WartaCrypt 2004

ON MODELING IND-CCA SECURITY IN CRYPTOGRAPHIC PROTOCOLS

D. Hofheinz†, J. Müller-Quade†, and R. Steinwandt†

† IAKS, Arbeitsgruppe Systemsicherheit, Prof. Dr. Th. Beth,
Fakultät für Informatik, Universität Karlsruhe, Germany.
E-Mail: {hofheinz,muellerq,steinwan}@ira.uka.de.

ABSTRACT. Two common notions of security for public key encryption schemes are shown to be equivalent: we prove that *indistinguishability against chosen-ciphertext attacks* (IND-CCA) is in fact polynomially equivalent to (yet "slightly" weaker than) *securely realizing the ideal functionality* $\mathcal{F}_{\text{PKE}}$ in the general modeling of cryptographic protocols of [Can01a]. This disproves in particular the claim that security in the sense of IND-CCA *strictly* implies security in the sense of realizing $\mathcal{F}_{\text{PKE}}$ (see [Can01a]). Moreover, we give concrete reductions among such security notions and show that these relations hold for both uniform and non-uniform adversarial entities.

1. Introduction

Judging the security of public key encryption schemes using formal methods has been introduced in the pioneering work [GM84], creating the notion of *semantic security* of a given public key cryptosystem. To treat situations in which an attacker does not remain "passive", but has access to a decryption facility, several notions of security for public key cryptosystems have been proposed subsequently; in particular, when not considering a random oracle available [BR95, Sho01], "indistinguishability of encryptions with respect to chosen-ciphertext attacks" (IND-CCA, see [RS92]) is the most stringent generally accepted security notion for public key cryptography (see, for example, [BDPR98]).

On the other hand, when considering *concrete* reductions of adversaries and comparing their *exact* complexities and running times, it turns out that a notion called "real-or-random security with respect to chosen-ciphertext attacks" (ROR-CCA) even implies IND-CCA strictly [BDJR97]. (Note that in [BDJR97], definitions and results are motivated by symmetric cryptography; however, as mentioned therein, all definitions and results immediately carry over to the setting of public key cryptography.)

Now in [Can01a], a general framework for describing security properties of multi-party protocols is proposed. In this framework the multi-party protocol in question is compared to an *ideal functionality* which represents what we ideally expect our protocol to do. In particular, to capture on a high level what we expect from a public key cryptosystem, in [Can01a] an ideal functionality $\mathcal{F}_{\text{PKE}}$ is described (see also Appendix A). Indeed, a public key cryptosystem can be regarded as a protocol aiming at securely realizing the ideal functionality $\mathcal{F}_{\text{PKE}}$,

Keywords: formal cryptography, cryptographic protocols, probabilistic encryption.

and in [Can01a] it is claimed that, for a public key cryptosystem, IND-CCA security *strictly* implies the property of securely realizing $\mathcal{F}_{\mathrm{PKE}}$.

Unfortunately, the proof of the implication in question assumes adversaries attacking in the IND-CCA sense to be *non-uniform* machines, in contrast to the common representation of such adversaries as algorithms *without* external input (see, e. g., [BDJR97, BDPR98]). Furthermore, in Section 2 we show that the counterexample given in [Can01a, Section 8.2.2] for the "strictly" statement does not apply. In fact, subsequently we prove that security in the ROR-CCA sense and securely realizing $\mathcal{F}_{\mathrm{PKE}}$ in the modeling of [Can01a] are equivalent notions of security for a public key cryptosystem, *if* we restrict completely to uniform or non-uniform adversarial entities. This implies (polynomial) equivalence with the notion of IND-CCA *with respect to the chosen class of adversaries*. More specifically, we give *concrete* reductions (cf. [BDJR97]) between adversaries attacking some public key cryptosystem P in the ROR-CCA sense and distinguishers between the ideal functionality $\mathcal{F}_{\mathrm{PKE}}$ and P in the sense of [Can01a]; it thereby turns out that we have a "tight" correspondence between them. As a technical tool which might be of interest in itself, we prove that the composition theorem of [Can01a] still holds when restricting to uniform environments with polynomial total running time.

2. Preliminaries

We start by fixing some notation. For more details on formal security notions like ROR-CCA and on multi-party computations we refer to [BDJR97] and [Can01a], respectively. A short restatement of the most relevant definitions for the sequel can also be found in the appendix.

To be able to compare adversarial entities in the sense of [Can01a] to adversaries attacking a public key cryptosystem P in the sense of [BDJR97], we will regard an adversary in the latter sense as a family $A = \{A_k\}$ of interactive Turing machines (ITMs) where ITM is to be understood as in [Can01a]. When interpreting algorithms as (families of) ITMs, we will assume a convenient definition of "code size" given. One could think here of a suitable combination of the number of states and the size of the alphabet of the Turing machine in question. Furthermore, a sequence $A = \{A_k\}_{k\in\mathbb{N}}$ of ITMs will be called *polynomially bounded*, if there is a single polynomial p, such that for *every* A_k we have

1. the code size of A_k is less than $p(k)$, and
2. when activated, A_k will enter either a waiting or a halt state after running at most $p(k)$ steps.

Finally, a sequence $A = \{A_k\}_{k\in\mathbb{N}}$ of ITMs will be called a *non-uniform family of ITMs*. If $A_1 = A_k$ for all $k \in \mathbb{N}$, then the family A is said to be *uniform*.

2.1. Security of public key schemes against chosen ciphertext attacks

At this point, we should clarify what exactly we mean by a public key encryption scheme: a public key encryption scheme is a triple $P = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ of algorithms which can be executed by a polynomially bounded, uniform family of ITMs. P consists of the *key generation* algorithm $\mathcal{K}$, which takes as input the *security*

parameter k and outputs a *private key-public key* pair (d, e). The *encryption* algorithm $\mathcal{E}$ (parametrized by the public key e) outputs on input of a *plaintext* m a corresponding *ciphertext* c. Finally, the *decryption* algorithm $\mathcal{D}$ (parametrized by the secret key d) outputs on input of a ciphertext c either a plaintext m or a special symbol indicating that the ciphertext c is invalid. We insist on $\mathcal{D}_d(\mathcal{E}_e(m)) = m$ for all private key-public key pairs (d, e), plaintexts m, and at all times. Also, we will freely identify a public key encryption scheme P with the corresponding protocol π_P geared towards realizing $\mathcal{F}_{\text{PKE}}$ (see [Can01a, Section 8.2.2] for more details on π_P).

For an *adversary* $A = \{A_k\}_{k\in\mathbb{N}}$ attacking some public key encryption scheme in the ROR-CCA sense—i. e., taking part in one of the respective experiments described in [BDJR97]—we define the *(total) running time* of A_k to be the worst-case number of steps *any* of the two ROR-CCA experiments defined in [BDJR97] runs (counting the steps used for key generation, encryptions, decryptions, and of course for executing A_k itself) *plus* the code size of A_k. This notion coincides with the notion of running time defined in [BDJR97].

The *advantage* for such an adversary A in the ROR-CCA game is defined in [BDJR97] through

$$\mathbf{Adv}^{\text{ror-cca}}_{P,A}(k) := \mathbf{P}(\mathbf{Exp}^{\text{ror-cca-1}}_{P,A}(k) = 1) - \mathbf{P}(\mathbf{Exp}^{\text{ror-cca-0}}_{P,A}(k) = 1).$$

For $U \in \{\text{uniform, non-uniform}\}$, we call a public key encryption scheme P secure in the sense of ROR-CCA with respect to U-adversaries if for *every* U-adversary A attacking P in the ROR-CCA sense and having a polynomial (in the security parameter k) total running time, $\mathbf{Adv}^{\text{ror-cca}}_{P,A}(k)$ is a negligible function in the security parameter k. At this, a function $f : \mathbb{N} \to \mathbb{R}$, $k \mapsto f(k)$ is *negligible* (in k), if for each $c \in \mathbb{N}$ there is a $k_c \in \mathbb{N}$ such that $|f(k)| < k^{-c}$ for all $k > k_c$.

One easily verifies that all the reductions of adversaries given in [BDJR97] still apply with these conventions, both for uniform and non-uniform adversaries; in particular, ROR-CCA security with respect to uniform adversaries is exactly the same notion as the one defined in [BDJR97], whereas ROR-CCA security with respect to non-uniform adversaries seems to be a stronger notion.

2.2. Security with respect to realizing $\mathcal{F}_{\text{PKE}}$

We will assume all participants in a protocol (including adversarial entities) to be polynomially bounded.[1] In [Can01a], non-uniformity is expressed via an external input $z = z_k$ (depending on the security parameter k). For relating security in the sense of [Can01a] to ROR-CCA or IND-CCA security, we assume the used extra input to be "hardwired", and therefore utilize families $A^{(z)} = \{A^{(z)}_k\}_{k\in\mathbb{N}}$ of ITMs without further input, where z_k is "hardwired" into $A^{(z)}_k$.

Originating in the idea of comparing a "real" protocol with an idealized version, in [Can01a], several equivalent definitions of what it means to securely realize an ideal functionality are given. For our purposes, it is convenient to use [Can01a, Section 4.4, Definition 4], where instead of an "arbitrary" adversary $\mathcal{A}$ only a so-called *dummy adversary* $\tilde{\mathcal{A}}$ is used. Basically, the latter

[1] Note that this definition of polynomially bounded refers only to a single activation. In principle it is possible to activate a polynomially bounded party exponentially often. Also it is worth pointing out, that enforcing a polynomial total "life-time" of each party through explicit life-time bounds can cause technical difficulties [Can02].

simply executes instructions of a predefined form, which are obtained from an *environment machine* $\mathcal{Z}$, modeled as a non-uniform family $\mathcal{Z} = \{\mathcal{Z}_k\}$ of ITMs. Essentially, the aim of an environment machine is to distinguish between

(a) running with parties $P_1, \ldots, P_n$ (modelled as uniform families of ITMs) which are executing protocol π and the (uniform) *dummy adversary* $\tilde{\mathcal{A}}$, and, on the other hand,

(b) running with (uniform) *dummy parties* $\tilde{P}_1, \ldots, \tilde{P}_n$, which act as a "communication relay" to the (uniform) ideal functionality $\mathcal{F}$, the ideal functionality $\mathcal{F}$ itself, and the simulator $\mathcal{S}$ (in place of the dummy adversary $\tilde{\mathcal{A}}$).

The capabilities of the simulator $\mathcal{S}$ in case (b) are rather limited (cf. [Can01a]) and used to model 'inevitable' attacks. Now, if a protocol securely realizes an ideal functionality $\mathcal{F}$, then for any fixed $\mathcal{Z}$ the respective output distributions in (a) and (b) may only differ by a function which is negligible in the security parameter; for a single protocol run, this security parameter k is fixed simultaneously for all participating ITMs. The former requirement reflects the desirable ability of the simulator $\mathcal{S}$ to "mimick" any attack carried out by the adversary $\tilde{\mathcal{A}}$ on protocol π well enough such that no environment can tell the difference between the ideal functionality $\mathcal{F}$ and protocol π.

For an environment machine $\mathcal{Z} = \{\mathcal{Z}_k\}$ that tries to distinguish between an "ideal" and a "real" protocol, for any fixed k we define the *(total) running time* of $\mathcal{Z}_k$ as follows: the *(total) running time* of $\mathcal{Z}_k$ is the worst-case *total* number of steps all ITMs participating in the protocol execution (including the adversary and $\mathcal{Z}_k$ itself) run in the *real model* (i. e., when the parties P_i behave according to π) *plus* the code size of $\mathcal{Z}_k$. Further on, the *advantage* of $\mathcal{Z}$ in distinguishing execution of π from $\mathcal{F}$ when running with simulator $\mathcal{S}$ in the ideal model and with the *dummy adversary* $\tilde{\mathcal{A}}$ in the real model is defined as

$$\mathbf{Adv}^{\mathcal{F},\pi}_{\mathcal{S},\mathcal{Z}}(k) := \left|\mathbf{P}(\mathcal{Z}_k \to 1 \mid \pi, \tilde{\mathcal{A}}) - \mathbf{P}(\mathcal{Z}_k \to 1 \mid \mathcal{F}, \mathcal{S})\right|,$$

where k denotes the security parameter. In other words, $\mathbf{Adv}^{\mathcal{F},\pi}_{\mathcal{S},\mathcal{Z}}(k)$ is the absolute value of the difference between the probabilities of $\mathcal{Z}$ outputting 1 in the real and in the ideal model. Saying that protocol π securely realizes functionality $\mathcal{F}$ now boils down to saying that there exists a simulator $\mathcal{S}$, so that for every environment $\mathcal{Z}$, the function $\mathbf{Adv}^{\mathcal{F},\pi}_{\mathcal{S},\mathcal{Z}}(k)$ is negligible in k. (This can be seen by comparing our modeling of non-uniformity by families of Turing machines to that of [Can01a], which employs additional environmental inputs z.)

2.3. On relating ROR-CCA, IND-CCA, and $\mathcal{F}_{\mathrm{PKE}}$

Especially when relating adversaries attacking a public key encryption scheme P in a sense similar to ROR-CCA and environments distinguishing between P (interpreted as a protocol) and $\mathcal{F}_{\mathrm{PKE}}$, it seems helpful to restrict the latter environments to the class of environments with *polynomial* total running time; otherwise, the total running time of an environment alone might not be bounded by any polynomial although confusingly it could be called "polynomially bounded": imagine an environment periodically querying the adversary

just for the sake of giving away control for a moment, thereby staying polynomially bounded in the sense above, yet doing this an exponential (in k) number of times.

Observe now that if we completely restrict to environments $\mathcal{Z}$ having polynomial total running time, inspection of the proof in [Can01a, Section 5.4] shows that the *composition theorem* still holds. The mentioned composition theorem is crucial in the work of [Can01a]; it enables us to formulate protocols τ which are using some ideal functionality $\mathcal{F}$ freely, *without* losing security when later substituting calls to $\mathcal{F}$ by invocations of some sub-protocol π which in turn realizes $\mathcal{F}$. Yet the proof of the composition theorem does not apply anymore if we completely restrict to *uniform* environments $\mathcal{Z}$; in the next section, we will give a small modification to the proof in question, so that it will still work when restricting to uniform environments with polynomial running time.

This variant of the composition theorem will turn out to be useful when trying to relate ROR-CCA and IND-CCA security with the ideal functionality $\mathcal{F}_{\mathrm{PKE}}$ already mentioned. Here, we will propose protocols realizing $\mathcal{F}_{\mathrm{PKE}}$ only with respect to *non-adaptive* adversaries; a non-adaptive adversary is not allowed to corrupt parties *during* the execution of the protocol in question. In particular, the non-adaptive dummy adversary is bound to ignore corruption requests from the environment during the execution of the protocol. In the sequel we show that realizing $\mathcal{F}_{\mathrm{PKE}}$ in the presence of non-adaptive adversaries is (polynomially) equivalent to security in the sense of IND-CCA.

Remark 2.1 *In [Can01a] it is claimed that, for a public key cryptosystem, IND-CCA security* strictly *implies the property of securely realizing $\mathcal{F}_{\mathrm{PKE}}$. To obtain a 'separating' example, an IND-CCA-secure encryption scheme P is slightly modified: to each ciphertext a* `1` *is appended after encryption, while decryption is preceded by stripping off the last bit of a ciphertext—without validating it to be a* `1`*. The modified scheme, which is clearly not secure in the IND-CCA sense, is claimed to be still realizing $\mathcal{F}_{\mathrm{PKE}}$; yet consider the following environment $\mathcal{Z}$: after invoking key-generation, $\mathcal{Z}$ activates some party P_i with* `(Encrypt,`id`,`e`,`r`)` *for a random r, thereby obtaining a ciphertext $c = \bar{c}$`1`*. Now decryption of $\bar{c}$`0` yields r only in the real model, hence it is possible to distinguish the real protocol from the ideal process and the modified scheme does not realize $\mathcal{F}_{\mathrm{PKE}}$.*

3. Composition in the uniform case

Here we will describe a small modification to the proof of the composition theorem given in [Can01a, Section 5.4], so that we are able to prove the latter theorem even in the case of uniform environment machines with polynomial total running time.

Proposition 3.1 *The composition theorem of [Can01a] holds even if we restrict the complete framework described in [Can01a] to uniform environments with polynomial total running time.*

Proof. We give a proof which works both for uniform and non-uniform environments. In fact, only a small modification of the construction used to

prove the composition theorem in [Can01a] is necessary. To see this, recall that, assuming an environment $\mathcal{Z}$ successfully distinguishing between an execution of protocol π in the $\mathcal{F}$-hybrid model and the execution of the composed protocol π^ρ (where protocol ρ securely realizes $\mathcal{F}$ with respect to a certain simulator $\mathcal{S}$ mimicking attacks on ρ carried out by $\tilde{\mathcal{A}}$), the idea is to construct an environment $\mathcal{Z}_\rho$ which successfully distinguishes between $\mathcal{F}$ and protocol ρ, thereby leading to a contradiction.

Let's fix a—possibly uniform—environment $\mathcal{Z} = \{\mathcal{Z}_k\}$ and a security parameter k. With respect to the simulator $\mathcal{H}$ explicitly constructed in [Can01a], let $\mathrm{HYB}^{\mathcal{F}^{(i)}}_{\pi,\mathcal{H},\mathcal{Z}}(k)$ denote the probability distribution of $\mathcal{Z}_k$'s output when running with protocol π, where calls to the first i instances of $\mathcal{F}$ invoked by π are "redirected" to ideal instances of $\mathcal{F}$, whereas the remaining instances of $\mathcal{F}$ are handled by protocol ρ. Let $m(k)$ be an upper bound for the number of $\mathcal{F}$-instances used during the execution of π. Note that $m(k)$ may be assumed to be a polynomial in the security parameter k. Thus, we can assume that the function m and so the value $m(k)$ is known to a, depending on the uniformity of $\mathcal{Z}$, possibly uniform environment $\mathcal{Z}'_\rho = \{(\mathcal{Z}'_\rho)_k\}$ where $(\mathcal{Z}'_\rho)_k$ first guesses a value $l \in \{1, \ldots, m(k)\}$ and then proceeds exactly as environment $(\mathcal{Z}_\rho)_k$ (described in [Can01a]) with input l. We find

$$
\begin{aligned}
\mathbf{Adv}^{\mathcal{F},\rho}_{\mathcal{S},\mathcal{Z}'_\rho}(k) &= \left|\mathbf{P}\left((\mathcal{Z}'_\rho)_k \to 1 \mid \rho, \tilde{\mathcal{A}}\right) - \mathbf{P}\left((\mathcal{Z}'_\rho)_k \to 1 \mid \mathcal{F}, \mathcal{S}\right)\right| \\
&= \frac{1}{m(k)}\left|\sum_{i=1}^{m(k)} \mathbf{P}\left((\mathcal{Z}'_\rho)_k \to 1 \mid \rho, \tilde{\mathcal{A}}, l = i\right) - \mathbf{P}\left((\mathcal{Z}'_\rho)_k \to 1 \mid \mathcal{F}, \mathcal{S}, l = i\right)\right| \\
&= \frac{1}{m(k)}\left|\sum_{i=1}^{m(k)} \mathrm{HYB}^{\mathcal{F}^{(i-1)}}_{\pi,\mathcal{H},\mathcal{Z}}(k) - \mathrm{HYB}^{\mathcal{F}^{(i)}}_{\pi,\mathcal{H},\mathcal{Z}}(k)\right| \\
&= \frac{1}{m(k)}\left|\mathrm{HYB}^{\mathcal{F}^{(0)}}_{\pi,\mathcal{H},\mathcal{Z}}(k) - \mathrm{HYB}^{\mathcal{F}^{(m(k))}}_{\pi,\mathcal{H},\mathcal{Z}}(k)\right| \\
&= \frac{1}{m(k)}\left|\mathbf{P}\left(\mathcal{Z}_k \to 1 \mid \pi^\rho, \tilde{\mathcal{A}}\right) - \mathbf{P}\left(\mathcal{Z}_k \to 1 \mid \pi^{\mathcal{F}}, \mathcal{H}\right)\right| \\
&= \frac{1}{m(k)}\mathbf{Adv}^{\pi^{\mathcal{F}},\pi^\rho}_{\mathcal{H},\mathcal{Z}}(k),
\end{aligned}
$$

which is, by assumptions about $\mathcal{Z}$ and $m(k)$, not negligible in k. □

4. Relating ROR-CCA and $\mathcal{F}_{\mathrm{PKE}}$

The specifications of the ideal functionalities $\mathcal{F}_{\mathrm{PKE}}$ and $\mathcal{F}_{\text{M-SMT}}$ from [Can01a], as used in the following, are given in Appendix A. We remark that in [Can01a] it is not specified how $\mathcal{F}_{\mathrm{PKE}}$ behaves when being asked multiple times for a key generation (possibly by different parties). Rather, a request for key generation is to be the first and only the first call to $\mathcal{F}_{\mathrm{PKE}}$. In effect, for $\mathcal{F}_{\mathrm{PKE}}$ to be securely realizable *at all* (no matter how we might "complete" its specification), we need to restrict to environments which use this functionality *as intended*; i.e. every environment trying to distinguish $\mathcal{F}_{\mathrm{PKE}}$ from some protocol P should only be allowed to send *one* key generation query to the functionality, and this

query has to be sent *before* any other queries.

Of course, in view of the composition theorem, that also imposes a limitation on the *use* of $\mathcal{F}_{\text{PKE}}$. Namely, in the case of protocol σ working in the $\mathcal{F}_{\text{PKE}}$-hybrid model as presented below, this translates into the following restriction: environments trying to distinguish execution of σ from the ideal functionality $\mathcal{F}_{\text{M-SMT}}$ should be forced to send some message `(receiver,`*id*`)` as the *first* query to the functionality, but *no* further such "initialization queries". (The ideal functionality $\mathcal{F}_{\text{M-SMT}}$ enables parties to communicate securely in the following sense: after being initialized by some party P_i, $\mathcal{F}_{\text{M-SMT}}$ allows *any other* party to send messages to P_i in a way that the adversary gains no other information than length information about the sent messages.) In particular, all the results presented in this section are to be seen in the light of these restrictions.[2]

The next proposition gives "tight" reductions between different types of attackers, i. e., there is an explicit relation between the respective advantages, and the reductions essentially preserve running time. As we did not fix, e. g., the notion of code size, we cannot obtain explicit formulæ relating running times (which by definition depend on the respective code size).

Proposition 4.1 *Let $P := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public key encryption scheme. Let $U \in \{$uniform, non-uniform$\}$. Then, in the following sense, we have a tight correspondence between adversaries attacking P in the ROR-CCA game and environments distinguishing $\mathcal{F}_{\text{PKE}}$ from protocol P in the presence of the non-adaptive dummy adversary:*

1. *For every U-adversary A in the ROR-CCA game, we can construct a U-environment $\mathcal{Z}$ so that for any simulator $\mathcal{S}$ we have*

$$\mathbf{Adv}^{\mathcal{F}_{\text{PKE}},P}_{\mathcal{S},\mathcal{Z}}(k) = \frac{|\mathbf{Adv}^{\text{ror-cca}}_{P,A}(k)|}{2}.$$

2. *There is a simulator $\mathcal{S}_{\text{ROR}}$, so that for any U-environment $\mathcal{Z}$ interacting with the non-adaptive dummy adversary, there exists a U-adversary A in the ROR-CCA game with*

$$|\mathbf{Adv}^{\text{ror-cca}}_{P,A}(k)| = \mathbf{Adv}^{\mathcal{F}_{\text{PKE}},P}_{\mathcal{S}_{\text{ROR}},\mathcal{Z}}(k).$$

Proof.

1. Let $A = \{A_k\}$ be a U-adversary attacking P in the ROR-CCA sense. From A, we will construct a U-environment $\mathcal{Z} = \{\mathcal{Z}_k\}$ distinguishing between $\mathcal{F}_{\text{PKE}}$ and P with the claimed advantage. For this, we define two experiments E_1 and E_2 (to be run by an environment in the setting of [Can01a]) as follows: for a given security parameter k, E_1 runs A_k as a black box with access to the facilities of $\mathcal{F}_{\text{PKE}}$ and outputs whatever A_k outputs. E_2 is identical to E_1, except for the responses to A_k upon encryption requests: if A_k requests encryption of a message m, E_2 responds

[2] Another approach to overcome these problems, thereby avoiding restrictions on environments and possible obstacles regarding the applicability of the composition theorem, could be based on ideas from [PW01]. Namely, a *family* of functionalities $\{\mathcal{F}_{\text{PKE},i}\}_{P_i}$ could be used, where $\mathcal{F}_{\text{PKE},i}$ enables only the party P_i to generate a key and to decrypt. A similar construction for $\mathcal{F}_{\text{M-SMT}}$ is possible.

with $\mathcal{F}_{\mathrm{PKE}}$'s encryption of some random plaintext of the same length as m (this random plaintext is chosen anew upon each encryption request).

We now describe the environment $\mathcal{Z}$: when activated, $\mathcal{Z}_k$ flips a coin $r \in \{1,2\}$. If $r = 1$, then $\mathcal{Z}_k$ runs experiment E_1 and outputs 1 if and only if E_1 outputs 1. On the other hand, if $r = 2$, then $\mathcal{Z}_k$ runs experiment E_2 and outputs 1 if and only if E_2 does *not* output 1.

For analysis, let's fix an arbitrary simulator $\mathcal{S}$ and a security parameter k and denote by ϵ_i^{R} the probability that experiment E_i yields output 1 while operating with P and the dummy adversary $\tilde{\mathcal{A}}$; define ϵ_i^{I} to be the probability for E_i to output 1 when running with $\mathcal{F}_{\mathrm{PKE}}$ and $\mathcal{S}$. Since $\epsilon_1^{\mathrm{I}} = \epsilon_2^{\mathrm{I}}$ (reflecting that $\mathcal{S}$'s responses to encryption queries cannot depend on the plaintext to be encrypted) and $|\epsilon_1^{\mathrm{R}} - \epsilon_2^{\mathrm{R}}| = |\mathbf{Adv}_{P,A}^{\text{ror-cca}}(k)|$ (by definition), it follows that $\mathcal{Z}$'s success in distinguishing $\mathcal{F}_{\mathrm{PKE}}$ from P is given by

$$\begin{aligned}
\mathbf{Adv}_{\mathcal{S},\mathcal{Z}}^{\mathcal{F}_{\mathrm{PKE}},P}(k) &= \left|\mathbf{P}\left(\mathcal{Z}_k \to 1 \mid P, \tilde{\mathcal{A}}\right) - \mathbf{P}\left(\mathcal{Z}_k \to 1 \mid \mathcal{F}_{\mathrm{PKE}}, \mathcal{S}\right)\right| \\
&= \left|\frac{1}{2}\left(\epsilon_1^{\mathrm{R}} + \left(1 - \epsilon_2^{\mathrm{R}}\right)\right) - \frac{1}{2}\left(\epsilon_1^{\mathrm{I}} + \left(1 - \epsilon_2^{\mathrm{I}}\right)\right)\right| \\
&= \frac{1}{2}\left|\epsilon_1^{\mathrm{R}} - \epsilon_2^{\mathrm{R}} + \epsilon_2^{\mathrm{I}} - \epsilon_1^{\mathrm{I}}\right| \;=\; \frac{|\mathbf{Adv}_{P,A}^{\text{ror-cca}}(k)|}{2}.
\end{aligned}$$

It should be clear that our reduction applies, no matter if A is uniform or not.

2. Let $\mathcal{Z}$ be an environment distinguishing $\mathcal{F}_{\mathrm{PKE}}$ from protocol P. We now describe the simulator $\mathcal{S}_{\mathrm{ROR}}$ in question. Encryption requests to $\mathcal{S}_{\mathrm{ROR}}$ are answered by a P-encryption of some random plaintext r of the respective length (as before, r is chosen anew upon each request). Decryption and key generation requests are handled just as P would do. (Of course, key generation requests by the environment are answered with the public key only.)

   Having said this, we can construct an adversary A attacking P in the ROR-CCA game in the obvious way and the claimed equality follows. Note that there is a small subtlety here: ROR-CCA adversaries are by definition not allowed to request decryptions of ciphertexts already obtained by the encryption facility. However, as the ciphertexts in question result from explicit encryption requests, A can obtain the same answers $\mathcal{Z}$ would have got in the setting of [Can01a] by feeding itself the respective arguments of these encryption requests. As before, our transformation applies to both uniform and non-uniform environments and adversaries.

□

Corollary 4.1 *Suppose we are in the situation of Proposition 4.1. Then, in the presence of non-adaptive adversaries, P securely realizes the functionality $\mathcal{F}_{\mathrm{PKE}}$ with respect to U-environments if and only if P is secure in the sense of ROR-CCA (interpreted in the public key sense) with respect to U-adversaries.*

Corollary 4.2 *Suppose we are in the situation of Proposition 4.1. Then, in the presence of non-adaptive adversaries, P securely realizes the functionality $\mathcal{F}_{\text{PKE}}$ with respect to U-environments if and only if P is secure in the sense of IND-CCA with respect to U-adversaries.*

In the case of non-uniform environments and non-uniform IND-CCA adversaries, the above corollary is nothing else but [Can01a, Claim 15].

Remark 4.1 *Observe that the reductions constructed in the proof of Proposition 4.1 are "tight" with respect to both total running time and advantage function, whereas there can be no "tight" reduction transforming $\mathcal{F}_{\text{PKE}}$-distinguishers to IND-CCA adversaries, since with respect to concrete reductions, IND-CCA-security is weaker compared to security in the ROR-CCA sense [BDJR97]. (In [BDJR97], the notion of IND-CCA is called FTG-CCA; there, it is also proven that FTG-CCA in turn is equivalent in some "tight" sense to SEM-CCA, an adaption of semantic security with respect to chosen-ciphertext attacks.)*

A remarkable feature of the framework of [Can01a] is the *composability* of functionalities; thus it is now worthwhile to ask how we can utilize the ideal functionality $\mathcal{F}_{\text{PKE}}$. For this we consider the ideal functionality $\mathcal{F}_{\text{M-SMT}}$ explained in [Can01a] (see also Appendix A). Again, we will only deal with non-adaptive adversaries.

Lemma 4.1 [Can01a, Claim 16]. *Let $U \in \{$uniform, non-uniform$\}$. Assuming ideally authenticated links, there exists a protocol σ which securely realizes $\mathcal{F}_{\text{M-SMT}}$ in the $\mathcal{F}_{\text{PKE}}$-hybrid model in the presence of non-adaptive adversaries. More specifically, for every non-adaptive adversary $\mathcal{A}$ attacking σ, there is a simulator $\mathcal{S}$ such that for every U-environment $\mathcal{Z} = \{\mathcal{Z}_k\}$ we have*

$$\mathbf{P}(\mathcal{Z}_k \to 1 \mid \sigma^{\mathcal{F}_{\text{PKE}}}, \mathcal{A}) = \mathbf{P}(\mathcal{Z}_k \to 1 \mid \mathcal{F}_{\text{M-SMT}}, \mathcal{S})$$

for every k.

Proof. This is shown in the proof of Claim 16 in [Can01a, Section 8.2.2]; this proof carries over to uniform environments. □

We can utilize the obtained results in order to make the security reductions of [Can01a, Claim 16] more explicit and apply the composition theorem in the uniform case:

Corollary 4.3 *Let $U \in \{$uniform, non-uniform$\}$. Assuming ideally authenticated links, any public key encryption scheme P which is secure in the ROR-CCA sense with respect to U-adversaries can be turned into a protocol σ^P securely realizing $\mathcal{F}_{\text{M-SMT}}$ with respect to U-environments in the presence of non-adaptive adversaries.*

In particular, there is a simulator $\mathcal{S}$ working in the $\mathcal{F}_{\text{M-SMT}}$-ideal model, such that every U-environment $\mathcal{Z}$ distinguishing $\mathcal{F}_{\text{M-SMT}}$ from execution of the composed protocol σ^P can be turned into a U-adversary A attacking P in the ROR-CCA game. We then have

$$|\mathbf{Adv}^{\text{ror-cca}}_{P,A}(k)| = \mathbf{Adv}^{\mathcal{F}_{\text{M-SMT}},\sigma^P}_{\mathcal{S},\mathcal{Z}}(k).$$

Proof. Of course, protocol σ is the protocol mentioned in Lemma 4.1. We construct a suitable simulator $\mathcal{S}$ emulating σ^P in the $\mathcal{F}_{\text{M-SMT}}$-ideal model. So let $\mathcal{H}$ be the simulator working in the $\mathcal{F}_{\text{PKE}}$-hybrid model used in the proof of the composition theorem of [Can01a], assuming composition of σ and P and simulation of P in the $\mathcal{F}_{\text{PKE}}$-ideal model through $\mathcal{S}_{\text{ROR}}$ (the following discussion also applies if we completely restrict ourselves to uniform environments). From $\mathcal{H}$, we construct $\mathcal{S}$ as mentioned in Lemma 4.1 and described in detail in the proof of Claim 16 in [Can01a, Section 8.2.2].

Now say that, with respect to the simulator $\mathcal{S}$ just described, $\mathcal{Z} = \{\mathcal{Z}_k\}$ is a U-environment distinguishing between σ^P and the ideal functionality $\mathcal{F}_{\text{M-SMT}}$. Observe that the output distribution of $\mathcal{Z}$ when interacting with $\mathcal{F}_{\text{M-SMT}}$ is *identical* to the one resulting from interaction with σ and $\mathcal{H}$ in the $\mathcal{F}_{\text{PKE}}$-hybrid model. On the other hand, we know from Proposition 3.1, that there is a U-environment $\mathcal{Z}'_P$ distinguishing $\mathcal{F}_{\text{PKE}}$ and $\mathcal{S}_{\text{ROR}}$ from P and $\tilde{\mathcal{A}}$, for which we have

$$\begin{aligned}
\mathbf{Adv}^{\mathcal{F}_{\text{PKE}},P}_{\mathcal{S}_{\text{ROR}},\mathcal{Z}'_P}(k) &= \left|\mathbf{P}\left(\mathcal{Z}_k \to 1 \mid \sigma^P, \tilde{\mathcal{A}}\right) - \mathbf{P}\left(\mathcal{Z}_k \to 1 \mid \sigma^{\mathcal{F}_{\text{PKE}}}, \mathcal{H}\right)\right| \\
&= \left|\mathbf{P}\left(\mathcal{Z}_k \to 1 \mid \sigma^P, \tilde{\mathcal{A}}\right) - \mathbf{P}\left(\mathcal{Z}_k \to 1 \mid \mathcal{F}_{\text{M-SMT}}, \mathcal{S}\right)\right| \\
&= \mathbf{Adv}^{\mathcal{F}_{\text{M-SMT}},\sigma^P}_{\mathcal{S},\mathcal{Z}}(k).
\end{aligned}$$

(By construction of protocol σ, the polynomial $m(k)$ used in the proof of Proposition 3.1 is the constant polynomial $m(k) = 1$.) The claimed equality then follows with Proposition 4.1 by interpreting $\mathcal{Z}'_P$ as a U-adversary attacking P in the ROR-CCA sense. □

5. Conclusion

We have shown that, for a public key encryption scheme, being secure in the ROR-CCA sense is in some "tight" sense equivalent to securely realizing $\mathcal{F}_{\text{PKE}}$ when interpreted as a protocol. In view of the results of [BDJR97], this means specifically that securely realizing $\mathcal{F}_{\text{PKE}}$ is a slightly stronger (yet polynomially equivalent) notion of security than indistinguishability with respect to chosen-ciphertext attacks.

Our results hold both for uniform and non-uniform adversarial entities, and in particular we have shown that the composition theorem of [Can01a] holds even with respect to uniform environments with polynomial total running time, thus enabling secure composition of protocols realizing $\mathcal{F}_{\text{PKE}}$. Specifically, one can use these results to justify the proposal in [Can01a] to "plug" any IND-CCA secure encryption scheme into protocols expecting access to $\mathcal{F}_{\text{PKE}}$. Furthermore, we have focused on concrete security reductions, thus allowing to speak of concrete security levels while preserving an intuitive modeling using the ideal functionality $\mathcal{F}_{\text{PKE}}$.

Note

After completing this manuscript, we learned from Ran Canetti, that in the independent work [CKN03a] also the equivalence between security in the sense of realizing $\mathcal{F}_{\text{PKE}}$ and in the IND-CCA sense is shown, but the focus and nature of the results obtained in [CKN03a] are quite different: whereas we focus on concrete security reductions for uniform and non-uniform settings, [CKN03a] investigates relaxed security notions that still preserve crucial security properties of public key encryption.

Acknowledgements

We thank Ran Canetti and Dominique Unruh for valuable discussions.

REFERENCES

[BDJR97] Mihir Bellare, Anand Desai, Eron Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science, Proceedings of FOCS 1997*, pages 394–403. IEEE Computer Society, 1997.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *Advances in Cryptology, Proceedings of CRYPTO '98*, number 1462 in Lecture Notes in Computer Science, pages 26–45. Springer-Verlag, 1998.

[BR95] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption—how to encrypt with RSA. In Alfredo de Santis, editor, *Advances in Cryptology, Proceedings of EUROCRYPT '94*, number 950 in Lecture Notes in Computer Science, pages 92–111. Springer-Verlag, 1995.

[Can01a] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Electronic Colloquium on Computational Complexity, October 2001. Full version of [Can01b].

[Can01b] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42th Annual Symposium on Foundations of Computer Science, Proceedings of FOCS 2001*, pages 136–145. IEEE Computer Society, 2001.

[Can02] Ran Canetti. E-mail communication with the authors, October 2002.

[CKN03a] Ran Canetti, Hugo Krawczyk, and Jesper B. Nielsen. Relaxing chosen-ciphertext security. Unpublished, later version published in [CKN03b], February 2003.

[CKN03b] Ran Canetti, Hugo Krawczyk, and Jesper B. Nielsen. Relaxing chosen-ciphertext security. In Dan Boneh, editor, *Advances in Cryptology, Proceedings of CRYPTO 2003*, number 2729 in Lecture Notes in Computer Science, pages 565–582. Springer-Verlag, 2003.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.

[PW01] Birgit Pfitzmann and Michael Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *IEEE Symposium on Security and Privacy, Proceedings of SSP 2001*, pages 184–200. IEEE Computer Society, 2001.

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *Advances in Cryptology, Proceedings of CRYPTO '91*, number 576 in Lecture Notes in Computer Science, pages 433–444. Springer-Verlag, 1992.

[Sho01] Victor Shoup. OAEP reconsidered. In Joe Kilian, editor, *Advances in Cryptology, Proceedings of CRYPTO 2001*, number 2139 in Lecture Notes in Computer Science, pages 239–259. Springer-Verlag, 2001.

A. The functionalities $\mathcal{F}_{\text{PKE}}$ and $\mathcal{F}_{\text{M-SMT}}$

For completeness, we describe the already mentioned ideal functionalities $\mathcal{F}_{\text{PKE}}$ and $\mathcal{F}_{\text{M-SMT}}$ introduced in [Can01a]; indeed, in the following two boxes, we simply reproduce the descriptions given in [Can01a].

Functionality $\mathcal{F}_{\text{PKE}}$

$\mathcal{F}_{\text{PKE}}$ proceeds as follows, running with parties $P_1, \ldots, P_n$ and an adversary $\mathcal{S}$.

1. In the first activation, expect to receive a value `(KeyGen,`id`)` from some party P_i. Then, do:
   (a) Hand `(KeyGen,`id`)` to the adversary.
   (b) Receive a value e from the adversary, and hand e to P_i.
2. Upon receiving a value `(Encrypt,`id`,`e'`,`m`)` from some party P_j, proceed as follows:
   (a) Hand `(Encrypt,`id`,`e'`,`$|m|$`)` to the adversary, where $|m|$ denotes the length of m. (If $e' \neq e$ or e is not yet defined then hand also the entire value m to the adversary.)
   (b) Receive a tag c from the adversary and hand c to P_j. In addition, if $e' = e$ then store the pair (c, m). (If the tag c already appears in a previously stored pair then halt.)
3. Upon receiving a value `(Decrypt,`id`,`c`)` from P_i (and P_i only), proceed as follows:
   (a) If there is a pair (c, m) stored in memory then hand m to P_i.
   (b) Otherwise, hand the value `(Decrypt,`id`,`c`)` to the adversary, receive a value m from the adversary, and hand m to P_i.

Functionality $\mathcal{F}_{\text{M-SMT}}$

$\mathcal{F}_{\text{M-SMT}}$ proceeds as follows, running with parties $P_1, \ldots, P_n$ and an adversary $\mathcal{S}$.

1. In the first activation, expect to receive a value `(receiver,`id`)` from some party P_i. Then, send `(receiver,`id`,`P_i`)` to all parties and the adversary. From now on, ignore all `(receiver,`id`,`P_i`)` values.
2. Upon receiving a value `(send,`id`,`m`)` from some party P_j, send `(`id`,`P_j`,`m`)` to P_i and `(`id`,`P_j`,`$|m|$`)` to the adversary.

B. Security in the ROR-CCA sense

For convenience, we also reproduce the criterion for security in the ROR-CCA sense. A detailed definition can be found in [BDJR97]; here we give a formula-

tion for the public key setting which is derived in a straightforward way from the private key case.

Let $P = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public key encryption scheme. Let $b \in \{0,1\}$ and $k \in \mathbb{N}$. Formally, for $(d, e) \leftarrow \mathcal{K}$ with public key e, we define the *real-or-random* oracle $\mathcal{E}_e(\mathcal{RR}(\cdot, b))$ to take input m and do the following: if $b = 1$ it computes $c \leftarrow \mathcal{E}_e(m)$ and returns c; else it computes $c \leftarrow \mathcal{E}_e(r)$ where $r \xleftarrow{R} \{0,1\}^{|m|}$ (i. e., r is a random bitstring of the same length as m) and returns c. Let A be an adversary that has access to the oracles $\mathcal{E}_e(\mathcal{RR}(\cdot, b))$ and $\mathcal{D}_d(\cdot)$. Now, we consider the following experiment:

Experiment $\mathbf{Exp}^{\text{ror-cca-b}}_{P,A}(k)$:
  $(d, e) \leftarrow \mathcal{K}(k)$
  $\tilde{b} \leftarrow A^{\mathcal{E}_e(\mathcal{RR}(\cdot,b)),\mathcal{D}_d(\cdot)}(k, e)$
  Return $\tilde{b}$

Above it is mandated that A never queries $\mathcal{D}_d(\cdot)$ on a ciphertext c output by the $\mathcal{E}_e(\mathcal{RR}(\cdot, b))$ oracle. We define the *advantage* of the adversary via

$$\mathbf{Adv}^{\text{ror-cca}}_{P,A}(k) := \mathbf{P}(\mathbf{Exp}^{\text{ror-cca-1}}_{P,A}(k) = 1) - \mathbf{P}(\mathbf{Exp}^{\text{ror-cca-0}}_{P,A}(k) = 1).$$

The scheme P is said to be *ROR-CCA secure* if the function $\mathbf{Adv}^{\text{ror-cca}}_{P,A}(\cdot)$ is negligible for any adversary A whose time complexity (including code size) is polynomial in k.