On Chosen Ciphertext Security of Multiple Encryptions

Oded Goldreich[*] Yoad Lustig Moni Naor[†]
Department of Computer Science
Weizmann Institute of Science
Rehovot 76100, ISRAEL

Preliminary version: February 2002
Slightly revised July 4, 2002

Abstract

We consider the security of *multiple* and possibly related plaintexts in the context of a chosen ciphertext attack. That is the attacker in addition and concurrently to obtaining encryptions of multiple plaintexts under the same key, may issue encryption and decryption queries and partial information queries. Loosely speaking, an encryption scheme is considered secure under such attacks if all that the adversary can learn from such attacks about the selected plaintexts can be obtained from the corresponding partial information queries.

The above definition extends the definition of semantic security under chosen ciphertext attacks (CCAs) which is also formulated in this work. The extension is in considering the security of multiple plaintexts rather than the security of a single plaintext. We prove that both these formulations are equivalent to the standard formulation of CCA, which refers to indistinguishability of encryptions. The good news is that any encryption scheme that is secure in the standard CCA sense is in fact secure in the extended model.

The treatment holds both for public-key and private-key encryption schemes.

Keywords: Security of Encryption Schemes, Chosen Ciphertext Attacks,

[*] Email: oded@wisdom.weizmann.ac.il. Supported by the MINERVA Foundation, Germany.

[†] Email: naor@wisdom.weizmann.ac.il. Supported in part by an IST Grant RAND-APX.

1 Introduction

In order to rigorously treat the security of a cryptographic scheme one must specify two things: (i) the power of the adversary both in terms of computation (time, memory etc.) and in terms of *access to the system*, and (ii) what it means to *break the cryptosystem*. To be more specific, in the context of an *encryption* scheme access to the system (as in (i)) means the type of attack (e.g., known or chosen plaintext, or chosen ciphertext). Breaking the encryption scheme (as in (ii)) should specify the functionality the adversary can gain with respect to the plaintext. Two examples of defining such functionalities are semantic security (see below) and non-malleability (which will not be the focus of the paper).

The rigorous treatment of the security of encryption schemes was initiated in the seminal work of Goldwasser and Micali [15]. Focusing on *passive* attacks, they introduced two fundamental notions of security, called *semantic security* and *indistinguishability of encryptions*.

Semantic security is a computational analogue of Shannon's definition of perfect secrecy [23]. It requires that whatever information about the plaintext one may compute from the ciphertext and some a-priori information, can be essentially computed as efficiently from the a-priori information alone.[1] This definition is the natural one, because it directly addresses the user's concerns (i.e., that nothing be gained by looking at the ciphertext). In retrospect, additional confidence in this definition may be gained by the fact that it actually follows the simulation paradigm [16], which underlies much of the later definitional work.

Indistinguishability of encryptions is a technical definition requiring that, for any two messages, it is infeasible to distinguish the encryption of the first message from the encryption of the second message.

The importance of the technical definition of indistinguishability of encryptions stems from the fact that it is equivalent to semantic security (cf. [15, 11]) while being easier to work with and reason about. In particular, it is easier to prove that an encryption scheme has indistinguishable encryptions and to deduce that it is semantically secure (cf. [15, 11]) than to directly prove that the encryption scheme is semantically secure. Note that all this discussion as well as the rest of paper is applicable both to the public-key and private-key models, where the difference amounts to whether or not the adversary is given the encryption-key.

As is well known and documented[2], it is often the case that the adversary may launch an *active* attack on the system. In particular, it might cause the system to decrypt ciphertexts of its choice for a while, i.e., launch a *chosen ciphertext attack*. Hence, stronger notions of attacks were considered[3], as well as definitions of security against such attacks and constructions of cryptosystem satisfying those definitions. Goldwasser, Micali, and Tong [17] investigated *interactive* public key cryptosystems secure against chosen ciphertext attacks. Naor and Yung [20] defined and constructed[4] schemes secure against *a priori chosen ciphertext attacks* also known as lunch-break attack[5]. Rackoff and Simon [24] defined the stronger type of attack, *a posteriori chosen ciphertext attacks* and Dolev, Dwork and Naor [9] constructed cryptosystems resistant to such attacks. Other constructions where given in [14, 6, 21, 22, 19]. Other works have explored the relationship

[1] This specific formulation was first suggested by Goldreich [10], and is equivalent to the one presented in [15].

[2] Consider, for instance, the attack Bleichenbacher suggested on PKCS # 1 [3].

[3] Also stronger requirements (from the implementation point of view) for *breaking* the cryptosystem were considered, namely malleability [9].

[4] The construction was based on Non-interactive Zero-Knowledge [5, 4].

[5] See more on the variants of chosen ciphertext attacks in Section 1.1.

between various types of active attacks [1, 9, 18]. These works have mostly dealt with the security of a *single* message and when discussing semantic security or indistinguishability of encryptions they have done so while referring to the latter, *technical definition* of security Note though that the non-malleability works have dealt directly with the security functionality.

New Attacks: It is relatively straightforward to generalize these definitions to deal with multiple message chosen simultaneously, and indeed such generalizations were considered in [9]. However, the main contribution of the current work is in treating the security of multiple messages chosen *adaptively and in a related manner* under a chosen ciphertext attack. In this new type of attack we consider a user holding some secret (as well as a secret key) being attacked by an interactive adversary as follows. The adversary can ask the user to encrypt (under said key) any partial information regarding its secret, i.e. the adversary specifies a function to be applied to the secret. The adversary can also ask the user to decrypt (under the corresponding key) any string of its choice. Note that the encryption and decryption requests are made adaptively and can be interleaved. Loosely speaking, we seek encryption schemes in which the adversary gains nothing by the above type of attacks (and call them multiple-messages CCA-secure). Our good news[6] is that we show that any encryption scheme that is CCA-secure in the standard sense satisfies the new notion of multiple-messages CCA-security.

We also provide a definition of single-message semantic-security under chosen ciphertext attacks, and show that it coincides with the standard notion of (single-message) CCA-security, which refers to the indistinguishability of encryptions (Section 3.) We note that independently of this work Watanabe, Shikata and Imai [26] considered the semantic security of a single message under chosen ciphertext attacks and showed its equivalence to indistinguishability of encryption. *For methodological reasons, although the treatment of multiple-messages CCA-security is the more important contribution of this work, we start with a treatment of single-message semantic-security under CCA.*

1.1 Semantic Security Under Chosen Ciphertext Attacks

Our first contribution is in suggesting a semantic security definition for the context of chosen ciphertext attacks, and in showing that this definition is equivalent to indistinguishability of encryptions under such attacks. Indeed, this is good news: It means that all schemes proven secure (in the technical sense) under chosen ciphertext attacks, are actually secure in the (more appealing) semantic security sense.

We treat both a-priori chosen ciphertext attacks (CCA1) and a-posteriori chosen ciphertext attacks (CCA2), and refer both to the public-key and private-key models. In all cases the attacker is given access to two oracles, one for encryption and the other for decryption. The attack is broken into two stages:

Stage 1: The attacker conducts some computation, using both its oracles, and terminates this stage by outputting a challenge template. We note that in the technical definition (i.e., indistinguishability of encryptions) the challenge template consists of a pair of (equal-length) plaintexts. In our definition of semantic security, the challenge template consists of three circuits (S, L, F), where S is a *sampling* circuit, and L (resp. F) are circuits with a number of input bits that equals the number of output bits in S. Loosely speaking, S specifies a probability space on plaintexts (i.e., by feeding S with a random input), L specifies partial

[6] We have no bad news; this is a good news only paper.

information (i.e., "information leak") regarding the plaintext that is given to the adversary, and F specifies partial information (regarding the plaintext) that the adversary claims to be able to learn.

Stage 2: In the second stage the adversary is given an encryption of a plaintext x along with $L(x)$, where x is selected according to S. In case of CCA1, at this stage, the adversary is only given access to the encryption oracle. In case of CCA2, at this stage, the adversary is also given access to the decryption oracle, under the restriction that it does not query the latter on the ciphertext obtained at the beginning of this stage. In both cases, the adversary halts with a guess for $F(x)$.

(Recall that in the technical definition, the adversary is given an encryption of one of the challenge plaintexts, and is outputting a bit (in attempt to distinguish the two cases).)

Loosely speaking, an encryption scheme is said to be semantically secure under CCAi (where $i \in \{1,2\}$) if for every efficient (i-type) attacker as above, there exists a corresponding benign adversary that "performs as well" without seeing the ciphertexts. Specifically, the benign adversary is given no oracle access, it produces a challenge template (S, L, F) (as above), is given only $L(x)$ (where x is selected according to S), and is supposed to guess $F(x)$. The benign adversary is required to produce challenge template according to the same distribution as the real adversary, and to be as successful as the real adversary in its guess of $F(x)$.

Note that the benign adversary models an ideal situation in which the adversary produces the same challenge template as the real adversary, but is given a "perfectly secure encryption" of the plaintext x (where given a "perfectly secure encryption" is equivalent to being given nothing).

1.2 Semantic Security Under Multiple-Challenge CCA

The above definition of semantic security seems most satisfactory, except that it refers only to the security of a single encrypted plaintext. Instead, one typically wants to consider the security of many plaintexts encrypted under the same key[7]. A simple way of addressing this concern is to generalize the notion of a challenge template, allowing to sample (via S) polynomially-many (possibly related) plaintexts, and letting L and F be applied to the resulting sequence of plaintexts. It is important to note that each of the plaintexts will be encrypted independently (i.e. using independent random bits) of the others. This is the notion considered in [9]. (Note that the definition there is for non-malleability which, in general, is a stronger requirement than semantic security, but the two notion coincide for CCA2.)

The above simple extension does not seem to provide an ultimate definition. The reason being that, especially in a context in which queries are allowed, producing a single challenge template (which refers to a sequence of plaintexts) is not equivalent to *adaptively* producing polynomially-many challenge template (each referring to a single new plaintext and answered by its encryption). Thus, the general notion of multiple plaintext security consists of allowing the generation of polynomially-many challenge templates, each answered analogously to a single challenge template, when the generation of these challenge templates may be interleaved with the encryption and decryption queries. When generalizing CCA1, we do not allow decryption queries after the first challenge template is issued. On the other hand, when generalizing CCA2, we allow arbitrary interleaving of (encryption and) decryption queries with the generation of challenge templates. (We

[7] We note that Bellare *et al.* [2] have considered the security of the same plaintext when encrypted under different (independently chosen) keys. Clearly, our treatment can be extended to handle multiple plaintexts each encrypted under various keys.

will even allow making a decryption query that refers to a challenge ciphertext; see details below.) For sake of concreteness, in the rest of this paper, we focus on the CCA2 case.

We now sketch our definition of multiple-challenge CCA2 security. The attack proceeds in iterations, where each iteration is of the following two types:

1. Based on the information it has gathered so far, the attacker makes either an encryption or a decryption query, which is answered by the corresponding oracle.

2. Based on the information it has gathered so far, the attacker issues a challenge template, of the form (S, L), which is answered as follows. As before, S is a sampling circuit, but here S takes as input the random choices made when answering previous challenge templates as well as a new sequence of random bits. Analogously, L is a circuit that computes information regarding all challenge plaintext produced so far (including the current one). Denoting the ith challenge template by (S^i, L^i) and the fresh coins it uses by r^i, this template is answered with the encryption of x^i along with $L^i(x^1, ..., x^i)$, where $x^i = S^i(r^1, ..., r^i)$. That is, x^i is generated by invoking S^i with the coins used in previous challenges (i.e., $r^1, ..., r^{i-1}$) along with the fresh coins r^i, and the "clear" information obtained (i.e., $L^i(x^1, ..., x^i)$) refers to all challenge plaintext produced so far.

After completing polynomially-many iterations of the above type, the adversary outputs a function F and a guess v of the value of F when applied to all challenge plaintexts (i.e., it tries to guess $F(x^1, ..., x^t)$, where t is the number of challenge plaintexts).

Loosely speaking, an encryption scheme will be said to be semantically secure under multiple-challenge CCA2 if for every efficient attacker as above, there exists a corresponding benign adversary that "performs as well" without seeing the ciphertexts. Specifically, the benign adversary is given no oracle access, it produces challenge templates (S^i, L^i)'s (as above), is given only $L^i(x^1, ..., x^i)$ (where the x^i are selected as above), and is supposed to guess $F(x^1, ..., x^t)$, for F of its choice. The benign adversary is required to produce the challenge templates and the function F according to the same distribution as the real adversary, and to be as successful as the real adversary in its guess of $F(x^1, ..., x^t)$.

Again, there are good news: We prove that an encryption schemes is semantically secure under multiple-challenge CCA2 if and only if it is secure under ordinary CCA2. Thus, all schemes proven secure under CCA2, are actually secure under multiple-challenge CCA2.

2 Preliminaries: Chosen Ciphertext Attacks

Chosen ciphertext attacks are attacks in which the adversary may obtain (from some legitimate user) plaintexts corresponding to ciphertexts of its choice (as well as ciphertexts corresponding to plaintexts of its choice). We consider two types of such attacks: In the milder type (cf. [20]), called *a-priori chosen ciphertext attacks*, decryption requests can be made only before the challenge ciphertext (for which the adversary should gain knowledge) is presented. In the stronger type (cf. [24, 9]), called *a-posteriori chosen ciphertext attacks*, decryption requests can be made also after the challenge ciphertext is presented, as long as one does not request to decrypt this very (challenge) ciphertext.

Following the outline provided in Section 1.1, we recall the technical definition of indistinguishability of encryptions under chosen ciphertext attacks. A few introductory technical comments are in place. Firstly, the attacker is decoupled into two parts, denoted A_1 and A_2, which correspond to the two stages in the discussion provided in Section 1.1. The string σ (below) is used for passing state information from A_1 to A_2. (Thus, also in the public-key case, it is unnecessary to provide A_2 with the encryption-key e, because A_1 may pass e to A_2 as part of σ.) The string z encodes (non-uniform) auxiliary information that may be a-priori known to the adversary (which is an important issue enabling modular composition).[8] The challenge template produced by A_1, denoted $(x^{(1)}, x^{(2)})$, consists of a pair of (equal-length) strings, and the challenge ciphertext is an encryption of one of these strings.

Definition 2.1 (indistinguishability of encryptions under chosen ciphertext attacks):

For public-key schemes: *A public-key encryption scheme, (G, E, D), is said to have* indistinguishable encryptions under a-priori chosen ciphertext attacks (CCA1) *if for every pair of probabilistic polynomial-time oracle machines, A_1 and A_2, for every positive polynomial $p(\cdot)$, and all sufficiently large n and $z \in \{0,1\}^{\mathrm{poly}(n)}$:*

$$|p^{(1)}_{n,z} - p^{(2)}_{n,z}| \;<\; \frac{1}{p(n)}$$

where

$$p^{(i)}_{n,z} \;\stackrel{\rm def}{=}\; \mathsf{Pr}\left[\begin{array}{ll} v = 1 & \text{where} \\ & (e,d) \leftarrow G(1^n) \\ & ((x^{(1)}, x^{(2)}), \sigma) \leftarrow A_1^{E_e, D_d}(e, z), \text{ where } |x^{(1)}| = |x^{(2)}|. \\ & c \leftarrow E_e(x^{(i)}) \\ & v \leftarrow A_2^{E_e}(\sigma, c) \end{array}\right]$$

Indistinguishability of encryptions under a-posteriori chosen ciphertext attacks (CCA2) *is defined analogously, except that A_2 is given oracle access to both E_e and D_d with the restriction that when given the challenge c, machine A_2 is not allowed to make the query c to the oracle D_d.*

For private-key schemes: *The definition is identical except that A_1 gets the security parameter 1^n instead of the encryption-key e.*

Clearly, the a-posteriori version of Definition 2.1 implies its a-priori version, which in turn implies the standard notion of security under passive attacks. All implications are strict [1, 18].

3 Semantic Security Under Chosen Ciphertext Attacks

In this section we provide a definition of semantic security under chosen ciphertext attacks and show that it is equivalent to the existing technical definition of security under chosen ciphertext attacks (i.e., Definition 2.1). Our definition is a natural extension of the definition of semantic security for passive attacks (cf. [15, 11]), alas the formulation is slightly more complex in the current context.

3.1 Definition

When defining the adversary, we follow the framework used in the technical definition (i.e., Definition 2.1), while adapting it to the adequate notion of a challenge template. Specifically, following the outline provided in Section 1.1, a challenge template consists of a triplet of circuits, denoted

[8] Indeed, in a uniform-complexity treatment, the string z must be taken from a polynomially-sampleable ensemble.

(S, L, F). Such a challenge is answered by selecting a plaintext x according to the distribution specified by S (i.e., $x = S(r)$ where r is uniformly selected in the set of strings of adequate length), and providing its encryption along with the leakage $L(x)$. The adversary's goal is to guess $F(x)$, and semantic security amount to saying that the adversary's success probability can be matched by a corresponding benign algorithm that is only given $L(x)$. It is crucial to require that the challenge template produced by the corresponding algorithm is distributed similarly to the challenge template produced by the adversary.[9] For simplicity, we require below that these distributions be identical, but it would have sufficed to require that they be computationally indistinguishable. (As in Definition 2.1, both the real adversary and its benign simulator are decoupled into two part (and the first part passes state information to the second part).)

Definition 3.1 (semantic security under chosen ciphertext attacks):

For public-key schemes: *A public-key encryption scheme, (G, E, D), is said to be* semantically secure under a-priori chosen ciphertext attacks (CCA1) *if for every pair of probabilistic polynomial-time oracle machines, A_1 and A_2, there exists a pair of probabilistic polynomial-time algorithms, A'_1 and A'_2, such that the following two conditions hold:*

1. *For every positive polynomial $p(\cdot)$, and all sufficiently large n and $z \in \{0,1\}^{\mathrm{poly}(n)}$:*

$$\begin{array}{cl} & \mathsf{Pr}\left[\begin{array}{rl} v = F(x) & \text{where} \\ & (e,d) \leftarrow G(1^n) \\ & ((S,L,F),\sigma) \leftarrow A_1^{E_e,D_d}(e,z) \\ & c \leftarrow (E_e(x), L(x))\,, \text{ where } x \leftarrow S(U_{\mathrm{poly}(n)}) \\ & v \leftarrow A_2^{E_e}(\sigma, c) \end{array}\right] \\ < & \mathsf{Pr}\left[\begin{array}{rl} v = F(x) & \text{where} \\ & ((S,L,F),\sigma) \leftarrow A'_1(1^n, z) \\ & x \leftarrow S(U_{\mathrm{poly}(n)}) \\ & v \leftarrow A'_2(\sigma, L(x)) \end{array}\right] + \frac{1}{p(n)} \end{array}$$

   where U_m denotes the uniform distribution over $\{0,1\}^m$.

2. *For every n and z, the first element* (i.e., the (S, L, F) part) *in the random variables $A'_1(1^n, z)$ and $A_1^{E_{G_1(1^n)},D_{G_2(1^n)}}(G_1(1^n), z)$ are identically distributed.*

Semantic security under a-posteriori chosen ciphertext attacks (CCA2) *is defined analogously, except that A_2 is given oracle access to both E_e and D_d with the restriction that when given the challenge $c = (c', c'')$, machine A_2 is not allowed to make the query c' to the oracle D_d.*

For private-key schemes: *The definition is identical except that algorithm A_1 gets the security parameter 1^n instead of the encryption-key e.*

Clearly, the a-posteriori version of Definition 3.1 implies its a-priori version, which in turn implies standard passive security.

[9] Thus if the real adversary asks for (and obtains) very informative leaks then the same is allowed to the corresponding benign algorithm, but if the real adversary asks for no informative leaks then the corresponding benign algorithm cannot ask for very informative leaks.

3.2 Equivalence of semantic security and ciphertext-indistinguishability

We show that the two formulations of CCA-security (i.e., semantic security and indistinguishable encryptions) are in fact equivalent.

Theorem 3.2 (equivalence of definitions for CCA): *A public-key* (resp., private-key) *encryption scheme (G, E, D) is semantically secure under a-priori CCA if and only if it has indistinguishable encryptions under a-priori CCA. An analogous claim holds for a-posteriori CCA.*

Proof Sketch: We adapt the known proof for the case of passive attacks (cf. [15, 11]) to the current setting. The adaptation is quite easy, and we focus on the case of a-posteriori CCA security (while commenting on the case of a-priori CCA security).

We start by showing that indistinguishable encryptions implies semantic security. Specifically, given an CCA-adversary (A_1, A_2) we construct the following matching algorithm A'_1, A'_2:

1. $A'_1(1^n, z) \overset{\text{def}}{=} (\tau, \sigma')$, where (τ, σ') is generated as follows:

   First, A'_1 generates an instance of the encryption scheme; that is, A'_1 lets $(e, d) \leftarrow G(1^n)$. Next, A'_1 invokes A_1, while emulating the oracles E_e and D_d, and obtains $((S, L, F), \sigma) \leftarrow A_1^{E_e, D_d}(1^n, z)$. Finally, A'_1 sets $\sigma' \overset{\text{def}}{=} (\sigma, e, d, 1^m)$, where m equals the number of output bits in S.

   (In case of a-priori CCA security, we may also set $\sigma' \overset{\text{def}}{=} (\sigma, e, 1^m)$. Note that the generated key-pair (e, d) allows A'_1 to emulate the encryption and decryption oracles E_e and D_d.)

2. $A'_2((\sigma, e, d, 1^m), \gamma) \overset{\text{def}}{=} A_2^{E_e, D_d}(\sigma, (E_e(1^m), \gamma))$, where typically $\gamma = L(x)$, $m = |x|$ and $x \leftarrow S(U_{\text{poly}(n)})$. Again, A'_2 uses the key-pair (e, d) in order to emulate the oracles E_e and D_d.

   (As in the previous item, in case of a-priori CCA security, we may also let $A'_2((\sigma, e, 1^m), \gamma) \overset{\text{def}}{=} A_2^{E_e}(\sigma, (E_e(1^m), \gamma))$.)

Since A'_1 merely emulates the generation of a key-pair and the actions of A_1 with respect to such a pair, the equal distribution condition (i.e., Item 2 in Definition 3.1) holds. Using the (corresponding) indistinguishability of encryption hypothesis, we show that (even in the presence of the encryption oracle E_e and a restricted decryption oracle D_d) the distributions $(\sigma, (E_e(x), L(x)))$ and $(\sigma, (E_e(1^{|x|}), L(x)))$ are indistinguishable (in particular by A_2), where $(e, d) \leftarrow G(1^n)$, $((S, L, F), \sigma) \leftarrow A_1^{E_e, D_d}(y, z)$ (with $y = e$ or $y = 1^n$ depending on the model), and $x \leftarrow S(U_{\text{poly}(n)})$. The main thing to notice is that the oracle queries made by a possible distinguisher of the above distributions can be handled by a distinguisher of encryptions (as in Definition 2.1), by passing these queries to its own oracles.[10] It follows that indistinguishable encryptions (as per Definition 2.1) implies semantic security (as per Definition 3.1).

[10] Suppose that given $((S, L, F), \sigma)$ generated by $A_1^{E_e, D_d}(y, z)$ and oracle access to E_e and D_d, where $(e, d) \leftarrow G(1^n)$ (and y is as above), one can distinguish $(\sigma, (E_e(x), L(x)))$ and $(\sigma, (E_e(1^{|x|}), L(x)))$, where $x \leftarrow S(U_{\text{poly}(n)})$ (and one does not query D_d on the input ciphertext). Then we obtain a distinguisher as in Definition 2.1 as follows. The first part of the distinguisher invokes A_1 (while answering its oracle queries by forwarding these queries to its own E_e and D_d oracle), and obtains $((S, L, F), \sigma) \leftarrow A_1^{E_e, D_d}(y, z)$. It sets $x^{(1)} \leftarrow S(U_{\text{poly}(n)})$ and $x^{(2)} = 1^{|x^{(1)}|}$. and outputs $((x^{(1)}, x^{(2)}), (\sigma, L(x^{(1)})))$. That is, $(x^{(1)}, x^{(2)})$ is the challenge template, and it is answered with $E_e(x^{(i)})$, where i is either 1 or 2. The second part of the new distinguisher, gets as input a challenge ciphertext $\alpha \leftarrow E_e(x^{(i)})$ and the state generated by the first part $(\sigma, L(x^{(1)}))$, and invokes the distinguisher of the contradiction hypothesis with input $(\sigma, (\alpha, L(x^{(1)})))$, while answering its oracle queries by forwarding these queries to its own E_e and D_d oracles. Indeed, the new distinguisher does not query D_d on α, because the original distinguisher was guaranteed not to do so. Thus, the new distinguisher violates the condition in Definition 2.1, in contradiction to the hypothesis that (G, E, D) has indistinguishable encryptions.

We now turn to the opposite direction. Here the construction of a challenge template (as per Definition 3.1) is analogous to the corresponding construction in passive attack case. Specifically, using the "indistinguishable-encryptions challenge template" $(x^{(1)}, x^{(2)})$, we construct the following "semantic security challenge" (S, H, F):

- The circuit S samples uniformly in $\{x^{(1)}, x^{(2)}\}$.
- The function F satisfies $F(x^{(1)}) = 1$ and $F(x^{(2)}) = 0$.
- The function L is defined arbitrarily subject to $L(x^{(1)}) = L(x^{(2)})$.

Again, the thing to notice is that the oracle queries made by a possible distinguisher of encryptions (as in Definition 2.1) can be handled by the semantic-security adversary, by passing these queries to its own oracles. We derive a contradiction to the hypothesis that (G, E, D) satisfies Definition 3.1, and the theorem follows. ■

4 Semantic Security Under Multiple-Challenge CCA2

We now consider general attacks during which several challenge templates may be produced at arbitrary times and possibly interleaved with decryption queries, continuing the discussion in Section 1.2, Each of these challenge templates will be answered similarly to the way such templates were answered above (i.e., by selecting a plaintext from the specified distribution and providing its encryption together with the specified partial information). Unlike Section 3, we will even allow attacks that make decryption queries regarding ciphertexts obtained as (part of the) answer to previous challenge templates. After such an attack, the adversary will try to obtain information about the unrevealed plaintexts, and security holds if its success probability can be met by a corresponding benign adversary that does not see the ciphertexts. Indeed, the above discussion requires clarification and careful formulation, provided next.

4.1 Definition of Multiple Message Attacks

We start with a description of *the actual attacks*. It will be convenient to change the formalism and consider the generation of challenge templates as challenge queries that are answered by a special oracle called the tester, and denoted $T_{e,r}$, where e is an encryption-key and r is a random string of adequate length.[11] On query a *challenge template* of the form (S, L), where S is a sampling circuit and L is a function (evaluation circuit), the (randomized) oracle $T_{e,r}$ returns the pair $(E_e(S(r)), L(r))$. (Indeed, we are further generalizing the attack by allowing the leak L to be an arbitrary function of r, rather only a function of the plaintext $S(r)$ (or all prior plaintexts).) Note that r is *not* known to the adversary, and that this formalism generalizes the one in Section 1.2. A multiple-challenge CCA is allowed queries to $T_{e,r}$ as well as *unrestricted* queries to both E_e and the corresponding D_d, including decryption queries referring to previously obtained challenge ciphertexts. It terminates by outputting a function F and a value v, hoping that $F(r) = v$. (Again, this generalizes the description in Section 1.2, where F was applied to the sequence of generated plaintexts $(x^1, ..., x^t)$.) Note that the description of F may encode various information gathered by the adversary during its attack (e.g., it may even encode its entire computation transcript).

[11] The formulation of Section 1.2 is obtained by letting $r = (r^1, ..., r^t)$, and making the i^{th} sampling circuit only refer to $(r^1, ..., r^i)$. Similarly, the i^{th} leak circuit should be restricted to depend only on $S^1(r^1), ..., S^i(r^1, ..., r^i)$. Actually, given the independence of S from L, one could have replaced the challenge queries by two types of queries: leak queries that correspond to the L's, and encrypted leak queries that correspond to the S's.

We now turn to describe *the benign adversary* (which does not see the ciphertexts). Such an adversary is given oracle access to a corresponding oracle, T_r, that behave as follows. On query a challenge template of the form (S, L), the oracle returns $L(r)$. (Again, r is not known to the adversary.) Like the real adversary, the benign adversary also terminates by outputting a function F and a value v, hoping that $F(r) = v$.

Security amounts to asserting that the effect of any efficient multiple-challenge CCA can be simulated by a efficient benign adversary that does not see the ciphertexts. As in Definition 3.1, the simulation has to satisfy two conditions:

1. The probability that $F(r) = v$ in the CCA must be met by the probability that a corresponding event holds in the benign model (where the adversary does not see the ciphertexts).

2. The challenge queries as well as the function F should be distributed similarly in the two models. Actually, each decryption query (of the real attacker) that refers to a ciphertext c that is contained in the answer given to a challenge query (S, L) is considered (or counted) as a (fictitious) challenge query (S, S). Note that this convention is justified by the fact that the challenge query (S, S) is equivalent to the decryption query c (followed by the encryption query $x = D_d(c)$). Put in other words, if the real adversary made a decryption query that refers to a ciphertext c contained in the answer given to the challenge (S, L) (and thus obtained $D_d(c) = D_d(E_e(S(r))) = S(r)$), then it is only fair that we allow the benign adversary (which sees no ciphertexts) to make the challenge query (S, S) and so obtain $S(r)$.

In order to obtain the actual definition, we need to define the trace of the execution of the above two types of adversaries. For a multiple-challenge CCA adversary, denoted A, the trace is defined as the sequence of challenge queries made during the attack, augmented by (fictitious) challenge queries such that the (fictitious challenge) query (S, S) is included if and only if the adversary made a decryption query c such that $(c, \cdot)$ is the answer given to a previous challenge query of the form $(S, \cdot)$. For the benign adversary, denoted B, the trace is defined as the sequence of challenge queries made during the attack.

Definition 4.1 (multiple-challenge CCA security):

For public-key schemes: *A public-key encryption scheme, (G, E, D), is said to be* secure under multiple-challenge chosen ciphertext attacks *if for every probabilistic polynomial-time oracle machine A there exists a probabilistic polynomial-time oracle machine B such that the following two conditions hold:*

1. *For every positive polynomial $p(\cdot)$, and all sufficiently large n and $z \in \{0,1\}^{\mathrm{poly}(n)}$:*

$$\mathsf{Pr}\left[\begin{array}{ll} v = F(r) & \text{where} \\ & (e,d) \leftarrow G(1^n) \text{ and } r \leftarrow U_{\mathrm{poly}(n)} \\ & (F,v) \leftarrow A^{E_e, D_d, T_{e,r}}(e,z) \end{array}\right]$$

$$< \quad \mathsf{Pr}\left[\begin{array}{ll} v = F(r) & \text{where} \\ & r \leftarrow U_{\mathrm{poly}(n)} \\ & (F,v) \leftarrow B^{T_r}(1^n, z) \end{array}\right] + \frac{1}{p(n)}$$

2. *The following two probability ensembles, indexed by $n \in \mathbb{N}$ and $z \in \{0,1\}^{\mathrm{poly}(n)}$, are computationally indistinguishable:*

   (a) The trace of $A^{E_{G_1(1^n)}, D_{G_2(1^n)}, T_{G_1(1^n), U_{\mathrm{poly}(n)}}}(G_1(1^n), z)$, augmented by its output.

(b) The trace of $B^{T_{U_{\mathrm{poly}(n)}}}(1^n, z)$ *augmented by its output.*

For private-key schemes: *The definition is identical except that machine* A *gets the security parameter* 1^n *instead of the encryption-key* e.

To get more comfortable with Definition 4.1, consider the special case in which the real CCA adversary does not make decryption queries to ciphertexts obtained as part of answers to challenge queries. (In the proof of Theorem 4.2, such adversaries will be called canonical and will be shown to be as powerful as the general ones.) The trace of such adversaries equals the sequence of challenge queries made during the attack, which simplifies Condition 2.

4.2 Relation to ordinary CCA2-security

It is easy to see that Definition 4.1 implies ordinary CCA2-security (e.g., Definition 2.1).[12] The more important fact (proven below) is that CCA2-security implies security under multiple-challenge CCA (i.e., Definition 4.1).

Theorem 4.2 (a-posteriori-CCA implies Definition 4.1): *Let* (G, E, D) *be a public-key* (resp., private-key) *encryption scheme that is secure under a-posteriori CCA. Then* (G, E, D) *is secure under multiple-challenge chosen ciphertext attacks.*

Proof Sketch: As a bridge between the multiple-challenge CCA and the corresponding benign adversary that does not see the ciphertext, we consider canonical adversaries that can *perfectly simulate* any multiple-challenge CCA without making decryption queries to ciphertexts obtained as part of answers to challenge queries. Instead, these canonical adversaries make corresponding queries of the form (S, S), where $(S, \cdot)$ is the challenge-query that was answered with the said ciphertext. Specifically, suppose that a multiple-challenge CCA has made the challenge query (S, L), which was answered by $(c, L(r))$, where $c = E_e(S(r))$, and at a later stage makes the decryption query c, which is to be answered by $D_d(c) = S(r)$. Then, the corresponding canonical adversary makes the challenge query (S, L) as the original adversary, receiving the same pair $(c, L(r))$, but later instead of making the decryption query c the canonical adversary makes the challenge query (S, S), which is answered by $S(r) = D_d(c)$. Note that the trace of the corresponding canonical adversary is identical to the trace of the original CCA adversary (and the same holds with respect to their outputs).

Thus, given an a-posteriori-CCA secure encryption scheme, it suffices to establish Definition 4.1 when the quantification is restricted to *canonical* adversaries A. Indeed, as in the proof of Theorem 3.2, we construct a benign adversary B in the natural manner: On input $(1^n, z)$, machine B generates $(e, d) \leftarrow G(1^n)$, and invokes A on input (y, z), where $y = e$ if we are in the public-key case and $y = 1^n$ otherwise. Next, B emulates all oracles expected by A, while using its own oracle T_r. Specifically, the oracles E_e and D_d are perfectly emulated by using the corresponding keys (known to B), and the oracle $T_{e,r}$ is (non-perfectly) emulated using the oracle T_r (i.e., the query (S, L) is forwarded to T_r, and the answer $L(r)$ is augmented with $E_e(1^m)$, where m is the number of output bits in S). Note that the latter emulation (i.e., the answer $(E_e(1^{|S(r)|}), L(r)))$ is non-perfect since the answer of $T_{e,r}$ would have been $(E_e(S(r)), L(r))$, yet (as we shall show) A cannot tell the difference.

[12] This can be shown by considering the special case (of Definition 4.1) in which the adversary makes a *single* challenge query, and does not make a decryption query that refers to the ciphertext provided as answer. Using ideas as in the second part of the proof of Theorem 3.2, this special case of Definition 4.1 implies Definition 2.1 (as a special case).

In order to show that B satisfies both conditions of Definition 4.1 (w.r.t the above A), we will show that the following two ensembles are computationally indistinguishable:

1. The global view in real attack of A on (G, E, D). That is, we consider the output of the following experiment:
   (a) $(e, d) \leftarrow G(1^n)$ and $r \leftarrow U_{\mathrm{poly}(n)}$.
   (b) $(F, v) \leftarrow A^{E_e, D_d, T_{e,r}}(y, z)$, where $y = e$ if we are in the public-key case and $y = 1^n$ otherwise. Furthermore, we let $((S^1, L^1), ..., (S^t, L^t))$ denote the trace of the execution $A^{E_e, D_d, T_{e,r}}(y, z)$.
   (c) The output is $((S^1, L^1), ..., (S^t, L^t)), (F, v), r$.
2. The global view in an attack emulated by B. That is, we consider the output of an experiment as above, except that $A^{E_e, D_d, T_{e,r}}(y, z)$ is replaced by $A^{E_e, D_d, T'_{e,r}}(y, z)$, where on query (S, L) the oracle $T'_{e,r}$ replies with $(E_e(1^{|S(r)|}), L(r))$ rather than with $(E_e(S(r)), L(r))$.

Note that computational indistinguishability of the above ensembles immediately implies Condition 2 of Definition 4.1, whereas Condition 1 also follows because using r we can determine whether or not $F(r) = v$ holds (for (F, v)). Also note that the above ensembles may be computationally indistinguishable only in case A is *canonical* (which we have assumed to be the case).[13]

The computational indistinguishability of the above ensembles is proven using a hybrid argument, which in turn relies on the hypothesis that (G, E, D) has indistinguishable encryptions under a-posteriori-CCAs. Specifically, we introduce $t + 1$ *mental experiments* that are hybrids of the above two ensembles (which we wish to relate). Each of these mental experiments is given oracle access to E_e and D_d, where $(e, d) \leftarrow G(1^n)$ is selected from the outside. The ith hybrid experiment uses these two oracles (in addition to y which equals e in the public-key case and 1^n otherwise), in order to emulate an execution of $A^{E_e, D_d, \Pi^i_{e,r}}(y, z)$, where r is selected by the experiment itself and $\Pi^i_{e,r}$ is a hybrid of $T_{e,r}$ and $T'_{e,r}$. Specifically, $\Pi^i_{e,r}$ is a history-dependent process that answers like $T_{e,r}$ on the first i queries and like $T'_{e,r}$ on the rest. Thus, for $i = 0, ..., t$, we define the ith hybrid experiment as a process that given y (which equals either e or 1^n) and oracle access to E_e and D_d, where $(e, d) \leftarrow G(1^n)$, behaves as follows:

1. The process selects $r \leftarrow U_{\mathrm{poly}(n)}$.
2. The process emulates an execution of $A^{E_e, D_d, \Pi^i_{e,r}}(y, z)$, where $y = e$ if we are in the public-key case and $y = 1^n$ otherwise, by using the oracles E_e and D_d. Specifically, the answers of $\Pi^i_{e,r}$ are emulated using the knowledge of r and oracle access to E_e: the jth query to $\Pi^i_{e,r}$, denoted (S^j, L^j), is answered by $(E_e(S^j(r)), L^j(S^j(r)))$ if $j \leq i$ and is answered by $(E_e(1^{|S^j(r)|}), L^j(S^j(r)))$ otherwise. (The process answers A's queries to E_e and D_d by forwarding them to its own corresponding oracles.)
3. As before, (F, v) denotes the output of $A^{E_e, D_d, \Pi^i_{e,r}}(y, z)$ and $((S^1, L^1), ..., (S^t, L^t))$ denotes its trace. The process outputs $((S^1, L^1), ..., (S^t, L^t)), (F, v), r$.

[13] Non-canonical adversaries can easily distinguish the two types of views by distinguishing the oracle $T_{e,r}$ from oracle $T'_{e,r}$. For example, suppose we make a challenge query with a sampling-circuit S that generates some distribution over $\{0,1\}^m \setminus \{1^m\}$, next make a decryption query on the ciphertext obtained in the challenge query, and output the answer. Then, in case we query the oracle $T_{e,r}$, we output $D_d(E_e(S(r))) \neq 1^m$; whereas in case we query the oracle $T'_{e,r}$, we output $D_d(E_e(1^m)) = 1^m$. Recall that, at this point, we are guaranteed that A is canonical (and indeed it might have been derived for perfectly-emulating some non-canonical A'). An alternative way of handling non-canonical adversaries is to let B handled the disallowed (decryption) queries by making the corresponding challenge query, and returning its answer rather than the decryption value. (Note that B that emulates $T'_{r,e}$ can detect which queries are disallowed.)

Note that that since A is *canonical*, none of the D_d-queries equals a ciphertext obtained as part of the answer of a $\Pi^i_{e,r}$-query.

Clearly, the distribution of the 0-hybrid is identical to the distribution of the global view in an attack emulated by B, whereas the distribution of the t-hybrid is identical to the distribution of the global view in a real attack by A. On the other hand, distinguishing the i-hybrid from the $(i+1)$-hybrid yields a successful *a-posteriori-CCA* (in the sense of distinguishing encryptions). That is, assuming that one can distinguish the i-hybrid from the $(i+1)$-hybrid, we construct a a-posteriori-CCA adversary (as per Definition 2.1) as follows. For $(e,d) \leftarrow G(1^n)$, given $y=e$ if we are in the public-key case and $y=1^n$ otherwise, the attacker (having oracle access to E_e and D_d) behaves as follows

1. The attacker selects $r \leftarrow U_{\mathrm{poly}(n)}$.
2. The attacker emulates an execution of $A^{E_e,D_d,\Pi^j_{e,r}}(y,z)$, where $j \in \{i,i+1\}$ (is unknown to the attacker), as follows. The queries to E_e and D_d are answered by using the corresponding oracles available to the attacker, and the issue is answering the queries to $\Pi^j_{e,r}$. The first i queries to $\Pi^j_{e,r}$ are answered as in both $\Pi^i_{e,r}$ and $\Pi^{i+1}_{e,r}$ (i.e., query (S,L) is answered by $(E_e(S(r)),L(r)))$, and the last $t-(i+1)$ queries are also answered as in both $\Pi^i_{e,r}$ and $\Pi^{i+1}_{e,r}$ (i.e. by $(E_e(1^{|S(r)|}),L(r))$, this time). The $i+1$ query, denoted (S^{i+1},L^{i+1}), is answered by producing the *challenge template* $(S^{i+1}(r),1^{|S^{i+1}(r)|})$, which is answered by the challenge ciphertext c (where $c \in \{E_e(S^{i+1}(r)),E_e(1^{|S^{i+1}(r)|})\}$), and replying with $(c,L^{i+1}(r))$.

   Note that if $c=E_e(S^{i+1}(r))$ then we emulate $\Pi^{i+1}_{e,r}$, whereas if $c=E_e(1^{|S^{i+1}(r)|})$ then we emulate $\Pi^i_{e,r}$.
3. Again, (F,v) denotes the output of $A^{E_e,D_d,\Pi^j_{e,r}}(y,z)$, and $((S^1,L^1),...,(S^t,L^t))$ denotes its trace. The attacker feeds $((S^1,L^1),...,(S^t,L^t)),(F,v),r$ to the hybrid distinguisher (which we have assumed to exist towards the contradiction), and outputs whatever the latter does.

The above is an a-posteriori-CCA as in Definition 2.1: it produces a *single* challenge (i.e., the pair of plaintexts $(S^{i+1}(r),1^{|S^{i+1}(r)|})$), and distinguishes the case it is given the ciphertext $c=E_e(S^{i+1}(r))$ from the case it is given the ciphertext $c=E_e(1^{|S^{i+1}(r)|})$, without querying D_d on the challenge ciphertext c. The last assertion follows by the hypothesis that A is *canonical*, and so none of the D_d-queries that A makes equals the ciphertext c obtained as (part of) the answer to the $i+1$st $\Pi^j_{e,r}$-query. Thus, distinguishing the $i+1$st and ith hybrids implies distinguishing encryptions under an a-posteriori-CCA, which contradicts our hypothesis regarding (G,E,D). The theorem follows. ■

Acknowledgments

We thank Shafi Goldwasser for useful discussions.

References

[1] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations among notions of security for public-key encryption schemes. Advances in Cryptology – CRYPTO'98 Proceedings, Springer, Lecture Notes in Computer Science (Vol. 1462), pages 26–45.

[2] M. Bellare, A. Boldyreva and S. Micali. *Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements*, Advances in Cryptology – Eurocrypt 2000 Proceedings, Lecture Notes in Computer Science Vol. 1807, Springer, 2000, pages 259–274.

[3] D. Bleichenbacher, *Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS # 1*, Advances in Cryptology - Crypto'98, Lecture Notes in Computer Science No. 1462, Springer Verlag, 1998, pp. 1–12.

[4] M. Blum, A. De Santis, S. Micali, and G. Persiano. Non-Interactive Zero-Knowledge Proof Systems. *SIAM Journal on Computing*, Vol. 20, No. 6, pages 1084–1118, 1991. (Considered the journal version of [5].)

[5] M. Blum, P. Feldman and S. Micali. Non-Interactive Zero-Knowledge and its Applications. In *20th ACM Symposium on the Theory of Computing*, pages 103–112, 1988. See [4].

[6] R. Cramer and V. Shoup. A Practical Public-Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attacks. In *Crypto98*, Springer-Verlag Lecture Notes in Computer Science (Vol. 1462), pages 13–25.

[7] A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano and A. Sahai. Robust Non-interactive Zero-Knowledge. In *Crypto01*, Springer Lecture Notes in Computer Science (Vol. 2139), pages 566–598.

[8] W. Diffie, and M.E. Hellman. New Directions in Cryptography. *IEEE Trans. on Info. Theory*, IT-22 (Nov. 1976), pages 644–654.

[9] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *SIAM Journal on Computing*, Vol. 30, pages 391–437, 2000. Preliminary version in *23rd ACM Symposium on the Theory of Computing*, 1991.

[10] O. Goldreich. A Uniform Complexity Treatment of Encryption and Zero-Knowledge. *Journal of Cryptology*, Vol. 6, No. 1, pages 21–53, 1993.

[11] O. Goldreich. *Encryption Schemes – fragments of a chapter*. December 1999. Available from `http://www.wisdom.weizmann.ac.il/~oded/foc-book.html`

[12] O. Goldreich. *Foundation of Cryptography – Basic Tools*. Cambridge University Press, 2001.

[13] O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions. *Journal of the ACM*, Vol. 33, No. 4, pages 792–807, 1986.

[14] O. Goldreich, S. Goldwasser, and S. Micali. On the Cryptographic Applications of Random Functions. In *Crypto84*, Springer-Verlag Lecture Notes in Computer Science (Vol. 263), pages 276–288, 1985.

[15] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Science*, Vol. 28, No. 2, pages 270–299, 1984. Preliminary version in *14th ACM Symposium on the Theory of Computing*, 1982.

[16] S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, Vol. 18, pages 186–208, 1989. Preliminary version in *17th ACM Symposium on the Theory of Computing*, 1985.

[17] S. Goldwasser, S. Micali and P. Tong. Why and How to Establish a Private Code in a Public Network. In *23rd IEEE Symposium on Foundations of Computer Science*, 1982, pages 134–144.

[18] J. Katz and M. Yung. Complete Characterization of Security Notions for Probabilistic Private-Key Encryption. In *32nd ACM Symposium on the Theory of Computing*, pages 245–254, 2000.

[19] Y. Lindell. A Simpler Construction of CCA2-Secure Public-Key Encryption Under General Assumptions. In preparation, 2002.

[20] M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure Against Chosen Ciphertext Attacks. In *22nd ACM Symposium on the Theory of Computing*, pages 427-437, 1990.

[21] A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Achieving Chosen-Ciphertext Security. In *40th IEEE Symposium on Foundations of Computer Science*, pages 543–553, 1999.

[22] A. Sahai. Improved Constructions Achieving Chosen-Ciphertext Security. In preparation, 2001. See [7].

[23] C.E. Shannon. *Communication Theory of Secrecy Systems, Bell Sys. Tech. J.*, Vol. 28, pages 656–715, 1949.

[24] C. Rackoff and D.R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto91*, Springer Verlag, Lecture Notes in Computer Science (Vol.), pages 433–444.

[25] R. Rivest, A. Shamir and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *CACM*, Vol. 21, Feb. 1978, pages 120–126.

[26] Y. Watanabe, J. Shikata and H. Imai *Equivalence between semantic security and indistinguishability against chosen ciphertext attacks*, Cryptology ePrint Archive: Report 2002/033

[27] A.C. Yao. Theory and Application of Trapdoor Functions. In *23rd IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.