Related Message Attacks to Public Key Encryption Schemes: Relations among Security Notions

María Isabel González Vasco and Angel L. Pérez del Pozo

Departamento de Matemática Aplicada, Universidad Rey Juan Carlos,
c/ Tulipán, s/n, 28933 Madrid, Spain
{mariaisabel.vasco, angel.perez}@urjc.es

Abstract. Consider a scenario in which an adversary, attacking a certain public key encryption scheme, gains knowledge of several ciphertexts which underlying plaintext are meaningfully related with a given target ciphertext. This kind of *related message attack* has been proved successful against several public key encryption schemes; widely known is the Franklin-Reiter attack to RSA with low exponent and its subsequent improvement by Coppersmith. However, to the best of our knowledge no formal treatment of these type of attacks has to date been done, and as a result, it has not been rigorously studied which of the "standard" security notions imply resilience to them.

We give formal definitions of several security notions capturing the resistance to this kind of attacks. For passive adversaries we prove that, for the case of *indistinguishability*, security against related message attacks is equivalent to standard CPA security. On the other hand, *one-wayness* robust schemes in this sense can be seen as strictly between OW-CPA and IND-CPA secure schemes. Furthermore, we prove that the same holds for active (CCA) adversaries.

1 Introduction

In a related message attack against a cryptosystem, the attacker obtains several ciphertexts such that there exists a known relation between the corresponding plaintexts. These ciphertexts may thus help the attacker to achieve a certain goal, for instance to gain useful information about the hidden plaintexts.

Such scenario is actually highly relevant in practise, as it is common that the apriori knowledge of the adversary on a message flow translates into a known relationship between the incoming messages. For example, as pointed in [13], this could happen if the content of an encrypted message is followed by a serial number. Then an attacker who pretends to be the legitimate recipient could ask for a message-resend and, if he is able to determine the increment in the serial number, he will obtain encryptions of two messages with a known relation between them.

Although related message attacks against several cryptosystems have been considered, a general formal study of the subject has never been proposed. This is somewhat surprising, given the intense research carried over on probable security notions and proof techniques for public key encryption schemes (see, for instance [3, 11]).

In this paper, we give security definitions for the above scenario, considering passive (CPA) and active (CCA) adversaries which aim at violating the *one-wayness* or the *indistinguishability* of a public key encryption scheme. We do not consider *non-malleability*, as this notion in our setting highly depends on the concrete relation taken into account; for our case studies, however, relations are rather general.

Furthermore, we insert our new definitions in the graph of known security notions, by proving its relationship to them. In particular, we prove that in the sense of *indistinguishability*, CPA (resp. CCA) security against related message attacks is equivalent to standard CPA (resp. CCA) security. For the case of *one-wayness* however, CPA (resp. CCA) security is weaker than our new notion, which is strictly between IND-CPA and OW-CPA (resp. IND-CCA and OW-CCA).

Related Work. Franklin and Reiter [6] proved that RSA with low encrypting exponent is vulnerable against this kind of attacks. Namely they showed that a message m can be efficiently recovered from its RSA ciphertext when the public exponent e equals 3 and two ciphertexts $c_1 = m^e$ and $c_2 = (am + b)^e$ are known together with the coefficients a and b . To make this attack work it is almost always enough to compute the gcd of two polynomials of degree 3. Then Coppersmith, Franklin, Patarin and Reiter [5] generalized this attack for an arbitrary low public exponent, an arbitrary number of related messages and polynomial relations of arbitrary degree between the messages. These generalizations require computing the gcd of polynomials of arbitrary degree and also the resultant of multivariate polynomials; thus, some of them are hardly relevant in practise. Recently Yacobi and Yacobi [13] proposed another related message attack against RSA when linear relations between sent messages are known. The tools they use are divided-differences and finite-differences. Their attack is in some sense more efficient than previous ones, though the adversary needs to intercept more ciphertexts.

The McEliece cryptosystem has also been attacked this way. Berson [4] found an attack which efficiently reveals the plaintext if a message is sent twice and, more generally, if two messages with a known linear relationship between them are sent. As the underlying field is of characteristic two, this means the adversary can decrypt two ciphertexts if he knows the XOR of the corresponding bitstring plaintexts.

Other related work in this direction we are aware of is that of Hastad [7]. In this case the same message is sent to several recipients and it is encrypted using a different RSA modulus each time. Hastad proved that, if certain conditions on the number of sent messages are met, then the plaintext can be efficiently recovered from the ciphertexts. He also showed that the attack can be extended to the case where there is an affine relation between the sent messages instead of all being the same. In this fashion, Pinch [10] extended the Hastad attack to

the LUC cryptosystem [9]. However, the scenario they both bear in mind differs from the one studied here, as we impose that the related messages available to the adversary are all encrypted under the same public key.
Paper Outline. After a brief preliminaries section, in which we recall the standard security notions needed to follow our argumentation, Section 3 is devoted to the introduction of our new definitions of security motivated by related-message attacks.

Furthermore, in Section 4 we focus on the case of passive adversaries, and prove that our new security notion (in the sense of *indistinguishability*) is equivalent to IND-CPA. For the case of *one-wayness* we demonstrate that the notion we introduce is strictly between OW-CPA and IND-CPA. Subsequently, identical results are obtained for active adversaries in Section 5. Our results are finally summarized on a last short section where we display the derived extension of the graph from Section 3.

2 Preliminaries

2.1 Public key encryption schemes and their security

Let us start with a brief revision of the main terminology and notion concerning the security of public key encryption schemes

As standard, by a public key encryption scheme we mean a triplet of probabilistic polynomial time algorithms $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where

- $\mathcal{K}$, the key generation algorithm, is randomized, takes the security parameter k as an input and outputs a pair of corresponding public and secret key,
- $\mathcal{E}$, the encryption algorithm, is also randomized, takes a message from the message space M and the public key as an input, and outputs a corresponding ciphertext (say, from a set C not necessarily equal to M);
- $\mathcal{D}$, the decryption algorithm, is deterministic, takes an element $c \in C$ and a secret key as input and outputs, if it exists, the plaintext m that encrypts to c with a corresponding public key. Otherwise, it outputs an error message $\perp$.

Security notions for public key encryption schemes are formulated in terms of the adversary's goals and capabilities. The adversary will be modelled, as standard, by a probabilistic polynomial time Turing machine $\mathcal{A}$. We will consider here two adversarial goals:

- violating the *one-wayness* (OW) of the scheme, namely, getting back *the whole* plaintext of a given ciphertext,
- violating the *indistinguishability* (IND) of the scheme, by gaining *any* information about the plaintext of a given ciphertext.

Furthermore, we will consider two types of adversaries in terms of their capabilities:

– a *passive* (CPA) adversary $\mathcal{A}$ is able to obtain a valid ciphertext from any given plaintext, that is, $\mathcal{A}$ has non-limited access to the encryption algorithm $\mathcal{E}$ and holds all public information (including the public key);
– an *active* (CCA) adversary $\mathcal{A}$ has, in addition, access to a decryption oracle, which gives the output of a legitimate decryption machine holding the secret key. Then, when presented with a target ciphertext, he will try to exploit the interaction with such an oracle to gain information on the corresponding plaintext. At this, we must impose the restriction that $\mathcal{A}$ will not be able to query this decryption oracle on this so-called challenge ciphertext (see bellow).

Summarizing, the "classical" security notions which we will adapt to the scenario we are considering are IND-CPA, OW-CPA, IND-CCA and IND-CPA. We refer the interested reader, for further details, to [11]. Here we briefly recall the corresponding definitions:

Definition 1. [OW-ADV] *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public key encryption scheme. Consider the following interaction between an adversary $\mathcal{A}$ of type* ADV $\in \{\text{CPA}, \text{CCA}\}$ *and a simulator $\mathcal{S}$.*

Step 1. *The simulator $\mathcal{S}$ runs the key generation algorithm on input a publicly known security parameter k, obtaining a pair (p_k, s_k) of corresponding public and secret keys.*
Step 2. *$\mathcal{S}$ selects uniformly at random a message $m_0 \in M_k$ encrypts it using p_k and $\mathcal{E}$.*
Step 3. *$\mathcal{S}$ hands the challenge ciphertext c_0 and the public key p_k to the adversary $\mathcal{A}$.*
Step 4. *$\mathcal{A}$ outputs a guess m for m_0.*

Now denote by $\mathsf{Succ}^{OW}(\mathcal{A})$ [1] the advantage probability over a random guess [2] that the adversary outputs the correct plaintext message m_0, taken over the space defined by the (u.a.r.) choices described above and the random coins of the encryption algorithm.

The scheme Π is secure in the sense of OW-ADV *if and only if for any adversary $\mathcal{A}$ of type* ADV *the probability $\mathsf{Succ}^{OW}(\mathcal{A})$ is a negligible function [3] of the security parameter k.*

Definition 2. [IND-ADV] *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public key encryption scheme. Consider the following experiment carried over between an adversary $\mathcal{A}$ of type* ADV $\in \{\text{CPA}, \text{CCA}\}$ *and a simulator $\mathcal{S}$.*

---

[1] $\mathsf{Succ}^{OW}(\mathcal{A})$ is often defined as the probability that the adversary outputs the correct plaintext message. The use of this definition carries over the implicit assumption that the probability of success with a random guess over the message space is negligible in the security parameter, which we don't make here.

[2] Without any information the adversary has probability $1/|M|$ of correctly guessing m_0.

[3] A function $\nu : \mathbb{N} \longmapsto \mathbb{R}$ is *negligible* if for any constant $c > 0$, there exists $n_0 \in \mathbb{N}$ such that $|\nu(n)| \leq n^{-c}$, $\forall n \geq n_0$.

Step 1. *The simulator $\mathcal{S}$ runs the key generation algorithm on input a publicly known security parameter k, obtaining a pair (p_k, s_k) of corresponding public and secret keys. It forwards the public key p_k to the adversary $\mathcal{A}$.*

Step 2. *The adversary $\mathcal{A}$ selects two messages $m_0, m_1 \in M_k$ and hands them to S.*

Step 3. *$\mathcal{S}$ chooses a bit b uniformly at random and encrypts m_b using p_k and $\mathcal{E}$; then it hands the corresponding challenge ciphertext c_b to $\mathcal{A}$.*

Step 4. *$\mathcal{A}$ outputs a guess b^* for the bit b.*

Now denote by $\mathsf{Succ}^{IND}(\mathcal{A})$ the advantage probability, over a random guess[4] *that the adversary outputs the correct bit b.*

This probability is again taken over the space defined by the (u.a.r.) choices described above and the random coins of the encryption algorithm.

The scheme Π is secure in the sense of IND-ADV *if and only if for any adversary $\mathcal{A}$ of type* ADV *the probability $\mathsf{Succ}^{IND}(\mathcal{A})$ is a negligible function of the security parameter k.*

2.2 Admissible Relations

Not all relations we may consider are appropiate for formalizing related message attacks, in particular, we shall not study relations with a somewhat non-homogeneous behavior in the message space. More precisely, information like "how many messages are related to m" should not leak useful information to the adversary; we want to quantify what he gains from eavesdropping ciphertexts of related clear messages, and assume such advantage does not lie in the mere fact that these ciphertexts exist.

Let M be a finite (message) space. Let $\mathcal{R}$ be a certain relation, which is publicly known and can be formalized as a subset of M^*, and for which moreover membership can be tested efficiently (in polynomial time). We can for instance assume that, for a fixed security parameter k, actually the considered message space is a subset $M_k \subseteq M$ and $\mathcal{R} \subseteq M_k^{n+1}$ for some (publicly known) n polynomial in k.

Moreover, given any $m \in M_k$, let $\mathcal{R}_m$ be the set of n-tuples $(m_1, \ldots, m_n)$ such that $(m, m_1, \ldots, m_n) \in \mathcal{R}$. Further, given a message $m \in M_k$, for $i = 0, \ldots, n$ define the sets $\mathcal{R}_m^i$ of n-tuples $(m_1, \ldots, m_n)$, such that $(m_1, \ldots, m_{i-1}, m, m_i, \ldots, m_n) \in \mathcal{R}$.[5]

Definition 3. [Admissible Relation] *Let $\mathcal{R}$ be as above. We say that $\mathcal{R}$ is* admissible *if the following properties hold for any $k \in \mathbb{N}$:*

i. for all $m \in M_k$, $i \in 1, \ldots, n+1$, $\mathcal{R}_m^i \neq \varnothing$;
ii. for all $m \in M_k$ it is possible to efficiently sample, uniformly at random, n-tuples from $\mathcal{R}_m$;

[4] Obviously, without any information the adversary has probability $\frac{1}{2}$ of guessing b, thus this advantage is meassured as $2Pr[b^* = b] - 1$.

[5] Note that $\mathcal{R}_m^0 = \mathcal{R}_m$.

iii. for random independent uniform choices of $m, m^* \in M_k$, and $i = 0, \dots, n$, it holds

$$\left| \frac{|\mathcal{R}^i_m|}{|\mathcal{R}^i_{m^*}|} - 1 \right| \leq \varepsilon(k)$$

where the function $\varepsilon : \mathbb{N} \longmapsto \mathbb{R}$ is negligible.

Note that condition iii. is imposed to prevent information leakage from the mere fact that there may exist significantly less messages related to a certain plaintext than to another; the relation must behave somewhat uniformly with respect to all messages.

3 Proposed security notions

We are now ready to introduce the new definitions. In the sequel, all considered adversaries are, as standard, modelled by probabilistic polynomial time turing machines. We start by the definition of security in the sense of *one-wayness*; at this, the adversarial goal is to decrypt a challenge ciphertext c_0, having captured several ciphertexts whose corresponding plaintexts are related to m_0 in the sense of the publicly known relation $\mathcal{R}$.

Remark 1. In the sequel, we restrict CCA adversaries further, in the sense that they shall not be allowed to query the decryption oracle on any ciphertext output by the simulator.

Definition 4. [$\mathcal{R}$-OW-RM-ADV] *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public key encryption scheme. Let $\mathcal{R}$ be an admissible relation defined on the corresponding message space M.*

Consider the following interaction between an adversary $\mathcal{A}$ of type ADV $\in$ $\{\mathrm{CPA}, \mathrm{CCA}\}$ *and a simulator $\mathcal{S}$.*

Step 1. *The simulator $\mathcal{S}$ runs the key generation algorithm on input a publicly known security parameter k, obtaining a pair (p_k, s_k) of corresponding public and secret keys.*

Step 2. *$\mathcal{S}$ selects uniformly at random a message $m_0 \in M_k$ and n messages* [6] *$(m_1, \dots, m_n) \in \mathcal{R}_{m_0}$.*

Step 3. *$\mathcal{S}$ encrypts all these messages using p_k and $\mathcal{E}$ and and hands the corresponding ciphertexts $(c_0, c_1, \dots, c_n)$ and the public key p_k to the adversary $\mathcal{A}$.*

Step 4. *$\mathcal{A}$ outputs a guess m for m_0.*

[6] recall that n is polynomial in the security parameter k and publicly known.

Now denote by $\mathsf{Succ}_{\mathcal{R}}^{OW}(\mathcal{A})$ *the advantage probability over a random guess* [7] *that the adversary outputs the correct plaintext message* m_0, *taken over the space defined by the (u.a.r.) choices described above and the random coins of the encryption algorithm.*

The scheme Π *is* secure in the sense of $\mathcal{R}$-OW-RM-ADV *if and only if for any adversary* $\mathcal{A}$ *of type* ADV *the probability* $\mathsf{Succ}_{\mathcal{R}}^{OW}(\mathcal{A})$ *is a negligible function of the security parameter* k.

We now introduce the definition of security in the sense of *indistinguishability*; at this, the adversarial goal is to retrieve *any* information from the plaintext corresponding to a challenge ciphertext having again at hand several ciphertexts whose corresponding plaintexts are related to the one hidden in the target, in the sense of the publicly known relation $\mathcal{R}$.

Definition 5. [$\mathcal{R}$-IND-RM-ADV] *Let* $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be a public key encryption scheme. Let* $\mathcal{R}$ *be an admissible relation defined on the corresponding message space* M..

Consider the following experiment carried over between an adversary $\mathcal{A}$ *of type* ADV $\in \{\text{CPA}, \text{CCA}\}$ *and a simulator* $\mathcal{S}$.

Step 1. *The simulator* $\mathcal{S}$ *runs the key generation algorithm on input a publicly known security parameter* k, *obtaining a pair* (p_k, s_k) *of corresponding public and secret keys. It forwards the public key* p_k *to the adversary* $\mathcal{A}$.
Step 2. *The adversary* $\mathcal{A}$ *selects two messages* $m_0, m_1 \in M_k$ *and hands them to* S.
Step 3. $\mathcal{S}$ *chooses a bit* b *uniformly at random and selects* n *messages* $(m_2, \ldots, m_{n+1})$ *uniformly at random from* R_{m_b}.
Step 4. $\mathcal{S}$ *encrypts all these messages using* p_k *and* $\mathcal{E}$ *and hands the corresponding ciphertexts* $(c_b, c_2, \ldots, c_{n+1})$ *to* $\mathcal{A}$.
Step 5. $\mathcal{A}$ *outputs a guess* b^* *for the bit* b.

Now denote by $\mathsf{Succ}_{\mathcal{R}}^{IND}(\mathcal{A})$ *the advantage probability, over a random guess*[8] *that the adversary outputs the correct bit* b.

This probability is again taken over the space defined by the (u.a.r.) choices described above and the random coins of the encryption algorithm.

The scheme Π *is* secure in the sense of $\mathcal{R}$-IND-RM-ADV *if and only if for any adversary* $\mathcal{A}$ *of type* ADV *the probability* $\mathsf{Succ}_{\mathcal{R}}^{IND}(\mathcal{A})$ *is a negligible function of the security parameter* k.

Now, our goal is actually to prove resistance against related messages attacks for general relations. We introduce the following definition:

---

[7] Without any information the adversary has probability $1/|M_k|$ of correctly guessing m_0.

[8] obviously, without any information the adversary has probability $\frac{1}{2}$ of guessing b, thus this advantage is meassured as $2Pr[b^* = b] - 1$.

Definition 6 (OW-RM-ADV / IND-RM-ADV). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public key encryption scheme, then it is* secure in the sense of OW-RM-ADV (resp. of IND-RM-ADV) *if it is* $\mathcal{R}$-OW-RM-ADV *(resp.* $\mathcal{R}$-IND-RM-ADV*) secure for every admissible relation $\mathcal{R}$.*

The following diagram summarizes some of the currently known relations among the main notions of security for public key schemes (see [3]). In the next two sections we will insert the new related-message attacks motivated notions in this diagram.Our results will in particular prove that resistance against related message attacks (in the OW sense) is strictly between the standard IND and OW notions, both for passive and active adversaries.

$$\begin{array}{ccc} IND-CPA & \substack{\longleftarrow \\ \not\longrightarrow} & IND-CCA \\ \downarrow\not\Uparrow & & \downarrow\not\Uparrow \\ OW-CPA & \substack{\longleftarrow \\ \not\longrightarrow} & OW-CCA \end{array}$$

Fig. 1. Relations from [3].

4 Relations between notions: The Passive Case

4.1 OW-CPA **versus** OW-RM-CPA

– OW-RM-CPA $\Rightarrow$ OW-CPA :
  Suppose an encryption scheme is not OW-CPA. This means, $\mathsf{Succ}^{OW}(\mathcal{A})$ is non-negligible in the security parameter; in other words, given a target ciphertext the adversary is able to recover the corresponding plaintext with non-neglibible probability. Clearly, the same holds if he is not only given the target ciphertext, but also n ciphertexts whose underlying plaintext are related to the hidden one.
– OW-CPA $\not\Rightarrow$ OW-RM-CPA
  1. Textbook RSA. Textbook RSA is known to be, for some specifications of the key generation algorithm, OW-CPA secure under the assumption that the RSA function is one-way. However, as proved by Franklin and Reiter [6], and later by Coppersmith [5] its one wayness can be overcome with non-negligible probability by a passive adversary taking advantage of related message attacks for some linear relation $\mathcal{R}$.
  Note that we have to assume above that the key generation is such that using Coppersmith's attack to try to invert the RSA function for a low public exponent does not compromise the OW-CPA security of the system. On the other hand, the related message attack of [5] relies on calculating the gcd of two polynomials of degree e, where e is the public exponent. This can

be computed in $O(e\,log^2\,e)$ time [12], which makes the attack very efficient for many exponents like the popular choice $e = 2^{16} + 1$.

2. Mc Eliece. Another example is the McEliece cryptosystem, whose OW-CPA security, as far as we know, remains unbroken for a suitable choice of parameters. On the other hand, Berson's related message attack [4] against the original scheme works no matter how the parameters are chosen (Therefore the cryptosystem is not OW-RMA-CPA secure:

Let us start by describing the McEliece cryptosystem in the usual way as a triple $(\mathcal{K}, \mathcal{E}, \mathcal{D})$:

- $\mathcal{K}$, on input the security parameter, generates three binary matrices: a $n \times n$ permutation matrix P, a $k \times k$ invertible matrix S and a $n \times k$ generator matrix G for a Goppa code with error correcting capacity t. The integers (n, k, t) depend on the security parameter. The secret key is $sk = (P, G, S)$ and the public key is the product matrix $pk = PGS$.
- $\mathcal{E}$, on input (pk, m), where $m \in \{0, 1\}^k$, selects uniformly at random an error vector $e \in (\mathbb{F}_2)^n$ with Hamming weight equal to t. Then computes the ciphertext $c = PGSm + e$.
- $\mathcal{D}$, on input (sk, c), computes $y = P^{-1}c$ and decodes y to a codeword x. Then returns $m = S^{-1}x$.

In a brute force attack against this cryptosystem, an adversary can guess a set E_0 consisting on k positions of the error vector e. If all these positions happen to equal 0 then $c_i = (PGSm)_i$ for every $i \in E_0$, where the subindex denotes the i-th coordinate. Then the message can be recovered from the ciphertext by inverting a submatrix of the public key PGS (corresponding to the rows with indices in E_0). However the probability of this guessing being correct is negligible in the security parameter.

Berson's attack is based on the fact that if a message is sent twice then it becomes much easier to guess coordinates of the ciphertext unaffected by the error vector. Namely, if an adversary $\mathcal{A}$ gets ciphertexts $c = PGSm + e$ and $c' = PGSm + e'$ then he can compute $c \oplus c'$ which equals $e \oplus e'$ and then he is able to determine the set $L_0 = \{1 \leq i \leq n : e_i = e'_i\}$. As t is always small when compared to n, the expected number of coincidences $e_i = e'_i = 1$ is low, therefore the probability of finding an unaffected position when choosing from L_0 is pretty high.

We give some concrete examples: the original proposal of McEliece has parameters $(n = 1024, k = 524, t = 50)$ and Berson's attack has success probability around 0.08 with only one guess (when the number of coincidences between positions of e and e' equals the expected value). For an example with bigger parameters we take the ones proposed in [8] for 256-bit security. These are $(n = 6624, k = 5129, t = 115)$. In this case the expected number of coincidences $e_i = e'_i = 1$ is between 1 and 2. If we assume it equals 2, then the probability of success of Berson's attack is around 0.04 with just one guess. The calculations are easy to make following Berson's paper.

4.2 IND-RM-CPA **versus** IND-CPA

– IND-RM-CPA $\Leftrightarrow$ IND-CPA

  Clearly, if the adversary can violate IND-CPA security then he can violate IND-RM-CPA security (he has at least the same information in the second case). Now, let us see how for any admissible relation $\mathcal{R}$ it holds IND-CPA $\Rightarrow$ $\mathcal{R}$-IND-RM-CPA.

  Let $\mathcal{R}$ be an admissible and publicly known relation for M. Let $\mathcal{A}$ be any pptm adversary who has non-negligibly above $\frac{1}{2}$ probability to win the $\mathcal{R}$-IND-RM-CPA challenge. We are going to argue he has non-negligibly above $\frac{1}{2}$ probability to win the IND-CPA challenge. Indeed, assume he choses messages m_0 and m_1 in the IND-CPA game and has received a challenge ciphertext c^*.

  Let $\mathcal{A}$ sample u.a.r. n messages $(m_2^0, \ldots, m_{n+1}^0) \in \mathcal{R}_{m_0}$. Further, he encrypts messages $m_2^0, \ldots, m_{n+1}^0$ with the public encryption algorithm yielding $(c_2^0, \ldots, c_{n+1}^0)$.

  Similarly, he samples u.a.r. n messages $(m_2^1, \ldots, m_{n+1}^1) \in \mathcal{R}_{m_1}$ and encrypts messages $m_2^1, \ldots, m_{n+1}^1$ with the public encryption algorithm yielding $(c_2^1, \ldots, c_{n+1}^1)$.

  The challenge ciphertext c^* is an encryption of m_b with b chosen u.a.r. from $\{0, 1\}$. Now the adversary plays the $\mathcal{R}-$IND-RM-CPA game with challenge ciphertexts: $(c^*, c_2^0, \ldots, c_{n+1}^0)$ and $(c^*, c_2^1, \ldots, c_{n+1}^1)$. If $b = 0$ then $\mathcal{A}$ has a non-negligible probability above $\frac{1}{2}$ to win first the game, while he has no advantage in the second one. If $b = 1$ then the reverse situation happens. This allows $\mathcal{A}$ to make a distinction between the two situations and break the IND-CPA security of the system: just let the adversary repeat the above experiment $T = T(k)$ times for each c^* and make a decision by majority.

4.3 IND-CPA versus OW-RM-CPA

– IND-RM-CPA$\Rightarrow$ OW-RM-CPA. This is obvious from the definitions.

– OW-RM-CPA $\not\Rightarrow$ IND-CPA

  Let us consider the following encryption scheme, which is a slight naive variation of El-Gamal encryption:

  For a fixed security parameter k, let p and q be k-bit primes and g a generator of the subgroup of prime order q of $\mathbb{Z}_p^*$. Let us denote that subgroup by G, and define a hash function H with domain G and image in $\{0, 1\}^{t(k)}$, for some polynomial t. We will assume this function H is an "ideal" hash function, following the so-called *random oracle paradigm* (see [1]).

  Now consider the following encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where

  - $\mathcal{K}$, the key generation algorithm, on the security parameter k outputs the secret key $sk := x$, which is chosen u.a.r. from $\mathbb{Z}_q$ and the public key, $g^x \mod p$.

- $\mathcal{E}$, the encryption algorithm, on input the public key and a message m in G, it selects u.a.r. $r \in \mathbb{Z}_q$ and outputs the ciphertext tuple

$$(g^r \mod p, mg^{xr} \mod p, H(m)).$$

- $\mathcal{D}$, the decryption algorithm, on input a triplet (a, b, c) will retrieve the message m as $(a^x)^{-1}b \mod p$ and check whether $H(m) = c$. If this check is correct, output m, otherwise, output $\perp$.

It is easy to see that this scheme is not IND-CPA secure; note that the adversary can compute the hash value for the two possible plaintexts m_0 and m_1 and see which one coincides with the hash value included in the target cyphertext, thus distinguishing with probability one an encryption of m_0 from an encryption of m_1.

Now, let us see that the scheme is OW-RM-CPA secure in the random oracle model. The idea behind the proof is that the third element on a ciphertext triplet should look like a random bit string unless you know the input message; thus, it should not help at all in breaking the one-wayness of the scheme.
We formalize the proof via a sequence of games or experiments, in which the adversary interacts with a simulator. This interaction will help us bounding his probability of success in violating the one wayness of the scheme. The first game is actually the true attack experiment. On the sequel, we will denote by S_i the event that the adversary wins in $\mathsf{Game}\ i$.

Game 0. This is the real game the adversary faces in the attack. Thus, the simulator runs the key generation algorithm and presents him a public key, p_k and $c_0, c_1, \ldots, c_n$ with n polynomial in k and each c_i an encryption of m_i under s_k and such that $(m_0, m_1, \ldots, m_n) \in \mathcal{R}$. Note that for each $i \in \{0, \ldots, n\}$, c_i is a triplet (x_i, y_i, z_i). At this, we should assume m_0 has been chosen u.a.r. from M_k and $(m_1, \ldots, m_n)$ u.a.r. from $\mathcal{R}_{m_0}$. Thus, $P[S_0] = \mathsf{Succ}_{\mathcal{R}}^{OW}(\mathcal{A})$.

Game 1. The adversary interacts with a simulator which instead of H uses a random-oracle simulation to the hash function.
Namely, the simulator keeps record of hash queries on a so called H-list, i.e. for every query $H(\mathsf{query})$ it outputs a value random selected uniformly at random from $\{0,1\}^{t(k)}$ and writes up the tuple $(\mathsf{query}, \mathsf{random})$ in his H-list. Subsequently, for a hash query $H(m)$ such that a record (m, z) appears in the H-list the answer of the simulator will be z.
Now, the random oracle assumption states that $P[S_0] = P[S_1]$.

Game 2. At this, we modify the simulator in the following sense: it selects u.a.r. two messages m_0 and m_0^* from M_k, and two n-tuples $(m_1, \ldots, m_n) \in \mathcal{R}_{m_0}$ and $(m_1^*, \ldots, m_n^*) \in \mathcal{R}_{m_0^*}$. It also runs the key generation algorithm and outputs p_k and $c_0, c_1, \ldots, c_n$ with each c_i a triplet (x_i, y_i, z_i) such that

- (x_i, y_i) is an ElGamal encryption of m_i under s_k

- z_i a hash query as described on the previous game, on input m_i^*.

At this, the amount of information the adversary may gather about a plaintext m_i from its ciphertext (x_i, y_i, z_i), comes strictly from the tuple (x_i, y_i), as z_i is constructed from m_i^* (and any correlation to m_i comes from the relation $\mathcal{R}$, and available a priori to the adversary).

Note that the output of this simulator may only be distinguished from the output of the simulator from Game 1 if

- the adversary queried, by chance, at some point the random oracle H on input m_i or m_i^* for some $i = 1, \ldots, n$. However, as $\mathcal{R}$ is admissible this happens with negligible probability only, wich we will denote $\varepsilon(k)$, or
- on the selected n-tuples $(m_0, m_1, \ldots, m_n)$ and $(m_0^*, m_1^*, \ldots, m_n^*)$ there exist $i \neq j \in \{0, \ldots, n\}$ such that $m_i = m_j$ and $m_i^* \neq m_j^*$ or vice versa. However, the adversary cannot perceive this difference, as given (x_i, y_i) and (x_j, y_j) he cannot tell whether they are encryptions of the same message or not - this follows easily from the fact that standard ElGamal is IND-CPA (see [14]).

Thus, $|P[S_2] - P[S_1]| \leq \varepsilon(k)$.

Now, as m_0^* has been chosen u.a.r from M_k and independently of m_0, only the first coordinates (x_i, y_i) on the output values c_i are correlated to the target plaintext m_0, i.e. of use for the adversary. Thus, his probability of success here is exactly the same as his probability of success against the OW-RM-CPA challenge against standard ElGamal. This is negligible, as standard ElGamal is IND-CPA$\Leftrightarrow$ IND-RM-CPA $\Rightarrow$ OW-RM-CPA. Therefore, $P[S_2]$ is negligible in the security parameter k. As

$$P[S_0] = \mathsf{Succ}_{\mathcal{R}}^{OW}(\mathcal{A}) \leq P[S_2] + \varepsilon(k)$$

this concludes the proof.

Note that the above simulation from Game 2 cannot be carried over by replacing each random oracle evaluation $H(m_i)$ with a truly random value. The reason is that the adversary expects n hash values $z_1, \ldots, z_n$ that come from related messages. Simple example, consider the relation $\mathcal{R}(m_1, \ldots, m_4) = 1$ iff $m_1 = m_3$ and $m_2 = m_4$; a 4-tuple of related messages is clearly distinguishable from a 4-tuple of random values.

It is worth mentioning that our proof does not use any specific feature of El-Gamal encryption scheme, just the fact that it is IND-CPA secure. Thus the result remains true if we choose any other IND-CPA secure scheme and modify the encryption algorithm by appending a hash image of the message to the corresponding ciphertext.

5 Relations between notions: The active case

Let us start by arguing with a classical example that, indeed, CCA adversaries are trivially more powerful than CPA adversaries:

5.1 **OW-RM-CPA $\nRightarrow$ OW-RM-CCA**

Indeed, let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be any OW-RM-CPA encryption scheme. Modify the decryption algorithm in that, for each public key, private key pair, on input a message c^* that is not a valid ciphertext with respect to the secret key in use, it outputs the error message $\perp = s_k$, i.e., the actual secret key!.

Indeed, the resulting ciphertext is no longer resisting any OW-RM-CCA attack, as the adversary trivially decrypts the target ciphertext with the secret key retrieved by querying the decryption oracle with any invalid ciphertext.

5.2 IND-RM-CCA **versus** IND-CCA

– IND-RM-CCA $\Leftrightarrow$ IND-CCA The argument is, for both implications, fully analogous to the pasive (CPA) case from Section 4.

5.3 OW-CCA **versus** OW-RM-CCA

– OW-RM-CCA $\Rightarrow$ OW-CCA :
Similarly as for the passive case, if an encryption scheme is not OW-CCA, given a target ciphertext the adversary is able to recover the corresponding plaintext with non-neglibible probability. Again, the same holds if he is not only given the target ciphertext, but also n ciphertexts whose underlying plaintext are related to the hidden one.
– OW-CCA $\nRightarrow$ OW-RM-CCA
Let us consider a "hashed" textbook RSA, that is, the textbook RSA encryption algorithm is modified in that, on input a public key (e, N) and a plaintext m, it oututs $(c, H(m))$, with c a standard textbook RSA encryption ($m^e \pmod N$) and H a random oracle image of m. Then, the decryption algorithm is modified in that, on input a ciphertext (c_1, c_2) it performs the usual textbook RSA decryption, retrieves m_1 and outputs m_1 iff $H(m_1) = c_2$ (and $\perp$ otherwise).
Moreover, consider the key generation procedure so that the resulting scheme is OW-CPA but not OW-RM-CCA, i.e., it is OW-CPA but the related message attacks of Franklin-Reiter/Coppersmith [6, 5] still apply.
We are going to prove this scheme is plaintext aware, i.e., the adversary can only construct a ciphertext if he knows the corresponding plaintexts. [9]
First, note that the adversarial probability of success against the one-wayness of the scheme does not augment by decrypting eavesdropped messages with the decryption oracle; indeed, a CPA adversary already has access to polynomially many (plaintext, ciphertext) pairs; and note that knowing such a pair (m, c) implies knowledge of the corresponding pair $(m, H(m))$. Following the standard RO simulation, we assume $\mathcal{A}$ keeps a two column table with entries

[9] In this case, as knowing a pair (plaintext, ciphertext) implies knowing the H-image of the plaintext, strong plaintext awarness and plaintext awareness in the RO, as defined by Bellare et al.(see [3, 2]) are equivalent.

$(x, H(x))$ corresponding to his random oracle queries. Furthermore, he may also write there pairs $(x, H(x))$ retrieved from eavesdropped messages. We will refer to this table as the H-list.
Assume $\mathcal{A}$, with polynomially many queries to H and to the decryption oracle, outputs a new (not eavesdropped!) valid ciphertext $c = (c_1, c_2)$. Then, by a standard argument $\mathcal{A}$ must know the corresponding plaintext m. Otherwise, there must exist a so called *plaintext extractor* $\mathcal{PE}$ which could have output m just from the public key and $\mathcal{A}$'s H-list. Indeed, on input a valid ciphertext $c = (x, y)$, $\mathcal{PE}$ searches on the H-list to see whether y is a response to an H query performed by $\mathcal{A}$. If this is the case, then he outputs the x-entry corresponding to this query. Note that this is the case except with negligible probability (the probability that A guessed the value c_2 without querying H).
Thus, the decryption oracle is of no use to the adversary and OW-CCA security is equivalent to OW-CPA. As a result, the scheme is OW-CCA in the random oracle model, while not OW-RM-CCA. This concludes the proof.

Note that in the proof we do not use any specific fact about textbook RSA, just that it is possible to find parameters such that the scheme is OW-CPA but not OW-RMA-CPA secure (see Section 4). Thus the proof works for any scheme with this property.

5.4 OW-RM-CCA **versus** IND-RM-CCA

- IND-RM-CCA $\Rightarrow$ OW-RM-CCA
  This follows trivially from the definitions.
- OW-RM-CCA $\not\Rightarrow$ IND-RM-CCA
  Let us construct an encryption scheme which we can prove OW-RM-CCA but which is not IND-RM-CCA. At this, we modify a well known construction of Bellare and Rogaway (see [1]), which is IND-CCA in the random oracle model. Let us start by describing this construction: Once the security parameter k is fixed, assume that we fix $\{0,1\}^k$ as plaintext space together with a function $f_k : \{0,1\}^k \to \{0,1\}^k$, chosen from a one-way trapdoor permutation family. Moreover, consider two hash functions

  $$G_k : \{0,1\}^k \to \{0,1\}^k \text{ and } H_k : \{0,1\}^{2k} \to \{0,1\}^{k_1},$$

  where k_1 is polynomial in k (for asymptotic arguments, we may assume k_1 grows roughly as k). In the sequel, we drop the subscript k unless necessary. The scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is described as follows:
  - $\mathcal{K}$, on input k outputs a description of the function f (the public key pk) and trapdoor information for computing its inverse f^{-1} (the secret key sk).
  - $\mathcal{E}$, on input (pk, m), where $m \in \{0,1\}^k$, it selects uniformly at random a value r in $\{0,1\}^k$ and computes

    $$x = f(r),\ y = m \oplus G(r) \text{ and } z = H(m||r)$$

    and outputs the ciphertext $c = (x, y, z)$.

- $\mathcal{D}$, on input (sk, c) computes

$$r = f^{-1}(x), \text{ and } m = y \oplus G(r).$$

Now, if $z = H(m||r)$ the algorithm returns m, otherwise, it returns $\perp$.

This scheme can be proven IND-CCA secure in the random oracle model. Namely, one should assume that H and G behave like random oracles, using the usual paradigm.

Now let us construct a related scheme, modifying the random oracle H (in that it is now defined on $\{0,1\}^k$) and the encryption algorithm. Consider $\hat{\Pi} = (\mathcal{K}, \hat{\mathcal{E}}, \hat{\mathcal{D}})$ such that:

- $\mathcal{K}$, as in Π.
- $\hat{\mathcal{E}}$, on input (pk, m), where $m \in \{0,1\}^k$, it selects uniformly at random a value r in $\{0,1\}^k$ and computes

$$x = f(r), \; y = m \oplus G(r) \text{ and } z = H(m)$$

and outputs the ciphertext $c = (x, y, z)$.
- $\hat{\mathcal{D}}$, on input (sk, c) computes

$$r = f^{-1}(x), \text{ and } m = y \oplus G(r).$$

Now, if $z = H(m)$ the algorithm returns m, otherwise, it returns $\perp$.

It is easy to see that $\hat{\Pi}$ is no longer IND-CCA (actually, not even IND-CPA), as an encryption of m_0 can be distinguished from an encryption of m_1 just by checking on the third element of the ciphertext triplet.

Let us however argue that this scheme is OW-RM-CCA.

Let $\mathcal{R}$ be any admissible relation. Again, we derive the proof via a sequence of games or experiments, in which the adversary interacts with a simulator. The first game is the true attack experiment and we will denote by S_i the event that the adversary wins in Game i.

Game 0. This is the true $\mathcal{R}$-OW-RM-CCA experiment. Thus, the simulator runs the key generation algorithm and presents him a public key, p_k and $c_0, c_1, \ldots, c_n$ with n polynomial in k and each c_i an encryption of m_i under s_k and such that $(m_0, m_1, \ldots, m_n) \in \mathcal{R}$. For each $i \in \{0, \ldots, n\}$, each ciphertext c_i is a triplet (x_i, y_i, z_i). At this, we should assume m_0 has been chosen u.a.r. from $\{0,1\}^k$ and $(m_1, \ldots, m_n)$ u.a.r. from $\mathcal{R}_{m_0}$. Thus, $P[S_0] = \mathsf{Succ}^{OW}_{\mathcal{R}}(\mathcal{A})$.

Game 1. Here, the simulator $\mathcal{S}$ simulates all his queries to the hash functions H and G using the random oracle paradigm, that is, he will as usual keep record of hash queries on two lists, which we will refer to as H-list and G-list. That is, i.e. for every query $H(\mathsf{query})$ (resp., $G(\mathsf{query})$ it outputs a value random selected uniformly at random from the corresponding image space and writes up the tuple $(\mathsf{query}, \mathsf{random})$ in the corresponding list. Subsequently, for a hash query $H(q)$ or $G(q)$ such that a record (q, z) appears in the corresponding H-list or G-list the answer of the simulator will be z. The random oracle paradigm states now that $P[S_0] = P[S_1]$.

Game 2. We now change how the simulator constructs the values $c_0, c_1, \ldots, c_n$ presented to the adversary. In particular, the first two elements of a ciphertext triplet c_i are constructed as in $\hat{\mathcal{E}}$, but not the last one. $\mathcal{S}$ chooses m_0^* u.a.r. from $\{0,1\}^k$ and $(m_1^*, \ldots, m_n^*)$ u.a.r. from $\mathcal{R}_{m_0^*}$. Now, each z_i is the response of the simulators H-query on input m_i^*.
Note that the output of this simulator may only be distinguished from the output of the simulator from Game 1 if

- the adversary or the decryption oracle queried, by chance, at some point the random oracle H on input m_i or m_i^* for some $i = 1, \ldots, n$. We know that the the decryption oracle rejects queries coming from m_i for $i = 1, \ldots, n$, and as $\mathcal{R}$ is admissible there is negligible probability that an encryption of any m_i^* has been fed into the decryption oracle. Thus, the probability P_2 that any of the above happens is negligible.
- on the selected n-tuples $(m_0, m_1, \ldots, m_n)$ and $(m_0^*, m_1^*, \ldots, m_n^*)$ there exist $i \neq j \in \{0, \ldots, n\}$ such that $m_i = m_j$ and $m_i^* \neq m_j^*$ or vice versa. However, the adversary cannot perceive this difference, as given $(f(r), m_i \oplus G(r))$ is and IND-CPA encryption of m_i.

Thus, $|P[S_2] - P[S_1]| \leq P_2$.
Game 3. Let us now change the simulation further, by imposing that also the second element in a ciphertext triplet c_i is constructed independently of the "unstarred" messages m_i. Namely, now the second component y_i will be a bitstring selected uniformly at random from $\{0,1\}^l$. Note that, the random oracle simulation of G guarantees that, if the adversary does not know r, $G(r)$ is for him a u.a.r. selected element in $\{0,1\}^k$, (and thus, so is $G(r) \oplus m_i$). As a result, given that f is a one-way permutation, an element y_i from one of the ciphertext triplets output by the simulator is only distinguishable from a random bitstring in $\{0,1\}^l$ if $\mathcal{A}$ queried, at some point, the random oracle G on r, which happens with negligible probability (P_3) only. As a result,

$$|P[S_3] - P[S_2]| \leq P_3.$$

Note moreover that

$$P[S_3] = \frac{1}{2^l},$$

as in this game the adversary obtains no information about the challenge ciphertext m_0 from the simulatior.
As a result,

$$P[S_0] \leq \frac{1}{2^l} + P_2 + P_3.$$

and thus $\mathsf{Succ}_{\mathcal{R}}^{OW}$ is negligible in the security parameter, which concludes the proof.

6 Summary and Further Remmarks

With the above results, we have been able to fully complete the graph of relations from [3], inserting the new notions we have defined:

$$
\begin{array}{lcl}
IND-CPA & \overset{\longleftarrow}{\not\longrightarrow} & IND-CCA \\
\quad\Updownarrow & & \quad\Updownarrow \\
IND-RM-CPA & & IND-RM-CCA \\
\quad\downarrow\not\Uparrow & & \quad\downarrow\not\Uparrow \\
OW-RM-CPA & \overset{\longleftarrow}{\not\longrightarrow} & OW-RM-CCA \\
\quad\downarrow\not\Uparrow & & \quad\downarrow\not\Uparrow \\
OW-CPA & \overset{\longleftarrow}{\not\longrightarrow} & OW-CCA
\end{array}
$$

Fig. 2. Completing the graph from [3].

It is somewhat not surprising that the indistinguishability notions of security imply resistance against related message attacks. On the other hand, it is as well natural that also in this scenario active adversaries are more powerful than passive ones. More surprising is perhaps the fact that OW-RM security is strictly between OW and IND security, which clearly indicates that, when using a *one-way* secure scheme, habits like repeating fixed headings on messages may not be safe.

It remains to investigate further what happens for the case of *non-malleability*. Recall that this property captures the resistance of an encryption scheme to attacks that aim at constructing a new valid ciphertext from eavesdropped ciphertexts of meaningfully related plaintexts. Already finding a suitable definition of what it means to be *non-malleable* in the related-message attacks scenario is non-trivial; in particular, determining what kind of relations we may consider admissible seems quite involved.

Acknowledgment

The first author is partially supported by projects MTM2007-67884-C04-01 and CCG07-UCM/ESP-2797. The second author is partially supported by projects MTM2005-02865 and Grupos UCM 910444. The authors are grateful to Jorge L. Villar for several enlightening discussions on the topic.

References

1. M. Bellare, and P. Rogaway, *Random Oracles are Practical: a Paradigm for Designing Efficient Protocols*, Proc. of the 1st CCS, ACM Press, New York, pp. 62–73, 1993.

2. M. Bellare, and P. Rogaway, *Optimal Asymmetric Encryption: How to Encrypt with RSA* , Proc. of EUROCRYPT'94, LNCS 950, pp. 92–111, 1994.
3. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, *Relations among notions for public key encryption schemes*, Proc. of CRYPTO'98, LNCS 1462, pp. 26–45, 1998.
4. T. A. Berson, *Failure of the McEliece Public-Key Cryptosystem Under Message-Resend and Related-Message Attack*, Proc. of CRYPTO'97, LNCS 1294, pp. 213–220, 1997.
5. D. Coppersmith, M. Franklin, J. Patarin and M. Reiter, *Low-Exponent RSA with Related Messages*, Proc. of EUROCRYPT'96, LNCS 1070, pp. 1–9, 1997.
6. M. Franklin and M. Reiter, *A linear protocol failure for RSA with exponent three*, presented at the CRYPTO'95 Rump Sesion., 1995.
7. J. Hastad, *Solving simultaneous modular equations of low degree*, SIAM J. Comput. 17, pp. 336–341, 1988.
8. D.J. Bernstein, T. Lange and C. Peters, *Attacking and Defending the McEliece Cryptosystem*, Proc. of PQCrypto 2008, LNCS 5299, pp. 31–46, 2008.
9. P. Smith and M.J.J. Lennon, *LUC: A new public key system*, in E.G. Douglas, Editor, Ninth IFIP Int. Symp. on Computer Security, Elsevier, pp. 103-117, 1994.
10. R. G.E. Pinch, *Extending the Hastad attack to LUC*, Electronic Letters vol. 31 issue 21, pp. 1827–1828, 1995.
11. D. Pointcheval, *Provable Security for Public Key Schemes*, in Contemporary Cryptology, Advanced Courses in Mathematics CRM Barcelona, Birkhäuser, pp. 133–191. 2005.
12. V. Strassen, *The computational complexity of continued fractions*, in SIAM J. Comput. 12(1), pp. 1–27, 1983.
13. O. Yacobi and Y. Yacobi, *A New Related Message Attack on RSA*, Proc. of PKC'2005, LNCS 3386, pp. 1–8, 2005.
14. Y. Tsiounis and M. Yung, *On the Security of ElGamal Based Encryption*, Proc. of PKC '98: Proceedings of the First International Workshop on Practice and Theory in Public Key Cryptography, LNCS 1431 pp. 117–134, 1998.