as $\mathcal{E}_{pk}^H$ and $\mathcal{D}_{sk}^H$).

### 4.1 Definition

Our definition of PA is from [4], except that we make one important refinement. An adversary $B$ for plaintext awareness is given a public key $pk$ and access to the random oracle $H$. We also provide $B$ with an oracle for $\mathcal{E}_{pk}^H$. (This is our refinement, and its purpose is explained later.) The adversary outputs a ciphertext $y$. To be plaintext aware the adversary $B$ should necessarily "know" the decryption $x$ of its output $y$. To formalize this it is demanded there exist some (universal) algorithm $K$ (the "plaintext extractor") that could have output $x$ just by looking at the public key, $B$'s $H$-queries and the answers to them, and the answers to $B$'s queries to $\mathcal{E}_{pk}^H$. (Note the extractor is not given the queries that $B$ made to $\mathcal{E}_{pk}^H$, just the answers received.) Let us now summarize the formal definition and then discuss it.

By $(hH, C, y) \leftarrow \mathsf{run}\, B^{H,\mathcal{E}_{pk}^H}(pk)$ we mean the following. Run $B$ on input $pk$ and oracles $H$ and $\mathcal{E}_{pk}^H$, recording $B$'s interaction with its oracles. Form into a list $hH = ((h_1, H_1), \ldots, (h_{q_H}, H_{q_H}))$ all of $B$'s $H$-oracle queries, $h_1, \ldots, h_{q_H}$, and the corresponding answers, $H_1, \ldots, H_{q_H}$. Form into a list $C = (y_1, \ldots, y_{q_E})$ the answers (ciphertexts) received as a result of $\mathcal{E}_{pk}^H$-queries. (The messages that formed the actual queries are *not* recorded.) Finally, record $B$'s output, $y$.

**Definition 3.** [**Plaintext Awareness** – PA] Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme, let $B$ be an adversary, and let $K$ be an algorithm (the "knowledge extractor"). For any $k \in \mathsf{N}$ let $\mathsf{Succ}_{K,B,\Pi}^{\mathrm{pa}}(k) \stackrel{\mathrm{def}}{=}$

$$\Pr\Big[H \leftarrow \mathsf{Hash}\ ;\ (pk, sk) \leftarrow \mathcal{K}(1^k)\ ;$$
$$(hH, C, y) \leftarrow \mathsf{run}\, B^{H,\mathcal{E}_{pk}^H}(pk) :\ K(hH, C, y, pk) = \mathcal{D}_{sk}^H(y)\Big]\ .$$

We insist that $y \notin C$; that is, $B$ never outputs a string $y$ which coincides with the value returned from some $\mathcal{E}_{pk}^{H}$-query. We say that $K$ is a $\lambda(k)$-extractor if $K$ has running time polynomial in the length of its inputs and for every adversary $B$, $\mathsf{Succ}_{K,B,\Pi}^{\mathrm{pa}}(k) \geq \lambda(k)$. We say that $\Pi$ is secure in the sense of PA if $\Pi$ is secure in the sense of IND-CPA and there exists a $\lambda(k)$-extractor $K$ where $1 - \lambda(k)$ is negligible. □

Let us now discuss this notion with particular attention to our refinement, which, as we said, consists of providing the adversary with an encryption oracle. At first glance this may seem redundant: since $B$ already has the public key, can't $B$ encrypt without making use of the encryption oracle? Absolutely. But in the RO model encrypting points oneself may involve making $H$-queries (remember that the encryption function now depends on $H$), meaning that $B$ will necessarily know any RO queries used to produce the ciphertext. (Formally, they become part of the transcript $\mathsf{run}\ B^{H,\mathcal{E}_{pk}^{H}}$.) This does not accurately model the real world, where $B$ may have access to ciphertexts via eavesdropping, in which case $B$ does not know the underlying RO queries. By giving $B$ an encryption oracle whose $H$-queries are *not* made a part of $B$'s transcript we get a stronger definition. Intuitively, should you learn a ciphertext $y_1$ for which you do not know the plaintext, *still* you should be unable to produce a ciphertext (other than $y_1$) whose plaintext you do not know. Thus the $\mathcal{E}_{pk}^{H}$ oracle models the possibility that $B$ may obtain ciphertexts in ways other than encrypting them herself.

We comment that plaintext awareness, as we have defined it, is *only* achievable in the random oracle model. (It is easy to see that if there is a scheme not using the random oracle for which an extractor as above exists then the extractor is essentially a decryption box. This can be formalized to a statement that an IND-CPA scheme cannot be plaintext aware in the above sense without using the random oracle.) It remains an interesting open question to find an analogous but achievable formulation of plaintext awareness for the standard model.

One might imagine that plaintext awareness coincides with semantic security coupled with a (non-interactive) zero-knowledge proof of knowledge [10] of the plaintext. But this is not valid. The reason is the way the extractor operates in the notion and scheme of [10]: the common random string (even if viewed as part of the public key) is under the extractor's control. In the PA notion, $pk$ is an input to the extractor and it cannot play with any of it. Indeed, note that if one could indeed achieve PA via a standard proof of knowledge, then it would be achievable in the standard (as opposed to random oracle) model, and we just observed above that this is not possible with the current definition.

### 4.2 Results

The proof of the following is in Section 4.3.

**Theorem 6.** [PA $\Rightarrow$ IND-CCA2] *If encryption scheme $\Pi$ is secure in the sense of* PA *then it is secure in the RO sense of* IND-CCA2.

**Corollary 1.** [PA $\Rightarrow$ NM-CCA2] *If encryption scheme $\Pi$ is secure in the sense of* PA *then $\Pi$ is secure in the RO sense of* NM-CCA2*.*

*Proof. Follows from Theorems 6 and the RO-version of Theorem 2.*

The above results say that PA $\Rightarrow$ IND-CCA2 $\Rightarrow$ NM-CCA2. In the other direction, we have the following, whose proof is in [2].

**Theorem 7.** [IND-CCA2$\not\Rightarrow$PA] *If there exists an encryption scheme $\Pi$ which is secure in the RO sense of* IND-CCA2*, then there exists an encryption scheme $\Pi'$ which is secure in the RO sense of* IND-CCA2 *but which is not secure in the sense of* PA*.*

### 4.3 Proof of Theorem 6

Intuition. The basic idea for proving chosen ciphertext security in the presence of some kind of proof of knowledge goes back to [15,16,7,10]. Let us begin by recalling it. Assume there is some adversary $A = (A_1, A_2)$ that breaks $\Pi$ in the IND-CCA2 sense. We construct an adversary $A' = (A'_1, A'_2)$ that breaks $\Pi$ in the IND-CPA sense. The idea is that $A'$ will run $A$ and use the extractor to simulate the decryption oracle. At first glance it may seem that the same can be done here, making this proof rather obvious. That is not quite true. Although we can follow the same paradigm, there are some important new issues that arise and must be dealt with. Let us discuss them.

The first is that the extractor cannot just run on any old ciphertext. (Indeed, if it could, it would be able to decrypt, and we know that it cannot.) The extractor can only be run on transcripts that originate from adversaries $B$ in the form of Definition 3. Thus to reason about the effectiveness of $A'$ we must present adversaries who output as ciphertext the same strings that $A'$ would ask of its decryption oracle. This is easy enough for the first ciphertext output by $A$, but not after that, because we did not allow our $B$s to have decryption oracles. The strategy will be to define a sequence of adversaries $B_1, \ldots, B_q$ so that $B_i$ uses the knowledge extractor $K$ for answering the first $i-1$ decryption queries, and then $B_i$ outputs what would have been its $i$-th decryption query. In fact this adversary $A'$ might not succeed as often as $A$, but we will show that the loss in advantage is still tolerable.

Yet, that is not the main problem. The more subtle issue is how the encryption oracle given to the adversary comes into the picture. Adversary $B_i$ will have to call its encryption oracle to "simulate" production of the challenge ciphertext received by $A_2$. It cannot create this ciphertext on its own, because to do so would incorrectly augment its transcript by the ensuing $H$-query. Thus, in fact, only one call to the encryption oracle will be required — yet this call is crucial.

Construction. For contradiction we begin with an IND-CCA2 adversary $A = (A_1, A_2)$ with a non-negligible advantage, $\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k)$ against $\Pi$. In addition, we know there exists a plaintext extractor, $K$, with high probability of success, $\mathsf{Succ}^{\text{pa}}_{K,B,\Pi}(k)$, for any adversary $B$. Using $A$ and $K$ we construct an IND-CPA adversary $A' = (A'_1, A'_2)$ with a non-negligible advantage, $\mathsf{Adv}^{\text{ind-cpa}}_{A',\Pi}(k)$ against

```
Algorithm $A'_1(pk; R)$
$hH \leftarrow ()$
Take $R_1$ from $R$
Run $A_1(pk; R_1)$, wherein
   When $A_1$ makes a query, $h$, to $H$:
      $A'_1$ asks *its* $H$-oracle $h$, obtaining $H(h)$
      Put $(h, H(h))$ at end of $hH$
      Answer $A_1$ with $H(h)$
   When $A_1$ makes its $j$th query, $y$, to $\mathcal{D}^H_{sk}$:
      $x \leftarrow K(hH, \varepsilon, y, pk)$
      Answer $A_1$ with $x$
Finally $A_1$ halts, outputting $(x_0, x_1, s)$
return $(x_0, x_1, (s, hH, pk))$
```

```
Algorithm $A'_2(x_0, x_1, (s, hH, pk), y; R)$
Take $R_2$ from $R$
Run $A_2(x_0, x_1, s, y; R_2)$, wherein
   When $A_2$ makes a query, $h$, to $H$:
      $A'_2$ asks *its* $H$-oracle $h$,
            obtaining $H(h)$
      Put $(h, H(h))$ at end of $hH$
      Answer $A_2$ with $H(h)$
   When $A_2$ makes its $j$th query, $y'$,
        to $\mathcal{D}^H_{sk}$:
      $x \leftarrow K(hH, (y), y', pk)$
      Answer $A_2$ with $x$
Finally $A_2$ halts, outputting bit, $d$
return $d$
```

**Fig. 2.** Construction of IND-CPA adversary $A' = (A'_1, A'_2)$ based on given IND-CCA2 adversary $A = (A_1, A_2)$ and plaintext extractor $K$.

$\Pi$. Think of $A'$ as the adversary $A$ with access only to a simulated decryption oracle rather than the real thing. Let $()$ denote the empty list. Recall that if $C(\cdot, \cdot, \cdots)$ is any probabilistic algorithm then $C(x, y, \cdots; R)$ means we run it with coin tosses fixed to $R$. The adversary $A'$ is defined in Figure 2.

Analysis. To reason about the behavior of $A'$ we define a sequence of adversaries $B_1, \ldots, B_q$, where $q$ is the number of decryption queries made by $A$. Using the existence of $B_1, B_2, \ldots$ we can lower bound the probability of the correctness of $K$'s answers in $A'_1$. The analysis can be found in [2].

## Acknowledgments

Following an oral presentation of an earlier version of this paper, Moni Naor suggested that we present notions of security in a manner that treats the goal and the attack model orthogonally [22]. We are indebted to him for this suggestion. We thank Hugo Krawczyk, Moti Yung, and the (other) members of the CRYPTO '98 program committee for excellent and extensive comments. Finally we thank Oded Goldreich for many discussions on these topics.

The first author was supported by a 1996 Packard Foundation Fellowship in Science and Engineering, and by NSF CAREER Award CCR-9624439. The second author was supported in part by the above mentioned grants of the first author. The fourth author was supported by NSF CAREER Award CCR-9624560 and a MICRO grant from RSA Data Security, Inc..

## 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. 30
2. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, Relations among notions of security for public-key encryption schemes. Full version of this paper, available via `http://www-cse.ucsd.edu/users/mihir/` 31, 35, 35, 37, 38, 39, 42, 43, 45, 46
3. M. Bellare and P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols. *First ACM Conference on Computer and Communications Security*, ACM, 1993. 29, 30
4. 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. 29, 29, 30, 40
5. M. Bellare and A. Sahai, private communication, May 1998. 29, 46, 46
6. D. Bleichenbacher, A chosen ciphertext attack against protocols based on the RSA encryption standard PKCS #1, *Advances in Cryptology — CRYPTO '98 Proceedings*, Lecture Notes in Computer Science, H. Krawczyk, ed., Springer-Verlag 1998. 30
7. M. Blum, P. Feldman and S. Micali, Non-interactive zero-knowledge and its applications. *Proceedings of the* 20th *Annual Symposium on Theory of Computing*, ACM, 1988. 42
8. 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, H. Krawczyk, ed., Springer-Verlag 1998. 30, 30
9. I. Damgård, Towards practical public key cryptosystems secure against chosen ciphertext attacks. *Advances in Cryptology – Crypto* 91 *Proceedings*, Lecture Notes in Computer Science Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991. 30
10. A. De Santis and G. Persiano, Zero-knowledge proofs of knowledge without interaction. *Proceedings of the* 33rd *Symposium on Foundations of Computer Science*, IEEE, 1992. 41, 41, 42
11. D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography. *Proceedings of the* 23rd *Annual Symposium on Theory of Computing*, ACM, 1991. 27, 27, 29, 30, 30, 31, 33, 33, 35, 45
12. D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography. *Technical Report CS95-27, Weizmann Institute of Science*, 1995. 27, 29, 29, 30, 33, 33, 35, 45
13. D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography. Manuscript, 1998. 27, 29, 29, 30, 33, 33, 35, 45, 46
14. O. Goldreich, A uniform complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, Vol. 6, 1993, pp. 21-53. 35
15. Z. Galil, S. Haber and M. Yung, Symmetric public key encryption. *Advances in Cryptology – Crypto* 85 *Proceedings*, Lecture Notes in Computer Science Vol. 218, H. Williams ed., Springer-Verlag, 1985. 30, 42
16. Z. Galil, S. Haber and M. Yung, Security against replay chosen ciphertext attack. *Distributed Computing and Cryptography*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 2, ACM, 1991. 42

17. S. Goldwasser and S. Micali, Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984. 27, 27, 29, 32
18. O. Goldreich, S. Goldwasser and S. Micali, How to construct random functions. *Journal of the ACM,* Vol. 33, No. 4, 1986, pp. 210–217. 39, 39
19. J. Håstad, R. Impagliazzo, L. Levin and M. Luby, Construction of a pseudorandom generator from any one-way function. Manuscript. Earlier versions in STOC 89 and STOC 90. 39
20. R. Impagliazzo and M. Luby, One-way functions are essential for complexity based cryptography. *Proceedings of the* 30th *Symposium on Foundations of Computer Science*, IEEE, 1989. 39
21. S. Micali, C. Rackoff and R. Sloan, The notion of security for probabilistic cryptosystems. *SIAM J. of Computing*, April 1988. 32
22. M. Naor, private communication, March 1998. 26, 43
23. 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. 27, 27, 30
24. 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. 27, 27, 30
25. 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/` 30
26. Y. Zheng and J. Seberry, Immunizing public key cryptosystems against chosen ciphertext attack. *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, 715–724 (1993). 30

## A Comparing our Notion of NM with Simulation NM

Let SNM refer to the original, simulation-based definition of non-malleability [11,12,13]. Its three forms are denoted SNM-CPA, SNM-CCA1, and SNM-CCA2. (In the full version of this paper [2] we recall DDN's definition. A key feature one must note here however is that the simulator is not allowed access to a decryption oracle, even in the CCA cases. We note that we are here discussing the version of SNM without "history"; we will comment on histories later.) The question we address here is how NM-ATK compares to SNM-ATK for each ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$.

It is easy to see that NM-CPA $\Rightarrow$ SNM-CPA. Intuitively, our definition can be viewed as requiring, for every adversary $A$, 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 an encryption of that message. Since $A$ does not have a decryption oracle, $A'$ can indeed do this.

If we continue to think in terms of the canonical simulator in the CCA cases, 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 our definition 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, NM-ATK implies SNM-ATK for all three types of attacks ATK, including CCA1 and CCA2. This was observed by Bellare and Sahai [5]. A proof of the following can be found in [2].

**Theorem 8.** [**5**] [NM-ATK ⇒ SNM-ATK] *If encryption scheme $\Pi$ is secure in the sense of* NM-ATK *then $\Pi$ is secure in the sense of* SNM-ATK *for any attack* $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$.

Are the definitions equivalent? For this we must consider whether SNM-ATK ⇒ NM-ATK. This is true for ATK = CCA2 (and thus the definitions are equivalent in this case) because [13] asserts that SNM-CCA2 implies IND-CCA2 and Theorem 2 asserts IND-CCA2 implies NM-CCA2. For $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA1}\}$ the question remains open.

Finally, on the subject of histories, we remark that all that we have discussed here is also true if we consider the history-inclusive versions of both definitions.