On the Definitions of Cryptographic Security: Chosen-Ciphertext Attack Revisited

Maxwell Krohn (advised by Michael O. Rabin and Michael Mitzenmacher)

April 5, 1999

Contents

1 Introduction

The Internet has brought the heretofore theoretical and obscure field of cryptography to the fore. Although computer scientists have contemplated applications such as electronic voting and electronic commerce for years, the sheer number of users connected to the global computer network lends these academic scenarios the potential to become reality. But as the Internet enables more applications for increasingly many honest users, it attracts increasingly many malicious and exploitative criminals. The global computer network, by design, is entirely insecure. Any party can intercept any message sent over the wires. Nobody's identity is known, and any machine might attempt to impersonate any other. As the number of applications for the Internet increases, so do the feasibility and profitability of Internet fraud.

Electronic voting is just one potential application for cryptography that could have profound importance. If people no longer had to leave work to vote, voter turnout might improve, and our democracy might more closely approximate the ideal. From a computer scientist's perspective, a problem such as electronic voting challenges the most advanced cryptographic constructions. To implement electronic voting, we would need to preserve anonymity, authenticity, privacy, autonomy, accuracy and so on. Voters would insist that no one, including government officials, could prove for whom they voted. The candidates would insist that the vote counts are accurate and that votes are not corrupted in transit. The public would insist that forgeries would be impossible, that no candidate could stuff the ballot box, and that no registered voter could vote more than once.

Before computer scientists can approach a problem so complex as electronic voting, they must first address more primitive cryptographic problems. One such problem, which this thesis will focus on, is privacy. Consider the example in which Cassius e-mailed Brutus over the Internet, "We assassinate Caesar on the ides of March." Call this message, for convenience, M. Computer networks are such that any associate in Cassius's office, any engineers performing maintenance to the computers that connect Cassius and Brutus, or any associate in Brutus's office could have recovered the complete message M. If one of these people happened to have been Caesar or a Caesar-sympathizer, the coup would have been summarily exposed and crushed. Rome might never have been freed of its tyrannical dictator.

Cryptographers take for granted the fact that Caesar can monitor any communications between Cassius and Brutus. Instead of controlling Caesar's physical access to the network, the cryptographer would instruct Cassius to encode his message M into another message N. Even if Caesar takes N off the network, he should not be able to learn anything about M. That is, to Caesar, N should look like gibberish, or random noise. Furthermore, Brutus should still be able to recover M, given N. This process of encoding and decoding messages in such a way that obscures information from malicious adversaries is called encryption and decryption, respectively. If we value our privacy and the secrecy of our information, we should encrypt any information that we send over the Internet.

Although schemes that encrypt and decrypt have been known for years, many with wartime applications, the modern model of cryptography is relatively young. Many questions persist, such as: When does an adversary (such as Caesar in the previous example) succeed?

When she uncovers the entire message M?[1] When she uncovers a significant segment of the message M? Or maybe when she figures out how to decode any future encryption that Cassius might send to Brutus? Furthermore, how much power does the adversary have? Does she only have the capability of eavesdropping? Or maybe she can sneak into Brutus's office during lunchtime, gain temporary access to his decryption scheme, and use this information later to decode N? Cryptographers must provide a formal and *mathematical* system of definitions to model such real-world events. Which events to model, and how to model them are questions whose answers are, even today, in flux. Needless to say, such definitions are central to analyzing the simple case in which Cassius desires to send a private message to Brutus, as well as more complicated scenarios such as electronic voting. Their mathematical accuracy and correspondence to real world situations are the foundations upon which all of cryptography is based.

This thesis's main focus is indeed cryptographic definitions. Bellare, Rogaway, DeSai and Pointcheval recently published a paper [2] whose goal was to coherently define and relate the different definition of cryptographic "security." This thesis offers many refinements to that paper's formal framework, including new notions of security, and important modifications to existing definitions.

1.1 Private and Public Key Cryptography

Most early methods of cryptography developed for military purposes are examples of a specific type of cryptography known as *secret key cryptography*, *private key cryptography* or *symmetric key cryptography*. Examples of such systems include the Enigma Code that Germany used in World War II, or the "secret decoder rings" included in cereal boxes. That is, if two parties Alice and Bob wish to communicate using a private key cryptosystem, they first have to share some secret information, called the *secret key*. They will use this same key for the purposes of encryption and decryption. Such a model makes sense if the two communicating parties have already established a trusted channel and are in frequent communication over an insecure channel. For instance, if a general wants to send secret messages to his corporal, they might have a meeting in which they share secret information (a face-to-face meeting is considered a secure channel). They can then use secret key cryptography to broadcast messages over the radio waves (an insecure channel), between the battlefield and the general's air-conditioned office. By encrypting messages, the two officers obscure their information from an enemy monitoring radio signals.

Such schemes are in widespread use today, the best known of which is called DES (Digital Encryption Standard). In general, secret key is the fastest and most computationally efficient form of encryption. Many such schemes, however, follow a classical and therefore less-rigorous notion of security. We accept the security of DES, for instance, only insofar as no efficient method of attacking DES currently exists.[2] However, there is no guarantee that a malicious adversary will not discover a means by which DES-encrypted data can

[1] It has become a convention in cryptographic papers to refer to adversaries with feminine pronouns. Although the adversary being considered here is "Caesar," all generic adversaries are assumed to be feminine.

[2] It is important to say that no *efficient* scheme exists because intractable ways of breaking cryptosystems always exist. In the case of DES, one might exhaustively try all possible 2^{56} secret keys, to see if any yields a valid decryption. However, no existing computer could carry out 2^{56} guesses in a reasonable amount of time, so this brute-force attack becomes intractable. Similarly, all cryptosystems should guarantee that the only possible attack against it is the exhaustive attack, and that such an attack takes a number of computations exponential in a sufficiently high security parameter (56 in the case of DES).

be quickly recovered. Moreover, secret key cryptography is inherently impractical given today's computer network models. Ideally, any two parties on the Internet should be able to share secret data, whether or not they have established a shared, secret key through other channels. For example, a consumer should be able to send his credit card number secretly to an online store, whether or not the store and the consumer have established a secret account number via telephone or regular mail. Indeed, consumers now expect to make secure transactions over the Internet through a series of clicks and keystrokes, without having to pick up the telephone.

In response to the inherent limitations of secret key cryptography, Diffie and Hellman in [9] propose a new paradigm: *public key cryptography*, or *asymmetric key cryptography*. Their idea is to have a pair of algorithms, encryption and decryption as usual, and a pair of keys: a public key and a secret key. Users widely publish their public key but keep their secret key private. To encrypt a message for Bob, one looks up Bob's public key in a public database and encrypts with a widely-published encryption algorithm, with Bob's public key and the message to be encrypted as inputs. The resulting message is decryptable by someone who has access to the secret key that corresponds to Bob's public key, namely Bob, and Bob alone. In the previous example, a consumer now has a convenient way of establishing a secure line of communication with an online store. The consumer can look up the store's public encryption key in a public database or can ask the store to send its public encryption key over the Internet. The consumer can then produce messages (more specifically, encryptions of credit card information) that the online store, but no one else on the Internet, can understand. Not long after Diffie and Hellman proposed the idea of public key encryption, several realizations emerged. Named after their inventors, the schemes include RSA [25], Rabin [22] and El Gamal [12].

Many modern schemes, such as Rabin and El Gamal, are furthermore "provably secure." The problem of "provable security" takes its cue from the more classical field of complexity theory. Complexity theory is a theoretical approach to classifying types of problems that computers and computer scientists face. For example, if a traveling salesman wants to visit all of the major cities in the United States, he might ask a computer to plot out his itinerary. He might ask the computer to simply output a route so that he visits each city at least once, or he might ask the computer to output the *shortest* route through all of the cities. A complexity theorist would argue that the two problems are fundamentally different. With current computational methods, the former problem might be solvable in under a second while the latter might be solvable in no less than 3 years. The former problem belongs to a class of problems known as $\mathcal{P}$ while the latter belongs to a class of problems known as $\mathcal{NP}$. Although such distinctions have existed for years, computer scientists are still unable to prove whether $\mathcal{P}$ and $\mathcal{NP}$ are really one in the same class of problems, or if there are some problems in the latter that are provably not in the former.

The basic idea of cryptography is to find a function that is fast to compute but nearly impossible to invert. In our above example, Cassius and Brutus can both compute a function that turns M into N or N into M, because they have certain secret information. Caesar, on the other hand, should not be able to deduce their secret information, or to decode efficiently M from N without the secret information. Caesar is attempting to invert the function, while Brutus and Cassius are merely computing it in the forward direction. Another example is the Gordian Knot. Someone was able to construct this knot without much trouble, yet nobody could find a way to undo it. Such a function — which is easy to compute but

hard to invert — is known as a "one-way function." Indeed, many papers (such as [18] and [17]) have shown that if one-way functions exist, then most goals in cryptography can be achieved. Conversely, in order for most goals in cryptography to be achieved, one-way functions must exist. However, one-way functions are only possible if $\mathcal{P} \neq \mathcal{NP}$. And even then, they still might not exist [14].

So how can any notion of "provable security" exist given all of this uncertainty in the field of complexity theory? Cryptographers have temporarily overcome the problem by basing their cryptosystems on that which is widely known to be unknown. Consider, for example, the problem of factoring integers. Mathematicians for years have struggled to find a practical method of factoring large numbers. The problem has attracted the attention of Gauss, Euler, Fermat, and many other great mathematicians, yet still no good solution exists. Thus, cryptographers have designed cryptosystems that are as hard to break as the factorization problem is to solve. With the Rabin cryptosystem, for instance, if one were able to fully decrypt messages without knowing the secret key, then one would be able to factor products of two large prime numbers. That is, breaking the Rabin scheme would be doing what thousands of mathematicians over hundreds of years could not. Similarly, systems such as El Gamal are tightly based on other difficult problems in mathematics. "Provably security," though somewhat of a misnomer, provides mathematical justification of a cryptosystem's security. Such reductions enable a more convincing notion of security than that of classical cryptography and are now considered standard by cryptographers.

1.2 Notions of Security

Consider a cryptosystem like RSA, by far the most popular means of public key encryption in use today. RSA underlies such systems as PGP (Pretty Good Privacy), which adds security features to e-mail messaging, and SET (Secure Electronic Transaction), which allows consumers to send their credit card numbers over the Internet with some sense of security. Unlike Rabin or El Gamal, other longstanding problems in mathematics are not reducible to RSA. Although it is based on factoring (if one could factor, one could break RSA), it is not tightly bound to factoring (if one could break RSA, one might still not be able to factor). But in its 22 years of existence, no one has uncovered an efficient means of "breaking" RSA. That is, if an adversary has access to an arbitrary sequence of messages that have been encrypted with RSA, no scheme presently exists whereby she might deduce the secret key that is used to decrypt those messages, or the decryptions of those messages. Thus RSA has come to be known as a difficult problem in mathematics, in and of itself. So by all accounts, RSA is secure, correct?

Not at all. The sense of security implicitly considered above turns out to be one of the easiest senses of security to achieve. One type of attack that the above definition does not preclude is that which reveals only partial information about an encrypted message. If an adversary intercepts a message y that has been encrypted with RSA, public key (e, n), she might use simple results of number theory to reveal information about one of the bits of decryption of y. In another type of attack, assume that Alice is sending Bob the encryption y of the number B, which turns out to be Alice's bid for a property at an auction in which all bids are private. An adversary Chloe might not be able to compute B given y, but she might instead send Bob the message $2^e \cdot y \pmod{n}$, where (e, n) is the public key. RSA has the property that the decryption of Chloe's message will be equal to $2 \cdot B$. Thus, Chloe will always be able to outbid Alice, even without knowing what Alice was bidding. Another

attack against RSA was recently discovered by Bleichenbacher [5]. Given a message y, intended for Alice, encrypted using an RSA-based scheme called PCKS#1, Bleichenbacher creates a series of encrypted messages distinct from y but based on y; call them $z_1, z_2, ...z_n$. Bleichenbacher then asks Alice for a small amount of information concerning $z_1, z_2, ...z_n$, so that Alice would never suspect that Bleichenbacher is hatching an evil plan. However, Bleichenbacher might then be able to compute the decryption of y in its entirety.

The above discussion of RSA motivates a very careful and formal treatment of cryptographic security. Indeed, we must do better than the most intuitive definition of security and consider many different adversaries, in many different scenarios, who wish to achieve varying degrees of success in their attacks.

1.3 Previous Work

This paper inherits much from the recent publication of Bellare, Desai, Pointcheval and Rogaway [2]. In that paper, the various kinds of cryptographic attacks are formalized and rigorously related. One novel approach of their paper is to treat the "strength" and "goals" of cryptographic attacks independently, or in their words, "orthogonally." To demonstrate what we mean by "strength" of attack, we return to the lunchtime attack discussed earlier. If Alice is sending encrypted messages to Bob, and Chloe is a malicious hacker who would like to intercept and decode these messages, we rate the strength of Chloe's attack on the basis of how much access she has to Bob's decryption equipment. Completely independent of when or how long she has access to this equipment, Chloe might want to recover just one bit from each of Alice's message, she might want to recover all of Alice's messages in their entirety, or she might want to forge messages related to Alice's, as in the case of the online auction. What Chloe hopes to achieve from her attack we intuitively call the "goal" of her attack.

This thesis will examine the traditional "strengths" of attack, while formalizing and analyzing new strengths of attack. The two "goals" this thesis considers are *distinguishability* and *malleability*. If Chloe wants to *distinguish* encrypted messages, then she must recover information about one bit of the corresponding decryption. If Chloe's goal is *malleability*, then she desires to forge related messages as demonstrated before.

Furthermore, this thesis also calls upon a relatively new concept in cryptography: *plaintext awareness*. In plaintext aware encryption systems, the decryptor, Bob, might not accept all encryptions from Alice. He might deem that Alice has forged certain encryptions, as opposed to encrypting them with her encryption engine. In case of forgery, Bob will refuse to output any decryption. Intuitively, a cryptosystem is plaintext aware if in order for Alice to produce encryptions that Bob will decrypt, Alice must use the encryption scheme that corresponds to Bob's decryption scheme. In the end, whatever plaintext Bob outputs, Alice must be able to compute on her own, without the benefit of Bob's decryption equipment.

Plaintext awareness, according to Bellare, Desai, Pointcheval and Rogaway, turns out to be one of the strongest formulations of security. If a scheme can be shown to be plaintext aware, then it also meets the strongest criteria for security; that is, it must be indistinguishable and non-malleable under the strongest form of cryptographic attack.

1.4 Original Contributions

In discussing and improving upon certain aspects of [2], this thesis offers several original contributions to the field, all of which are intimately related:

A Revival of A Criterion for Cryptosystem Semantics. The first section of this thesis, Section 2, offers a summary of the latest definitions of cryptographic semantics and definitions of security. By cryptographic semantics, we refer to the basic operation and logistics of encryption and decryption engines. We formally require, for instance, that any messages encrypted under a certain scheme can be decrypted to the exact original message. Such requirements seem obvious and are either stated or implicitly assumed in most cryptographic papers. The issues become somewhat complicated, however, when we introduce the concept of an invalid encryption, one which a decryption engine can choose not to decrypt. In its definitional framework, [2] offers no indication as to what is considered an "invalid encryption." This thesis reinstates the older convention of [4], which simply defines an "invalid encryption" as that which an encryption engine would never produce. This slight change has important implications and calls into question some of the more important results of [2].

New Definitions of Security. Attacks such as the lunchtime attack discussed earlier have appeared in cryptography papers since [20] in 1991. Despite their theoretical importance, these attacks do not correspond to the real world security concerns of the Internet. An example of a more practical attack is the Bleichenbacher attack [5], which can be carried out across the wires — as opposed to across the hall. As Bleichenbacher argues, some information about encryptions and cryptosystems can be readily obtained through conventional Internet communication, and we would want any cryptosystem to be secure against such attacks. This paper offers a novel formal definition for the Bleichenbacher-type attack and proves its relations to existing notions of security. As it turns out, this new notion of security is weaker than those of [2]. But we might imagine a situation in which security against attacks such as the lunchtime attack and computational efficiency are at odds. Then, we might settle for a less stringent definition of security but should not settle for anything less than security against Bleichenbacher-type attacks. In Section 4, this thesis provides a formal framework for this practical security credo.

Improvements Upon Old Definitions of Security. The lunchtime attack and variations thereof are known as chosen ciphertext attacks. The lunchtime attack turns out to be one of its weaker formulations, and cryptographers have arrived at more secure – if less practical – paradigms. Because the strongest formulation of chosen ciphertext attack, called adaptive chosen ciphertext attack, has no practical realization, cryptographers have constantly changed its very theoretical formulation to different ends. The definition offered in [2], the most up-to-date definition, leaves something to be desired. Under this definition, a cryptosystem might be secure in the sense of adaptive chosen message attack, but some cryptosystems which are closely related and intuitively just as secure fail to meet the security criteria. In Section 5, this thesis offers improvements upon the indistinguishability formalizations of [2], while still preserving the delicate results of that paper. Similarly, in Section 6, the same is done for non-malleability.

All of these adjustments, improvements and contributions to the definitions of cryptography serve to bridge the gap between mathematical rigor and practical applications. This thesis offers new definitions of security that more closely model today's computing environment and adjusts existing definitions accordingly. The resulting definitional framework should in turn be considered as the revised foundation for all cryptographic research.

2 Existing Definitions of Security

The basic definitions of security have been continuously recast and revised in the literature. The first definitional approaches to cryptography were anything but mathematically rigorous. For example, the notion of security used in the original RSA [25] paper seems to be that "there is no obvious way of recovering plaintext from ciphertext, even given the public encryption key." In this paper, the security of the scheme is not even tightly bound to the underlying computationally intractable problem, which is factoring a composite number. To date, there is no reduction of factorization to the RSA problem, and one could break the RSA scheme without making progress toward factoring large composite integers.

The next step in making definitions for cryptosystems more rigorous was done by Rabin, whose cryptosystem is provably related to the problem of factoring large composite integers [22]. However, even this more careful construction and definition of security left something to be desired. For example, even if an adversary could not recover the entire plaintext given a ciphertext, might it still be possible for an adversary to recover one bit of the plaintext? In the case of RSA, cryptographers have shown that all but $\log_2 n$ bits of the encryptions are secure (where n is the length of the encryption), while conceding that some bits are entirely insecure [19]. Furthermore, schemes such as RSA and Rabin encrypt a given message x the same way every time. Some attacks have been suggested to exploit this fact. For instance, if Bob is encrypting the message "buy" and sending it to his stock broker, an eavesdropper (Alice) could record the message and discover through other means whether Bob told the broker to "buy" or "sell". Then, the next time Bob tells the broker to "buy," Alice will know exactly what Bob had in mind, and she might use this information maliciously.

The solution to both of these problems — security against replay attacks and bit-by-bit security — came with the advent of probabilistic security, as seen in [15]. This paper introduces the novel idea of random encryptions, or a one-to-many encryption scheme whereby a plaintext message x is randomly mapped to one of many possible encryptions of x. Furthermore, Goldwasser and Micali propose a high standard of security. A successful adversary need not recover the entire message from the ciphertext, but rather just one bit of information from the ciphertext.

2.1 Public Key Encryption Semantics

Before we can talk about the definitions of security, it is important to establish a basic set of encryption semantics, which are true of all cryptosystems, whether secure or insecure. These semantics define the mathematical models that are used in contemporary cryptographic research.

The following semantic and notational conventions are used in [2], and will be used for the most part in this thesis as well. Any encryption scheme consists of three algorithms: a key generation algorithm, an encryption algorithm and a decryption algorithm. All three should run in time polynomial with the length of their inputs, and in most modern cryptographic settings, they are probabilistic algorithms. This triple is denoted as $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$.

Rule 1. $\mathcal{K}$ is known as the *key generation algorithm*. Given a security parameter represented in unary, 1^k, $\mathcal{K}(1^k)$ will output a keypair (pk, sk), known as the public key and the private (or secret) key, respectively.

Rule 2. $\mathcal{E}$ is known as the *encryption algorithm*, that given a public key pk and a message $x \in \{0,1\}^*$ will output $\mathcal{E}_{pk}(x) \to y$, where $y \in \{0,1\}^*$ is known as the *ciphertext*.

Rule 3. $\mathcal{D}$ is known as the *decryption algorithm*, that given the corresponding private key sk and a message y encrypted by $\mathcal{E}_{pk}$ will output a message $x \in \{0,1\}^*$.

We require that every ciphertext can be accurately decrypted to the original plaintext, for every keypair generated by $\mathcal{K}$, except in an negligibly small number of cases. Formally, for all $k \in \mathbb{N}$

$$\Pr[(pk, sk) \leftarrow \mathcal{K}(1^k);\ x \leftarrow \{0,1\}^*;\ y \leftarrow \mathcal{E}_{pk}(x);\ x' \leftarrow \mathcal{D}_{sk}(y) :\ x' \neq x] = f(k) \qquad (2.1)$$

where $f(k)$ is negligible in k.[3] This probabilistic formulation of the relationship between the encryption and decryption algorithms is called for, considering that many of the cryptosystems considered in concert with this definition are probabilistic and can fail in a negligibly small number of cases. The above requirement was given in [19]. The exact notation of Equation (2.1) is discussed in Section 2.3.1. Briefly, an experiment is run: a key pair is chosen at random, then a message is chosen at random, then the message is encrypted, then it is decrypted. The probability of the original message and the decrypted ciphertext not being equal should be negligible in k.

For notational convenience, these semantical conventions are combined into a single definition:

Definition 2.1 (Public Key Semantics) *A triple of probabilistic polynomial-time algorithms $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ comprise a public key cryptosystem if they conform to Rules 1 through 3 above and to Equation (2.1).*

Note that in the above definition, any ciphertext $y \in \{0,1\}^*$ is treated as if it were output from $\mathcal{E}_{pk}$, or, alternatively, as if it were considered a "legal" ciphertext. This is the paradigm used for any number of cryptosystems in the literature, such as RSA [25], Goldwasser-Micali [15], Rabin [22], El Gamal [12], and so on. Any y submitted to $\mathcal{D}_{sk}$ will be decrypted as if it were a legitimate encryption, even if it is a phony ciphertext forged by a malicious adversary. As we will discuss in this thesis, certain cryptosystems refine the existing paradigm to solve this problem. In these cryptosystems, for a given keypair, there is a notion of those inputs to the decryption box that were output by the encryption box (called *legal ciphertexts*) and those that would never be output by the encryption box (called *illegal ciphertexts*).

The Legality Criterion. Definition 2.1 is generally assumed of public key cryptosystems. However, the exact behavior of the decryption box varies from paper to paper. In general, how can we determine which messages are legal and which messages are illegal? Unfortunately, [2] makes no mention of which ciphertexts the decryption box will accept. In this thesis, we return to the very logical convention of [4]. Namely, we require that the decryption box refuses to decrypt those messages that were not encrypted using the corresponding public key, and for it to indicate in these cases that the decryption failed. This requirement will be called *The Legality Criterion* and is an addition to the above system

[3] According to [2], a function $f : \mathbb{N} \to \mathbb{R}$ is negligible if for every constant $c \geq 0$ there exists an integer k_c such that $f(k) \leq k^{-c}$ for all $k \geq k_c$.

of semantics. To this effect, we modify the semantics of the decryption box from Rule 3 above.

Assume an arbitrary keypair (pk, sk). Then:

Rule 3′. $\mathcal{D}$ is known as the *decryption algorithm*, that given a ciphertext $y \in \{0,1\}^*$ will output a message $x \in \{0,1\}^* \cup \perp$, where $\perp$ indicates an illegal encryption.

As in Equation (2.1), we express the informal statement: "The decryption box rejects illegal encryptions" in terms of a probabilistic experiment. First, we define the set of illegal encryptions for a given encryption algorithm $\mathcal{E}$ and a given keypair (pk, sk), which we will call $I_{\mathcal{E}_{pk}}$. From our above discussion, we have that $y \in I_{\mathcal{E}_{pk}}$ if for all $x \in \{0,1\}^*$, $y \notin \mathcal{E}_{pk}(x)$.[4] Then, for all $k \in \mathbb{N}$:

$$\Pr[(pk, sk) \leftarrow \mathcal{K}(1^k);\ y \leftarrow I_{\mathcal{E}_{pk}} :\ \mathcal{D}_{sk}(y) \neq \perp] = g(k) \tag{2.2}$$

where $g(k)$ is negligible in k. That is, given a cryptosystem $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, first randomly pick a keypair. Then randomly pick an illegal encryption for that public key. Then in all but negligibly many cases, we expect that the decryption engine will refuse to decrypt this illegal encryption. Note that the intuitive idea behind Equation (2.2) is given in [4], but the probabilistic formulation of the idea appears here for the first time. We stress again that it is quite important to formulate any definitions in terms of probabilistic experiments, since the algorithms to which we will eventually apply these definitions are inherently probabilistic.

With this new restriction on the decryption oracle, a different set of semantics are defined:

Definition 2.2 (Public Key Semantics With Legal Criterion) *A triple of probabilistic polynomial-time algorithms $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ comprise a public key cryptosystem under the Legality Criterion if they conform to Rules 1, 2, 3′ and to Equations (2.1) and (2.2).*

Although the Legality Criterion might seem like a trivial addition to Definition 2.1, the close connection it establishes between the encryption and the decryption boxes has deep implications. Certain results that hold under Definition 2.1 might not hold under Definition 2.2. For example, refer to Theorems 3.6 and Theorems 3.7 from [2]. Both of these proofs rely upon a constructed encryption box $\mathcal{E}'_{pk}$, which on input x will output messages of the form $(0 \parallel \mathcal{E}_{pk}(x))$.[5] By the Legality Criterion, any messages of the form $(1 \parallel y)$ would be illegal encryptions, and this makes sense, because $\mathcal{E}'_{pk}$ would have never output something of this form. However, [2] allows an adversary to query the decryption box with messages of the form $(1 \parallel y)$, thereby gaining information about the secret key. The decryption oracle indeed answers some queries of the form $\mathcal{D}_{sk}(1 \parallel y)$ with responses $x \neq \perp$. Π' cannot be considered a valid cryptosystem under Definition 2.2, and thus the results based upon Π' might not hold for cryptosystems that uphold the Legality Criterion.

The Legality Criterion is indeed a very natural restriction to cryptosystem semantics, one which this thesis will try to uphold. However, proofs of certain theorems seem impossible with Definition 2.2 (such as those mentioned above), and in those cases, we will revert to Definition 2.1.

An efficient and very secure scheme that conforms to both definitions can be found in [8].

[4] We write $y \notin \mathcal{E}_{pk}(x)$ to mean that y is not a possible output of the one-to-many function $\mathcal{E}_{pk}(x)$.

[5] Throughout this thesis, we adopt the notation that $(a \parallel b)$ denotes a concatenated with b .

2.2 The Strength and Goals of Cryptographic Attack

Now that we have defined the most basic semantics of cryptosystem operation, we desire to define what we mean by security. One of the novel approaches of [2] is to treat the power and the goals of different cryptographic attacks completely orthogonally. That is, the goals of certain attacks are different: some might try to gain information about ciphertexts, while others might attempt to forge new ciphertexts that are related to existing ones (for instance, given $\mathcal{E}_{pk}(x)$, we might desire to produce $\mathcal{E}_{pk}(x')$ where $x' = 2x$; such a goal is clearly possible with RSA and others). The former goal is known as *distinguishability*, the ability to determine properties about a message given its encryption, and the ability to distinguish between it and other encrypted messages. The latter is known as *malleability*. As shown in [2] and others, the two goals are closely related.

Another issue to consider when formulating notions of security is the strength of the adversary. The weakest adversary in cryptological settings is one that can only tap a line of communication and overhear encrypted messages. But we might also imagine "stronger" attacks, such as the case in which an adversary wants to learn about messages sent with the keypair (pk, sk), and has temporary access to the decryption box $\mathcal{D}_{sk}$ — the lunchtime attack.

2.2.1 Strength of Attack

Here we assume the method of notation and attack formulation developed in [2]. For every cryptosystem Π, we first imagine a large set of adversaries, those who wish to attack the system and uncover information that Π intends to keep secret. An adversary to Π is given by two probabilistic polynomial time algorithms, $A = (A_1, A_2)$. The general idea of any attack against an public key cryptosystem is that an adversary is presented with a "challenge ciphertext" y and must determine certain properties about that y, or better yet, the underlying encryption scheme. The relative strength of any adversary A, attempting to learn about y or the cryptosystem Π running with the keypair (pk, sk), is given by how much access A has to the decryption box, $\mathcal{D}_{sk}$. The more access the adversary has, the stronger his attack will be.

How should we go about qualifying "how much access" an adversary has to the decryption box? [2] and others before have broken down an adversary's attack into two stages. In the first stage, the "find" stage according to [4], the adversary analyzes the public key and tries to determine which plaintexts, when encrypted, are vulnerable to attack. This is the job of A_1. In the second stage, the "guess" stage, the adversary (A_2) will be presented with a challenge ciphertext y, an encryption of one of the plaintexts he found in stage 1. The adversary will then be challenged either to determine information about $\mathcal{D}_{sk}(y)$ (in the case of distinguishing), or to forge a new message y' so that $\mathcal{D}_{sk}(y)$ and $\mathcal{D}_{sk}(y')$ are related in some useful way (in the case of malleating). Given this paradigm for adversarial attacks, "how much access" an adversary has to the decryption box is determined by whether A_1, A_2 or both can access $\mathcal{D}_{sk}$.

This two-stage formulation of adversaries might seem somewhat contrived, and unrelated to real-world scenarios. In reality, an adversary does not have the liberty to choose those plaintexts which he will later be challenged to decrypt. Rather, the adversary — in sniffing packets off of a public network — will be challenged with a ciphertext y, which could be the encryption of any $x \in \{0,1\}^*$. The theory behind the two-stage attack given above is

still justified. If a given scheme is secure against a two-stage adversary who can carefully choose the space of candidate plaintexts, then it is certainly secure against the real-world adversary, who is challenged with the encryption of any $x \in \{0,1\}^*$.

We now enumerate the three strengths of attack:

1. The weakest is called *chosen-plaintext attack* (CPA), in which A has no access to the decryption box. A can, however, make queries to the encryption box $\mathcal{E}_{pk}$ bounded only by computational requirements (i.e., A is a polynomial time algorithm). Schemes that are secure against a CPA attack might be completely susceptible to the next two attacks.

2. The second strongest attack is called *non-adaptive chosen ciphertext attack* (CCA1). In this attack, the adversary is given access to the decryption box *before* she receives the challenge ciphertext y. First formalized in [20], this attack is often called "the lunchtime attack:" it may come about when an adversary gains access to $\mathcal{D}_{sk}$ for a short period of time (i.e., lunchtime), and submits carefully chosen queries to $\mathcal{D}_{sk}$, in attempt to learn valuable properties about sk. It is important to note that certain IND-CPA secure cryptosystems, like Blum-Goldwasser [6], are breakable under the CCA1 attack. Discussion in [19] revealed that a careful attack can reveal the secret key.

3. The strongest attack is known as *adaptive chosen ciphertext attack* (CCA2). In this model, A has access to the decryption box both before and after it receives the challenge ciphertext y. That is, both A_1 and A_2 can make queries to $\mathcal{D}_{sk}$, with the only restriction being that A_2 cannot submit the query y to $\mathcal{D}_{sk}$. This attack is also known as the Rackoff-Simon Attack [23], and is "adaptive" insofar as the queries to $\mathcal{D}_{sk}$ can depend on the ciphertext y. Unlike CCA1, this attack is somewhat contrived and has no obvious real-world realization. However, it represents the strongest possible attack model in this setting, and proving that a scheme is secure against CCA2 is a strong testament to its overall security. As shown in [2], if Π is known to be polynomially indistinguishable in the context of this attack, then it follows that Π is also non-malleable. This a strong and useful result in the domain of protocol design.

2.3 Formalizations

2.3.1 Indistinguishability

We now present the most up-to-date formalizations of the above attacks, where the goal of the attack is for an adversary to distinguish encryptions. Using the same nomenclature as in [2], there are three types of security to be considered: IND-CPA, IND-CCA1, and IND-CCA2. Here, IND represents the goal of security, to be INDistinguishable. Likewise, CPA, CCA1 and CCA2 represent the strength of attack, whether passive adversary, or (adaptive) chosen ciphertext attack.

All three definitions are based upon the same definitional model. Namely, an experiment is carried out in which an adversary interacts with the cryptosystem. The adversary's ability to succeed in the experiment is measured, and if the likelihood of her successes are sufficiently small we can conclude that the cryptosystem is secure against her attack. If all adversaries have a low probability of success, then we deem the cryptosystem secure in the general sense.

In particular, the experiment consists of three abstract parties: the adversary, given by the algorithm A, the cryptosystem, given by Π, and the umpire. The adversary corresponds to the "bad-guy," she who is attempting to gain information that other parties intend to keep secret. Similarly, Π corresponds to the "good-guy," he who enables parties to interact securely over the network. The umpire is the one who will coordinate an interaction, or an "experiment," between the two, to determine what capacity that adversary has to defeat the cryptosystem. The steps of the experiment, in order, are:

1. The umpire runs the key generation algorithm $\mathcal{K}$, to generate a keypair, with security parameter k. Call the keypair generated (pk, sk).

2. The umpire then gives the public key to the adversary. The adversary $A = (A_1, A_2)$ has two stages. The first, A_1, intends to find a pair of messages that the second, A_2, will later on distinguish. Thus $(x_0, x_1, s) \leftarrow A_1(pk)$ means that A_1 will output two plaintexts, x_0 and x_1, plus additional internal state information that it can pass on to A_2, which is given by s. Note that in the case of chosen ciphertext attack (CCA1 or CCA2), the adversary A_1 has access to the decryption oracle $\mathcal{D}_{sk}$. That is, he can query the decryption oracle to gain information about sk.

3. The umpire then picks a random bit, b, and keeps this bit secret. If the random bit is 0, he will encrypt $y \leftarrow \mathcal{E}_{pk}(x_0)$. Otherwise, he will encrypt $y \leftarrow \mathcal{E}_{pk}(x_1)$.

4. The umpire then challenges A_2 to discover b. He gives the adversary x_0, x_1, the challenge ciphertext y and the internal state information from A_1 given by s. A_2 must now decide whether y is an encryption of x_0 or of x_1. In the case of adaptive chosen ciphertext attack, A_2 will have access to the decryption oracle $\mathcal{D}_{sk}$. A_2 will query this oracle to gain more information about y without explicitly asking for $\mathcal{D}_{sk}(y)$. A_2 will then output a guess bit, call it b'. That is, $b' \leftarrow \mathcal{D}_{sk}(x_0, x_1, y, s)$.

5. The umpire then checks to see if $b' = b$. If so, then A has succeeded. If not, then she has failed.

In this context, successful adversaries are those that can win the experiment with a probability of $\frac{1}{2} + \frac{1}{Q(k)}$, where Q is non-negligible function. Note that an adversary that guesses randomly will win with probability $\frac{1}{2}$. A successful adversary must have a non-negligible advantage (given by Q) over the uninformed adversary. Again, if there are no successful adversaries for a particular cryptosystem, then that cryptosystem meets the criteria for security.

Cryptographers have invented a symbolic vocabulary for expressing such interactions. Experiments take the general form:

$$\Pr[x_1 \leftarrow T_1(y_1);\ x_2 \leftarrow T_2(y_2); \ldots x_n \leftarrow T_n(y_n) : R(x')] \tag{2.3}$$

The series of n assignments are carried out by the umpire. He will coordinate the different probabilistic during machines T_i, feeding them input x_i and collecting output y_i as the experiment dictates. Each of the x_i and the y_i can be a scalar or a vector of values. Note that some of the T_i might be algorithms that pick randomly from a distribution. It is common for a stage of an experiment to be of the form $b \leftarrow \{0, 1\}$, which is shorthand for "pick b from $\{0, 1\}$ at random, with uniform probability." Also, statements such as $a \leftarrow b$ can denote simple assignment operations.

After the series of n algorithms have been run in order, the umpire tests to see if relation $R(x')$ holds, where x' is a vector of values derived from the outputs $x_1, \ldots x_n$. Note that R will output either true or false. In many cases, this relation is simple equality. The probability is taken over the random coin flips that determine the randomness of the probabilistic turning machines T_i and the random selection of values from distributions. A success is given by an instance of the experiment after which $R(x')$ holds.

With these notational conventions and the experiment described above, [2] defines indistinguishability as follows:

Definition 2.3 *[IND-CPA, IND-CCA1, IND-CCA2] from [2].*

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 $i \in \{0,1\}$, we mean $\mathcal{O}_i$ is the function which, on any input, returns the empty string, ε.

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 $k \in \mathbb{N}$ let $\mathsf{Adv}^{ind\text{-}atk}_{A,\Pi}(k) =$

$$2 \cdot \Pr[(sk, pk) \leftarrow \mathcal{K}(1^k);\ (x_0, x_1, s) \leftarrow A_1^{\mathcal{O}_1};\ b \leftarrow \{0,1\};\\ y \leftarrow \mathcal{E}_{pk}(x_b) :\ A_2^{\mathcal{O}_2}(x_0, x_1, s, y) = b] - 1 \tag{2.4}$$

where

- *If atk = cpa, then $\mathcal{O}_1(\cdot) = \varepsilon$ and $\mathcal{O}_2(\cdot) = \varepsilon$.*
- *If atk = cca2, 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 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 $\mathsf{Adv}^{ind\text{-}atk}_{A,\Pi}(\cdot)$ is negligible.

To clarify notation, we notice that $\mathsf{Adv}^{\text{ind-atk}}_{A,\Pi}(\cdot)$ is the advantage that the adversary A has in defeating Π under the attack given by atk. It is also a notational convention for $A_1^{\mathcal{O}_1}$ to mean that "algorithm A_1 has access to oracle $\mathcal{O}_1$." This notation will be used repeatedly throughout this thesis.

Although bit-by-bit security is not explicit from the above definitions, reductions in [15] prove that if a cryptosystem meets the indistinguishability criteria, then its encryptions are bit-by-bit secure. The converse also holds.

2.3.2 Non-Malleability

The definition [2] proposes for non-malleability will also be important to any arguments about indistinguishability. The most canonical definitions of malleability can be found in the papers of Dolev, Dwork and Naor [10, 11]. However, [2] offers a significant simplification to the existing definition, one which is at least as strong as DDN. It is an open problem to show whether the definition of [2] is equivalent to or stronger than that of DDN.

The definition of [2] brings the rather simple of idea of malleability — deforming a challenge ciphertext to another ciphertext so that the respective decryptions are related —

to a new level of generality. In their model, the adversary $A = (A_1, A_2)$ has considerable flexibility. She can pick the message space from which all messages are chosen, based on the public key. Based on an encryption of a plaintext taken from this message space, the adversary gets to choose a relation and a vector of malleated ciphertexts. The adversary then hopes that the decryption of the malleated ciphertexts she generated and the decryption of the challenge ciphertext, when input into the relation she generated, make the relation true. The adversary further hopes that there is a significant difference between this "Real Experiment" and a "Placebo Experiment," in which the umpire effectively misleads her. In particular, the steps of The Real Experiment and The Placebo Experiment are as follows:

The Real Experiment

Step 1. As usual, the umpire starts off with a security parameter k, and generates a keypair $(pk, sk) \leftarrow \mathcal{K}(1^k)$ at random.

Step 2. The umpire then passes the public key pk to the adversary's first algorithm A_1. As in the distinguishability experiments, A_1 has access to the decryption oracle $\mathcal{D}_{sk}$ in the case of CCA1 or CCA2. A_1 then computes a distribution of messages M such that she will be able to malleate the encryption of messages chosen from M. She also outputs internal state information s that will be passed on later to A_2.

Step 3. The umpire then picks a message x at random from M, and encrypts $y \leftarrow \mathcal{E}_{pk}(x)$. This encryption, y, now is the challenge ciphertext. A_2's challenge will be to malleate y.

Step 4. The umpire now gives (M, s, y) to A_2 as a challenge. A_2 attempts to output a relation R and a vector of malleated ciphertexts given by $\mathbf{y}$ so that $R(x, \mathcal{D}_{sk}(\mathbf{y}))$ holds, and $R(\tilde{x}, \mathcal{D}_{sk}(\mathbf{y}))$ does not hold for random $\tilde{x} \in M$, $x \neq \tilde{x}$. Note that in CCA2, A_2 has access to the decryption oracle $\mathcal{D}_{sk}$. Although not explicitly stated in [2], it is clear that A_2 should not ask for $\mathcal{D}_{sk}(y)$, the decryption of the challenge ciphertext; this would enable trivial attacks.

Step 5. The umpire decrypts each element of the vector that A_2 output; this is notated $\mathbf{x} = \mathcal{D}_{sk}(\mathbf{y})$. He then determines whether or not A succeeds in defeating Π in the sense of malleability. A succeeds if $y \notin \mathbf{y}$ and $\perp \notin \mathbf{x}$ and finally that the relationship holds, that $R(x, \mathbf{x})$ is true. The umpire demands that $y \notin \mathbf{y}$ so as not to attribute success to the adversary for the trivial attack: otherwise, the adversary could always output $\mathbf{y} = (y)$ and R such that $R(a, b)$ is true if and only if $a = b$. Clearly, this attack does not pose a threat to the security of Π. By similar logic, the umpire does not give the adversary credit for corrupting y into an illegal ciphertext. Thus, the umpire insists that no member of $\mathbf{y}$ is an illegal ciphertext, or equivalently, that $\perp \notin \mathbf{x}$.

The Placebo Experiment

Steps 1 to 4. Same as in The Real Experiment

Step 5. In this step we desire to prevent the adversary A from outputting a relation R such that R is always true. Indeed, A should only get credit for successfully mounting an NM attack against Π if she succeeds in producing ciphertexts that are related to the

challenge ciphertext y by a non-trivial relation R. Thus, in The Placebo Experiment, the umpire will pick another message from the message space $\tilde{x} \in M$. It will then test to see if $y \in \mathbf{y}$ and $\perp \in \mathbf{x}$. As in the above experiment, this experiment will be a failure if either of these conditions holds. Next, the umpire will see if $R(\tilde{x}, \mathbf{x})$ holds. If it does, then A has succeed in The Placebo Experiment. A successful adversary A should not succeed in The Placebo Experiment.

The overall advantage of an adversary A is then given by its real advantage in the above experiments. Indeed, A succeeds in malleating encryptions of Π whenever she succeeds in The Real Experiment. However, A only adds random noise when she succeeds in The Placebo Experiment. Thus, we can conclude:

$$\begin{aligned}\Pr[A \text{ succeeds in NM-attacking } \Pi] = &\Pr[A \text{ succeeds in The Real Experiment}] \\ &- \Pr[A \text{ succeeds in The Placebo Experiment}]\end{aligned} \tag{2.5}$$

This formulation has further advantages over previous ones: namely, the two-step adversary mirrors that in the distinguishability experiments, so it will again be possible to concisely formulate NM-CPA, NM-CCA1, and NM-CCA2 — that is, non-malleability attacks with no access to the decryption oracle (NM-CPA), NM attacks with access to the decryption oracle only before the challenge ciphertext is given (NM-CCA1), and NM attacks with access to the decryption oracle throughout (NM-CCA2).

Definition 2.4 *[NM-CPA, NM-CCA1, NM-CCA2] from [2].*

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 $k \in \mathbb{N}$ define

$$\mathsf{Adv}^{nm\text{-}atk}_{A,\Pi} \stackrel{def}{=} \left|\mathsf{Succ}^{nm\text{-}atk}_{A,\Pi}(k) - \mathsf{Succ}^{nm\text{-}atk}_{A,\Pi,\$}(k)\right| \tag{2.6}$$

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

$$\begin{aligned}\Pr[&(pk, sk) \leftarrow \mathcal{K}(1^k);\ (M, s) \leftarrow A_1^{\mathcal{O}_1}(pk);\ x \leftarrow M;\ y \leftarrow \mathcal{E}_{pk}(x); \\ &(R, \mathbf{y}) \leftarrow A_2^{\mathcal{O}_2}(M, s, y);\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) :\ y \notin \mathsf{y} \wedge \perp \notin \mathbf{x} \wedge R(x, \mathbf{x})]\end{aligned} \tag{2.7}$$

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

$$\begin{aligned}\Pr[&(pk, sk) \leftarrow \mathcal{K}(1^k);\ (M, s) \leftarrow A_1^{\mathcal{O}_1}(pk);\ x, \tilde{x} \leftarrow M;\ y \leftarrow \mathcal{E}_{pk}(x); \\ &(R, \mathbf{y}) \leftarrow A_2^{\mathcal{O}_2}(M, s, y);\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) :\ y \notin \mathsf{y} \wedge \perp \notin \mathbf{x} \wedge R(\tilde{x}, \mathbf{x})]\end{aligned} \tag{2.8}$$

where:

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

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 relations R computable in time $p(k)$, then $\mathsf{Adv}^{nm\text{-}atk}_{A,\Pi}(\cdot)$ is negligible.

In addition we note that A_2 must not submit y to $\mathcal{O}_2$ in the case that atk=cca2, again, to avoid the trivial attack.

We will accept these definitions at face value now but will later develop some important revisions, especially concerning CCA2.

3 The Random Oracle Methodology

In this section, we introduce the Random Oracle Methodology and with it tools relevant to our discussion of cryptographic definitions.

Any formalizations in the standard computational model have analogues in what is called the Random Oracle Model. First proposed by Bellare and Rogaway [3], the Random Oracle Model assumes that fair players and adversaries alike have access to a shared, truly random function known as a random oracle. On input of the string $x \in \{0,1\}^*$, a random oracle R that has not been given x before will generate and output a random string r, so that $R(x) = r$ for all subsequent queries. One way to think of a random oracle is as an ideal hash function, one that has no statistical properties that might be exploited by a malicious adversary. In the Random Oracle Model, we prove certain things about algorithms assuming access to a perfect random oracle R. In practice, a polynomial-time computable function is substituted for R, such as the Secure Hash Algorithm (SHA) [21] or MD5 Message Digest [24]. The practical implementations have no formal proofs of security, but rather security that can be related to provable security in an ideal model.

It is important to note that the Random Oracle Model represents an entirely different computational model from what we have so far discussed. Up to now, we have made only one assumption: that trapdoor functions — those that behave like RSA, Rabin or El Gamal — exist. As shown in [13], given such a function, it is possible to construct any number of cryptographic primitives. These assumptions are often regarded as the "standard assumptions," or the "real world," especially when compared to the Random Oracle Model. The Random Oracle Model goes on to make an additional assumption: that ideal random oracles can be effectively approximated, for it is clear from a computational complexity perspective that they cannot exist.

3.1 Formalizations

Formulation of definitions in the Random Oracle Model closely follows those of the standard computational model. The only difference is that all algorithms and adversaries will have access to a random oracle, that is itself randomly chosen at the beginning of the definitional experiment. For example, in Definition 2.3, we would only need to change Equation (2.4) to the following experiment:

$$\begin{aligned} 2 \cdot \Pr[H \leftarrow \mathsf{Hash};\, (sk, pk) \leftarrow \mathcal{K}^H(1^k);\, (x_0, x_1, s) \leftarrow A_1^{\mathcal{O}_1, H};\, b \leftarrow \{0,1\}; \\ y \leftarrow \mathcal{E}_{pk}^H(x_b) :\, A_2^{\mathcal{O}_2, H}(x_0, x_1, s, y) = b] - 1 \end{aligned} \tag{3.1}$$

If an encryption box $\mathcal{E}_{pk}$ operates in the Random Oracle Model and has access to the random oracle R, then we notate it $\mathcal{E}_{pk}^H$. By the notation $H \leftarrow \mathsf{Hash}$, we mean pick random function H at random from the family of random functions $\{f : \{0,1\}^* \rightarrow \{0,1\}^*\}$. Although the above equation does not mention the decryption box $\mathcal{D}_{sk}^H$ explicitly, it is clear to see that it, too, has access to H. Works such as [4] have developed very efficient cryptosystems that are provably secure in this random oracle sense of IND-CPA. Such systems betray the motivation behind the Random Oracle Model, which is the marriage between computational efficiency and theoretical security.

However, recent analysis of the Random Oracle Model [7] has revealed that certain cryptosystems that are provably secure in the Random Oracle Model are in fact provably

insecure in every practical implementation. That is, implementing these schemes using any tractable hash function (such as SHA or MD5) will result in insecure schemes. The schemes that [7] uses to arrive these results are admittedly very contrived and would never be implemented. But the point remains; according to [7], "the lesson is that the mere fact that a scheme is secure in the Random Oracle Model does not necessarily imply that a particular implementation of it (in the real world) is secure, or even that this scheme does not have any 'structural flaws.' " Such results cast serious doubt on the strength of security attainable within the bounds of the Random Oracle Model.

3.2 Plaintext Awareness

Putting these objections aside for now, a result is discussed that has to date only been formulated in the Random Oracle Model. In this section, we introduce a new problem, namely *plaintext awareness* from [4], which is related to Professor Rabin's "proof of plaintext knowledge" [1]. As usual, we consider a public key cryptosystem $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ but add an additional restriction.

Consider again the case of Alice and Bob communicating over a network. Alice has access to Bob's public key and encrypts her plaintexts with $\mathcal{E}_{pk}$, while Bob decrypts with $\mathcal{D}_{sk}$. But Bob is more cautious than many. He wants to be absolutely certain that Alice is not a malicious adversary, and that she is not sending carefully prepared messages that might yield information about his secret key or other encryptions that have been encrypted with his public key. Thus, he requires Alice, for every encryption $\mathcal{E}_{pk}(x)$ she sends over the network, to send an additional "proof" that she knows what x is. Clearly, this proof must not yield any information about x to an eavesdropper listening over the network, so as not to defeat the purpose of the encryption scheme. Once Bob is certain that Alice indeed can compute x on her own, then his decryption scheme $\mathcal{D}_{sk}$ will go ahead and output x. Otherwise, his decryption box will suppress output.

Hence, in the case in which Alice is indeed a malicious adversary, who is attempting some sort of chosen ciphertext attack, one of two situations might occur. First, Alice might forge a message y for which she cannot compute $\mathcal{D}_{sk}(y)$, thus she will not be able to prove her knowledge of $\mathcal{D}_{sk}(y)$, and Bob's decryption box will not output any information. In the second case, Alice forges a message y for which she can compute $x = \mathcal{D}_{sk}(y)$ — perhaps just by encrypting x. Thus she can prove to Bob that she knows x, and Bob's decryption box will indeed output x. However, Alice cannot possibly gain any knowledge from access to $\mathcal{D}_{sk}$, because as she already proved to Bob, she can compute x on her own.

Although various proof of knowledge protocols have existed in the literature for over a decade, the first practical and efficient marriage of proof of knowledge and encryption schemes appears in the work of Bellare and Rogaway [4].

It is important to note, however, that the proposed definition and scheme are only valid in the Random Oracle Model, and no similar claims have been produced in the realm of standard cryptographic assumptions. Despite any difficulties with the Random Oracle Model, recent work has finally stated, formalized and proven the relations among different ideas of cryptographic security, including that of plaintext awareness [2]. This work has formally established that plaintext aware schemes are secure against the strongest form of chosen ciphertext attack, and that the converse is false.

3.3 Definitions

The original definition for plaintext awareness appeared in [4] but was later revised for [2]. The difference between the two versions is subtle, but the latter proves to be a stronger definition and is the one we shall consider here.

Plaintext aware schemes are defined with the standard syntax for encryption schemes, in terms of a triple of algorithms, given by $(\mathcal{K}, \mathcal{E}^H, \mathcal{D}^H)$: key generation, encryption and decryption algorithms. Again, we write $\mathcal{E}^H$ and $\mathcal{D}^H$ to denote that the two algorithms have access to the random oracle H. In this setting, a malicious adversary B has a new goal: to forge a message that could have been output by $\mathcal{E}_{pk}$ for which she does not know the corresponding plaintext. A scheme is thus plaintext aware if in order for B to formulate a legal encryption y, she must have "known" x such that $\mathcal{E}_{pk}(x) = y$. The formalizations of "knowledge" come from the language of zero-knowledge proofs, as formulated by [16] and summarized by [13]. In this context, B "knows" x if B can compute it in polynomial time.

We first consider an arbitrary adversary B. Given the public key of a cryptosystem, pk, B desires output y such that $\exists x \in \{0,1\}^*$ where $y \in \mathcal{E}_{pk}(x)$. It is also required that B does not "know" x. In computing such an y, B has access to both H, the random oracle, and $\mathcal{E}^H_{pk}$, the encryption oracle. The latter represents information that B might have acquired through eavesdropping: she might know (x, y) such that $\mathcal{E}^H_{pk}(x) = y$, but she might not have access to the internal queries $\mathcal{E}^H_{pk}$ made to the random oracle H.

Plaintext awareness demands the existence of a universal "plaintext extractor," denoted K.[6] Roughly speaking, for an arbitrary pk, K should be able to rederive the plaintext x that corresponds to an encryption y output by an adversary B, if it is given enough information about the internal computations of B. If it were not given any additional information, K would simply be a decryption box. The language of random oracles provides a convenient way to formalize what we mean by "internal computations:" in this case, it means all of the oracle queries made by B to H, and all of the responses to the queries made to H and $\mathcal{E}^H_{pk}$. Note that we do not include the inputs to $\mathcal{E}^H_{pk}$ to model the amount of information acquired by eavesdropping (ciphertexts without corresponding plaintexts). If given this information, K can compute the decryption of the message output by B in polynomial time, then B, having access to all of the same information, can compute the plaintext as well in polynomial time. This is what is meant by "knowledge" of the plaintext.

We can now formalize our definitions. First, we introduce notation to model a transcript of an adversary's queries. Using the notation of [2], $(hH, C, y) \leftarrow \mathsf{run}\, B^{H, \mathcal{E}^H_{pk}}(pk)$ means that B runs with public key pk, outputting ciphertext y such that $\mathcal{D}^H_{sk}(y) \neq \perp$ (where sk is the corresponding secret key to pk). While B is running, a transcript is also recorded, given by hH and C, where $hH = ((h_1, H_1), (h_2, H_2), ..., (h_{q_H}, H_{q_H}))$, where the h_i represent B's queries to the H oracle, and H_i represent H's responses to B. Similarly, $C = (y_1, y_2, ..., y_{q_\mathcal{E}})$ represent $\mathcal{E}^H_{pk}$'s responses to B's queries. Again, the definition of the success of an adversary is given by a probabilistic experiment. The three parties of the experiment are now the adversary who attempts to forge a ciphertext, the extractor who attempts to foil any attempt at forgery, and the umpire, who will judge the success of the two parties.

1. The umpire picks a random oracle function at random from the set of all possible random oracle functions. This is notated: $H \leftarrow \mathsf{Hash}$.

[6] Note the difference between $\mathcal{K}$, the key generation algorithm, and K, the plaintext extractor. This is the notational convention adapted in [2].

2. Given a security parameter k, the umpire picks a keypair at random, running $(pk, sk) \leftarrow \mathcal{K}(1^k)$.

3. The umpire submits the challenge public key, pk, to the adversary B. $B(pk)$ runs in attempt to find a legal ciphertext y while making minimal queries to H and to $\mathcal{E}_{pk}$. The adversary is, after all, required to report the sets of all of her queries (hH and C) to the umpire.

4. The umpire, in turn, submits the ciphertext y the adversary B has output, and all of the adversary's oracle queries to the plaintext extractor. The umpire runs $K(hH, C, y, pk) \rightarrow x$. K's challenge is now to determine $\mathcal{D}_{sk}(y)$. Because we usually assume that Π is an IND-CPA cryptosystem, K must use information aside from just y to carry out its task. If K did not need hH and C, then it could decrypt any encryption, and the cryptosystem would not be secure. Thus, on the basis of the internal computations performed by B, which can be deduced from the sets hH, and C, K runs in attempt to recover $x = \mathcal{D}_{sk}(y)$.

5. The umpire then checks the results, deeming that K is successful when it outputs x such that $x = \mathcal{D}_{sk}(y)$.

If K is almost always successful against all adversaries, then we can conclude that K is a plaintext extractor for Π. The existence of such an algorithm assures us that if an adversary computes a ciphertext y, then the adversary must necessarily be able to compute $\mathcal{D}_{sk}(y)$ and thus could not gain any useful information from a chosen ciphertext attack.

We take the following definition directly from [2]:

Definition 3.1 (Plaintext Awareness from [2]) *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 $k \in \mathbb{N}$ define:*

$$\begin{aligned}\mathsf{Succ}_{K,B,\Pi}(k) = \Pr[H \leftarrow \mathsf{Hash};\ (pk, sk) \leftarrow \mathcal{K}(1^k);\ (hH, C, y) \leftarrow \mathsf{run}B^{H,\mathcal{E}^H_{pk}}(pk) :\\ K(hH, C, y, pk) = \mathcal{D}^H_{sk}(y)]\end{aligned} \tag{3.2}$$

We do not consider "replay attacks," thus $y \notin C$. K is known as a $\lambda(k)$-extractor if K has running time polynomial in the length of its inputs and for every adversary B, $\mathsf{Succ}_{K,P,\Pi}(k) \geq \lambda(k)$. Define that Π is secure in the sense of Plaintext Awareness if there exists a $\lambda(k)$ extractor K where $1 - \lambda(k)$ is negligible.

3.4 An Example of a Plaintext Aware Scheme

There are no published public key schemes that meet the very stringent requirements of Definition 3.1. The only existing scheme in the literature that even approximates the plaintext aware paradigm is given by Bellare and Rogaway in [4]. Given any trapdoor permutation, an encryption scheme is constructed that is semantically secure and plaintext aware in a weaker sense.

We are given a cryptosystem $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ that describes a public key cryptosystem based on a trapdoor permutation. We now construct $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$, which has access to 2 random functions, given by G and H. For the purposes of this section, it is assumed that

all algorithms have access to G and H. We do not notate this explicitly for the purposes of clarity.

For any security parameter k, we pick two other security parameters based on k, k_0 and k_1, so that $k_0 + k_1 < k$. In the constructed encryption scheme, k_0 roughly corresponds to security against an decryption adversary, and k_1 roughly corresponds to security against a message forging adversary. To determine k_0 and k_1 given the security parameter k, we define two functions, $k_0(\cdot)$ and $k_1(\cdot)$ so that for all $k > 1$, $k_0(k) + k_1(k) < k$. For a security parameter k, the encryption scheme takes input x such that $|x| = n(k)$, where $n(k) = k - k_0(k) - k_1(k)$. We now define the triple of algorithms:

1. $\mathcal{K}' = \mathcal{K}$.
2. Algorithm $\mathcal{E}'_{pk}$: Given security parameter k, accept input of length $n(k)$. For any $x \in \{0,1\}^{n(k)}$, select r at random of length $k_0(k)$, and output:

$$\mathcal{E}'_{pk}(x) = \mathcal{E}_{pk}(x0^{k_1(k)} \oplus G(r) \parallel r \oplus H(x0^{k_1(k)} \oplus G(r))) \tag{3.3}$$

   Note that $\oplus$ denotes bitwise-XOR, $\parallel$ denotes concatenation and $x0^n$ denotes x padded with n zeros.

3. Algorithm $\mathcal{D}'_{sk}$: On input y such that $|y| = k$, compute $w \leftarrow \mathcal{D}_{sk}(y)$. Set s to the first $n(k) + k_1(k)$ bits of w and t to the last $k_0(k)$ bits of w. Now recover $r \leftarrow t \oplus H(s)$, and $z \leftarrow s \oplus G(r)$. If the last $k_1(k)$ bits of z are not all 0, then reject y; output $\perp$. Otherwise, x is set equal to the first $n(k)$ bits of z, and is output.

This scheme is semantically secure under the normal assumptions of the Random Oracle Model. This scheme is also shown to be plaintext aware, but only if we assume that the adversary B does not have access to messages from $\mathcal{E}'_{pk}$ via eavesdropping. This scheme, therefore, has not been proven secure in the random oracle sense of IND-CCA2. The weaker proofs of security are omitted, but it is helpful to provide an intuitive argument for the plaintext awareness of Π'.

For an adversary B to construct a ciphertext y that is accepted by a decryption box $\mathcal{D}'_{sk}$, y must be such that $z = \mathcal{D}_{sk}(y)$ is of the appropriate form, namely, that there exists an r and x so that

$$y = (x0^{k_1(k)} \oplus G(r) \parallel r \oplus H(x0^{k_1(k)} \oplus G(r))) \tag{3.4}$$

where k is the security parameter such that $n(k) = |z|$. The proof of Π''s plaintext awareness argues that for B to have constructed such a y, it must have queried the G oracle for $G(r)$ and the H oracle for $H(x0^{k_1(k)} \oplus G(r))$. Therefore, B must have known r and $x0^{k_1(k)} \oplus G(r)$, which implies that B can readily compute x. Thus, for B to submit a legal ciphertext y to the decryption box $\mathcal{D}'_{sk}$, it must, with good probability, be able to compute $x = \mathcal{D}'_{sk}(y)$.

4 Illegal Ciphertext Attack

This section of the paper is an original contribution to cryptographic definitions. As we mentioned above, IND-CCA2 represents one of the strongest possible types of attack against a cryptosystem. Although it does not correspond to real world attack models, it is a useful criterion for measuring the security of a cryptosystem. If Π is secure in the sense of IND-CCA2, then we expect it to be secure against other, more practical attacks. A perfect example of a more practical attack is one that was recently presented by Bleichenbacher [5]. He calls his attack an "adaptive chosen ciphertext attack against certain protocols based on RSA."

The Bleichenbacher attack on RSA-PKCS#1 proceeds as follows: the adversary wants to find $m \equiv c^d \pmod{n}$, where c is an arbitrary integer (or the ciphertext), n is the public key modulus, and d is the secret key decryption exponent. The attacker then produces a series of messages of the form $c' \equiv cs^e \pmod{n}$, where e is the public key encryption exponent. Based on whether or not c' is accepted by the decryption oracle as a legal ciphertext, the adversary can narrow down the range of possible values for m. As Bleichenbacher discusses in his paper, this form of attack has immediate practical applications. Many protocols such as RSA-PCKS#1 inform the adversary over the network if it ever submits an illegal encryption, while tacitly accepting any legal encryptions. Thus, the access the adversary needs to the decryption box is much less than that given by IND-CCA1 or IND-CCA2. Indeed, a Bleichenbacher attack can be mounted over the Internet, and does not require an adversary to have physical access to the decryption box (as do IND-CCA1 and IND-CCA2).

This type of attack motivates a new definition of security, which we will call *Illegal Ciphertext Attack* (ICA). Such a definition will help to bridge the gap between the practical considerations discussed in [5] and the more formal and theoretical considerations discussed in [2].

4.1 The Definition

The definition of ICA will closely resemble that of CCA, the major difference being that the adversary will not have access to the full decryption oracle but rather an oracle that will simply output whether or not a ciphertext is legal or illegal (a judge oracle).

Given a cryptosystem $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and a public key (pk, sk) generated by $\mathcal{K}$, define the judge oracle as follows:

- Oracle $\mathcal{J}_{sk}^{\Pi}(y)$ outputs Illegal if $\mathcal{D}_{sk}(y) = \perp$; else it will output Legal.

The strength of ICA will be determined by how much access an adversary has to the judge oracle. That is, we can formulate IND-ICA1 and IND-ICA2 in the same way that we formulated IND-CCAi. In the former attack, the adversary has access to the oracle $\mathcal{J}_{sk}^{\Pi}$ as it determines the two messages x_0, x_1 it will have to distinguish. In the latter, the adversary has at least this much access to the oracle, but also access to the oracle after the challenge bit b has been chosen randomly and $y \leftarrow \mathcal{E}_{pk}(x_b)$ has been generated.

Definition 4.1 (IND-ICA1, IND-ICA2) *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $A = (A_1, A_2)$ be an adversary. For atk $\in$ {ica1,ica2}, and $k \in \mathbb{N}$ let* $\mathsf{Adv}_{A,\Pi}^{ind\text{-}atk}(k) =$

$$2 \cdot \Pr[(sk, pk) \leftarrow \mathcal{K}(1^k);\ (x_0, x_1, s) \leftarrow A_1^{\mathcal{O}_1};\ b \leftarrow \{0,1\};\ y \leftarrow \mathcal{E}_{pk}(x_b) :\ A_2^{\mathcal{O}_2}(x_0, x_1, s, y) = b] - 1 \tag{4.1}$$

where

- *If atk = ica1, then* $\mathcal{O}_1(\cdot) = \mathcal{J}_{sk}^{\Pi}(\cdot)$ *and* $\mathcal{O}_2(\cdot) = \varepsilon$.
- *If atk = ica2, then* $\mathcal{O}_1(\cdot) = \mathcal{J}_{sk}^{\Pi}(\cdot)$ *and* $\mathcal{O}_2(\cdot) = \mathcal{J}_{sk}^{\Pi}(\cdot)$.

We insist, above, that A_1 outputs x_0, x_1 with $|x_0| = |x_1|$. We say that Π is secure in the sense of IND-ATK if A being polynomial time implies that $\mathsf{Adv}_{A,\Pi}^{\text{ind-atk}}(\cdot)$ is negligible. Note that here we make no restrictions on what queries A_2 might make to $\mathcal{J}_{sk}^{\Pi}$.

4.2 Relating ICA to Other Notions of Security

Given this new definition, we expect that it fits logically within the framework of our existing definitions. For instance, if Π is secure in the sense of IND-ICAi, then it is also secure in the sense of IND-CPA; we would also expect that if Π is secure in the sense of IND-ICA2, then it is also secure in the sense of IND-ICA1.

We would also eventually like to formalize what the Bleichenbacher attack against PCKS#1 has shown in practice, that security in the sense of IND-CPA does not imply security in the sense of IND-ICA2 (see Corollary 4.2). Note that the Bleichenbacher attack cannot serve as a formal result, because PCKS#1 is not known to be secure in the sense of IND-CPA. In the following sections, we will use new results and those of [2] to construct a more complete picture; we will find either implications (security in the sense of Definition A implies security in the sense of Definition B) or separations (there exist cryptosystems that are secure in the sense of Definition A but are not secure in the sense of Definition B) between ICAi, CCAi and CPA.

4.2.1 Results from [2]

Because [2] does not use the Legality Criterion discussed in Section 2.1, we must approach the results of that paper as either those that conform to the Legality Criterion, and those that do not.

Results for Definition 2.1 and 2.2 The following results from [2] are true whether or not the Legality Criterion in enforced. They are trivial results, which are assumed without proof.

Theorem 4.1 (IND-CCA2 ⇒ IND-CCA1) *If a cryptosystem* Π *is secure in the sense of IND-CCA2, then it is secure in the sense of IND-CCA1.*

Theorem 4.2 (IND-CCA1 ⇒ IND-CPA) *If a cryptosystem* Π *is secure in the sense of IND-CCA1, then it is secure in the sense of IND-CPA.*

A Result for Definition 2.1 Only

Theorem 4.3 (IND-CPA ⇏ IND-CCA1) *There exist cryptosystems that are secure in the sense of IND-CPA but are not secure in the sense of IND-CCA1. (Theorem 3.6 from [2])*

As discussed in Section 2.1, the proof of the Theorem 4.3 uses "legal" encryptions that never could be output by the encryption algorithm in question. Thus, the proof is invalid for cryptosystems following Definition 2.2. This result has technically been shown through practical instantiation. As mentioned above, the Blum-Goldwasser Cryptosystem is secure in the sense of IND-CPA [6] but susceptible to IND-CCA1 attack [19]. Such a result, however, depends upon the Factoring Intractability Assumption and would not hold if an efficient means of factoring were discovered.

Rather than using existing "provably secure" cryptosystems to prove whether or not one notion of security implies another, this thesis will use a more theoretical approach, such as that used in [2]. With this approach, it is possible to prove implications or separations by making simpler assumptions: for instance, that a scheme secure in the sense of IND-CPA exists. It is left as an open question whether or not Theorem 4.3 holds with Definition 2.2.

Theorem 4.3 with Definition 2.1 is also proven in this thesis using an approach different from that of [2]. See Corollary 4.4.

4.2.2 Trivial Notions

It is clear to see that full access to the judge oracle is more useful in an attack than is partial access, and similarly, partial access to the oracle is better than no access at all. To this effect, it is trivial to show the following two theorems, so the proofs are omitted:

Theorem 4.4 (IND-ICA2 $\Rightarrow$ IND-ICA1) *If* Π *is secure in the sense of IND-ICA2, then it is also secure in the sense of IND-ICA1.*

Theorem 4.5 (IND-ICA1 $\Rightarrow$ IND-CPA) *If* Π *is secure in the sense of IND-ICA1, then it is also secure in the sense of IND-CPA.*

By necessity, the judge oracle is weaker than is the complete decryption oracle. For an adversary attacking a cryptosystem, the information the decryption oracle reveals should be a superset of the information that the judge oracle reveals. This follows directly from the construction of the judge oracle.

Theorem 4.6 (IND-CCAi $\Rightarrow$ IND-ICAi for $i \in \{1, 2\}$) *If a cryptosystem* Π *is secure in the sense of IND-CCA2, then it is also secure in the sense of IND-ICA2. Similarly, if a cryptosystem* Π *is secure in the sense of IND-CCA1, then it is also secure in the sense of IND-ICA1.*

4.3 New Results

We now offer four new theorems that are not trivial notions. They will provide the information necessary to complete the relationships between ICA and other notions of security.

4.3.1 IND-CCA1 $\not\Rightarrow$ IND-ICA2

Theorem 4.7 (IND-CCA1 $\not\Rightarrow$ IND-ICA2) *There exist cryptosystems that meet the Legality Criterion of Definition 2.2, that are secure in the sense of IND-CCA1 but are not secure in the sense of IND-ICA2.*

Proof. We make the minimal assumption that there exists a cryptosystem $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ that is secure in the sense of IND-CCA1 and meets the semantical requirements of Definition 2.2. We construct a new cryptosystem $\Pi' = (\mathcal{K}, \mathcal{E}', \mathcal{D}')$ as follows. Given a security parameter k and a keypair $(pk, sk) \leftarrow \mathcal{K}(1^k)$:

1. Algorithm $\mathcal{E}'_{pk}(x)$
   (a) $R_1 \leftarrow \{0,1\}^k$
   (b) If $R_1 = 0$, then set $b = x[0]$, the 0th bit of x; else, $b \leftarrow \{0,1\}$
   (c) Output $(R_1 \parallel b \parallel \mathcal{E}_{pk}(x))$
2. Algorithm $\mathcal{D}'_{sk}(y)$
   (a) Split $(A \parallel b \parallel z) \leftarrow y$, where b is a single bit
   (b) $x \leftarrow \mathcal{D}_{sk}(z)$
   (c) If $A = 0$ and $b \neq x[0]$, output $\perp$; else, output x

This cryptosystem clearly meets all the semantical requirements given by Definition 2.1. The proof will be completed when the following claims are proven:

Claim 4.1 Π' *meets Definition 2.2 (the Legality Criterion).*

Claim 4.2 Π' *is secure in the sense of IND-CCA1.*

Claim 4.3 Π' *is not secure in the sense of IND-ICA2.*

Proof of Claim 4.1 Consider all of the possible y for which $\mathcal{D}'_{sk}(y) = \perp$. Assume that all ciphertexts are of the form, $y = (A \parallel b \parallel z)$.

Case 1. $A \neq 0$ and $\mathcal{D}_{sk}(z) = \perp$. $\mathcal{D}'_{sk}(y)$ will output $\perp$ by the "else" clause of Step 2c. However, since we are given that Π meets Definition 2.2, it follows that with very good probability,[7] for all x, $z \notin \mathcal{E}_{pk}(x)$. Thus, by Step 1c, it follows that with very good probability, for all x, $y \notin \mathcal{E}'_{pk}(x)$.

Case 2. $A = 0$ and $b \neq x[0]$, where $x = \mathcal{D}_{sk}(z)$. In this case, $\mathcal{D}'_{sk}(y)$ will output $\perp$ by the "if" clause of Step 2c. However, it is clear from Step 1b that these conditions will not be true for any y output by $\mathcal{E}'_{pk}$.

Thus, we have shown that with very good probability, for all y output by $\mathcal{E}_{pk}$, that $\mathcal{D}_{sk}(y) \neq \perp$. We can conclude that Π' meets the requirements of Definition 2.2. ■

[7] By very good probability, we wean $1 - \epsilon(k)$, where k is the security parameter and $\epsilon(\cdot)$ is negligible.

Proof of Claim 4.2 The intuition behind this construction is that in very rare situations, the modified encryption box $\mathcal{E}'_{pk}$ will give away important information about the plaintext it is to encrypt. In all other situations, the modified encryption box will encrypt regularly, by some box whose IND-CCA1 security has already been established.

More formally, consider any adversary $B = (B_1^{\mathcal{D}'_{sk}}, B_2)$, that attacks Π' in the sense of IND-CCA1, with advantage given by $\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k)$. Recall that $\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k)$ is defined as

$$2 \cdot \Pr[(pk, sk) \leftarrow \mathcal{K}(1^k);\ (x_0, x_1, s) \leftarrow B_1^{\mathcal{D}'_{sk}}(pk);\ b \leftarrow \{0,1\};\ y \leftarrow \mathcal{E}'_{pk}(x_b) : B_2(x_0, x_1, s, y) = b] - 1 \tag{4.2}$$

We now construct an adversary $A = (A_1^{\mathcal{D}_{sk}}, A_2)$ that calls upon B and attacks Π in the sense of IND-CCA1. Since Π is assumed to be secure in the sense of IND-CCA1, the advantage A has in attacking it must be negligible. We will show that $\mathsf{Adv}^{\text{ind-cca1}}_{A,\Pi}(k)$ is an upper bound for $\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k)$, and thus $\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k)$ must also be negligible in k.

The adversary A is defined as follows:

1. Algorithm $A_1^{\mathcal{D}_{sk}}(pk)$
   (a) Run $(x_0, x_1, s) \leftarrow B_1^{\mathcal{D}'_{sk}}(pk)$. Whenever B_1 submits z to $\mathcal{D}'_{sk}$, simulate $\mathcal{D}'_{sk}(z)$ as follows:
      i. Split $(R \parallel b \parallel y) \leftarrow z$, where $|b| = 1$
      ii. Run $x \leftarrow \mathcal{D}_{sk}(y)$
      iii. If $R = 0$ and $x[0] \neq b$, then output $\perp$
      iv. Else output x
   (b) Output (x_0, x_1, s)
2. Algorithm $A_2(x_0, x_1, s, y)$
   (a) Pick $R_1 \leftarrow \{0,1\}^k$ at random
   (b) Pick $b' \leftarrow \{0,1\}$ at random
   (c) Run $b \leftarrow B_2(x_0, x_1, s, (R_1 \parallel b' \parallel y))$
   (d) Output b

It is clear to see that if B is a pair of probabilistic polynomial time turning machines (PPT), then A is PPT as well, for A_1 and A_2 call on B constantly many times.

It is also clear to see that A_1 will always simulate $\mathcal{D}'_{sk}$ successfully. This is a simple result of the construction of $\mathcal{D}'_{sk}$ and Step 1a of the algorithm.

Now it remains to show that B succeeds with negligible probability. We put an upper bound on $\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k)$ that will be closely related to $\mathsf{Adv}^{\text{ind-cca1}}_{A,\Pi}(k)$. First, we notice that in most cases, running the IND-CCA1 experiment on Π using the adversary A is nearly indistinguishable from running the IND-CCA1 experiment on Π' using the adversary B. The only difference is that, on average, 1 out of every 2^{k+1} encryptions A_2 submits to B_2 will be an illegal encryption, one that never could have been generated by $\mathcal{E}_{pk}$. After all, if B_2 chooses $R_1 = 0$, then it must also pick a b' such that it equals the first bit of $\mathcal{D}_{sk}(y)$. Since we are trying to put an upper bound on the probability that A_2 will submit

an illegal encryption to B_2, we can just assume that should such a case in which $R_1 = 0$ arise, A_2 will always pick the incorrect b'. Taking probability over B_2's random coin flips, $\Pr[R_1 = 0] = 2^{-k}$, thus

$$\Pr[A_2 \text{ encypts illegally}] \leq 2^{-k} \tag{4.3}$$

With the exception of these very rare cases, running the CCA1 experiment on Π with the adversary A will be statistically indistinguishable from running the CCA1 experiment on Π' with the adversary B. This is a direct result of the construction of Π' and A. To facilitate our analysis, we consider the following events:

- Legal is the event in which A_2 submits a legal encryption to B_2. Note that from Equation (4.3), $\Pr[\mathsf{Legal}] \geq 1 - 2^{-k}$.
- Succ is the event in which B_2 is successful in guessing the encryption. Note that $\Pr[\mathsf{Succ} \mid \mathsf{Legal}] = \mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k)$ by the definition of $\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k)$. It follows from the construction of A that A succeeds only when B succeeds. Hence, $\mathsf{Adv}^{\text{ind-cca1}}_{A,\Pi}(k) = \Pr[\mathsf{Succ}]$.

By conditioning:

$$\begin{aligned}
\mathsf{Adv}^{\text{ind-cca1}}_{A,\Pi}(k) &= \Pr[\mathsf{Succ}] \\
&= \Pr[\mathsf{Succ} \mid \mathsf{Legal}] \cdot \Pr[\mathsf{Legal}] + \Pr[\mathsf{Succ} \mid \neg\mathsf{Legal}] \cdot \Pr[\neg\mathsf{Legal}] \\
&\geq \Pr[\mathsf{Succ} \mid \mathsf{Legal}] \cdot \Pr[\mathsf{Legal}]
\end{aligned} \tag{4.4}$$

Now, we use the definitions of Legal and Succ to show that:

$$\begin{aligned}
\mathsf{Adv}^{\text{ind-cca1}}_{A,\Pi}(k) &\geq (1 - 2^{-k}) \Pr[\mathsf{Succ} \mid \mathsf{Legal}] \\
&\geq (1 - 2^{-k}) \mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k)
\end{aligned} \tag{4.5}$$

Rearranging, we have that:

$$\frac{\mathsf{Adv}^{\text{ind-cca1}}_{A,\Pi}(k)}{1 - 2^{-k}} \geq \mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k) \tag{4.6}$$

Since Π is secure in the sense of IND-CCA1, it follows that $\mathsf{Adv}^{\text{ind-cca1}}_{A,\Pi}(k)$ is negligible in k. Thus, the left hand side of Equation (4.6) is negligible in k, and finally, $\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k)$ is negligible in k. Since we have shown that an arbitrary adversary B cannot defeat Π' in the sense of IND-CCA1, it follows that Π' is secure in the sense of IND-CCA1. ■

Proof of Claim 4.3 Construct an adversary $A = (A_1, A_2^{\mathcal{J}_{sk}^{\Pi'}})$. A_1 will always generate the same two messages, the bits 0 and 1. Thus, $(0, 1, \varepsilon)$ will be output by $A_1(pk)$, for any pk. A_2 is defined as follows:

- $\mathsf{Algorithm}\ A_2^{\mathcal{J}_{sk}^{\Pi'}}(x_0, x_1, y, \varepsilon)$
  1. $z \leftarrow (0 \parallel 0 \parallel y)$
  2. If $\mathcal{J}_{sk}^{\Pi'}(z)$ outputs Legal, then output 0; else output 1

It is clear that $A_2^{\mathcal{J}_{sk}^{\Pi'}}$ will always be successful in guessing b such that $\mathcal{D}_{sk}(y) = x_b$. In the case that $b = 0$ and $x_b = x_0 = 0$, it follows that $x_b[0] = 0$. Thus the encryption z will be a legal encryption, and the judge oracle will output Legal. In the other case, $x_b[0] = 1$, and the encryption z will be an illegal encryption. Hence, the judge oracle will output Illegal. ■

Consequently, the following corollaries hold with either Definition 2.1 or 2.2:

Corollary 4.1 (IND-ICA1 $\not\Rightarrow$ IND-ICA2) *There exist cryptosystems that are secure in the sense of IND-ICA1 but are not secure in the sense of IND-ICA2.*

Proof. Assume for the purposes of contradiction that all cryptosystems Π that are secure in the sense of IND-ICA1 are also secure in the sense of IND-ICA2. By Theorem 4.6, we have that Π secure in the sense of IND-CCA1 implies that Π is secure in the sense of IND-ICA1, and therefore is secure in the sense of IND-ICA2. But this contradicts Theorem 4.7, so our assumption must be false. ■

Corollary 4.2 (IND-CPA $\not\Rightarrow$ IND-ICA2) *There exist cryptosystems that are secure in the sense of IND-CPA but are not secure in the sense of IND-ICA2.*

Proof. Assume that this is not the case. Then all cryptosystems Π that are secure in the sense of IND-CPA are also secure in the sense of IND-ICA2. Hence, by Theorems 4.4 and 4.5, it follows that the three definitions IND-CPA, IND-ICA1 and IND-ICA2 are equivalent. This contradicts Corollary 4.1, thus our assumption is false. ■

Corollary 4.3 (IND-CCA1 $\not\Rightarrow$ IND-CCA2) *There exist cryptosystems that are secure in the sense of IND-CCA1 but are not secure in the sense of IND-CCA2.*

Proof. If this were not the case, then we would have IND-CCA1 $\Rightarrow$ IND-CCA2, and by Theorem 4.6, we have that IND-CCA2 $\Rightarrow$ IND-ICA2. Hence, we have IND-CCA1 $\Rightarrow$ IND-ICA2, which contradicts Theorem 4.7. ■

Note that [2] also proves this theorem, but for Definition 2.1 only. Our corollary holds whether or not the Legality Criterion is enforced.

4.3.2 IND-ICA2 $\not\Rightarrow$ IND-CCA2

Theorem 4.8 (IND-ICA2 $\not\Rightarrow$ IND-CCA2) *There exist cryptosystems that meet the Legality Criterion of Definition 2.2, that are secure in the sense of IND-ICA2 but are not secure in the sense of IND-CCA2.*

As usual, a cryptosystem $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ that meets the Legality Criterion and is secure in the sense of IND-ICA2 is assumed to exist. Our challenge now is to construct a cryptosystem whose decryption box will give away information related to a ciphertext and will not give away information on the basis of legal/illegal queries. To this effect, construct the cryptosystem $\Pi' = (\mathcal{K}, \mathcal{E}', \mathcal{D}')$ as follows: [8]

1. Algorithm $\mathcal{E}'_{pk}(x)$

[8] Note that $\overline{x}$ refers to the bitwise negation of x.

(a) $R \leftarrow \{0,1\}^{k+1}$

(b) If $R = 0$, then $b \leftarrow 0$; else $b \leftarrow 1$

(c) If $b = 0$, then $x' \leftarrow \overline{x}$; else $x' \leftarrow x$

(d) Output $(b \parallel \mathcal{E}_{pk}(x'))$

2. Algorithm $\mathcal{D}'_{sk}(y)$

(a) Split $(b \parallel z) \leftarrow y$

(b) $x \leftarrow \mathcal{D}_{sk}(z)$

(c) If $x = \perp$, then output $\perp$

(d) If $b = 1$, then output x; else output $\overline{x}$

We first argue that Π' meets the Legality Criterion. The decryption algorithm $\mathcal{D}'_{sk}$ can only output $\perp$ at Step 2c. Thus, in order for $\mathcal{D}'_{sk}(y)$ to output $\perp$, $\mathcal{D}_{sk}(z)$ must output $\perp$. Because Π meets the Legality Criterion, $\mathcal{D}_{sk}(z)$ will, with very good probability, only output $\perp$ if for all x, $z \notin \mathcal{E}_{pk}(x)$. This condition is exactly equivalent to $\forall x,\ y \notin \mathcal{E}'_{pk}(x)$, because $\mathcal{E}'_{pk}$ will not alter the legality of encryptions. Therefore, with very good probability, $\mathcal{D}'_{sk}(y)$ will output $\perp$ if and only if for all x, $y \notin \mathcal{E}_{pk}(x)$. Hence, the Legality Criterion is satisfied.

The intuition behind our construction is rather simple. The modified cryptosystem, Π', does not change the legality of messages of the underlying cryptosystem, Π. Thus, as long as Π is secure in the sense of IND-ICA2, we can expect Π' to be secure in this manner as well. However, it is easy to modify a given encryption y, yielding a new encryption y' such that $\mathcal{D}'_{sk}(y)$ and $\mathcal{D}'_{sk}(y')$ are closely related. That is:

Claim 4.4 *Π' is secure in the sense of IND-ICA2.*

Claim 4.5 *Π' is not secure in the sense of IND-CCA2.*

Proof of Claim 4.4 Given any polynomial-time adversary $B = (B_1^{\mathcal{J}_{sk}^{\Pi'}}, B_2^{\mathcal{J}_{sk}^{\Pi'}})$ that defeats Π' in the sense of IND-ICA2 with an advantage given by $\mathsf{Adv}^{\text{ind-ica2}}_{B,\Pi'}(k)$. Our goal will be to show that $\mathsf{Adv}^{\text{ind-ica2}}_{B,\Pi'}(k)$ is negligible in k.

Note that in this notation, the oracle $\mathcal{J}_{sk}^{\Pi'}$ is based upon the decryption box $\mathcal{D}'_{sk}$. We now construct adversary $A = (A_1^{\mathcal{J}_{sk}^{\Pi}}, A_2^{\mathcal{J}_{sk}^{\Pi}})$ which attacks Π in the sense of IND-ICA2. The oracle $\mathcal{J}_{sk}^{\Pi}$ is similarly based upon the decryption box $\mathcal{D}_{sk}$.

- Algorithm $A_1^{\mathcal{J}_{sk}^{\Pi}}(pk)$

  1. Run $(x_0, x_1, s) \leftarrow B_1^{\mathcal{J}_{sk}^{\Pi'}}(pk)$
     (a) Whenever B_1 submits y to $\mathcal{J}_{sk}^{\Pi'}$, return $\mathcal{J}_{sk}^{\Pi}(y)$
  2. Output (x_0, x_1, s)

- Algorithm $A_2^{\mathcal{J}_{sk}^{\Pi}}(x_0, x_1, y, s)$

  1. Set b, a bit, equal to 0

2. $y' \leftarrow (b \parallel y)$
3. Run $b' \leftarrow B_2^{\mathcal{J}_{sk}^{\Pi'}}(x_0, x_1, y', s)$
   (a) Whenever B_2 submits z to $\mathcal{J}_{sk}^{\Pi'}$, return $\mathcal{J}_{sk}^{\Pi}(z)$
4. Output b'

First, it is seen that A can perfectly simulate $\mathcal{J}_{sk}^{\Pi'}$ by calling upon $\mathcal{J}_{sk}^{\Pi}$. By construction of Π', y is such that $\mathcal{D}_{sk}(y) = \perp$ if and only if $\mathcal{D}'_{sk}(b \parallel y) = \perp$, where $b = 0$ or $b = 1$.

We now consider two different experiments. First,

$$\begin{aligned}\mathsf{Exp} = [&(pk, sk) \leftarrow \mathcal{K}(1^k);\ (x_0, x_1, s) \leftarrow A_1^{\mathcal{J}_{sk}^{\Pi}}(pk);\ b \leftarrow \{0,1\};\ y \leftarrow \mathcal{E}_{pk}(x_b) : \\ &A_2^{\mathcal{J}_{sk}^{\Pi}}(x_0, x_1, y, s) = b]\end{aligned} \tag{4.7}$$

By definition $\mathsf{Adv}^{\text{ind-ica2}}_{A,\Pi}(k) = 2 \cdot \Pr[\mathsf{Exp}] - 1$. Compare this experiment to:

$$\begin{aligned}\mathsf{Exp}' = [&(pk, sk) \leftarrow \mathcal{K}(1^k);\ (x_0, x_1, s) \leftarrow B_1^{\mathcal{J}_{sk}^{\Pi'}}(pk);\ b \leftarrow \{0,1\};\ y \leftarrow \mathcal{E}'_{pk}(x_b) : \\ &B_2^{\mathcal{J}_{sk}^{\Pi'}}(x_0, x_1, y, s) = b]\end{aligned} \tag{4.8}$$

Similarly, $\mathsf{Adv}^{\text{ind-ica2}}_{B,\Pi'}(k) = 2 \cdot \Pr[\mathsf{Exp}'] - 1$. The two experiments, given by Exp and Exp' will be identical, except when $\mathcal{E}'_{pk}(x_b)$ outputs an encryption of the form $(1 \parallel y)$. Call this event RevEnc, which will happen with probability 2^{-k-1} by construction of $\mathcal{E}'_{pk}$. Thus we have that:

$$\Pr[\mathsf{Exp}] = \Pr[\mathsf{Exp}' \mid \neg\mathsf{RevEnc}] \tag{4.9}$$

By conditioning, we argue that:

$$\begin{aligned}\Pr[\mathsf{Exp}'] &= \Pr[\neg\mathsf{RevEnc}] \cdot \Pr[\mathsf{Exp}' \mid \neg\mathsf{RevEnc}] + \Pr[\mathsf{RevEnc}] \cdot \Pr[\mathsf{Exp}' \mid \mathsf{RevEnc}] \\ &\leq \Pr[\neg\mathsf{RevEnc}] \cdot \Pr[\mathsf{Exp}' \mid \neg\mathsf{RevEnc}] + \Pr[\mathsf{RevEnc}]\end{aligned} \tag{4.10}$$

We deduced above that $\Pr[\mathsf{RevEnc}] = 2^{-k-1}$. Thus:

$$\Pr[\mathsf{Exp}'] \leq \left(1 - 2^{-k-1}\right) \cdot \Pr[\mathsf{Exp}' \mid \neg\mathsf{RevEnc}] + 2^{-k-1} \tag{4.11}$$

Rearranging terms:

$$\Pr[\mathsf{Exp}' \mid \neg\mathsf{RevEnc}] \geq \frac{\Pr[\mathsf{Exp}'] - 2^{-k-1}}{1 - 2^{-k-1}} \tag{4.12}$$

Because $k > 0$, it follows that $(1 - 2^{-k-1}) < 1$. Hence,

$$\Pr[\mathsf{Exp}' \mid \neg\mathsf{RevEnc}] \geq \Pr[\mathsf{Exp}'] - 2^{-k-1} \tag{4.13}$$

Combining Equations (4.9) and (4.13), we have that:

$$\Pr[\mathsf{Exp}] \geq \Pr[\mathsf{Exp}'] - 2^{-k-1} \tag{4.14}$$

Multiplying both sides by 2, and subtracting 1:

$$2 \cdot \Pr[\mathsf{Exp}] - 1 \geq (2 \cdot \Pr[\mathsf{Exp}'] - 1) - 2^{-k} \tag{4.15}$$

We now apply the definition of Exp and Exp' and rearrange terms:

$$\mathsf{Adv}^{\text{ind-ica2}}_{A,\Pi}(k) + 2^{-k} \geq \mathsf{Adv}^{\text{ind-ica2}}_{B,\Pi'}(k) \tag{4.16}$$

Since Π is secure in the sense of IND-ICA2 by assumption, $\mathsf{Adv}^{\text{ind-ica2}}_{A,\Pi}(k)$ is negligible in k. This implies that the left hand side of Equation (4.16) is negligible in k, and therefore, $\mathsf{Adv}^{\text{ind-ica2}}_{B,\Pi'}(k)$ is negligible in k. We have thus shown that an arbitrary adversary B has negligible advantage in IND-ICA2 attacking Π', which proves the claim. ■

Proof of Claim 4.5 Consider the following adversary $A = (A_1, A_2)$:

- Algorithm $A_1(pk)$
  1. $x_0, x_1 \leftarrow \{0,1\}^k$ at random
  2. Repeat above until $x_0 \neq x_1$ and $\overline{x_0} \neq x_1$
  3. Output (x_0, x_1, ε)
- Algorithm $A_2^{\mathcal{D}'_{sk}}(x_0, x_1, y, \varepsilon)$
  1. Split $(b \parallel z) \leftarrow y$ where $|b| = 1$
  2. $x' \leftarrow \mathcal{D}'_{sk}(\overline{b} \parallel z)$
  3. $x' \leftarrow \overline{x'}$
  4. If $x' = x_0$, then output 0; else output 1

Assume that the adversary A_2 is given y such that $y = (b \parallel z)$, where $|b| = 1$. There are two obvious cases to consider. In the first, $b = 1$. That is, y is such that $\mathcal{D}_{sk}(z) = \mathcal{D}'_{sk}(y)$. However, in this case, $y' = (0 \parallel z)$ is submitted to the decryption oracle, whose decryption will be $\overline{\mathcal{D}'_{sk}(y)}$. Thus, by bitwise-reversing the output of the decryption oracle, A_2 will retrieve the decryption of y without having explicitly submitted y to the decryption oracle, and without having received a plaintext of the form x_b from the decryption oracle. As we will see later, A_2 has thus far been successful in an IND-CCA2 attack, both in the sense of [2], and the improved definition of IND-CCA2 offered later in this thesis (see Definition 5.1).

In the other case, $b = 0$, and y is such that $\mathcal{D}_{sk}(z) = \overline{\mathcal{D}'_{sk}(y)}$. Thus, when $y' = (1 \parallel z)$ is submitted to the decryption oracle, it will return $\overline{\mathcal{D}'_{sk}(y)}$. Again, A_2 need only bitwise reverse the output of the decryption oracle and it will retrieve the decryption of y.

A will therefore be close to 100% effective in passing its challenge, and it is expected that A will run in time polynomial in the size of its inputs. This concludes the proof of the claim. ■

4.3.3 IND-CPA $\not\Rightarrow$ IND-ICA1 By Definition 2.1

If cryptosystems are not restricted to the Legality Criterion defined earlier, the following holds:

Theorem 4.9 *[IND-CPA $\not\Rightarrow$ IND-ICA1] There exist cryptosystems that are secure in the sense of IND-CPA but are not secure in the sense of IND-ICA1.*

As before, assume that there exists $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, a cryptosystem that is secure in the sense of IND-CPA. We now construct $\Pi' = (\mathcal{K}, \mathcal{E}', \mathcal{D}')$ as follows:

- Algorithm $\mathcal{E}'_{pk}(x)$
  1. $b_2 \leftarrow \{0, 1\}$
  2. $b_1 \leftarrow 0$
  3. Output $(b_1 \parallel b_2 \parallel \mathcal{E}_{pk}(x))$, where $|b_1| = |b_2| = 1$
- Algorithm $\mathcal{D}'_{sk}(y)$
  1. $(b_1 \parallel b_2 \parallel z) \leftarrow y$
  2. If $b_1 = 0$, then output $\mathcal{D}_{sk}(z)$
  3. Else if $z \leq |sk|$ and $b_2 = \mathbf{sk}[z]$ (that is if b_2 is equal to the zth bit of sk), then output $\mathcal{D}_{sk}(z)$
  4. Else output $\perp$

This cryptosystem does not meet the Legality Criterion, as some ciphertexts whose first bits are "1" are not rejected by the decryption box, although the encryption box would clearly never produce such encryptions. Using this construction, we can now prove the following claims:

Claim 4.6 *Π' is secure in the sense of IND-CPA.*

Claim 4.7 *Π' is not secure in the sense of IND-ICA1.*

Proof of Claim 4.6 Assume that we are given $B = (B_1, B_2)$, an adversary that attacks Π' in the sense of IND-CPA. Then we construct an adversary $A = (A_1, A_2)$ that attacks Π in the same sense with the same success rate, thereby proving that $\mathsf{Adv}^{\text{ind-cpa}}_{A,\Pi}(k) = \mathsf{Adv}^{\text{ind-cpa}}_{B,\Pi'}(k)$, and $\mathsf{Adv}^{\text{ind-cpa}}_{B,\Pi'}(k)$ must be negligible.

The construction is simple. $A_1 = B_1$, and define A_2 as follows:

- Algorithm $A_2(x_0, x_1, y, s)$
  1. $b_2 \leftarrow \{0, 1\}$
  2. $b_1 \leftarrow 0$
  3. $z \leftarrow (b_1 \parallel b_2 \parallel y)$
  4. Output $B_2(x_0, x_1, z, s)$

From our construction of A and of Π, it follows that running A against Π is the same experiment as running B against Π'. The only difference is that in the former case, A adds the 0-bit and the random bit to the ciphertext, while in the latter, $\mathcal{E}$ adds the 0-bit and the random bit to the ciphertext. Hence, we expect the two experiments to have the same probability of success. Since Π was assumed to be secure in the sense of IND-CPA, $\mathsf{Adv}^{\text{ind-cpa}}_{A,\Pi}(k)$ is negligible; therefore, $\mathsf{Adv}^{\text{ind-cpa}}_{B,\Pi'}(k) = \mathsf{Adv}^{\text{ind-cpa}}_{A,\Pi}(k)$ is negligible. ■

Proof of Claim 4.7 Construct an adversary $A = (A_1^{\mathcal{J}_{sk}^{\Pi'}}, A_2)$, and define A_1 as follows:

- Algorithm $A_1^{\mathcal{J}_{sk}^{\Pi'}}(pk)$
  1. $\mathbf{sk} \leftarrow 0$
  2. For $i \leftarrow 0$ to ∞:
     (a) $b_0 \leftarrow \mathcal{J}_{sk}^{\Pi'}(1 \parallel 0 \parallel i)$
     (b) $b_1 \leftarrow \mathcal{J}_{sk}^{\Pi'}(1 \parallel 1 \parallel i)$
     (c) If $b_0 = b_1 = \mathsf{Illegal}$, then break
     (d) Else if $b_0 = \mathsf{Legal}$, then $\mathbf{sk}[i] \leftarrow 0$
     (e) Else $\mathbf{sk}[i] \leftarrow 1$
  3. $x_0 \leftarrow \{0,1\}^k$
  4. $x_1 \leftarrow \{0,1\}^k$
  5. Output $(x_0, x_1, \mathbf{sk})$

This algorithm will always output the secret key sk that corresponds to pk. Hence, A_2 will always be able to perform the distinguishing task required of it. The success of A_1 is a result of the contrived construction of Π'. Every time a message is submitted to $\mathcal{D}_{sk}$ whose first bit is 1, the box $\mathcal{D}_{sk}$ will give away one bit of knowledge about the secret key. Note that implicitly, $\mathcal{D}_{sk}$ will also reveal the length of the secret key. For if $\mathcal{D}_{sk}(1 \parallel 0 \parallel i) = \mathcal{D}_{sk}(1 \parallel 1 \parallel i) = \perp$, then it follows that $i > |sk|$. This feature of the decryption box is used in Step 2c. This algorithm will make $2(l+1)$ queries to $\mathcal{J}_{sk}^{\Pi'}$, where $l = |sk|$, thus will run in time polynomial in the size of its inputs. ■

We must note, however, that if the Legality Criterion is enforced, then Theorem 4.9 does not hold. We intuitively claim that no such result holds for cryptosystems that meet this criterion. For the decryption oracle to reveal any information about the secret key through a legal/illegal query, the Legality Criterion mandates that the encryption oracle must also be privy to this information. If this were the case, then the cryptosystem could not be secure in the sense of IND-CPA, since the encryption box and the public key would reveal information about the secret key. This line of reasoning, however, is speculative. It implies that all information gained in a IND-ICA1 attack is about the secret key. A formal proof that security in the sense of IND-CPA implies security in the sense of IND-ICA1 would require a construction that simulates the judge oracle, which we leave as an open problem.

The following corollary holds only with Definition 2.1:

Corollary 4.4 (IND-CPA $\not\Rightarrow$ IND-CCA1) *There exist cryptosystems that are secure in the sense of IND-CPA but are not secure in the sense of IND-CCA1.*

Proof. This corollary follows from Theorems 4.6 and 4.9. Note that this corollary is equivalent to Theorem 3.6 of [2]. ■

4.3.4 IND-ICA2 $\not\Rightarrow$ IND-CCA1 By Definition 2.1

If cryptosystems are not restricted to the Legality Criterion defined earlier, the following holds:

Theorem 4.10 (IND-ICA2 $\not\Rightarrow$ IND-CCA1 By Definition 2.1) *There exist cryptosystems that are secure in the sense of IND-ICA2, but are not secure in the sense of IND-CCA1.*

Proof. As usual, we assume that a cryptosystem $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ exists that is secure in the sense of IND-ICA2. We now construct the cryptosystem $\Pi' = (\mathcal{K}, \mathcal{E}', \mathcal{D}')$, where:

- Algorithm $\mathcal{E}'_{pk}(x)$
  1. $b \leftarrow 0$
  2. Output $(b \parallel \mathcal{E}_{pk}(x))$
- Algorithm $\mathcal{D}'_{sk}(y)$
  1. Split $(b \parallel z) \leftarrow y$, where $|b| = 1$
  2. If $b = 1$, then output sk
  3. Else output $\mathcal{D}_{sk}(z)$

From the construction of Π', it follows that $\mathcal{D}'_{sk}$ will give away secret information that $\mathcal{J}^{\Pi'}_{sk}$ will not. For whenever an IND-ICA2 adversary submits a query of the form $(1 \parallel y)$ to $\mathcal{J}^{\Pi'}_{sk}$, it will simply respond with Legal and thus will not convey information about the secret key. We have the following two claims, which will complete the proof:

Claim 4.8 Π' *is secure in the sense of IND-ICA2.*

Claim 4.9 Π' *is not secure in the sense of IND-CCA1.*

Proof of Claim 4.8. Consider an arbitrary adversary $B = (B_1^{\mathcal{J}^{\Pi'}_{sk}}, B_2^{\mathcal{J}^{\Pi'}_{sk}})$ that attacks Π' in the sense of IND-ICA2. Clarifying our notation, $\mathcal{J}^{\Pi'}_{sk}$ is the judge oracle based on $\mathcal{D}'_{sk}$. We now construct the adversary $A = (A_1^{\mathcal{J}^{\Pi}_{sk}}, A_2^{\mathcal{J}^{\Pi}_{sk}})$ that attacks Π in the sense of IND-ICA2 as follows:

1. Algorithm $A_1^{\mathcal{J}^{\Pi}_{sk}}(pk)$
   (a) Run $(x_0, x_1, s) \leftarrow B_1^{\mathcal{J}^{\Pi'}_{sk}}(pk)$. Whenever B_1 queries $\mathcal{J}^{\Pi'}_{sk}(y)$, do the following:
      i. Split $(b \parallel z) \leftarrow y$, where $|b| = 1$
      ii. If $b = 1$, then output Legal
      iii. Else, output $\mathcal{J}^{\Pi}_{sk}(z)$
   (b) Output (x_0, x_1, s)
2. Algorithm $A_2^{\mathcal{J}^{\Pi}_{sk}}(x_0, x_1, y, s)$
   (a) Run $b \leftarrow B_2^{\mathcal{J}^{\Pi'}_{sk}}(x_0, x_1, y, s)$. Whenever B_2 queries $\mathcal{J}^{\Pi'}_{sk}(y)$, do the following:

i. Split $(c \parallel z) \leftarrow y$, where $|c| = 1$
ii. If $c = 1$, then output Legal
iii. Else, output $\mathcal{J}_{sk}^{\Pi}(z)$

(b) Output b

It is clear to see from the above construction and from the definition of $\mathcal{J}_{sk}^{\Pi'}$ that A will be able to simulate $\mathcal{J}_{sk}^{\Pi'}$ in all situations. By construction of $\mathcal{D}'_{sk}$, all encryptions that begin with the bit $b = 1$ will be legal encryptions. Hence, Steps 1(a)ii and 2(a)ii will always be accurate in simulating the behavior of $\mathcal{J}_{sk}^{\Pi'}$.

Once we have established this fact, it follows that $\mathsf{Adv}_{A,\Pi}^{\text{ind-ica2}}(k) = \mathsf{Adv}_{B,\Pi'}^{\text{ind-ica2}}(k)$, for the behavior of A attacking Π and the behavior of B attacking Π' are exactly equivalent. Since Π is assumed to be secure in the sense of IND-ICA2, $\mathsf{Adv}_{A,\Pi}^{\text{ind-ica2}}(k)$ is negligible in k. Therefore, $\mathsf{Adv}_{B,\Pi'}^{\text{ind-ica2}}(k)$ is also negligible in k, and the claim is proven. ■

Proof of Claim 4.9 Consider the adversary $A = (A_1^{\mathcal{D}'_{sk}}, A_2)$. $A_1^{\mathcal{D}'_{sk}}$ will simply pick a random number R and submit $(1 \parallel R)$ to its decryption oracle, $\mathcal{D}'_{sk}$, which will answer with sk, the secret key. Once $A_1^{\mathcal{D}'_{sk}}$ has recovered the secret key, it will clearly succeed in any attack it mounts against Π'. ■

This theorem results in the following corollary, which holds only for Definition 2.1:

Corollary 4.5 (IND-ICA1 $\not\Rightarrow$ IND-CCA1) *There exist cryptosystems that are secure in the sense of IND-ICA1 but are not secure in the sense of IND-CCA1.*

Proof. Assume that this is not the case. Then we have that security in the sense of IND-ICA1 implies security in the sense of IND-CCA1 for all cryptosystems. By Theorem 4.4, we have that security in the sense of IND-ICA2 implies security in the sense of IND-ICA1. Combining these two claims, it follows that security in the sense of IND-ICA2 implies security in the sense of IND-CCA1, which contradicts Theorem 4.10. Therefore, our assumption is false. ■

4.4 The Big Pictures

The above implications and separations can be summarized in Figures 4.1 and 4.2. In both, an arrow between Definition A and Definition B is taken to mean that security in the sense of A implies security in the sense of B. Similarly, a hatched arrow between Definition A and Definition B is taken to mean that security in the sense of A does not necessarily imply security in the sense of B. In both, the definitions for cryptosystem security are strongest in the upper right-hand corner, and weakest in the lower left-hand corner. Each number refers to the theorem within this section that justifies the implication (regular arrow) or separation (hatched arrow). All non-numbered connections are corollaries that can be deduced from the relationships summarized by the figures.

Figure 4.1 encapsulates the results associated with Definition 2.1, those cryptosystems that do not necessarily meet the Legality Criterion. Note that this is a complete picture, and all definitions are connected through either known implications or known separations.

Figure 4.2 holds for those cryptosystems that conform to the Legality Criterion. This diagram is incomplete, with open problems marked by dashed lines.

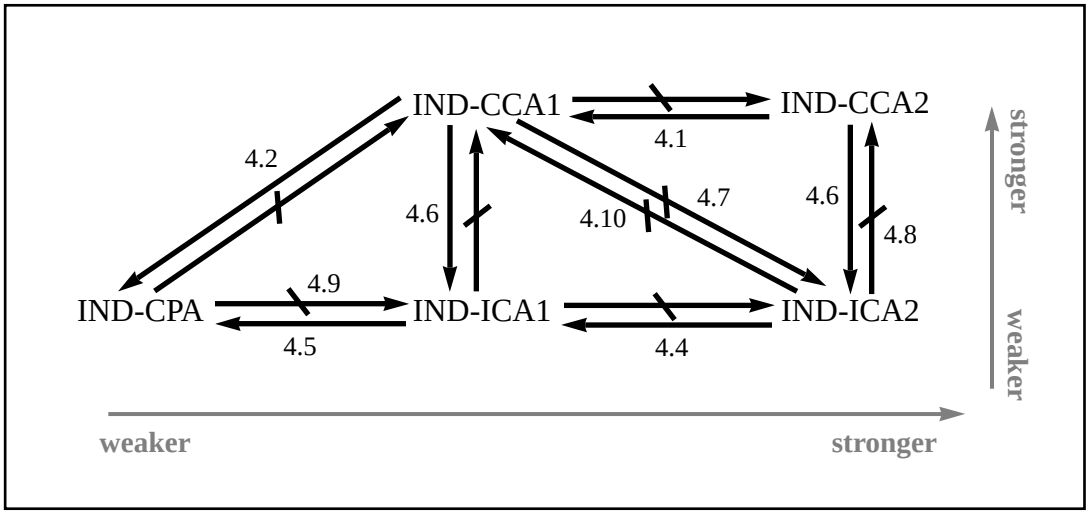

Figure 4.1: Relationships Among Definitions (Legality Criterion Enforced)

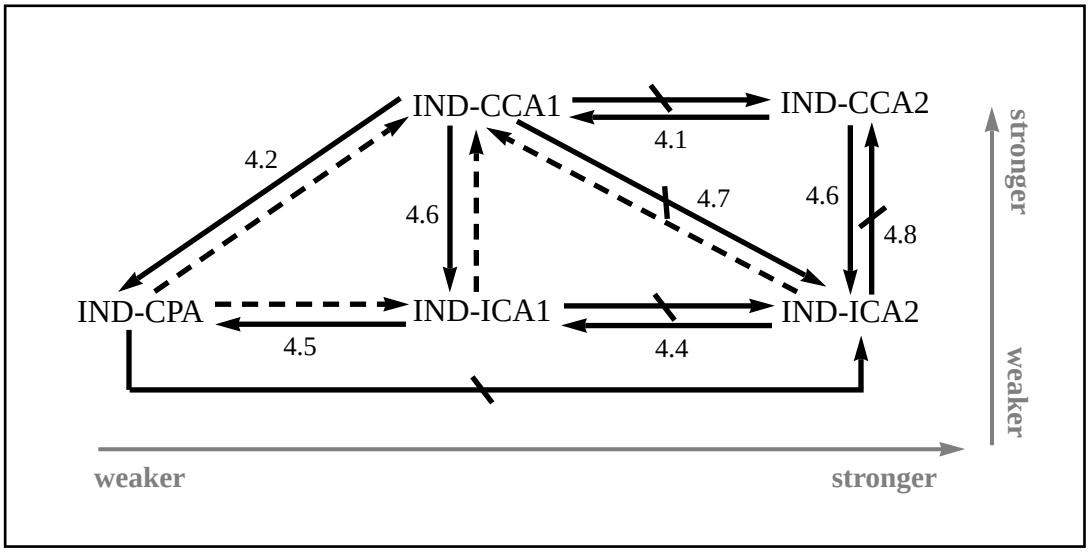

Figure 4.2: Relationships Among Definitions (Legality Criterion Not Enforced)

5 Adaptive Chosen Ciphertext Attack Revisited

Adaptive chosen ciphertext attack is such an appealing concept to cryptographers because it models one of the strongest notions of security. Although the attack that IND-CCA2 models is entirely impractical (if an adversary had access to the decryption oracle, why not just decrypt the challenge ciphertext?), if we know that a certain cryptosystem is secure in the sense of IND-CCA2, we expect it to be secure against weaker and more practical attacks. We would definitely expect that if Π is secure in the sense of IND-CCAi, then it would be secure in the sense of IND-ICAi, for in the former, an adversary would have access to the decryption oracle and in the latter that access is restricted.

It is our opinion that the existing definition for IND-CCA2 is too restrictive. In this chapter, the definition of IND-CCA2 is reformulated, so that intuitive results such as Theorem 4.6 remain.

5.1 Restrictions on Decryption Oracle Queries: A Motivating Example

Our working definition for IND-CCA2 (Definition 2.3) states that "we further insist that A_2 does not ask its oracle to decrypt y." To this effect, assume that there exists a cryptosystem $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ that is secure in the sense of IND-CCA2. Now construct a cryptosystem $\Pi' = (\mathcal{K}, \mathcal{E}', \mathcal{D}')$ as follows:

- Algorithm $\mathcal{E}'_{pk}(x)$
  1. $b \leftarrow \{0,1\}$
  2. Output $(b \parallel \mathcal{E}_{pk}(x))$
- Algorithm $\mathcal{D}'_{sk}(y)$
  1. Split $(b \parallel z) \leftarrow y$
  2. Output $\mathcal{D}_{sk}(z)$

It is clear that Π' does not reveal anything more than Π except for the random bit it appends to all encryptions. From an information theoretic standpoint, Π' reveals nothing about the plaintext or the secret key that Π does not. Our definition of IND-CCA2 should be such that Π is IND-CCA2-secure implies that Π' is IND-CCA2-secure. But this is clearly not the case. Consider the adversary $A = (A_1^{\mathcal{D}_{sk}}, A_2^{\mathcal{D}_{sk}})$, defined as follows: $A_1^{\mathcal{D}_{sk}}(pk)$ outputs two random plaintexts of equal length, x_0, x_1, combined with the empty string to form the triple (x_0, x_1, ε). $A_2^{\mathcal{D}_{sk}}(x_0, x_1, y, \varepsilon)$ will simply reverse the first bit of y to get y' and submit y' to the decryption oracle $\mathcal{D}_{sk}$. By our construction, $\mathcal{D}_{sk}(y) = \mathcal{D}_{sk}(y')$, but $y \neq y'$. Thus, A_2 will output b such that $x_b = \mathcal{D}_{sk}(y')$ and succeed in its challenge with probability 1.

One might argue that the new scheme Π' is malleable, and therefore, if we were to consider it secure in the sense of IND-CCA2, then the results from [2] that establish IND-CCA2 $\Leftrightarrow$ NM-CCA2 would be contradicted. As we will show in Chapter 6, this depends upon our definition of non-malleability. Certain canonical definitions, such as those from [10] and [11], would indeed define Π' to be non-malleable.

5.2 A Failed Definition of IND-CCA2

If we would like cryptosystems such as Π' to be considered secure against IND-CCA2, then we must restrict not what is input to the decryption oracles, but rather what is output. That is, if the A_1 in our definitional experiment outputs two candidate plaintexts x_0 and x_1, then A_2 cannot submit z to $\mathcal{D}_{sk}$ such that $x_b = \mathcal{D}_{sk}(z)$, for $b = 0$ or $b = 1$. But now an interesting question arises. The adversary no longer knows whether or not it will make an illegal query of the decryption box. By illegal query, we mean a query about a ciphertext that will decrypt to either x_0 or x_1 and will hence render the experiment useless. Note the difference between an illegal query, described here, and the illegal ciphertext described earlier, those ciphertexts that could not have been produced by the real encryption engine. Returning to the problem at hand, what happens if A_2 submits such a z in the definitional experiment? Does A fail if it makes such an error? Or is the instance of the experiment thrown out? The wording of the [2] definition does not seem to favor one interpretation over the other. We define the following events:

- Let $\mathsf{GoodOut}_A$ (Good Output) be the event in which A outputs guess correctly.
- Let $\mathsf{IllglQry}_A$ (Illegal Query) be the event in which A makes at least one illegal query to its oracle during the experiment. A submits z to $\mathcal{D}_{sk}$ such that $\mathcal{D}_{sk}(z) = x_b$ for $b = 0$ or $b = 1$.

Should we define the advantage of the adversary as:

$$\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) = \Pr[\mathsf{GoodOut}_A \mid \neg\mathsf{IllglQry}_A] \tag{5.1}$$

or should we define a failure of A as either an illegal query or an incorrect guess? That is, should we define:

$$\begin{aligned}\Pr[\neg\mathsf{GoodOut}_a] = &\Pr[\mathsf{IllglQry}_A] \\ &+ \Pr[\neg\mathsf{IllglQry}_A] \cdot \Pr[\neg\mathsf{GoodOut}_A \mid \neg\mathsf{IllglQry}_A]\end{aligned} \tag{5.2}$$

And consequently:

$$\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) = \Pr[\neg\mathsf{IllglQry}_A] \cdot \Pr[\mathsf{GoodOut}_A \mid \neg\mathsf{IllglQry}_A] - \Pr[\mathsf{IllglQry}_A] \tag{5.3}$$

The problem here is that intuition about real world attack scenarios is useless, because, as we have argued many times, there is no real-world equivalent of IND-CCA2. Rather, IND-CCA2 is an abstraction that cryptographers use to capture the strongest type of cryptographic attack. To resolve this question, we will invoke our definition of IND-ICA2. As seen in Definition 4.1, there is no problem of illegal queries in the IND-ICA2 attack model. It is clear, furthermore, that whichever definition of IND-CCA2 we decide to use, security in the sense of IND-CCA2 should imply security in the sense of IND-ICA2. Note that as we change the experiment that defines IND-CCA2, and the behavior of the decryption oracle $\mathcal{D}_{sk}$, we are potentially upsetting Theorem 4.6, which establishes IND-CCA2 $\Rightarrow$ IND-ICA2. That is, if we assume Equation (5.3), then oracle $\mathcal{D}_{sk}$ no longer provides a superset of the functionality of $\mathcal{J}^{\Pi}_{sk}$. After all, queries to the former might cause the adversary to automatically fail in her attempt to foil a cryptosystem, and no equivalent failure scenario is given by the latter oracle.

We thus proceed by assuming Equation (5.3) and proving a contradiction of the intuitive relation that should hold in any definitional system: that IND-CCA2 $\Rightarrow$ IND-ICA1.

Theorem 5.1 *Assume that there exists a cryptosystem* Π *that is plaintext aware in the sense of Definition 3.1, and that IND-CCA2 is defined using Equation (5.3). Then there exists a cryptosystem* Π' *that is not secure in the sense of IND-ICA2 but secure in the sense of IND-CCA2.*

Proof. Given a cryptosystem Π that is secure in the sense IND-CPA and Plaintext Aware (plaintext extractor K, with success rate $\lambda(k)$). Then by Theorem 4.2 of [2], Π is secure in the sense of IND-CCA2. We use the same construction as in Section 4.3.1. Recalling that construction, define $\Pi' = (\mathcal{K}, \mathcal{E}', \mathcal{D}')$:

- Algorithm $\mathcal{E}'_{pk}(x)$
  1. $R_1 \leftarrow \{0,1\}^k$
  2. If $R_1 = 0$, then set $b = x[0]$, the 0th bit of x; else, $b \leftarrow \{0,1\}$
  3. Output $(R_1 \parallel b \parallel \mathcal{E}_{pk}(x))$
- Algorithm $\mathcal{D}'_{sk}(y)$
  1. Split $(A \parallel b \parallel z) \leftarrow y$, where b is a single bit
  2. $x \leftarrow \mathcal{D}_{sk}(z)$
  3. If $A = 0$ and $b \neq x[0]$, then output $\perp$; else, output x

We already known from Claim 4.3 that Π' is not secure in the sense of IND-ICA2. This is true regardless of our definition for IND-CCA2. However, assuming the proposed definition for IND-ICA2, that of Equation (5.3), we prove of Π' the following claim:

Claim 5.1 Π' *is secure in the sense of IND-CCA2.*

Proof of Claim 5.1 The intuition behind the proof is rather simple. An IND-CCA2 adversary cannot gain any useful information about ciphertexts solely on the basis of their legality. The construction of Π' favors the type of attacks detailed in Claim 4.3, whereby an adversary takes the challenge ciphertext y and formulates related messages whose legality or non-legality according to the decryption box would reveal information about the original message y. However, assume that the adversary is using this approach to decide what the first bit of $\mathcal{D}_{sk}(y)$ is. If the adversary already has evidence about what the first bit of $\mathcal{D}_{sk}(y)$ might be, then she has no reason to gain further information from decryption queries. If she is completely unsure, then she cannot gain any information from the legality of messages related to y, because in half of the cases, she will submit legal encryptions to $\mathcal{D}_{sk}$, which will count as illegal queries, and hence she will fail the experiment according to the proposed definition.

Consider an adversary $B = (B_1, B_2)$ that attacks Π' in the sense of IND-CCA2 with non-negligible probability. We desire to construct an adversary $A = (A_1, A_2)$ that attacks Π in the sense of IND-CCA2, and to bound B's probability of success by A's. Note that we will be using the definition of IND-CCA2 given by Equation (5.3). Given B_1, B_2, construct A_1, A_2 as follows.

1. Algorithm $A_1^{H,\mathcal{D}_{sk}}(pk)$

(a) Run $(x_0, x_1, s) \leftarrow B_1^{H,\mathcal{D}'_{sk}}(pk)$

i. When $B_1^{H,\mathcal{D}'_{sk}}$ queries H with query h add $(h, H(h))$ to the hH-list, and answer with $H(h)$

ii. When $B_1^{H,\mathcal{D}'_{sk}}$ queries $\mathcal{D}'_{sk}$ with y:

A. Split $(A \parallel b \parallel z) \leftarrow y$

B. $x \leftarrow \mathcal{D}_{sk}(z)$

C. If $A = 0$ and $\mathbf{x}[0] \neq b$, then output $\perp$

D. Else output x

(b) Output $(x_0, x_1, s \parallel pk)$

2. Algorithm $A_2^{H,\mathcal{D}_{sk}}(x_0, x_1, y, s \parallel pk)$

(a) Pick $R \leftarrow \{0,1\}^k$ at random

(b) Pick $b \leftarrow \{0,1\}$ at random

(c) Run $b \leftarrow B_2^{H,\mathcal{D}'_{sk}}(x_0, x_1, (R \parallel b \parallel y), s)$

i. When $B_2^{H,\mathcal{D}'_{sk}}$ queries H with query h add $(h, H(h))$ to the hH-list and answer with $H(h)$

ii. When $B_2^{H,\mathcal{D}'_{sk}}$ queries $\mathcal{D}'_{sk}(y')$:

A. Split $(A \parallel b' \parallel z) \leftarrow y'$

B. If $(z = y)$, then output $\perp$

C. Else output $K(hH, (y), z, pk)$

(d) Output b

It is first observed that A_1 can simulate $\mathcal{D}'_{sk}$ with 100% accuracy. This is a result of the construction of the cryptosystem Π' and the adversary A_1. To determine the success of A_2 we must determine how accurately A_2 can simulate the oracle queries B_2 makes to $\mathcal{D}'_{sk}$; this simulation is represented by Steps 2(c)iiB and 2(c)iiC of the algorithm. We first consider the case of Step 2(c)iiB, where $B_2^{H,\mathcal{D}'_{sk}}$ might request an encryption related to that of the challenge ciphertext, though not exactly equal. If the challenge ciphertext is $(R \parallel b \parallel y)$ and the decryption query in this step is $(A \parallel b' \parallel y)$, B_2 is in danger of submitting an illegal query to the decryption oracle. Let c be equal to the 0th bit of $\mathcal{D}_{sk}(y)$. If $(A \neq 0)$ or if $(A = 0 \ \wedge \ b' = c)$ then $\mathcal{D}'_{sk}(A \parallel b' \parallel y) = \mathcal{D}_{sk}(y) = x_b$ for $b = 0$ or $b = 1$. This is exactly the definition of an illegal query to $\mathcal{D}'_{sk}$. However, in these cases, A_2 will simulate B_2's calls to $\mathcal{D}'_{sk}$ by simply outputting $\perp$. Thus, A_2 will be incorrectly simulating B_2's query, but B_2 would be making an illegal query anyway. In the case that $(A = 0 \ \wedge \ b' \neq c)$, then $\mathcal{D}_{sk}(A \parallel b' \parallel y) = \perp$, and Step 2(c)iiB will correctly simulate $\mathcal{D}'_{sk}$. We conclude that in the case of Step 2(c)iiB, A_2 simulates $\mathcal{D}'_{sk}$ incorrectly only when B_2 makes an illegal query.

Now Step 2(c)iiC is examined. K will not always output the correct plaintext that corresponds to its input ciphertext z. As shown in [2], we can expect that if A_2 calls on K q times, then the probability of K failing during the execution of A_2 is $q \cdot (1 - \lambda(k))$.

This analysis of A_2 leads to some conclusions about its failure rate. Summarizing the above arguments, consider the following events:

- Let BadSimulation be the event in which A_2 does not simulate $\mathcal{D}'_{sk}$ properly.

- Let $\mathsf{WrongOutput}$ be the event in which A_2 wrongly outputs $\perp$ in Step 2(c)iiB of the algorithm.
- Let $\mathsf{ExtractorError}$ be the event in which K incorrectly responds to one of the queries made to it by A_2.

These events can be related as follows:

$$\Pr[\mathsf{BadSimulation}] = \Pr[\mathsf{WrongOutput}] + \Pr[\mathsf{ExtractorError}] \tag{5.4}$$

Our reasoning about the logic of Step 2(c)iiB in the algorithm can be expressed as:

$$\Pr[\mathsf{WrongOutput}] \leq \Pr[\mathsf{IllglQry}_B] \tag{5.5}$$

That is, A_2 only outputs incorrectly if B_2 would have made an illegal query, but B_2 still might make other illegal queries that A_2 can answer legally using its plaintext extractor K.

Finally, our reasoning about the logic of Step 2(c)iiC in the algorithm can be expressed as:

$$\Pr[\mathsf{ExtractorError}] \leq q \cdot (1 - \lambda(k)) \tag{5.6}$$

By our construction of Π' and B, there are only three cases in which A will fail. The first is if A_2 produces an incorrect ciphertext in step 2c. We showed in Section 4.3.1 that this happens only with probability less than 2^{-k}. The second is if B_2 runs to completion but fails in its task of guessing the challenge ciphertext. This happens with probability less than $1 - \Pr[\neg\mathsf{IllglQry}_B] \cdot \Pr[\mathsf{GoodOut}_B \mid \neg\mathsf{IllglQry}_B]$. Finally, we will assume that if A_2 simulates $\mathcal{D}'_{sk}$ incorrectly, then B_2 will fail. This happens with probability equal to $\Pr[\mathsf{BadSimulation}]$. Thus,

$$\begin{aligned}\Pr[\neg\mathsf{GoodOut}_A \mid \neg\mathsf{IllglQry}_A] &\leq 2^{-k} + (1 - \Pr[\mathsf{IllglQry}_B] \cdot \Pr[\mathsf{GoodOut}_B \mid \neg\mathsf{IllglQry}_B]) \\ &\quad + \Pr[\mathsf{BadSimulation}] \\ &\leq 2^{-k} + (1 - \Pr[\neg\mathsf{IllglQry}_B] \cdot \Pr[\mathsf{GoodOut}_B \mid \neg\mathsf{IllglQry}_B]) \\ &\quad + \Pr[\mathsf{IllglQry}_B] + q \cdot (1 - \lambda(k))\end{aligned} \tag{5.7}$$

Notice, however, that A will never make an illegal query to its decryption oracle. Thus, we have that $\Pr[\neg\mathsf{IllglQry}_A] = 1$, and by Equation (5.3):

$$\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) = \Pr[\mathsf{GoodOut}_A \mid \neg\mathsf{IllglQry}_A] \tag{5.8}$$

We also have by simple probability that:

$$\Pr[\mathsf{GoodOut}_A \mid \neg\mathsf{IllglQry}_A] = 1 - [\neg\mathsf{GoodOut}_A \mid \neg\mathsf{IllglQry}_A] \tag{5.9}$$

Combining Equations (5.8), (5.7) and (5.9), it follows that:

$$\begin{aligned}\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) \geq{}& \Pr[\neg\mathsf{IllglQry}_B] \cdot \Pr[\mathsf{GoodOut}_B \mid \neg\mathsf{IllglQry}_B] - \Pr[\mathsf{IllglQry}_B] - 2^{-k} \\ &- q \cdot (1 - \lambda(k))\end{aligned} \tag{5.10}$$

Finally, invoking Equation (5.3) for adversary B attacking cryptosystem Π', Equation (5.10) becomes:

$$\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) \geq \mathsf{Adv}^{\text{ind-cca2}}_{B,\Pi'}(k) - 2^{-k} - q \cdot (1 - \lambda(k)) \tag{5.11}$$

Rearranging terms:

$$\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) + 2^{-k} + q \cdot (1 - \lambda(k)) \geq \mathsf{Adv}^{\text{ind-cca2}}_{B,\Pi'}(k) \tag{5.12}$$

Recall that q refers to the number of $\mathcal{D}'_{sk}$ queries made by B_2. This will be polynomially many in the security parameter k; by an argument in [2], $q \cdot (1 - \lambda(k))$ will be negligible in k. Because Π is secure in the sense of IND-CCA2, it follows that $\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k)$ is negligible in k. And clearly, 2^{-k} is negligible in k. Since the left hand side of Equation (5.12) is negligible in k, it follows that the right hand side must be negligible in k. Thus, we have shown that for an arbitrary adversary B, $\mathsf{Adv}^{\text{ind-cca2}}_{B,\Pi'}(k)$ is negligible in k. This completes the proof. ■

Note that it was necessary to make some important assumptions to reach this result. We implicitly assumed that given a definition of IND-ICA2 in the standard computational model, the equivalent definition of IND-ICA2 must hold in the Random Oracle model. We have also assumed the existence of a plaintext aware cryptosystem. These are not known to exist as defined in [2]. The result might very well hold in the standard computational model, without assuming the existence of a plaintext extractor K, but no proof is known at this time. However, this example has given us good reason to consider definitions of IND-CCA2 other than that given by Equation (5.3), namely that of Equation (5.1). This new definition is formulated naturally in the next section.

5.3 A New Definition for IND-CCA2

As shown in the previous section, in defining IND-CCA2, we do not wish the legality of an adversary's queries to affect its success rate. Rather, if an adversary makes an illegal query to the decryption oracle, it should not gain any useful information but, at the same time, should not automatically fail in its attack. The most natural way to overcome any problems in our definitions is to alter the decryption oracle so that it will never give away any information it should not be giving away. If this were the case, then the adversary could not make illegal queries.

More formally, define a new decryption oracle:[9]

- Oracle $\mathcal{S}^{x_0,x_1}_{sk}(y)$:
  1. Run $x \leftarrow \mathcal{D}_{sk}(y)$
  2. If $x = x_0$ or $x = x_1$, then output "$*$"
  3. Else output x

Given this new oracle, IND-CCA2 becomes very natural to define:

Definition 5.1 (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 any $k \in \mathbb{N}$ let* $\mathsf{Adv}^{ind\text{-}cca2}_{A,\Pi}(k) =$

$$\begin{aligned} 2 \cdot \Pr[(sk, pk) \leftarrow \mathcal{K}(1^k);\ (x_0, x_1, s) \leftarrow A_1^{\mathcal{O}_1};\ b \leftarrow \{0,1\}; \\ y \leftarrow \mathcal{E}_{pk}(x_b) :\ A_2^{\mathcal{O}_2}(x_0, x_1, s, y) = b] - 1 \end{aligned} \tag{5.13}$$

[9] $\mathcal{S}$ is for "saint", as this oracle would never do anything illegal.

$\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ *as before, but now* $\mathcal{O}_2(\cdot) = \mathcal{S}_{sk}^{x_0,x_1}(\cdot)$. *We insist, above, that* A_1 *outputs* x_0, x_1 *with* $|x_0| = |x_1|$. *We say that* Π *is secure in the sense of IND-ATK if* A *being polynomial-time implies that* $\mathsf{Adv}_{A,\Pi}^{ind\text{-}atk}(\cdot)$ *is negligible.*

With this new definition of IND-CCA2, the intuitive result given by Theorem 4.6 in the old definitional model is now easy enough to prove.

Theorem 5.2 (IND-CCA2 ⇒ IND-ICA2) *If* Π *is secure in the sense of IND-CCA2 then it is secure in the sense of IND-ICA2.*

Proof. Given any adversary $B = (B_1^{\mathcal{J}_{sk}^{\Pi}}, B_2^{\mathcal{J}_{sk}^{\Pi}})$ that attacks Π in polynomial time in the sense of IND-ICA2, we make a corresponding adversary $A = (A_1^{\mathcal{D}_{sk}}, A_2^{\mathcal{S}_{sk}})$ that attacks Π in the sense of IND-CCA2. The only challenge here is to simulate the behavior of $\mathcal{J}_{sk}^{\Pi}$ given access to $\mathcal{D}_{sk}$ or $\mathcal{S}_{sk}$, and this is trivial. Whenever a message y is submitted by B to $\mathcal{J}_{sk}^{\Pi}$, submit y to $\mathcal{D}_{sk}$ or $\mathcal{S}_{sk}$ instead. If the output is a real message or the message "$*$", then the input was a legal encryption, and $\mathcal{J}_{sk}^{\Pi}$ would have output Legal; thus output Legal. And if $\mathcal{D}_{sk}$ or $\mathcal{S}_{sk}$ returns $\perp$, then input was an illegal encryption, and $\mathcal{J}_{sk}^{\Pi}$ would have output $\mathsf{Illegal}$; thus output $\mathsf{Illegal}$. With this simple construction, the advantage of B in attacking Π in the sense of IND-ICA2 will be exactly the same as A attacking Π in the sense of IND-CCA2, and the theorem is proven. ■

6 Non-Malleability

In this section, we will make similar changes to Definition 2.4, the non-malleability definition from [2]. Again, we want to make the definition a little less rigid, to accommodate intuitively non-malleable cryptosystems disregarded by Definition 2.4. Another goal is to reformulate non-malleability so that NM-ATK implies IND-ATK, for all ATK, and furthermore, that IND-CCA2 implies NM-CCA2.

6.1 A Motivating Example

As before, assume that we a have a cryptosystem $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ that is secure in the sense of NM-ATK. As usual, we construct a very similar $\Pi' = (\mathcal{K}, \mathcal{E}', \mathcal{D}')$ as follows:

- Algorithm $\mathcal{E}'_{pk}(x)$
  1. $y \leftarrow \mathcal{E}_{pk}(x)$
  2. $b \leftarrow \{0, 1\}$
  3. Output $(b \parallel y)$
- Algorithm $\mathcal{D}'_{sk}(y)$
  1. Split $(b \parallel z) \leftarrow y$ where $|b| = 1$
  2. Output $\mathcal{D}_{sk}(z)$

We argue that intuitively, Π is secure in the sense of NM-CCA2 should imply that Π' is secure in the sense of NM-CCA2. After all, an adversary attacking Π' has only one method of attack not available to an adversary attacking Π, which is to forge different encryptions of the same plaintext. That is, given any encryption y, including the challenge encryption, an adversary attacking Π' can easily formulate y' such that $y' \neq y$ but $\mathcal{D}_{sk}(y) = \mathcal{D}_{sk}(y')$, by simply reversing the first bit of the encryption. In any real world situation, this capacity would not give the adversary an advantage. For this reason, the definitions of [10, 11] do not consider those relations R for which $R(a, b) = 1$ if and only if $a = b$.

However, under Definition 2.4 from [2], cryptosystems such as Π' are not secure in the sense of NM-CCA2. Consider the following adversaries:

- Algorithm $A_1^{\mathcal{D}_{sk}}(pk)$
  1. Set $M := \{x_0, x_1\}$, where $x_0 \neq x_1$ and $\overline{x_0} \neq x_1$
  2. Output (M, ε)
- Algorithm $A_2^{\mathcal{D}_{sk}}(M, y, \varepsilon)$
  1. Reverse the first bit of y to get y'
  2. $x \leftarrow \mathcal{D}_{sk}(y')$
  3. $z \leftarrow \mathcal{E}_{pk}(\overline{x})$
  4. Output $(R, \mathbf{y})$ such that $R(a, b) = 1$ if and only if $\overline{a} = b$, and $\mathbf{y} = (z)$

Whatever challenge ciphertext y is presented to A_2 by the umpire, A_2 will be able to successfully find its decryption by slightly modifying y to get y', and then submitting y' to the decryption oracle. Once the adversary has determined the decryption of the challenge y, coming up with a related message z is trivial. Thus, as before, we would like to change the restrictions on which queries A_2 can and cannot make to the decryption oracle, to allow cryptosystems like Π' to be considered as non-malleable.

Furthermore, we have to add a simple restriction to the types of relations that A_2 is allowed to output. If A_2 is allowed to simply output R such that $R(a, b) = 1$ if and only if $a = b$, further trivial attacks are possible. Consider the adversary $B = (B_1, B_2)$ where $B_1 = A_1$ from above and:

- Algorithm $B_2^{\mathcal{D}_{sk}}(M, y, \varepsilon)$
  1. Reverse the first bit of y to get y'
  2. $\mathbf{y} \leftarrow (y')$
  3. Output $(R, \mathbf{y})$, so that $R(a, b) = 1$ if and only if $a = b$

Whatever the challenge ciphertext y, A_2 will be able to output a y' so that $y \neq y'$ and $\mathcal{D}_{sk}(y) = \mathcal{D}_{sk}(y')$. Such an attack has real world applications in such cryptosystems as Goldwasser-Micali [15]. One can always pick a random $r \in \mathbb{Z}_{p \cdot q}$, set $s \leftarrow r^2 \pmod{p \cdot q}$, and then multiply every encrypted bit by s. The outcome is an encryption y' not equal to the original encryption y but that decrypts to the same plaintext.

In order to change our definitions of IND-CCA2 and NM-CCA2 to include cryptosystems such as Π', we will have to allow the above attack. It is hard to imagine, however, a case in which such an attack would be useful. Indeed most protocol systems that rely on non-malleability — such as online auctions — do not assume that such an attack is impossible. By lessening the restrictions of the definition, we do not consider security against this trivial attack, but might allow for certain intuitively non-malleable cryptosystems that do not meet the definition proposed in [2].

6.2 A New Definition of Non-Malleability

We now present the necessary adjustments to Definition 2.4. As before, we will define an oracle for use in CCA2 attacks that restricts possible outputs that the adversary might attain. Assume that the adversary's first algorithm outputs the message space M, from which the umpire might sample. Then the second algorithm is forbidden to receive the decryption of messages of the form $\mathcal{E}_{pk}(x)$, where $x \in M$. Note that the adversary then has a motivation to output a smaller message space M so as to get more information from the modified decryption oracle. However, it is always in the adversary's best interest to output a small distribution M, for the smaller the distribution, the better the chance that the adversary knows which plaintext the umpire pulled from the distribution.

To achieve the above goals, we will further require of the adversary a polynomial time algorithm[10] $\mathcal{C}^M$ such that $\mathcal{C}^M(x) = \mathsf{True}$ if and only if $x \in M$. An adversary must be able to provide such an algorithm if he can provide M, a distribution of messages that can be sampled in time polynomial in k.

Thus, the restricted oracle is as follows:

[10] That is, polynomial in the security parameter k.

- Oracle $\mathcal{S}_{sk}^{M}(y)$
  1. Run $x \leftarrow \mathcal{D}_{sk}(y)$
  2. If $\mathcal{C}^{M}(x) = \mathsf{True}$, then output "$*$"
  3. Else output x

Given this new oracle, the new definition of NM follows:

Definition 6.1 (NM-CPA, NM-CCA1, NM-CCA2) *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $A = (A_1, A_2)$ be an adversary. For atk $\in$ {cpa,cca1,cca2} and $k \in \mathbb{N}$ define*

$$\mathsf{Adv}_{A,\Pi}^{nm\text{-}atk}(k) \stackrel{def}{=} \left|\mathsf{Succ}_{A,\Pi}^{nm\text{-}atk}(k) - \mathsf{Succ}_{A,\Pi,\$}^{nm\text{-}atk}(k)\right| \tag{6.1}$$

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

$$\begin{aligned}\Pr[(pk, sk) \leftarrow \mathcal{K}(1^k);\ (M, \mathcal{C}^M, s) \leftarrow A_1^{\mathcal{O}_1}(pk);\ x \leftarrow M;\ y \leftarrow \mathcal{E}_{pk}(x);\\ (R, \mathbf{y}) \leftarrow A_2^{\mathcal{O}_2}(M, s, y);\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) : \perp \notin \mathbf{x} \wedge \mathbf{x} \cap M = \emptyset \wedge R(x, \mathbf{x})]\end{aligned} \tag{6.2}$$

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

$$\begin{aligned}\Pr[(pk, sk) \leftarrow \mathcal{K}(1^k);\ (M, \mathcal{C}^M, s) \leftarrow A_1^{\mathcal{O}_1}(pk);\ x, \tilde{x} \leftarrow M;\ y \leftarrow \mathcal{E}_{pk}(x);\\ (R, \mathbf{y}) \leftarrow A_2^{\mathcal{O}_2}(M, s, y);\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) : \perp \notin \mathbf{x} \wedge \mathbf{x} \cap M = \emptyset \wedge R(\tilde{x}, \mathbf{x})]\end{aligned} \tag{6.3}$$

where:

- *If atk = cpa, then $\mathcal{O}_1(\cdot) = \varepsilon$ and $\mathcal{O}_2(\cdot) = \varepsilon$.*
- *If atk = cca1, then $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}$ and $\mathcal{O}_2(\cdot) = \varepsilon$.*
- *If atk = cca2, then $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}$ and $\mathcal{O}_2(\cdot) = \mathcal{S}_{sk}^{M}$.*

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 also insist that R is such that $R(x, \boldsymbol{x}) = 0$ whenever $x \in \boldsymbol{x}$. 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 relations R computable in time $p(k)$, then $\mathsf{Adv}_{A,\Pi}^{nm\text{-}atk}(\cdot)$ is negligible.

Unfortunately, we have had to make our definition of non-malleability *ostensibly* weaker than that of [2, 10, 11]. The extra requirement that ($\mathbf{x} \cap M = \emptyset$) is needed for Theorem 6.2 and might appear restrictive. However, we can informally argue that if there exists an adversary A that defeats Π in the sense of NM-ATK, when the clause ($\mathbf{x} \cap M = \emptyset$) is not enforced, then there most certainly exists an adversary B that can defeat Π in the sense of NM-ATK when the clause is enforced; therefore, the clause in question does not make the new definition any weaker. Again, it is just a matter of B_1's judicious narrowing down of the distribution M, so that it does not contain any of the decryptions of malleated ciphertexts.

6.3 Relating IND-CCA2 To Non-Malleability

Since we have altered the definitions of both IND-CCA2 and NM, we would like to preserve the strong results relating the two. These theorems appear in the following sections.

6.3.1 NM-ATK ⇒ IND-ATK

The proof offered in this section follows the construction and argument of [2], with some refinements to reflect the new formulation of IND-CCA2 and NM.

Intuitively, this theorem states that non-malleability attacks are easier to mount against a cryptosystem Π than are distinguishability attacks. If an adversary A has the capacity to attack Π in the sense of distinguishability, and to succeed in partially decrypting messages, then her malleability attack becomes simple. That is, if an adversary is given a challenge ciphertext y, and can learn some information about $\mathcal{D}_{sk}(y)$, then it is easy for her to produce a message y' such that $\mathcal{D}_{sk}(y)$ and $\mathcal{D}_{sk}(y')$ are related. The adversary computes x so that x approximates $\mathcal{D}_{sk}(y)$ — this is the distinguishing phase — then forms the plaintext x' related to x and then encrypts $y' \leftarrow \mathcal{E}_{pk}(x')$.

By contrapositive, this theorem proves that a system secure against non-malleability is indeed more secure than a system secure only against distinguishability.

Theorem 6.1 (NM-ATK ⇒ IND-ATK) *If Π is secure in the sense of NM-ATK, then it is secure in the sense of IND-ATK, for ATK $\in$ {CPA, CCA1, CCA2}.*

Proof. We start out with a cryptosystem Π that is secure in the sense of NM-ATK. To show that Π is secure in the sense of IND-ATK, we show that for any adversary $B = (B_1, B_2)$, that $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k)$ is negligible in k. To do this, we construct an adversary A closely related to B that attacks Π in the sense of NM-ATK. Π secure in the sense of NM-ATK implies that $\mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(k)$ is negligible, and this should be enough to prove that $\mathsf{Adv}^{\text{ind-atk}}_{A,\Pi}(k)$ must also be negligible. To this effect, construct A as follows:

- Algorithm $A_1^{\mathcal{O}_1}(pk)$
  1. Run $(x_0, x_1, s) \leftarrow B_1^{\mathcal{O}_1}(pk)$
  2. Set $M := \{x_0, x_1\}$
  3. Define $\mathcal{C}^M$ such that $\mathcal{C}^M(x) = \mathsf{True}$ if $x = x_0$ or $x = x_1$; else, $\mathcal{C}^M(x) = \mathsf{False}$.
  4. $s' \leftarrow (x_0, x_1, pk, s)$
  5. Return $(M, \mathcal{C}^M, s')$
- Algorithm $A_2^{\mathcal{O}_2}(M, s', y)$ where $s' = (x_0, x_1, pk, s)$
  1. $c \leftarrow B_2^{\mathcal{O}'_2}(x_0, x_1, s, y)$
  2. $y' \leftarrow \mathcal{E}_{pk}(\overline{x_c})$
  3. Return (R, y') where $R(a, b) = 1$ if $\overline{a} = b$

By $M := \{x_0, x_1\}$, we mean that M is the probability space for which x_0 and x_1 can be chosen with equal probability, namely $1/2$.

Note that in the case of NM-CCA2, A_2 will have access to the oracle $\mathcal{S}^M_{sk}$, and B_2 will need access to the oracle $\mathcal{O}'_2 = \mathcal{S}^{x_0,x_1}_{pk}$. By definition of $\mathcal{S}$, $\mathcal{O}_2 = \mathcal{O}'_2$. In the case of NM-CCA2, the oracle used by A_1 and that used by B_1 are the exact same oracles.

To proceed, we recall the definitions of NM-ATK and IND-ATK. That is, the advantage by A is given by:

$$\mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(k) = \left|\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) - \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k)\right| \tag{6.4}$$

And in particular,

$$\begin{aligned}
\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) = \Pr[&(pk,sk) \leftarrow \mathcal{K}(1^k);\ (M,\mathcal{C}^M,s) \leftarrow A_1^{\mathcal{O}_1}(pk);\ x \leftarrow M;\ y \leftarrow \mathcal{E}_{pk}(x);\\
&(R,y') \leftarrow A_2^{\mathcal{O}_2}(M,s,y);\ x' \leftarrow \mathcal{D}_{sk}(y') : \perp \neq x' \wedge x' \notin M \wedge R(x,x')]\\
\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) = \Pr[&(pk,sk) \leftarrow \mathcal{K}(1^k);\ (M,\mathcal{C}^M,s) \leftarrow A_1^{\mathcal{O}_1}(pk);\ x,\tilde{x} \leftarrow M;\ y \leftarrow \mathcal{E}_{pk}(x);\\
&(R,y') \leftarrow A_2^{\mathcal{O}_2}(M,s,y);\ x' \leftarrow \mathcal{D}_{sk}(y') : \perp \neq x' \wedge x' \notin M \wedge R(\tilde{x},x')]
\end{aligned} \tag{6.5}$$

Our aim is to relate the above probabilities to $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k)$, the advantage that B has in mounting an IND-ATK attack against Π. Adopting the same notation as in [2], $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) = 2 \cdot p_k - 1$, where

$$\begin{aligned}
p_k = \Pr[&(sk,pk) \leftarrow \mathcal{K}(1^k);\ (x_0,x_1,s) \leftarrow A_1^{\mathcal{O}_1};\ b \leftarrow \{0,1\};\\
&y \leftarrow \mathcal{E}_{pk}(x_b) :\ A_2^{\mathcal{O}_2}(x_0,x_1,s,y) = b]
\end{aligned} \tag{6.6}$$

We use Proposition 3.8 of [2] to assume, without loss of generality, that $x_0 \neq x_1$. That proposition proves that for any adversary $C = (C_1, C_2)$ that attacks Π in the sense of IND-ATK, there exists a closely related adversary $B = (B_1, B_2)$ that attacks Π with a comparable advantage; however, with B, we are guaranteed that for all inputs pk, $(x_0, x_1, s) \leftarrow B_1^{\mathcal{O}_1}(pk)$ is such that $x_0 \neq x_1$. A similar argument justifies the assertion that without loss of generality, $\overline{x_0} \neq x_1$.

Our first claim establishes the connection between B attacking Π in the sense of IND-ATK and A attacking Π in the sense of NM-ATK:

Claim 6.1 $\mathsf{Succ}^{\textit{nm-atk}}_{A,\Pi}(k) = p_k$.

Proof. The condition for $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k)$ is that $\perp \neq x'$, $x' \notin M$, and $R(x,x')$ is true. We first observe that R is a legal relation: that for all z, $R(z,z)$ is false, as $z \neq \overline{z}$. Note the difference between our proof and that of [2]. That paper shows instead that $R(x,x')$ implies $x \neq x'$, and hence by unique decryptions, that $y \neq y'$.

By construction of the adversary A, $R(x,x')$ will be true exactly when $\overline{x} = x'$. Since $y' = \mathcal{E}_{pk}(x')$ by the last step in the definitional experiment, and $y' = \mathcal{E}_{pk}(\overline{x_c})$ by Step 2 of Algorithm A_2, it follows naturally that $x' = \overline{x_c}$. Combining these results, $\overline{x_c} = x' = \overline{x}$, and thus $x = x_c$. Also, $\overline{x} = x'$, coupled with $\overline{x_0} \neq x_1$, gives us that $x' \notin M$.

By Step 2, it follows that the malleated encryption output by A_2 will always be a valid encryption, and hence, its decryption will never be $\perp$. Thus, it is always the case that $\perp \neq x'$.

So far we have shown that $(\perp \neq x' \wedge x' \notin M \wedge R(x,x'))$ if and only if $x = x_c$, where $x = \mathcal{D}_{sk}(y)$, the decryption of the challenge ciphertext. In both the experiment defining $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k)$ and the experiment defining p_k, the challenge ciphertext y is chosen under the same circumstances: it is the encryption of a message randomly chosen from the distribution

$\{x_0, x_1\}$. If we assume that $x_0 \neq x_1$, by Proposition 3.8 of [2], then we can argue that $x = x_c$ if and only if $b = c$; this is by the experiment that defines p_k. Thus, we have that

$$\Pr[b = c] = \Pr[\perp \neq x' \wedge x' \notin M \wedge R(x, x')] \tag{6.7}$$

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

Claim 6.2 $\mathsf{Succ}^{\textit{nm-atk}}_{A,\Pi,\$}(k) = 1/2$.

Proof. See [2]. This is a result of "an information theoretic fact, namely that A has no information about the message $\tilde{x}$ with respect to which its success is measured." In other words, the selection of $\tilde{x}$ and the processing of algorithm A are independent events. Thus the order in which they are executed is irrelevant. The definitional experiment chooses $\tilde{x} \leftarrow M$ and then runs A_2. Since A_2 is not fed $\tilde{x}$, it can also be run before its selection.

From above, we have that $R(\tilde{x}, x')$ holds whenever $\tilde{x} = x_c$. In effect, A_2 will run to completion and output something that corresponds to $x_c \in \{x_0, x_1\}$. After A_2 is all finished, the umpire will randomly choose $\tilde{x} \in \{x_0, x_1\}$. A will succeed whenever $R(\tilde{x}, x')$ holds, or whenever $\tilde{x} = x_c$, and it is clear to see this is determined only by the selection of $\tilde{x}$, and hence, happens with probability 1/2.

Given these two claims, the analysis from [2] shows:

$$\begin{aligned} \mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) &= 2 \cdot \left(p_k - \frac{1}{2} \right) \\ &= 2 \cdot \left(\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) - \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) \right) \\ &\leq 2 \cdot \left| \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) - \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) \right| \end{aligned} \tag{6.8}$$

Hence, $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) \leq 2 \cdot \mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(k)$. Because Π is secure in the sense of NM-ATK, $\mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(k)$ must be negligible in k. Consequently $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k)$ is also negligible in k, and our theorem is proven.

6.3.2 IND-CCA2 ⇒ NM-CCA2

Again, we follow the constructions and outline of the proof offered in [2], paying close attention to the details of our new definitions. This result, after all, is a strong result about security against CCA2. A similar result does not exist for IND-CCA1, which leads cryptographers to believe that IND-CCA2, the adaptive chosen ciphertext attack, is the more useful adversarial model, despite its lack of correspondence to practical attacks.

The intuition behind the theorem is as follows. An adversary who mounts a malleability attack against a scheme Π has less work to do than does an adversary who mounts a distinguishability attack against Π. After all, the former adversary does not have to determine information about the plaintext but rather can succeed by altering ciphertexts. Such a capacity for malleating messages might not be useful at all in distinguishing two messages, for computing results in the realm of decrypted, plaintext messages. However, the lower standard for success in the realm of malleability actually conforms to the higher standard for success in the realm of distinguishability when the adversary has access to the decryption oracle (NM-CCA2). If an adversary can exploit a relationship between two encrypted

messages (non-malleability), then he can decrypt these messages using the decryption box and determine explicitly a corresponding relationship between the two plaintexts (distinguishability). In other words, a malleability adversary might "know" less about the system than does a distinguishability adversary, but the former's access to the decryption box in the case of NM-CCA2 makes up for any handicaps, thereby establishing their equal strength.

Theorem 6.2 (IND-CCA2 $\Rightarrow$ NM-CCA2) *If Π is secure in the sense of IND-CCA2, then it is also secure in the sense of NM-CCA2.*

Proof. As usual, we begin with a cryptosystem $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ that is secure in the sense of IND-CCA2. An arbitrary adversary who attacks Π in the sense of NM-CCA2 is now considered; call it $B = (B_1, B_2)$. Note that B must meet the conditions of Definition 6.1; namely, B_2 must output R so that $R(x, \mathbf{x}) = 0$ if $x \in \mathbf{x}$. Our goal is thus to construct an algorithm $A = (A_1, A_2)$, which calls upon B and attacks Π in the sense of IND-CCA2. If we can prove that $\mathsf{Adv}_{A,\Pi}^{\text{ind-cca2}}(k)$ negligible implies that $\mathsf{Adv}_{B,\Pi}^{\text{nm-cca2}}(k)$ is negligible, then our theorem is proven.

The construction follows. Note that it is based on that of [2], but some modifications have been made to simulate the $\mathcal{S}_{sk}^{M}$ oracle required by B_2.

1. Algorithm $A_1^{\mathcal{D}_{sk}}(pk)$
   (a) Run $(M, \mathcal{C}^M, s) \leftarrow B_1^{\mathcal{D}_{sk}}(pk)$
   (b) Chose $x_0, x_1 \leftarrow M$ at random
   (c) $s' \leftarrow (s, M, \mathcal{C}^M)$
   (d) Output (x_0, x_1, s')
2. Algorithm $A_2^{\mathcal{S}_{sk}^{x_0,x_1}}(x_0, x_1, y, s')$ where $s' = (s, M, \mathcal{C}^M)$
   (a) Define $\mathcal{S}_{sk}^{M}(y)$ as follows:
      i. $z \leftarrow \mathcal{S}_{sk}^{x_0,x_1}(y)$
      ii. If $z = *$, or $\mathcal{C}^M(z) = \mathsf{True}$, then output "$*$"
      iii. Else output z
   (b) Run $(R, \mathbf{y}) \leftarrow B_2^{\mathcal{S}_{sk}^{M}}(M, s, y)$
   (c) Run $\mathbf{x} \leftarrow \mathcal{S}_{sk}^{x_0,x_1}(\mathbf{y})$
   (d) If $* \in \mathbf{x}$, or there is an element $v \in \mathbf{x}$ such that $\mathcal{C}^M(v) = \mathsf{True}$, then $F \leftarrow \mathsf{False}$; else $F \leftarrow \mathsf{True}$
   (e) Else if $\perp \notin \mathbf{x}$, and F and $R(x_0, \mathbf{x})$ are true, then return 0
   (f) Else, output $d \leftarrow \{0, 1\}$

We first argue that A_2 is able to accurately simulate $\mathcal{S}_{sk}^{M}$ in all cases. If B_2 submits y to the simulation of $\mathcal{S}_{sk}^{M}$ such that $\mathcal{D}_{sk}(y)$ is x_0 or x_1 (which are both in M by the definition of A_1), then $\mathcal{S}_{sk}^{x_0,x_1}(y) = *$, and therefore the simulation of $\mathcal{S}_{sk}^{M}$ will output "$*$" by Step 2(a)i. If B_2 submits y so that $\mathcal{D}_{sk}(y) = x$, where $x \in M - \{x_0, x_1\}$, then by Step 2(a)ii, $\mathcal{S}_{sk}^{M}$ will output "$*$". Finally, for any y such that $y \notin M$, $\mathcal{S}_{sk}^{M}$ will output $\mathcal{D}_{sk}(y)$ by Step 2(a)ii. Thus, the simulation of $\mathcal{S}_{sk}^{M}$ exactly mirrors the behavior of the real oracle.

The rest of the proof now follows from [2], with some added elucidation. The proof proceeds by considering four very closely related events. The first two are in the domain of distinguishability. Namely, consider the experiment that defines $\mathsf{Adv}^{\text{ind-ica2}}_{A,\Pi}(k)$, which tests A's capacity to differentiate encrypted ciphertexts as usual. Consider all experiments in which A_2 outputs a guess bit of 0; here there are two events. In the first, the umpire actually encrypted x_0 and A guesses successfully. The clear complement is the case in which the umpire encrypted x_1 and hence A guessed incorrectly. These two events are given by $p_k(0)$ and $p_k(1)$ respectively, where:

$$\begin{aligned} p_k(b) = \Pr[(sk, pk) \leftarrow \mathcal{K}(1^k);\ (x_0, x_1, s) \leftarrow A_1^{\mathcal{D}_{sk}};\ y \leftarrow \mathcal{E}_{pk}(x_b) : \\ A_2^{\mathcal{S}_{sk}^{x_0,x_1}}(x_0, x_1, s, y) = 0] \end{aligned} \tag{6.9}$$

It is clear to see that the success of the adversary can be given by:

$$\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) = p_k(0) - p_k(1) \tag{6.10}$$

Equation (6.10) is the difference between the probability of successfully guessing b and the probability of introducing incorrect results into the experiment. Again, because Π is assumed to be secure in the sense of IND-CCA2, we expect $p_k(0) - p_k(1)$ to be negligible in k.

The next two events are in the realm of malleability. We consider the cases in which the success conditions for the non-malleability experiment hold, namely that $\perp \notin \mathbf{x}$, $\mathbf{x} \cap M = \emptyset$ and $R(x, \mathbf{x})$ hold. The first event to consider, $p'_k(0)$ is the case of the honest experiment, in which the x used in the test conditions is the same x whose encryption the adversary examined. The second event, $p'_k(1)$, is the placebo experiment in which $\tilde{x}$ is used in the test conditions, but an encryption of $x \neq \tilde{x}$ is given to the adversary. We therefore have:

$$\begin{aligned} p'_k(b) = \Pr[(pk, sk) \leftarrow \mathcal{K}(1^k);\ (M, \mathcal{C}^M, s) \leftarrow A_1^{\mathcal{D}_{sk}}(pk);\ x_0, x_1 \leftarrow M;\ y \leftarrow \mathcal{E}_{pk}(x_b); \\ (R, \mathbf{y}) \leftarrow A_2^{\mathcal{S}_{sk}^M}(M, s, y);\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) : \perp \notin \mathbf{x} \wedge \mathbf{x} \cap M = \emptyset \wedge R(x_0, \mathbf{x})] \end{aligned} \tag{6.11}$$

In the case of $p'_0(k)$, the challenge encryption $y = \mathcal{E}_{pk}(x_0)$ is indeed the encryption of the message x_0 used in the test condition of the experiment. However, in the case of $p'_1(k)$, the challenge encryption $y = \mathcal{E}_{pk}(x_1)$ may not be the encryption of the message x_0 used in the test condition. Thus, by direct application of Definition 6.1, we have that:

$$\mathsf{Adv}^{\text{nm-cca2}}_{B,\Pi}(k) = p'_k(0) - p'_k(1) \tag{6.12}$$

Now that we have established these four experiments and their relations to definitions of security, the theorem is just a matter of showing relationships between p_k and p'_k, which will follow from the construction. Consider $p_k(0)$. If the random bit b chosen by the umpire is 0, then the challenge encryption given to A_2 is $y = \mathcal{E}_{pk}(x_0)$. Now let us look at the code of A_2. Consider the case in which "$*$" is output in Step 2c. Then there exists either x_0 or x_1 in the vector $\mathcal{D}_{sk}(\mathbf{y})$. Thus, $\mathcal{D}_{sk}(\mathbf{y}) \cap M \neq \emptyset$. Furthermore, if there exists a v such that $v \in \mathcal{D}_{sk}(\mathbf{y})$ and $v \in M$, then $\mathcal{C}^M(v)$ will output True in Step 2d. Hence, F will be set to True if and only if $\mathcal{D}_{sk}(\mathbf{y}) \cap M = \emptyset$.

The probability that $(R, \mathbf{y}) \leftarrow B_2(M, s, y)$ is output such that $\perp \notin \mathcal{D}_{sk}(\mathbf{y})$, F and $R(x_0, \mathcal{D}_{sk}(\mathbf{y}))$ are true is therefore given exactly by $p'_0(k)$. Thus, the probability that A_2

outputs 0 on Step 2(a)ii is given by $p'_0(k)$. The probability that the program will get to Step 2f is given by $(1-p'_k(0))$, and only one half of these cases will result in an output of 0. Hence, the probability of the A_2 outputting 0 from Step 2f is given by $\frac{1}{2}(1-p'_k(0))$. Finally, the probability that A_2 outputs 0 when $b=0$ is given by:

$$p_k(0) = p'_k(0) + \frac{1}{2}(1 - p'_k(0)) = \frac{1}{2}\left[1 + p'_k(0)\right] \tag{6.13}$$

The same exact argument is applied for the case in which the random bit b chosen by the umpire is 1, and the challenge ciphertext given to A_2 is $y = \mathcal{E}_{pk}(x_1)$. That is:

$$p_k(1) = \frac{1}{2}\left[1 + p'_k(1)\right] \tag{6.14}$$

Putting Equations (6.13) and (6.14) together, it is clear to see that:

$$p_k(0) - p_k(1) = \frac{1}{2}\left[p'_0(k) - p'_1(k)\right] \tag{6.15}$$

With the help of Equations (6.10) and (6.12), it is concluded that:

$$\mathsf{Adv}^{\text{nm-cca2}}_{B,\Pi}(k) = 2 \cdot \mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) \tag{6.16}$$

Because Π is secure in the sense of NM-CCA2, $\mathsf{Adv}^{\text{ind-cca2}}_{B,\Pi}(k)$ is negligible in k. Therefore, $\mathsf{Adv}^{\text{nm-cca2}}_{A,\Pi}(k)$ is negligible in k, and the result is proven. ■

7 Conclusion

It is important to note that all cryptographic algorithms, protocols and applications are based on the notions of security most recently summarized in [2] and revised here. Clearly, these definitions are always open to further analysis and revision, in attempt to find general and strong definitions that can be applied to efficient algorithms. In the name of clarity and inclusiveness, this thesis recommends a revised definition of IND-CCA2 and NM. The existing definitions were too exclusive, and it is entirely conceivable that an efficient public key cryptosystem might be developed that is secure in our sense of IND-CCA2 and NM-CCA2, but insecure in the sense of older definitions.

Moreover, the introduction of illegal ciphertext attack in this thesis serves to find a marriage between practical methods of cryptographic attack, such as the Bleichenbacher attack [5], and theoretical definitions of security, as presented in [2]. This new kind of security can also help to provide more intuition on existing definitions of security, IND-CCA2 in particular. As seen in Section 5.2, an awareness of ICA security has led to further intuitive understanding of adaptive chosen ciphertext attack. Illegal ciphertext attack and Theorem 4.7 also serve to support IND-CCA2 as the "correct" model of chosen ciphertext security.

This thesis hardly represents a complete discussion of the topic. Indeed, there is more work to be done in relating non-malleability and non-adaptive chosen ciphertext attack to illegal ciphertext attack. However, the arguments presented here — and any others pertaining to illegal ciphertext attack — point to a new standard of security for practical cryptosystems. Indeed, many inefficient schemes exist in the literature that are secure against IND-CCA2 [23], but new applications compel cryptographers to develop cryptosystems that are efficient and meet a standard of security higher that that of IND-CPA. As we have shown in this thesis, IND-ICA2 is a type of security that although weaker than IND-CCA2 is stronger than IND-CPA and even IND-CCA1. Furthermore, it provides security against the most feasible types of active attack, such as the Bleichenbacher attack. It is highly recommended that any cryptosystem for use over the Internet meets the IND-ICA2 standard of security. Further work in cryptography might result in an efficient cryptographic scheme, which though insecure against adaptive chosen ciphertext attack, is still secure in the sense of IND-ICA2. Such a system would be ideal for secure and practical transactions over the Internet. But whether cryptographers consider illegal ciphertext attack as a practical guideline for algorithm design or a theoretical tool, it is an important notion of security that should stand alongside the more canonical definitions.

8 Acknowledgments

I would like to thank my advisors, Professor Michael Rabin and Professor Michael Mitzenmacher, for their continual guidance and assistance. Although this thesis is based largely upon [2], Professor Rabin's [1] profoundly influenced my research and thought. Additional thank yous to my concentration advisor, Professor Les Valiant, to my dedicated team of proofreaders, and to Professor Sylvio Micali for his formative lectures on cryptography.

References

[1] Y. Aumann and M. Rabin. Proof of plaintext knowledge and chosen ciphertext attack, 1999. In preparation.

[2] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology – Crypto '98*, Lecture Notes in Computer Science. Springer-Verlag, 1998. Full version can be found at `http://www-cse.ucsd.edu/~mihir/papers/relations.ps`.

[3] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*, ACM, 1993.

[4] M. Bellare and P. Rogaway. Optimal asymmetric encryption – how to encrypt with RSA. In A. De Santis, editor, *Advances in Cryptology – Crypto '93*, volume 950 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994. Full version can be found at `http://www-cse.ucsd.edu/~mihir/papers/pke.ps`.

[5] D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PCKS#1. In H. Krawczyk, editor, *Advances in Cryptology – Crypto '98*, Lecture Notes in Computer Science. Springer-Verlag, 1998.

[6] M. Blum and S. Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In G. R. Blakley and D. C. Chaum, editors, *Advances in Cryptology – Crypto '84*, pages 289–302. Springer-Verlag, 1985. Lecture Notes in Computer Science No. 196.

[7] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited (preliminary version). Available at `ftp://theory.lcs.mit.edu/pub/people/oded/rom.ps`, March 1998.

[8] R. Cramer and V. Shoop. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *Advances in Cryptology – Crypto '98*, Lecture Notes in Computer Science, pages 12–25. Springer-Verlag, 1998.

[9] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

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

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

[12] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4), July 1985.

[13] O. Goldreich. Foundations of cryptography (fragments of a book), February 1995. In preparation. Available online at: `http://philby.ucsd.edu/BOOKS/oded-frag.html`.

[14] O. Goldreich and S. Goldwasser. On the possibility of basing cryptography on the assumption that $\mathcal{P} \neq \mathcal{NP}$. Available at `ftp://theory.lcs.mit.edu/pub/people/oded/rom.ps`, February 1998.

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

[16] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal of Computer Science*, 18(1):186 – 208, February 1989.

[17] J. Hastad, 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.

[18] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *Proceedings of the 30th Symposium on Foundations of Computer Science*, IEEE, 1989.

[19] S. Micali. From class lectures, MIT 6.875J, Fall 1997.

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

[21] National Institute of Standards. Secure hash algorithm. Technical Report 180, FIPS, 1993.

[22] M. Rabin. Digitalized signatures and public key functions as intractable as factoring. Technical Report MIT/LCS/TR-212, MIT, 1979.

[23] C. Rackoff and D. Simon. Noninteractive zero-knowledge proof of knowledge and chosen ciphertext attack. *22nd Annual ACM Symposium on Theory of Computing*, pages 427 – 437, 1990. Available at: `http://research.microsoft.com/crypto/papers/ni.ps`.

[24] R. Rivest. The MD5 message-digest algorithm. *IETF Network Working Group, RFC 1321*, April 1992.

[25] R. Rivest, A. Shamir, and L. Adelman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21(2):120 – 126, February 1978.