An extended abstract of this paper appears in *Advances in Cryptology – Crypto 98 Proceedings*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998. This is the full paper.

Relations Among Notions of Security for Public-Key Encryption Schemes

M. Bellare* A. Desai* D. Pointcheval† P. Rogaway‡

June 1998

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.

*Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. E-Mail: {mihir, adesai}@cs.ucsd.edu URL: http://www-cse.ucsd.edu/users/{mihir, adesai}/
Supported in part by NSF CAREER Award CCR-9624439 and a 1996 Packard Foundation Fellowship in Science and Engineering.

†Laboratoire d'Informatique de l'École Normale Supérieure, 45 rue d'Ulm, F – 75230 Paris Cedex 05. E-mail: david.pointcheval@ens.fr URL: http://www.dmi.ens.fr/~pointche/ and GREYC, Dépt d'Informatique, Université de Caen, Esplanade de la paix, F – 14032 Caen Cedex.

‡Dept. of Computer Science, Engineering II Bldg., One Shields Avenue, University of California at Davis, Davis, CA 95616, USA. E-mail: rogaway@cs.ucdavis.edu URL: http://www.cs.ucdavis.edu/~rogaway/
Supported by NSF CAREER Award CCR-9624560 and a MICRO grant from RSA Data Security, Inc..

Contents

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 [25].

We consider two different goals: *indistinguishability of encryptions*, due to Goldwasser and Micali [21], and *non-malleability*, due to Dolev, Dwork and Naor [13]. 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 [26], 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 [27], 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 $\{\mathrm{IND}, \mathrm{NM}\}$ and attacks $\{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$ 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 [21];[1] IND-CCA1 is the notion of [26]; IND-CCA2 is the notion of [27]; NM-CPA, NM-CCA1 and NM-CCA2 are from [13, 14, 15].

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

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

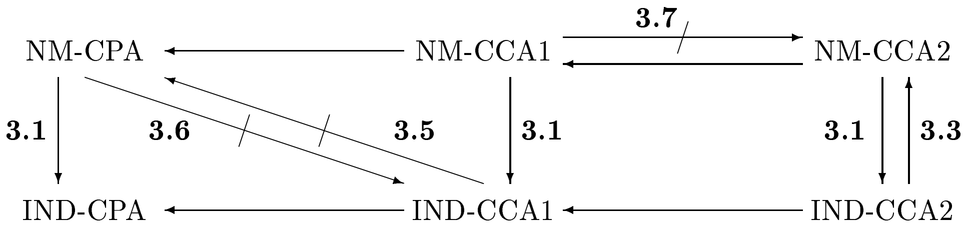

Figure 1: *An arrow is an implication, and in the directed graph given by the arrows, there is a path from* **A** *to* **B** *if and only* **A** $\Rightarrow$ **B**. *The hatched arrows represent separations we actually prove; all others follow automatically. The number on an arrow or hatched arrow refers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

the following:

- **A** $\Rightarrow$ **B**: A proof that if Π is any encryption scheme meeting notion of security **A** then Π also meets notion of security **B**.
- **A** $\not\Rightarrow$ **B**: A construction of an encryption scheme Π 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.

1.3 Plaintext Awareness

Another adversarial goal we will consider is *plaintext awareness* (PA), first defined by Bellare and Rogaway [6]. 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.)

[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.

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 [5] 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 4.2 shows that PA (together with the RO version of IND-CPA) implies the RO version of IND-CCA2. In the other direction, Theorem 4.4 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 [13, 14, 15] 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, the definitions are equivalent [7], under any form of attack.

Thus the results in this paper are not affected by the definitional change. 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 [21] 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 [6]. 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 [28]. 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

RELATIONS. The most recent version of the work of Dolev, Dwork and Naor, the manuscript [15], has, independently of our work, considered the question of relations among notions of encryptions beyond IND-CPA. It 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 [13] and the 1995 technical report [14], do not contain these claims.)

FOUNDATIONS. The theoretical treatment of public-key encryption begins with Goldwasser and Micali [21] and continues with Yao [29], Micali, Rackoff and Sloan [24], and Goldreich [18, 19]. These works treat privacy under chosen-plaintext attack (the notion we are capturing via IND-CPA). They show that various formalizations of it are equivalent, in various models. Specifically, Goldwasser and Micali introduced, and showed equivalent, the notions of indistinguishability and semantic security; Yao introduced a notion based on computational entropy; Micali, Rackoff and Sloan showed that appropriate variants of the original definition are equivalent to this; Goldreich [18] made important refinements to the notion of semantic security and showed that the equivalences still held; and Goldreich [19] provided definitions and equivalences for the case of uniform adversaries. We build on these foundations both conceptually and technically. In particular, this body of work effectively justifies our adopting one particular formulation of privacy under chosen-plaintext attack, namely IND-CPA.

None of the above works considered chosen-ciphertext attacks and, in particular, the question of whether indistinguishability and semantic security are equivalent in this setting. In fact, semantic security under chosen-ciphertext attack seems to have not even been defined. As mentioned earlier, definitions for semantic security under CCA can be obtained along the lines of our new definition of non-malleability. We expect (and hope) that, after doing this, the equivalence between semantic security and indistinguishability continue to hold with respect to CCA, but this has not been checked.

RECENT WORK ON SIMPLIFYING NON-MALLEABILITY. As noted above, Bellare and Sahai [7] have shown that the definition of non-malleability given in this paper is equivalent to the original one of [13, 14, 15]. In addition, they provide a novel formulation of non-malleability in terms of indistinguishability, showing that non-malleability is just a form of indistinguishability under a certain type of attack they call a parallel attack. Their characterization can be applied to simplify some of the results in this paper.

SCHEMES. 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 [26], which meets IND-CCA1, that of [13], which meets IND-CCA2, and the much more efficient recent scheme of Cramer and Shoup [10], which also meets IND-CCA2. Next are the schemes proven secure in a random-oracle model; here we have those of [5, 6], which meet PA and are as efficient as schemes in current standards. Then there are schemes without proofs, such as those of [11, 30]. Finally, there are schemes for non-standard models, like [16, 27].

We comment that it follows from our results that the above mentioned scheme of [10], shown to meet IND-CCA2, is also non-malleable, even under an adaptive chosen-ciphertext attack.

Symmetric encryption. This paper is about relating notions of security for public-key (ie. asymmetric) encryption. The same questions can be asked for private-key (ie. symmetric) encryption. Definitions for symmetric encryption scheme privacy under CPA were given by [2]. Those notions can be lifted to deal with CCA. Definitions for non-malleability in the private-key setting can be obtained by adapting the public-key ones. Again we would expect (and hope) that, if properly done, the analogs to the relations we have proven remain.

One feature of definitions in this setting is worth highlighting. Recall that in the public-key setting, nothing special had to be done to model CPA; it corresponds just to giving the adversary the public key. Not so in a private-key setting. The suggestion of [3] is to give the adversary an oracle for encryption under the private key. This must be done in all definitions, and it is under this notion that we expect to see an analog of the results for the public-key case.

Goldreich, in discussions on this issue, has noted that in the private-key case, one can consider an attack setting weaker than CPA, where the adversary is not given an encryption oracle. He points out that under this attack it will not even be true that non-malleability implies indistinguishability.

Encryption scheme security which goes beyond indistinguishability is important in the private-key case too, and we feel it deserves a full treatment of its own which would explore and clarify some of the above issues.

Further remarks. We comment that non-malleability is a general notion that applies to primitives other than encryption [13]. Our discussion is limited to its use in asymmetric encryption.

Bleichenbacher [8] has recently shown that a popular encryption scheme, RSA PKCS #1, does not achieve IND-CCA1. He also describes a popular protocol for which this causes problems. His results reinforce the danger of assuming anything beyond IND-CPA which has not been demonstrated.

A preliminary version of this paper appeared as [3]. We include here material which was omitted from that abstract due to space limitations.

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

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 CCAi 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 [21, 24], 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 cpa, cca1, cca2, while ATK is then the corresponding formal symbol from CPA, CCA1, 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, ε.

Definition 2.1 [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 atk $\in$ {cpa, cca1, 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

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

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 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. ▮

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 [13], with refinements in [14, 15]. 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 [13, 14, 15], 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 is equivalent to DDN's [7].

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 2.2 [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 $\in \{\text{cpa}, \text{cca1}, \text{cca2}\}$ and $k \in \mathsf{N}$ define

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

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

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

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

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

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 Π 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.

Remark 2.3 [Histories] One might want to strengthen the notion to require that the adversary's advantage remains small even if it obtains, somehow, some a priori information about the message x.

Such incorporation of message "history" was made in Goldreich's formalizations of semantic security [19]. The DDN definitions similarly incorporate history in the context of non-malleability. The same can be done for our definition. Whether or not one uses histories does not affect the results in this paper, so for simplicity we have omitted this feature in the formal definition above, and discuss it only in remarks.

Let us briefly sketch our way of adding histories to our definition. We simply change the meaning of the message space M output by A during the first phase of her execution: make M a distribution on pairs (x, a) consisting of messages and their associated auxiliary information (history). Now modify the definition of $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k)$ so as follows. The sampling from M in the experiment becomes $(x, a) \leftarrow M$, and, later, a is given as an additional input to A_2. Everything else is the same. Similarly modify $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k)$ as follows. The sampling from M becomes $(x, a), (\tilde{x}, \tilde{a}) \leftarrow M$, and, later, A_2 gets $\tilde{a}$ (not a) as an additional input. Everything else is the same.

We recall that the traditional approach of incorporating histories followed in [19, 14] is via a fixed history function $\mathsf{hist}(x)$ that is then universally quantified at the start. Our approach would seem to be simpler and also more general, since it allows one to associate with messages probabilistic information efficiently computable only knowing secret coins associated to the message.

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 [13] 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 Section 3.3.

Theorem 3.1 [NM-ATK $\Rightarrow$ IND-ATK] *If encryption scheme* Π *is secure in the sense of* NM-ATK *then* Π *is secure in the sense of* IND-ATK, *for any attack* ATK $\in \{\text{CPA}, \text{CCA1}, \text{CCA2}\}$.

Remark 3.2 Recall that the relation R in Definition 2.2 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 3.4.

Theorem 3.3 [IND-CCA2 $\Rightarrow$ NM-CCA2] *If encryption scheme* Π *is secure in the sense of* IND-CCA2 *then* Π *is secure in the sense of* NM-CCA2.

Remark 3.4 Theorem 3.3 coupled with Theorem 3.1 and Remark 3.2 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 3.3. In contrast, the following says that non-adaptive chosen-ciphertext security does *not* imply non-malleability. The proof is in Section 3.5.

Theorem 3.5 [IND-CCA1$\not\Rightarrow$NM-CPA] *If there exists an encryption scheme* Π *which is secure in the sense of* IND-CCA1, *then there exists an encryption scheme* Π' *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 3.6.

Theorem 3.6 [NM-CPA$\not\Rightarrow$IND-CCA1] *If there exists an encryption scheme* Π *which is secure in the sense of* NM-CPA, *then there exists an encryption scheme* Π' *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 3.7 for the proof of the following.

Theorem 3.7 [NM-CCA1$\not\Rightarrow$NM-CCA2] *If there exists an encryption scheme* Π *which is secure in the sense of* NM-CCA1, *then there exists an encryption scheme* Π' *which is secure in the sense of* NM-CCA1 *but which is not secure in the sense of* NM-CCA2.

3.2 Notation and Preliminaries

For relations R which could be of arbitrary arity we use the simplifying notation $R(a, b)$ as a shorthand for $R(a, \mathbf{b})$ when it is clear that $\mathbf{b}[1] = b$ and $|\mathbf{b}| = 1$. We let $\overline{a}$ denote the bitwise complement (namely the string obtained by flipping each bit) of a.

For an IND-ATK adversary $A = (A_1, A_2)$ we will, whenever convenient, assume that the messages x_0, x_1 that A_1 outputs are distinct. Intuitively this cannot decrease the advantage because the contribution to the advantage in case they are equal is zero. Actually one has to be a little careful. The claim will be that we can modify A to make sure that the output messages are distinct, and one has to be careful to make sure that when A outputs equal messages the modified adversary does not get any advantage, so that the advantage of the modified adversary is the same as that of the original one. For completeness we encapsulate the claim in the following proposition.

Proposition 3.8 *Let* $A = (A_1, A_2)$ *be any adversary attacking encryption scheme* Π *in the sense of* IND-ATK. *Then there exists another adversary* $B = (B_1, B_2)$ *attacking* Π *in the sense of* IND-ATK *such that the two (equal length) messages that* B_1 *outputs are always distinct,* $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) = \mathsf{Adv}^{\text{ind-atk}}_{A,\Pi}(k)$, *and the running time of* B *is within a constant factor of that of* A.

Proof: Adversaries A and B have access to an oracle $\mathcal{O}_1$ in their first stage and an oracle $\mathcal{O}_2$ in their second stage, these oracles being instantiated according to the attack ATK as described in the definitions. The adversary $B = (B_1, B_2)$ is as follows:

`Algorithm` $B_1^{\mathcal{O}_1}(pk)$	`Algorithm` $B_2^{\mathcal{O}_2}(x'_0, x'_1, s', y)$ where $s' = s \,\|\, d$
$(x_0, x_1, s) \leftarrow A_1^{\mathcal{O}_1}(pk)$	`if` $d = 0$ `then` $c \leftarrow A_2^{\mathcal{O}_2}(x'_0, x'_1, s, y)$
`if` $x_0 \neq x_1$ `then` $d \leftarrow 0$ `else` $d \leftarrow 1$	`else` $c \leftarrow \{0,1\}$
$x'_0 \leftarrow x_0$; $s' \leftarrow s \,\|\, d$	`return` c
`if` $d = 0$ `then` $x'_1 \leftarrow x_1$ `else` $x'_1 \leftarrow \overline{x_0}$	
`return` (x'_0, x'_1, s')	

Note that by defining x'_0, x'_1 this way we always have $x'_0 \neq x'_1$. Also note that when $x_0 = x_1$ we have B_2 output a random bit c to make sure its advantage in that case is zero.

It is easy to see that the running time of B is within a constant factor of that of A. Now we claim that $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) = \mathsf{Adv}^{\text{ind-atk}}_{A,\Pi}(k)$. To justify this, consider the experiments underlying the definitions of the advantages of A and B, respectively:

$$\begin{aligned}
\mathsf{Experiment1} &\stackrel{\text{def}}{=} (pk, sk) \leftarrow \mathcal{K}(1^k) \; ; \; (x_0, x_1, s) \leftarrow A_1(pk) \; ; \; b \leftarrow \{0,1\} \; ; \\
&\qquad y \leftarrow \mathcal{E}_{pk}(x_b) \; ; \; c \leftarrow A_2^{\mathcal{O}_2}(x_0, x_1, s, y) \\
\mathsf{Experiment2} &\stackrel{\text{def}}{=} (pk, sk) \leftarrow \mathcal{K}(1^k) \; ; \; (x_0, x_1, s) \leftarrow A_1(pk) \; ; \; b \leftarrow \{0,1\} \; ; \\
&\qquad y \leftarrow \mathcal{E}_{pk}(x_b) \; ; \; c \leftarrow B_2^{\mathcal{O}_2}(x'_0, x'_1, s \,\|\, d, y) \; .
\end{aligned}$$

In the last experiment, x'_0, x'_1, d are defined in terms of x_0, x_1 as per the code of B_1. Let $\Pr_1[\,\cdot\,] = \Pr[\mathsf{Experiment1} : \,\cdot\,]$ be the probability function under $\mathsf{Experiment1}$ and $\Pr_2[\,\cdot\,] = \Pr[\mathsf{Experiment2} : \,\cdot\,]$ be that under $\mathsf{Experiment2}$. By definition

$$Adv^{\text{ind-atk}}_{A,\Pi}(k) \;=\; 2 \cdot \Pr_1[\, b = c \,] - 1 \qquad \text{and} \qquad \mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) \;=\; 2 \cdot \Pr_2[\, b = c \,] - 1 \; .$$

Thus it suffices to show that $\Pr_1[\, b = c \,] = \Pr_2[\, b = c \,]$. Let E denote the event that $x_0 = x_1$, or, equivalently, that $d = 1$. Then

$$\begin{aligned}
\Pr_1[\, b = c \,] &= \Pr_1[\, b = c \mid E \,] \cdot \Pr_1[\, E \,] + \Pr_1\left[\, b = c \mid \overline{E} \,\right] \cdot \Pr_1\left[\, \overline{E} \,\right] \\
\Pr_2[\, b = c \,] &= \Pr_2[\, b = c \mid E \,] \cdot \Pr_2[\, E \,] + \Pr_2\left[\, b = c \mid \overline{E} \,\right] \cdot \Pr_2\left[\, \overline{E} \,\right] .
\end{aligned}$$

That $\Pr_1[\, b = c \,] = \Pr_2[\, b = c \,]$ now follows by putting together the following observations:

- $\Pr_1[\, E \,] = \Pr_2[\, E \,]$ since E depends only on A_1.
- $\Pr_1[\, b = c \mid E \,] = 1/2$ because when E is true, A_2 has no information about b. On the other hand $\Pr_2[\, b = c \mid E \,] = 1/2$ because when E is true we have B_2 output a random bit.
- $\Pr_1\left[\, b = c \mid \overline{E} \,\right] = \Pr_2\left[\, b = c \mid \overline{E} \,\right]$ because in this case the experiments are the same, namely we are looking at the output of A_2.

This completes the proof of Proposition 3.8. ▮

3.3 Proof of Theorem 3.1: NM-ATK $\Rightarrow$ IND-ATK

We are assuming that encryption scheme Π is secure in the NM-ATK sense. We will show it is also secure in the IND-ATK sense. Let $B = (B_1, B_2)$ be a IND-ATK adversary attacking Π. We want to show that $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(\cdot)$ is negligible. To this end, we describe a NM-ATK adversary $A = (A_1, A_2)$ attacking Π. Adversaries A and B have access to an oracle $\mathcal{O}_1$ in their first stage and an oracle $\mathcal{O}_2$ in their second stage, these oracles being instantiated according to the attack ATK as per the definitions. Recall that $\overline{z}$ denotes the bitwise complement of a string z.

`Algorithm` $A_1^{\mathcal{O}_1}(pk)$ $(x_0, x_1, s) \leftarrow B_1^{\mathcal{O}_1}(pk)$ $M := \{x_0, x_1\}$ $s' \leftarrow (x_0, x_1, pk, s)$ `return` (M, s')	`Algorithm` $A_2^{\mathcal{O}_2}(M, s', y)$ where $s' = (x_0, x_1, pk, s)$ $c \leftarrow B_2^{\mathcal{O}_2}(x_0, x_1, s, y)$ $y' \leftarrow \mathcal{E}_{pk}(\overline{x_c})$ `return` (R, y') where $R(a, b) = 1$ iff $a = \overline{b}$

The notation $M := \{x_0, x_1\}$ means that M is being assigned the probability space which assigns to each of x_0 and x_1 a probability of $1/2$. $A_2^{\mathcal{O}_2}$ outputs (the description of) the complement relation R, which for any arguments a, b is 1 if $a = \overline{b}$ and 0 otherwise.

We consider the advantage of A, given by

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

where

$$\begin{aligned}
\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) \;&=\; \Pr\Big[(pk,sk) \leftarrow \mathcal{K}(1^k)\;;\; (M,s') \leftarrow A_1^{\mathcal{O}_1}(pk)\;;\; x \leftarrow M\;;\; y \leftarrow \mathcal{E}_{pk}(x)\;; \\
&\qquad (R,y') \leftarrow A_2^{\mathcal{O}_2}(M,s',y)\;;\; x' \leftarrow \mathcal{D}_{sk}(y') :\; y \neq y' \wedge \bot \neq x' \wedge R(x,x')\Big] \\
\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) \;&=\; \Pr\Big[(pk,sk) \leftarrow \mathcal{K}(1^k)\;;\; (M,s') \leftarrow A_1^{\mathcal{O}_1}(pk)\;;\; x,\tilde{x} \leftarrow M\;;\; y \leftarrow \mathcal{E}_{pk}(x)\;; \\
&\qquad (R,y') \leftarrow A_2^{\mathcal{O}_2}(M,s',y)\;;\; x' \leftarrow \mathcal{D}_{sk}(y') :\; y \neq y' \wedge \bot \neq x' \wedge R(\tilde{x},x')\Big] \;.
\end{aligned}$$

Recall the advantage of B is given by $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) = 2 \cdot p_k - 1$, where

$$\begin{aligned}
p_k \;&=\; \Pr\Big[\,(pk,sk) \leftarrow \mathcal{K}(1^k)\;;\; (x_0,x_1,s) \leftarrow B_1^{\mathcal{O}_1}(pk)\;;\; b \leftarrow \{0,1\}\;; \\
&\qquad y \leftarrow \mathcal{E}_{pk}(x_b)\;;\; c \leftarrow B_2^{\mathcal{O}_2}(x_0,x_1,s,y) :\; c = b\Big] \;.
\end{aligned}$$

By Proposition 3.8 we may assume here, without loss of generality, that we always have $x_0 \neq x_1$. This turns out to be important below.

Claim 1: $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) = p_k$.

Proof: Look first at the code of A_2. Note that $R(x,x')$ is true iff $\mathcal{D}_{sk}(y) = x_c$. Also note that when $R(x,x')$ is true it must be that $x \neq x'$ and hence, by the unique decryptability of the encryption scheme, that $y \neq y'$. Also we always have $\bot \neq x'$.

Now, consider the experiment defining p_k. An important observation is that $\mathcal{D}_{sk}(y) = x_c$ iff $b = c$. (This uses the fact that $x_0 \neq x_1$, and would not be true otherwise.) Now one can put this together with the above and see that $b = c$ in the experiment underlying p_k exactly when $y \neq y' \wedge \bot \neq x' \wedge R(x,x')$ in the experiment underlying $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k)$. □

Claim 2: $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) = 1/2$.

Proof: This follows from an information theoretic fact, namely that A has no information about the message $\tilde{x}$ with respect to which its success is measured. □

Now we can apply the claims to get

$$\begin{aligned}
\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) \;&=\; 2 \cdot \left(p_k - \frac{1}{2}\right) \\
&=\; 2 \cdot \left(\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) - \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k)\right) \\
&\leq\; 2 \cdot \left|\, \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) - \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) \,\right| \\
&=\; 2 \cdot \mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(k) \;.
\end{aligned}$$

But since Π is secure in the NM-ATK sense we know that $\mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(\cdot)$ is negligible, and hence the above implies $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(\cdot)$ is negligible too. This concludes the proof of Theorem 3.1.

The claim of Remark 3.2 is clear from the above because the relation R output by A is binary.

3.4 Proof of Theorem 3.3: IND-CCA2 ⇒ NM-CCA2

We are assuming that encryption scheme Π 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 Π. We must show that $\mathsf{Adv}^{\text{nm-cca2}}_{B,\Pi}(\cdot)$ is negligible. To this end, we describe an IND-CCA2 adversary $A = (A_1, A_2)$ attacking Π.

`Algorithm` $A_1^{\mathcal{D}_{sk}}(pk)$	`Algorithm` $A_2^{\mathcal{D}_{sk}}(x_0, x_1, s', y)$ where $s' = (M, s)$
$(M, s) \leftarrow B_1^{\mathcal{D}_{sk}}(pk)$	$(R, \mathbf{y}) \leftarrow B_2^{\mathcal{D}_{sk}}(M, s, y)$; $\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$
$x_0 \leftarrow M$; $x_1 \leftarrow M$	`if` $(y \notin \mathbf{y} \wedge \perp \notin \mathbf{x} \wedge R(x_0, \mathbf{x}))$ `then` $d \leftarrow 0$
$s' \leftarrow (M, s)$	`else` $d \leftarrow \{0,1\}$
`return` (x_0, x_1, s')	`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

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

Also for $b \in \{0,1\}$ we let

$$
\begin{aligned}
p'_k(b) \;=\; \Pr\Big[\, & (pk, sk) \leftarrow \mathcal{K}(1^k) \;;\; (M, s) \leftarrow B_1^{\mathcal{D}_{sk}}(pk) \;;\; x_0, x_1 \leftarrow M \;;\; y \leftarrow \mathcal{E}_{pk}(x_b) \;; \\
& (R, \mathbf{y}) \leftarrow B_2^{\mathcal{D}_{sk}}(M, s, y) \;;\; \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) : \; y \notin \mathbf{y} \wedge \perp \notin \mathbf{x} \wedge R(x_0, \mathbf{x}) \,\Big] .
\end{aligned}
$$

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,

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

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 $\mathsf{Succ}^{\text{nm-cca2}}_{B,\Pi,\$}(k)$. On the other hand, in case it is x_0, we are looking at the experiment defining $\mathsf{Succ}^{\text{nm-cca2}}_{B,\Pi}(k)$. So

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

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

3.5 Proof of Theorem 3.5: IND-CCA1 $\not\Rightarrow$ NM-CPA

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 Π 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` $\mathcal{K}'(1^k)$	`Algorithm` $\mathcal{E}'_{pk}(x)$	`Algorithm` $\mathcal{D}'_{sk}(y_1 \| y_2)$
$(pk, sk) \leftarrow \mathcal{K}(1^k)$	$y_1 \leftarrow \mathcal{E}_{pk}(x)$; $y_2 \leftarrow \mathcal{E}_{pk}(\overline{x})$	`return` $\mathcal{D}_{sk}(y_1)$
`return` (pk, sk)	`return` $y_1 \| y_2$	

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. It is now quite easy to see that:

Claim 3.9 *Π' is not secure in the* NM-CPA *sense.*

Proof: Given a ciphertext $y_1 \,\|\, y_2$ of a message x, it is easy to create a ciphertext of $\overline{x}$: just output $y_2 \,\|\, y_1$. Thus, the scheme is malleable.

Formally, we can specify a polynomial time adversary $A = (A_1, A_2)$ that breaks Π' in the sense of NM-CPA, with probability almost one, as follows. $A_1(pk)$ outputs (M, ϕ) where M puts a uniform distribution on $\{0,1\}^k$. Then algorithm $A_2(M, \phi, y_1 \,\|\, y_2)$ outputs $(R, y_2 \,\|\, y_1)$ where R describes the binary relation defined by $R(m_1, m_2) = 1$ iff $m_1 = \overline{m_2}$. It is easy to see that the plaintext, x', corresponding to the ciphertext that A outputs is R-related to x with probability 1. Observe that the probability of some random plaintext $\tilde{x}$ being R-related to x' is at most 2^{-k}. Thus $\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi'}(k)$ is $1 - 2^{-k}$ which is not negligible. (In fact it is close to one.) Hence A is a successful adversary and the scheme is not secure in the sense of NM-CPA. ▮

On the other hand, a hybrid argument establishes that Π' retains the IND-CCA1 security of Π:

Claim 3.10 *Π' is secure in the sense of* IND-CCA1.

Proof: Let $B = (B_1, B_2)$ be some polynomial time adversary attacking Π' in the IND-CCA1 sense. We want to show that $\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k)$ is negligible. To do so, consider the following probabilities, defined for $i, j \in \{0,1\}$:

$$\begin{aligned} p_k(i,j) \;=\; \Pr\Big[\, & (pk, sk) \leftarrow \mathcal{K}(1^k) \;;\; (x_0, x_1, s) \leftarrow B_1^{\mathcal{D}_{sk}}(pk) \;;\; y_1 \leftarrow \mathcal{E}_{pk}(x_i) \;;\; y_2 \leftarrow \mathcal{E}_{pk}(\overline{x_j}) : \\ & B_2(x_0, x_1, s, y_1 \| y_2) = 1 \,\Big] \,. \end{aligned}$$

We know that $\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k) = p_k(1,1) - p_k(0,0)$. The following lemmas state that, under our assumption that Π is IND-CCA1-secure, it must be that the differences $p_k(1,1) - p_k(1,0)$ and $p_k(1,0) - p_k(0,0)$ are both negligible. This will complete the proof since

$$\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k) \;=\; p_k(1,1) - p_k(0,0) \;=\; [p_k(1,1) - p_k(1,0)] + [p_k(1,0) - p_k(0,0)] \;,$$

being the sum of two negligible functions, will be negligible. So it remains to (state and) prove the lemmas.

Lemma 1: $p_k(1,1) - p_k(1,0)$ is negligible.

Proof: We can construct an adversary $A = (A_1, A_2)$ that attacks the scheme Π in the IND-CCA1 sense, as follows:

`Algorithm` $A_1^{\mathcal{D}_{sk}}(pk)$	`Algorithm` $A_2(m_0, m_1, s, y)$
$\quad (x_0, x_1, s) \leftarrow B_1^{\mathcal{D}'_{sk}}(pk)$	$\quad y_1 \leftarrow \mathcal{E}_{pk}(\overline{m_1}) \;;\; y_2 \leftarrow y$
$\quad m_0 \leftarrow \overline{x_0} \;;\; m_1 \leftarrow \overline{x_1}$	$\quad d \leftarrow B_2(\overline{m_0}, \overline{m_1}, s, y_1 \,\|\, y_2)$
$\quad$ **return** (m_0, m_1, s)	$\quad$ **return** d

The computation $B_1^{\mathcal{D}'_{sk}}(pk)$ is done by A_1 simulating the $\mathcal{D}'_{sk}$ oracle. It can do this by replying to query $y_1 \,\|\, y_2$ via $\mathcal{D}_{sk}(y_1)$, using its own $\mathcal{D}_{sk}$ oracle and the definition of $\mathcal{D}'_{sk}$. This adversary is polynomial time. One can now check the following:

$$\Pr\Big[(pk, sk) \leftarrow \mathcal{K}(1^k) \;;\; (m_0, m_1, s) \leftarrow A_1^{\mathcal{D}_{sk}}(pk) \;;\; y \leftarrow \mathcal{E}_{pk}(m_1) :\; A_2(m_0, m_1, s, y) = 1\Big] \;=\; p_k(1,1)$$

$$\Pr\Big[(pk, sk) \leftarrow \mathcal{K}(1^k) \;;\; (m_0, m_1, s) \leftarrow A_1^{\mathcal{D}_{sk}}(pk) \;;\; y \leftarrow \mathcal{E}_{pk}(m_0) :\; A_2(m_0, m_1, s, y) = 1\Big] \;=\; p_k(1,0)$$

Thus $\mathsf{Adv}^{\text{ind-cca1}}_{A,\Pi}(k) = p_k(1,1) - p_k(1,0)$. The assumed security of Π in the IND-CCA1 sense now implies the latter difference is negligible. $\square$

Lemma 2: $p_k(1,0) - p_k(0,0)$ is negligible.

Proof: We can construct an adversary $A = (A_1, A_2)$ that attacks the scheme Π in the IND-CCA1 sense, as follows:

Algorithm $A_1^{\mathcal{D}_{sk}}(pk)$	Algorithm $A_2(x_0, x_1, s, y)$
$(x_0, x_1, s) \leftarrow B_1^{\mathcal{D}'_{sk}}(pk)$	$y_1 \leftarrow y$ and $y_2 \leftarrow \mathcal{E}_{pk}(\overline{x_0})$
return (x_0, x_1, s)	$d \leftarrow B_2(x_0, x_1, s, y_1 \| y_2)$
	return d

Again A is polynomial time and can simulate $\mathcal{D}'_{sk}$ given $\mathcal{D}_{sk}$. We observe that

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

Thus $\mathsf{Adv}^{\text{ind-cca1}}_{A,\Pi}(k) = p_k(1,0) - p_k(0,0)$. The assumed security of Π in the IND-CCA1 sense now implies the latter difference is negligible. $\square$

This completes the proof of Claim 3.10. ▮

Remark 3.11 We could have given a simpler scheme Π' than the one above that would be secure in the IND-CCA1 sense but not in the NM-CPA sense. Let $\mathcal{K}'$ be as above, let $\mathcal{E}'_{pk}(x) \leftarrow y \,\|\, b$ where $y \leftarrow \mathcal{E}_{pk}(x)$ and $b \leftarrow \{0,1\}$ and $\mathcal{D}'_{sk}(b \,\|\, y) \leftarrow \mathcal{D}_{sk}(y)$. The malleability of Π' arises out of the ability of the adversary to create another ciphertext from the challenge ciphertext $y \,\|\, b$, by returning $y \,\|\, \overline{b}$. This is allowed by Definition 2.2 since the only restriction is that the vector of ciphertexts $\mathbf{y}$ the adversary outputs should not contain $y \,\|\, b$. However, the definition of [13] did not allow this, and, in order to have a stronger separation result that also applies to their notion, we gave the above more involved construction.

3.6 Proof of Theorem 3.6: NM-CPA $\not\Rightarrow$ IND-CCA1

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 Π 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:

Algorithm $\mathcal{K}'(1^k)$	Algorithm $\mathcal{E}'_{pk \,\|\, u}(x)$	Algorithm $\mathcal{D}'_{sk \,\|\, u \,\|\, v}(b \,\|\, y)$ where $b \in \{0,1\}$
$(pk, sk) \leftarrow \mathcal{K}(1^k)$	$y \leftarrow \mathcal{E}_{pk}(x)$	if $b = 0$ then return $\mathcal{D}_{sk}(y)$
$u, v \leftarrow \{0,1\}^k$	return $0 \,\|\, y$	else if $y = u$ then return v
$pk' \leftarrow pk \,\|\, u$		else if $y = v$ return sk
$sk' \leftarrow sk \,\|\, u \,\|\, v$		else return $\perp$
return (pk', sk')		

Analysis. The proof of Theorem 3.6 is completed by establishing that Π' is vulnerable to a IND-CCA1 attack but remains NM-CPA secure.

Claim 3.12 *Π' is not secure in the sense of* IND-CCA1.

Proof: The adversary queries $\mathcal{D}'_{sk \,\|\, u \,\|\, v}(\cdot)$ at $1 \,\|\, u$ to get v, and then queries it at the point $1 \,\|\, v$, to get sk. At this point, knowing the secret key, she can obviously perform the distinguishing task we later require of her.

If you wish to see it more formally, the find stage A_1 of the adversary gets pk as above and outputs any two distinct, equal length messages x_0, x_1. In the next stage, it receives a ciphertext $0 \,\|\, y \leftarrow \mathcal{E}'_{pk \,\|\, u}(x_b)$ where b was a random bit. Now it can compute $\mathcal{D}_{sk}(y)$ to recover the message and thus determine b with probability one. It is obviously polynomial time. ▌

Remember that Π is assumed secure in the sense of NM-CPA. We will use this to establish the following:

Claim 3.13 *Π' is secure in the sense of* NM-CPA.

Proof: To prove this claim we consider a polynomial time adversary B attacking Π' in the NM-CPA sense. We want to show that $\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(\cdot)$ is negligible. To do this, we construct an adversary $A = (A_1, A_2)$ that attacks Π in the NM-CPA sense. The idea is that A can run B as a subroutine and simulate the choosing of u, v by the key generation algorithm $\mathcal{K}'$ for B.

$$
\begin{array}{l|l}
\texttt{Algorithm } A_1(pk) & \texttt{Algorithm } A_2(M, s', y) \text{ where } s' = (s, u, v, pk) \\
\quad u, v \leftarrow \{0,1\}^k & \quad (R, \mathbf{z}) \leftarrow B_2(M, s, 0 \,\|\, y) \\
\quad pk' \leftarrow pk \,\|\, u & \quad \texttt{for } 1 \le i \le |\mathbf{z}| \texttt{ do} \text{ parse } \mathbf{z}[i] \text{ as } b_i \,\|\, z_i \text{ where } b_i \text{ is a bit} \\
\quad (M, s) \leftarrow B_1(pk') & \quad \texttt{for } 1 \le i \le |\mathbf{z}| \texttt{ do} \\
\quad s' \leftarrow (s, u, v, pk) & \quad\quad \texttt{if } b_i = 0 \texttt{ then } \mathbf{y}[i] \leftarrow z_i \\
\quad \texttt{return } (M, s') & \quad\quad \texttt{else if } (b_i = 1) \wedge (z_i = u) \texttt{ then } \mathbf{y}[i] \leftarrow \mathcal{E}_{pk}(v) \\
 & \quad\quad\quad\quad \texttt{else } \mathbf{y}[i] \leftarrow y \\
 & \quad \texttt{return } (R, \mathbf{y})
\end{array}
$$

We now define two experiments. The first is the one under which $\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k)$ is evaluated, and the second is the one under which $\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k)$ is evaluated:

$$
\begin{aligned}
\mathsf{Experiment1} \stackrel{\text{def}}{=}\ & (pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (M, (s, u, v, pk)) \leftarrow A_1(pk)\ ;\ x, \tilde{x} \leftarrow M\ ;\ y \leftarrow \mathcal{E}_{pk}(x)\ ; \\
& (R, \mathbf{y}) \leftarrow A_2(M, (s, u, v, pk), y)\ ;\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) \\
\mathsf{Experiment2} \stackrel{\text{def}}{=}\ & (pk \,\|\, u, sk \,\|\, u \,\|\, v) \leftarrow \mathcal{K}'(1^k)\ ;\ (M, s) \leftarrow B_1(pk \,\|\, u)\ ;\ x, \tilde{x} \leftarrow M\ ; \\
& 0 \,\|\, y \leftarrow \mathcal{E}'_{pk \,\|\, u}(x)\ ;\ (R, \mathbf{z}) \leftarrow B_2(M, s, 0 \,\|\, y)\ ;\ \mathbf{w} \leftarrow \mathcal{D}'_{sk\|u\|v}(\mathbf{z})\ .
\end{aligned}
$$

Let $\Pr_1[\cdot] = \Pr[\mathsf{Experiment1} : \cdot\,]$ be the probability function under $\mathsf{Experiment1}$ and $\Pr_2[\cdot] = \Pr[\mathsf{Experiment2} : \cdot\,]$ be that under $\mathsf{Experiment2}$. Let E_1,E_2, and E_3 be the following events:

$$
\begin{aligned}
E_1 &\stackrel{\text{def}}{=} \forall i : \ (b_i = 0) \vee (b_i = 1 \wedge z_i = u) \\
E_2 &\stackrel{\text{def}}{=} \exists i : \ (b_i = 1 \wedge z_i = v \wedge u \neq v) \\
E_3 &\stackrel{\text{def}}{=} \exists i : \ (b_i = 1 \wedge z_i \neq u \wedge z_i \neq v)
\end{aligned}
$$

For $j = 1, 2, 3$ let

$$
\begin{aligned}
p(1,j) &= \Pr_1[\, y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(x, \mathbf{x}) \mid E_j \,] - \Pr_1[\, y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(\tilde{x}, \mathbf{x}) \mid E_j \,] \\
p(2,j) &= \Pr_2[\, 0 \,\|\, y \notin \mathbf{z} \wedge \bot \notin \mathbf{w} \wedge R(x, \mathbf{w}) \mid E_j \,] - \Pr_2[\, 0 \,\|\, y \notin \mathbf{z} \wedge \bot \notin \mathbf{w} \wedge R(\tilde{x}, \mathbf{w}) \mid E_j \,]\ .
\end{aligned}
$$

By conditioning we have:

$$
\begin{aligned}
\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k) &= \left| \textstyle\sum_{j=1}^{3} p(1,j) \cdot \Pr_1[\, E_j \,] \right| \\
\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k) &= \left| \textstyle\sum_{j=1}^{3} p(2,j) \cdot \Pr_2[\, E_j \,] \right|\ .
\end{aligned}
$$

We now upper bound $\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k)$ in terms of $\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k)$ by a series of lemmas. The first observation is that the probability of our three events is the same in both experiments.

Lemma 1: $\Pr_1[\, E_j \,] = \Pr_2[\, E_j \,]$ for $j = 1, 2, 3$.

Proof: These events depend only on the keys and B. □

Let q be a polynomial which bounds the running time of B. In particular we can assume $|\mathbf{z}| < q(k)$.

Lemma 2: $p(2,1) \ \le\ p(1,1) + q(k) \cdot 2^{-k}$.

Proof: By event E_1 every $\mathbf{z}[i] = b_i \,\|\, z_i$ has either $(b_i = 0)$ or $(b_i = 1 \wedge z_i = u)$.

If $b_i = 0$ then A will output z_i in $\mathsf{Experiment1}$, while B would be outputting $0 \,\|\, z_i$ in $\mathsf{Experiment2}$. But $\mathcal{D}'_{sk\,\|\,u\,\|\,v}(0 \,\|\, z_i) = \mathcal{D}_{sk}(z_i)$, and furthermore $y = z_i$ (the challenge to A is equal to this component of A's output) iff $0 \,\|\, y = 0 \,\|\, z_i$ (the challenge to B is equal to this component of B's output). Thus A properly simulates B.

If $b_i = 1$ and $z_i = u$ then $\mathcal{D}'_{sk \,\|\, u \,\|\, v}(b_i \,\|\, z_i) = v$ is random and independent of the execution of B. To "simulate" it we have A output an encryption of random v. But, A will only be successful if the created ciphertext is different from y. The probability of this not happening can be upper bounded by the probability that $v = \mathcal{D}_{sk}(y)$, which is at most 2^{-k}. The worst case in this event is when $\forall i :\ (b_i = 1 \wedge z_i = u)$. Since $|\mathbf{z}| \le q(k)$, the probability, under this event, that A does not match the advantage of B, is at most $q(k) \cdot 2^{-k}$. □

Lemma 3: $\Pr_1[\, E_2 \,] \ \le\ q(k) \cdot 2^{-k}$.

Proof: B has no information about v since the latter was chosen independently of its execution, and also u has a 2^{-k} chance of equaling v. The Lemma follows since $|\mathbf{z}| < q(k)$. □

Lemma 4: $p(1,3) = p(2,3) = 0$.

Proof: When event E_3 happens in $\mathsf{Experiment1}$, one of the ciphertexts $\mathbf{y}[i]$ that A_2 outputs equals y and hence there is no contribution to the success probability. When event E_3 happens in $\mathsf{Experiment2}$, the definition of $\mathcal{D}'_{sk \,\|\, u \,\|\, v}$ says that the decryption of some $\mathbf{z}[i]$ is $\perp$ and hence again there is no contribution to the success probability. In other words, in both cases, there is no success in either the "real" or the "random" experiment. □

From Lemmas 1,2,3,4 we get

$$\begin{aligned}
\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k) &= \left| \textstyle\sum_{j=1}^{3} p(2,j) \cdot \Pr_1[\, E_j \,] \right| \\
&\le q(k) \cdot 2^{-k} + |\, p(1,1) \cdot \Pr_1[\, E_1 \,] + p(2,2) \cdot \Pr_1[\, E_2 \,] + p(1,3) \cdot \Pr_1[\, E_3 \,] \,| \\
&\le q(k) \cdot 2^{-k} + |\, p(1,1) \cdot \Pr_1[\, E_1 \,] + p(1,2) \cdot \Pr_1[\, E_2 \,] + p(1,3) \cdot \Pr_1[\, E_3 \,] \,| \\
&\qquad + |\, p(2,2) - p(1,2) \,| \cdot \Pr_1[\, E_2 \,] \\
&\le q(k) \cdot 2^{-k} + \left| \textstyle\sum_{j=1}^{3} p(1,j) \cdot \Pr_1[\, E_j \,] \right| + \Pr_1[\, E_2 \,] \\
&\le 2q(k) \cdot 2^{-k} + \mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k)\ .
\end{aligned}$$

The assumption that Π is secure in the sense of NM-CPA implies that $\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k)$ is negligible, and hence it follows that $\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k)$ is negligible. ▌

3.7 Proof of Theorem 3.7: NM-CCA1 $\not\Rightarrow$ NM-CCA2

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 3.6 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 [20]. (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 [23] which in turn implies the existence of a family of pseudorandom functions [22, 20].) 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: $\{0,1\}^k \to \{0,1\}^k$. We define the new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ as follows. Recall that ε is the empty string.

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

ANALYSIS. The proof of Theorem 3.7 is completed by establishing that Π' is vulnerable to a NM-CCA2 attack but remains NM-CCA1 secure.

Claim 3.14 *Π' is not secure in the sense of* NM-CCA2.

Proof: The idea is that while the adversary may not ask for the decryption of the challenge ciphertext $0\|y \,\|\, \varepsilon$ in its second stage, it may ask for the decryption of $1\|y\|F_K(y)$. This is in fact exactly the decryption of $0\|y \,\|\, \varepsilon$. The adversary first needs to compute $F_K(y)$ without access to K. This is easily done by calling the decryption oracle on $1\|y\|\varepsilon$.

More precisely, the adversary $A = (A_1, A_2)$ works like this. In the first stage it outputs a message space M consisting of two distinct strings x_0, x_1, each having probability $1/2$. A_2, given challenge ciphertext $0\|y \,\|\, \varepsilon$, makes query $1\|y\|\varepsilon$ to get $F_K(y)$, and outputs (R, Z) where $R(a,b) = 1$ iff $a = b$ is the equality relation, and $Z = 1 \,\|\, y \,\|\, F_K(y)$. Notice that $Z \neq 0\|y \,\|\, \varepsilon$ so this is a valid output, but $\mathcal{D}'_{sk \,\|\, K}(Z) = \mathcal{D}'_{sk \,\|\, K}(0\|y \,\|\, \varepsilon)$ so $\mathsf{Succ}^{\text{nm-cca2}}_{A,\Pi}(k) = 1$. On the other hand, $\mathsf{Succ}^{\text{nm-cca2}}_{A,\$,\Pi}(k) \leq 1/2$. So $\mathsf{Adv}^{\text{nm-cca2}}_{A,\Pi}(k) \geq 1/2$, which is certainly not negligible. ∎

Remember that Π is assumed secure in the sense of NM-CCA1. We will use this to establish the following:

Claim 3.15 *Π' is secure in the sense of* NM-CCA1.

Let us first give some intuition and then the proof. The key point 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.

Proof of Claim 3.15: To prove this claim we consider a polynomial time adversary B attacking Π' in the NM-CCA1 sense. We want to show that $\mathsf{Adv}^{\text{nm-cca1}}_{B,\Pi'}(\cdot)$ is negligible. To do this, we consider the following adversary $A = (A_1, A_2)$ attacking Π in the NM-CCA1 sense. The idea is that A can choose the key K for the key generation algorithm $\mathcal{K}'$ of B and thus provide a simulation of the decryption oracle of B.

```
Algorithm A_1^{D_sk}(pk)
    K ← {0,1}^k
    (M, s) ← B_1^{D'_{sk || K}}(pk)
    s' ← (s, K, pk)
    return (M, s')
```

```
Algorithm A_2(M, s', y) where s' = (s, K, pk)
    (R, z) ← B_2(M, s, 0 || y || ε)
    for 1 ≤ i ≤ |z| do parse z[i] as b_i || u_i || v_i where b_i is a bit
    for 1 ≤ i ≤ |z| do
        if (b_i = 0) ∧ (v_i = ε) then y[i] ← u_i
        else if (b_i = 1) ∧ (v_i = ε) then y[i] ← E_pk(F_K(u_i))
            else if (b_i = 1) ∧ (v_i = F_K(u_i)) then y[i] ← u_i
                else y[i] ← y
    return (R, y)
```

The analysis follows in spirit that in the proof of Claim 3.13; the key new element is the pseudorandom function. Roughly we seek to recapture the lemmas in that proof modulo the security of the pseudorandom function family.

For the proof, we define two experiments. The first is the one under which $\mathsf{Adv}^{\text{nm-cca1}}_{A,\Pi}(k)$ is evaluated, and the second is the one under which $\mathsf{Adv}^{\text{nm-cca1}}_{B,\Pi'}(k)$ is evaluated:

$$\begin{aligned}
\mathsf{Experiment1} \ &\stackrel{\text{def}}{=}\ (pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (M,(s,K,pk)) \leftarrow A_1^{\mathcal{D}_{sk}}(pk)\ ;\ x, \tilde{x} \leftarrow M\ ;\ y \leftarrow \mathcal{E}_{pk}(x)\ ; \\
&\qquad (R, \mathbf{y}) \leftarrow A_2(M,(s,K,pk),y)\ ;\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) \\
\mathsf{Experiment2} \ &\stackrel{\text{def}}{=}\ (pk, sk \,\|\, K) \leftarrow \mathcal{K}'(1^k)\ ;\ (M,s) \leftarrow B_1^{\mathcal{D}'_{sk \,\|\, K}}(pk)\ ;\ x, \tilde{x} \leftarrow M\ ; \\
&\qquad 0 \,\|\, y \,\|\, \varepsilon \leftarrow \mathcal{E}'_{pk \,\|\, u}(x)\ ;\ (R, \mathbf{z}) \leftarrow B_2(M, s, 0 \,\|\, y \,\|\, \varepsilon)\ ;\ \mathbf{w} \leftarrow \mathcal{D}'_{sk \,\|\, K}(\mathbf{z})\ .
\end{aligned}$$

Let $\Pr_1[\,\cdot\,] = \Pr[\mathsf{Experiment1} : \cdot\,]$ be the probability function under $\mathsf{Experiment1}$ and $\Pr_2[\,\cdot\,] = \Pr[\mathsf{Experiment2} : \cdot\,]$ be that under $\mathsf{Experiment2}$. Let E_1, E_2, and E_3 be the following events:

$$\begin{aligned}
E_1 &\stackrel{\text{def}}{=} \forall i : (v_i = \varepsilon) \vee (b_i = 1 \wedge v_i = F_K(u_i) \wedge u_i \neq y) \\
E_2 &\stackrel{\text{def}}{=} \exists i : (b_i = 1 \wedge v_i = F_K(u_i) \wedge u_i = y \wedge v_i \neq \varepsilon) \\
E_3 &\stackrel{\text{def}}{=} \exists i : (b_i = 1 \wedge v_i \neq F_K(u_i) \wedge v_i \neq \varepsilon) \vee (b_i = 0 \wedge v_i \neq \varepsilon)
\end{aligned}$$

For $j = 1, 2, 3$ let

$$\begin{aligned}
p(1,j) &= \Pr_1[\, y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(x,\mathbf{x}) \mid E_j \,] - \Pr_1[\, y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(\tilde{x},\mathbf{x}) \mid E_j \,] \\
p(2,j) &= \Pr_2[\, 0 \,\|\, y \,\|\, \varepsilon \notin \mathbf{z} \wedge \bot \notin \mathbf{w} \wedge R(x,\mathbf{w}) \mid E_j \,] - \Pr_2[\, 0 \,\|\, y \,\|\, \varepsilon \notin \mathbf{z} \wedge \bot \notin \mathbf{w} \wedge R(\tilde{x},\mathbf{w}) \mid E_j \,]\ .
\end{aligned}$$

By conditioning we have:

$$\begin{aligned}
\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k) &= \left| \textstyle\sum_{j=1}^{3} p(1,j) \cdot \Pr_1[\, E_j \,] \right| \\
\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k) &= \left| \textstyle\sum_{j=1}^{3} p(2,j) \cdot \Pr_2[\, E_j \,] \right|\ .
\end{aligned}$$

We now upper bound $\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k)$ in terms of $\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k)$ by a series of lemmas.

Lemma 1: $\Pr_1[\, E_j \,] = \Pr_2[\, E_j \,]$ for $j = 1, 2, 3$.

Proof: These events depend only on the keys and B. □

Let q be a polynomial which bounds the running time of B and in particular so that $|\mathbf{z}| < q(k)$.

Lemma 2: $p(2,1) \ \leq\ p(1,1) + \nu(k)$ for some negligible function ν depending on B.

Proof: We consider two possible cases for values of $\mathbf{z}[i] = b_i \,\|\, u_i \,\|\, v_i$, given event E_1.

First suppose $(b_i = 1 \wedge v_i = F_K(u_i) \wedge u_i \neq y)$. Note that $v_i = F_K(u_i)$ implies $v_i \neq \varepsilon$ since the output of F_K is always k bits long. Now, from the code of A_2, we see that in this case A_2 sets $\mathbf{y}[i]$ to u_i. Observe that if ciphertext $\mathbf{y}[i]$ (respectively $\mathbf{z}[i]$) that A (respectively B) creates equals y (respectively $0 \,\|\, y \,\|\, \varepsilon$) then there is no contribution to the success probability. Since $b_i = 1$ we know that $\mathbf{z}[i] \neq 0 \,\|\, y \,\|\, \varepsilon$. On the other hand the condition $u_i \neq y$ means that $\mathbf{y}[i] \neq y$ too. From the definition of $\mathcal{D}'$ we have $\mathcal{D}'_{sk \,\|\, K}(1 \,\|\, u_i \,\|\, F_K(u_i)) = \mathcal{D}_{sk}(u_i)$, so A is properly simulating B. (Meaning the contribution to their respective success probabilities is the same.)

For the second case, namely $v_i = \varepsilon$, we consider the two possible values of b_i.

If $b_i = 0$ then A will set $\mathbf{y}[i] = u_i$, and from the definition of $\mathcal{D}'$ we have $\mathcal{D}'_{sk \,\|\, K}(0 \,\|\, u_i \,\|\, \varepsilon) = \mathcal{D}_{sk}(u_i)$. Observe that A will output a ciphertext $\mathbf{y}[i]$ that equals y if and only if B outputs a ciphertext $\mathbf{z}[i]$ that equals $0 \,\|\, y \,\|\, \varepsilon$. So again A is properly simulating B.

If $b_i = 1$ then $\mathcal{D}'_{sk \,\|\, K}(1 \,\|\, u_i \,\|\, \varepsilon) = F_K(u_i)$ by definition of $\mathcal{D}'$. A correctly "simulates" this by outputting an encryption of $F_K(u_i)$. This choice of A contributes to the success probability as long as it is different from y. The probability of this not happening can be upper bounded by the probability that $\mathcal{E}_{pk}(F_K(u_i)) = y$. We must consider the worst case, which is when $\forall i : (b_i = 1 \wedge v_i = \varepsilon)$, so we are interested in bounding the probability that there is some i such that $\mathcal{E}_{pk}(F_K(u_i)) = y$. Intuitively, such "ciphertext collisions" are unlikely since otherwise the scheme would not be secure even in the sense of IND-CCA1. Formally, one can show that the probability of such collisions is at most $\nu(k)$, where $\nu(\cdot)$ is a negligible function depending on B, by showing that if not, we could design an adversary A' that would break the scheme in the sense of IND-CCA1. This is standard, and a sketch of the details follows.

In the first stage A' does what A does, picking a key K so that it can provide a simulation of the decryption oracle of B, similar to the simulation provided by A. It runs the first stage of B and picks a pair of messages uniformly from the message space output by B. In the second stage it is given an encryption of one of these messages as the challenge. It then obtains a polynomial number of encryptions of one of the messages and checks if any of the resulting ciphertexts match the challenge ciphertext. If it does then it bets that the challenge ciphertext corresponds to this message, otherwise it decides by flipping a coin. Observe that the success of A' is exactly one half the probability of there being some i such that $\mathcal{E}_{pk}(F_K(u_i)) = y$ since the experiments defining the success of A' and the upper bound on the probability in question are similar. Since Π is given to be secure in the NM-CCA1 sense (and therefore in the IND-CCA1 sense, see Theorem 3.1), we get a bound of $\nu(k)$ where ν is a negligible function depending on B. □

Notice that in the above we did not use the security of the pseudorandom function family. That comes up only in the next lemma. Accordingly, in the following, for any polynomial f we let $\delta_f(k)$ be a negligible function which upper bounds the advantage obtainable by any adversary in distinguishing F from a family of random functions when the running time of this adversary is at most $f(k)$.

Lemma 3: $\Pr_1[\, E_2 \,] \;\leq\; q(k) \cdot [\delta_q(k) + \nu(k)]$ for some negligible function ν that depends on B.

Proof: Event E_2 occurs if B outputs $1 \,\|\, u_i \,\|\, v_i$ where $u_i = y$ and $v_i = F_K(y)$. The claim is that this happens with only a small probability.

Note that it is not impossible for B to compute the value of F_K on a point, even though F is pseudorandom, because it can compute $F_K(m)$ on a point m of its choice simply by querying its decryption oracle on $1 \,\|\, m \,\|\, \varepsilon$. However, this oracle is only available in the first stage, and in that stage B does not know y. When she does get to know y (in the second stage) she no longer has the decryption oracle. The pseudorandomness of F then says her chance of computing $F_K(y)$ is small.

To turn this intuition into a formal proof, first imagine that we use, in the role of F_K, a random function g. (Imagine that $\mathcal{D}_{sk \,\|\, K}$ has oracle access to g and uses it in the role of F_K.) In the resulting scheme and experiment, it is clear that the chance that B computes $g(y)$ is at most 2^{-k} plus the chance that she made a query involving y to the decryption oracle in the first stage. Since y is a ciphertext created after the first stage, we claim that the chance that B could make a query involving y in her first stage is negligible. This is true because if not, we would contradict the fact that Π is IND-CCA1. (This can be argued analogously to the argument in the previous Lemma. We omit the details.)

Let $\nu(k)$ then be the negligible probability of computing $g(y)$. Now given that F is pseudorandom in nature we can bound the probability of B correctly computing $F_K(y)$ by $\delta_q(k) + \nu(k)$ for some polynomial q which depends on B. (Justified below.) So while B could always pick u_i to be y, she would have a negligible probability of setting v_i to be $F_K(y)$. In the worst case this event could happen with probability at most $|\mathbf{z}| \cdot [\delta_q(k) + \nu(k)]$.

The bound of $\delta_q(k) + \nu(k)$ mentioned above is justified using the assumed security of F as a pseudorandom function family. If the event in question had a higher probability, we would be able to construct a distinguisher between F and the family of random functions. This distinguisher would get an oracle g for some function and has to tell whether g is from F^k or is a random function of k bits to k bits. It would itself pick the secret keys underlying $\mathsf{Experiment1}$ or $\mathsf{Experiment2}$ and run the adversaries A or B. It can test whether or not the event happens because it knows all decryption keys. If it happens it bets that g is pseudorandom, because the chance under a random function is at most $2^{-k} + \nu(k)$. Since this kind of argument is standard, we omit the details. □

Lemma 4: $p(1,3) = p(2,3) = 0$.

Proof: When event E_3 happens in $\mathsf{Experiment1}$, one of the ciphertexts $\mathbf{y}[i]$ that A_2 outputs equals y and hence there is no contribution to the success probability. When event E_3 happens in $\mathsf{Experiment2}$, the definition of $\mathcal{D}'_{sk \,\|\, K}$ says that the decryption of some $\mathbf{z}[i]$ is $\perp$ and hence again there is no contribution to the success probability. In other words, in both cases, there is no success in either the "real" or the "random" experiment. □

From Lemmas 1,2,3,4 we get

$$
\begin{aligned}
\mathsf{Adv}^{\text{nm-cca1}}_{B,\Pi'}(k) &= \left| \textstyle\sum_{j=1}^{3} p(2,j) \cdot \Pr_1[\,E_j\,] \right| \\
&\le \nu(k) \;+\; |\, p(1,1) \cdot \Pr_1[\,E_1\,] \;+\; p(2,2) \cdot \Pr_1[\,E_2\,] \;+\; p(1,3) \cdot \Pr_1[\,E_3\,]\,| \\
&\le \nu(k) \;+\; |\, p(1,1) \cdot \Pr_1[\,E_1\,] \;+\; p(1,2) \cdot \Pr_1[\,E_2\,] \;+\; p(1,3) \cdot \Pr_1[\,E_3\,]\,| \\
&\qquad + \;|\, p(2,2) - p(1,2)\,| \cdot \Pr_1[\,E_2\,] \\
&\le \nu(k) \;+\; \left| \textstyle\sum_{j=1}^{3} p(1,j) \cdot \Pr_1[\,E_j\,] \right| \;+\; \Pr_1[\,E_2\,] \\
&\le \nu(k) + q(k) \cdot [\delta_q(k) + \nu(k)] \;+\; \mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k)\;.
\end{aligned}
$$

Since $\delta_q(k)$ and $\nu(k)$ are negligible quantities, the assumption that Π is secure in the sense of NM-CCA1 implies that $\mathsf{Adv}^{\text{nm-cca1}}_{A,\Pi}(\cdot)$ is negligible, and hence it follows that $\mathsf{Adv}^{\text{nm-cca1}}_{B,\Pi'}(\cdot)$ is negligible. ▮

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 2.1 and 2.2, begin the specified experiment (the experiment which defines advantage) by choosing a random function H from the set of all functions from some appropriate domain to appropriate range. (These sets might change from scheme to scheme.) 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 [6], 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. 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 $\mathcal{E}_{pk}^H$. 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 4.1 [**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 $k \in \mathsf{N}$ define

$$\begin{aligned}\mathsf{Succ}^{\mathrm{pa}}_{K,B,\Pi}(k) \;\stackrel{\mathrm{def}}{=}\; \Pr\Big[&H \leftarrow \mathsf{Hash}\ ;\ (pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (hH, C, y) \leftarrow \mathsf{run}\, B^{H,\mathcal{E}_{pk}^H}(pk) :\\ &K(hH, C, y, pk) = \mathcal{D}_{sk}^H(y)\Big]\ .\end{aligned}$$

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, $\mathsf{Succ}^{\mathrm{pa}}_{K,B,\Pi}(k) \geq \lambda(k)$. We say that Π is secure in the sense of PA if Π 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 the oracle for $\mathcal{E}_{pk}^H$. At first glance this may seem redundant: since B has the public key, can it not encrypt on its own? It can. But, in the random-oracle model, encrypting such points oneself involves making H-queries (remember that $\mathcal{E}_{pk}^H$ itself makes H queries), meaning B knows the oracle queries used by $\mathcal{E}_{pk}^H$ 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, where B's state of knowledge does not include the underlying oracle queries. By giving B an encryption oracle $\mathcal{E}_{pk}^H$ whose H-queries (if any) 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 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 [12] of the plaintext. But this is not valid. The reason is the way the extractor operates in the notion and scheme of [12]: 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 4.2 [PA $\Rightarrow$ IND-CCA2] *If encryption scheme* Π *is secure in the sense of* PA *then it is secure in the RO sense of* IND-CCA2.

Corollary 4.3 [PA $\Rightarrow$ NM-CCA2] *If encryption scheme* Π *is secure in the sense of* PA *then* Π *is secure in the RO sense of* NM-CCA2.

Proof: Follows from Theorems 4.2 and the RO-version of Theorem 3.3. ▌

The above results say that PA $\Rightarrow$ IND-CCA2 $\Rightarrow$ NM-CCA2. In the other direction, we have the following, whose proof is in Section 4.4.

Theorem 4.4 [IND-CCA2$\not\Rightarrow$PA] *If there exists an encryption scheme* Π *which is secure in the RO sense of* IND-CCA2*, then there exists an encryption scheme* Π' *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 4.2: PA $\Rightarrow$ IND-CCA2

Intuition. The basic idea for proving chosen-ciphertext security in the presence of some kind of proof of knowledge goes back to [16, 17, 9, 12]. Let us begin by recalling it. Assume there is some adversary $A = (A_1, A_2)$ that breaks Π in the IND-CCA2 sense. We construct an adversary $A' = (A'_1, A'_2)$ that breaks Π 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 4.1. 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 Bs 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 Π. 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 Π. Think of A' as the adversary A with access only to a simulated decryption oracle rather than the real thing. If $A(\cdot,\cdot,\cdots)$ is any probabilistic algorithm then $A(x,y,\cdots;R)$ means we run it with coin tosses fixed to R. Let ε denote the empty list. The adversary is defined as follows:

`Algorithm` $A'_1(pk;R)$	`Algorithm` $A'_2(x_0,x_1,(s,hH,pk),y;R)$
$hH \leftarrow \varepsilon$	
Take R_1 from R	Take R_2 from R
Run $A_1(pk;R_1)$, wherein	Run $A_2(x_0,x_1,s,y;R_2)$, wherein
When A_1 makes a query, h, to H:	When A_2 makes a query, h, to H:
A'_1 asks *its* H-oracle h, obtaining $H(h)$	A'_2 asks *its* H-oracle h, obtaining $H(h)$
Put $(h,H(h))$ at end of hH	Put $(h,H(h))$ at end of hH
Answer A_1 with $H(h)$	Answer A_2 with $H(h)$
When A_1 makes its jth query, y, to $\mathcal{D}^H_{sk}$:	When A_2 makes its jth query, y', to $\mathcal{D}^H_{sk}$:
$x \leftarrow K(hH,\varepsilon,y,pk)$	$x \leftarrow K(hH,(y),y',pk)$
Answer A_1 with x	Answer A_2 with x
Finally A_1 halts, outputting (x_0,x_1,s)	Finally A_2 halts, outputting bit, d
`return` $(x_0,x_1,(s,hH,pk))$	`return` d

Analysis. To reason about the behavior of A' we describe adversaries $B_1, \ldots, B_q$, where q is the number of decryption queries made by A.

Adversary B_1 runs $A)1$, answeriing A_1's H-oracle queries using its own H-oracle, being careful to collect up the questions and their answers, forming a list of these, hH. When A_1 finally makes its first decryption query, y_1, algorithm B_1 halts, outputting y_1.

Algorithm B_2 likewise runs A_1. As before, H-queries (and their answers) are recorded in hH. When the first query y_1 to $\mathcal{D}^H_{sk}$ is made, B_2 passes y_1 to K along with the transcript hH and pk. Since A_1 does not have access to an encryption oracle, the ciphertext list C that K expects will be empty ($C = \varepsilon$). Algorithm B_2 then passes on K's answer to A_1 and continues running A_1, appropriately updating hH, until the second query, y_2, is made to $\mathcal{D}^H_{sk}$. Then B_2 outputs y_2.

This process continues in this way to construct each B_i for $i \in \{1, \ldots, q_1\}$, where q_1 is the number of $\mathcal{D}^H_{sk}$-queries made by A_1. This is described by the left-hand column below.

Algorithm $B_i^{H,\mathcal{E}_{pk}^H}(pk;R)$ // $i \in \{1,\ldots,q\}$
  $hH \leftarrow \varepsilon$
  Let R_1, R_2 be taken from R.
  Run $A_1(pk;R_1)$, wherein
    When A_1 makes a query, h, to H:
      B_i 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 jth query, y, to $\mathcal{D}_{sk}^H$:
      if $j = i$ then return y and halt
      else $x \leftarrow K(hH, \varepsilon, y, pk)$
        Answer A_1 with x
  Finally, A_1 halts, outputting (x_0, x_1, s)

// *Algorithm B_i, continued*
$d \leftarrow \{0,1\}$
Using B_i's encryption oracle, let $y \leftarrow \mathcal{E}_{pk}^H(x_d)$
Run $A_2(x_0, x_1, s, y; R_2)$, wherein
  When A_2 makes a query, h, to H:
    B_i 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 $\mathcal{D}_{sk}^H$:
    if $i = j + q_1$ then return y' and halt
    else $x \leftarrow K(hH, (y), y', pk)$
      Answer A_2 with x

Having defined adversaries corresponding to each decryption query made by A_1, we now need to do this for A_2. Recall that adversary A_2 gets as input (x_0, x_1, s, y) where, in the experiment defining advantage, y is selected according to $y \leftarrow \mathcal{E}_{pk}^H(x_d)$ for a random bit d. Remember that A_2 is prohibited from asking $\mathcal{D}_{sk}^H(y)$, although A_2 may make other (possibly related) decryption queries. How then can we pass y to our decryption simulation mechanism? This is where the encryption oracle and the ciphertext list C come in. We define adversaries $B_{q_1+1}, \ldots, B_q$ just like we defined $B_1, \ldots, B_{q_1}$, except that this time $C = (y)$ rather than being empty. This is shown above in the righ-hand column.

Let us now see how good a simulation A_1' is for $A_1^{\mathcal{D}_{sk}^H}$. Note that the values (x_0, x_1, s) produced by A_1' are not necessarily the same as what A_1 would have output after the analagous interactions with $\mathcal{D}_{sk}^H$, since one of K's answers may not be the correct plaintext. Let D be the event that at least one of K's answers to A_1's decryption queries was not the correct plaintext. 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' by

$$\Pr[A_1'(pk) = A_1^{\mathcal{D}_{sk}^H}(pk)] \geq 1 - \Pr[\mathsf{D}] \geq 1 - q_1 \cdot (1 - \lambda(k)) .$$

Letting q_2 be the number of decryption oracle queries made by A_2, we similarly have for A_2' that and that

$$\Pr[A_2'(x_0, x_1, (s, hH), y) = A_2^{\mathcal{D}_{sk}^H}(x_0, x_1, s, y) \mid A_1'(pk) = A_1^{\mathcal{D}_{sk}^H}(pk)] \geq 1 - q_2 \cdot (1 - \lambda(k)) .$$

Now using the above, one can see that

$$\mathsf{Adv}_{A',\Pi}^{\text{ind-cpa}}(k) \quad \geq \quad \mathsf{Adv}_{A,\Pi}^{\text{ind-cca2}}(k) - 2q \cdot (1 - \lambda(k)),$$

where $q = q_1 + q_2$ and represents the total number of decryption oracle queries made by the adversary A. A_1' runs A_1, asking for q_1 executions of K. Similarly A_2' runs A_2, asking for q_2 executions of K. Hence the running time of our new adversary A' is equal to $t_A + q \cdot t_K$, where t_A and t_K are the running times of A and K respectively, which is polynomial if A and K are polynomial time. Under our assumptions $\mathsf{Adv}_{A,\Pi}^{\text{ind-cca2}}(k)$ is non-negligible and $1 - \lambda(k)$ is negligible, so $\mathsf{Adv}_{A',\Pi}^{\text{ind-cpa}}(k)$ is non-negligible, and Π is not secure in the sense of IND-CPA security.

In concrete security terms, the advantage drops linearly in q while the running time grows linearly in q. Note that it was important in the proof that K almost always succeeded; it would not have worked with $\lambda(k) = 0.5$, say.

4.4 Proof of Theorem 4.4: IND-CCA2$\not\Rightarrow$PA

Assume there exists some IND-CCA2 secure encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, since otherwise the theorem is vacuously true. We now modify Π to a new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ which is also IND-CCA2 secure but not secure in the PA sense. This will prove the theorem. The new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ is defined as follows:

`Algorithm` $\mathcal{K}'(1^k)$	`Algorithm` $\mathcal{E}'^{H}_{pk \,\|\, a}(x)$	`Algorithm` $\mathcal{D}'^{H}_{sk \,\|\, b}(y)$
$(pk, sk) \leftarrow \mathcal{K}(1^k)$	`return` $\mathcal{E}^{H}_{pk}(x)$	`return` $\mathcal{D}^{H}_{sk}(y)$
$b \leftarrow \{0,1\}^k$; $a \leftarrow \mathcal{E}^{H}_{pk}(b)$		
$pk' \leftarrow pk \,\|\, a$; $sk' \leftarrow sk \,\|\, b$		
`return` (pk', sk')		

In other words, the only difference is that in the new scheme, the public key contains a random ciphertext a whose decryption is in the secret key. Our two claims are that Π' remains IND-CCA2 secure, but is not PA. This will complete the proof.

Claim 4.5 *Π' is secure in the sense of* IND-CCA2.

Proof: Recall our assumption is that Π is IND-CCA2 secure. To prove the claim we consider a polynomial time adversary B attacking Π' in the IND-CCA2 sense. We want to show that $\mathsf{Adv}^{\text{ind-cca2}}_{B,\Pi'}(\cdot)$ is negligible. To do this, we consider the following adversary $A = (A_1, A_2)$ attacking Π in the IND-CCA2 sense. The idea is that A can simulate the choosing of a by the key generation algorithm $\mathcal{K}'$ for B, and thus has access to the corresponding secret b. Note that having an oracle for $\mathcal{D}^{H}_{sk}$, it is indeed possible for A to reply to any queries to the $\mathcal{D}'^{H}_{sk \,\|\, b}$ oracle made by B: to query y it simply returns $\mathcal{D}^{H}_{sk}(y)$.

`Algorithm` $A_1^{\mathcal{D}^{H}_{sk}}(pk)$	`Algorithm` $A_2^{\mathcal{D}_{sk}}(x_0, x_1, s', y)$ where $s' = (s, a, b)$
$b \leftarrow \{0,1\}^k$; $a \leftarrow \mathcal{E}^{H}_{pk}(b)$	$pk' \leftarrow pk \,\|\, a$
$pk' \leftarrow pk \,\|\, a$	$d \leftarrow B_2^{\mathcal{D}'_{sk \,\|\, b}}(x_0, x_1, s, y)$
$(x_0, x_1, s) \leftarrow B_1^{\mathcal{D}'^{H}_{sk \,\|\, b}}(pk \,\|\, a)$	`return` d
$s' \leftarrow (s, a, b)$	
`return` (x_0, x_1, s')	

It is clear that A is polynomial time and that $\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) = \mathsf{Adv}^{\text{ind-cca2}}_{B,\Pi'}(k)$. The assumption that Π is secure in the sense of IND-CCA2 implies that $\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k)$ is negligible, and hence it follows that $\mathsf{Adv}^{\text{ind-cca2}}_{B,\Pi'}(k)$ is negligible. ▌

Claim 4.6 *Π' is not plaintext-aware.*

Proof: We consider the following specific adversary B that outputs as her ciphertext the value a in her public key:

`Algorithm` $B^{H, \mathcal{E}^{H}_{pk'}}(pk')$ where $pk' = pk \,\|\, a$
  `return` a

Intuitively, this adversary defeats any aspiring plaintext extractor: It will not be possible to construct a plaintext extractor for this B as long as Π' is secure in the sense of IND-CPA. Hence there does not exist a plaintext extractor for Π'.

The formal proof is by contradiction. Assume Π' is PA. Then there exists a plaintext-extractor K' for Π'. We now define an adversary $A = (A_1, A_2)$ that attacks Π in the sense of IND-CPA. the empty list.

Algorithm $A_1(pk)$	Algorithm $A_2(x_0, x_1, pk, y)$
$x_0 \leftarrow \{0,1\}^k$	$pk' \leftarrow (pk, y)$
$x_1 \leftarrow \{0,1\}^k$	$x' \leftarrow K'(\varepsilon, \varepsilon, y, pk')$
return (x_0, x_1, pk)	if $x' = x_0$ then $d \leftarrow 0$
	else if $x' = x_1$ then $d \leftarrow 1$
	else $d \leftarrow \{0,1\}$
	return d

Consider the experiment defining the success of (A_1, A_2) in attacking Π in the sense of IND-CPA. In this experiment, y is the encryption of a random k-bit string. This means that in the input $(\varepsilon, \varepsilon, y, pk')$ given to K, the distribution of $(\varepsilon, \varepsilon, y)$ is exactly that of $\mathsf{run}\, B^{\mathcal{E}_{pk'}}(pk')$. This is because B, the adversary we defined above, has no interaction with its oracles, and the value a in the public key pk' is itself the encryption of a random k-bit string. Thus, our assumption that K' works means that the extraction is successful with probability $\mathsf{Succ}^{\mathrm{pa}}_{K',B,\Pi'}(k)$. Thus

$$\mathsf{Adv}^{\text{ind-cpa}}_{A,\Pi}(k) \;\geq\; \mathsf{Succ}^{\mathrm{pa}}_{K',B,\Pi'}(k) - \frac{1}{2^k} - \frac{1 - \mathsf{Succ}^{\mathrm{pa}}_{K',B,\Pi'}(k)}{2}\;.$$

The first term is a lower bound on the probability that A_2 outputs 0 when the message was x_0. The second term is an upper bound on the probability that it outputs 1 when the message was x_0. Now since K' is assumed to be a good extractor we know that $\mathsf{Succ}^{\mathrm{pa}}_{K',B,\Pi'}(k) = 1 - \lambda(k)$ for some negligible function $\lambda(\cdot)$ and hence $\mathsf{Adv}^{\text{ind-cpa}}_{A,\Pi}(k)$ is not negligible. (In fact is of the form $1 - \lambda'(k)$ for some negligible function $\lambda'(\cdot)$.) This contradicts the indistinguishability of Π, as desired. ▮

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 [25]. We are indebted to him for this suggestion. We also 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.

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, E. Jokipii and P. Rogaway, A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation. *Proceedings of the* 38th *Symposium on Foundations of Computer Science*, IEEE, 1997.

[3] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, Relations among notions of security for public-key encryption schemes. Preliminary version of this paper. *Advances in Cryptology — Crypto '98 Proceedings*, Lecture Notes in Computer Science, H. Krawczyk, ed., Springer-Verlag 1998.

[4] M. Bellare, R. Impagliazzo and M. Naor, Does parallel repetition lower the error in computationally sound protocols? *Proceedings of the* 38th *Symposium on Foundations of Computer Science*, IEEE, 1997.

[5] 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.

[6] 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.

[7] M. Bellare and A. Sahai, Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. Manuscript, February 1999.

[8] 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 Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography. *Proceedings of the* 23rd *Annual Symposium on Theory of Computing*, ACM, 1991.

[14] D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography. *Technical Report CS95-27, Weizmann Institute of Science*, 1995.

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

[16] 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.

[17] 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.

[18] O. Goldreich, Foundations of cryptography. Class notes, Spring 1989, Technion University.

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

[20] O. Goldreich, S. Goldwasser and S. Micali, How to construct random functions. *Journal of the ACM,* Vol. 33, No. 4, 1986, pp. 210–217.

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

[22] J. Håstad, R. Impagliazzo, L. Levin and M. Luby, Construction of a pseudo-random generator from any one-way function. Manuscript. Earlier versions in STOC 89 and STOC 90.

[23] 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.

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

[25] M. Naor, private communication, March 1998.

[26] 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] 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.

[28] 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/`

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

[30] 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).