Full version of the extended abstract, which appeared in
the proceedings of the Fourth Conference on Security in Communication Networks 2004
(8 – 10 september 2004, Amalfi, Italy)
C. Blundo Ed. Springer-Verlag, LNCS ????, pages ???–???.

On the Security Notions for Public-Key Encryption Schemes

Duong Hieu Phan and David Pointcheval

École normale supérieure – Dépt d'informatique
45 rue d'Ulm, 75230 Paris Cedex 05, France.
{duong.hieu.phan,david.pointcheval}@ens.fr

Abstract. In this paper, we revisit the security notions for public-key encryption, and namely indistinguishability. We indeed achieve the surprising result that no decryption query before receiving the challenge ciphertext can be replaced by queries (whatever the number is) after having received the challenge, and vice-versa. This remark leads to a stricter and more complex hierarchy for security notions in the public-key setting: the (i, j)-IND level, in which an adversary can ask at most i (j resp.) queries before (after resp.) receiving the challenge. Excepted the trivial implications, all the other relations are strict gaps, with no polynomial reduction (under the assumption that IND-CCA2 secure encryption schemes exist.) Similarly, we define different levels for non-malleability (denoted (i, j)-NM.)

Keywords: public-key encryption, semantic security, non-malleability, pseudo-random functions.

1 Introduction

Relations between security notions for public-key encryption scheme have been deeply studied, namely in the recent papers of Bellare *et al.* [2] and of Bellare and Sahai [4]. These papers are based on the seminal works of Goldwasser and Micali [8] which defined the notions of polynomial security, or indistinguishability denoted IND; Noar and Yung [12] and Rackoff and Simon [14], which introduced stronger scenarios of attacks, and Dolev, Dwork and Noar [5, 6] which proposed a stronger security notion: the non-malleability.

It is now clear that the security notions (indistinguishability and non-malleability) have to be studied under specific attacks: the basic scenario in the public-key setting in the chosen-plaintext attacks (CPA), but more interesting situations are captured by the chosen-ciphertext attacks. Chosen-ciphertext attacks have been split in two families, for historical reasons explained below, the non-adaptive ones (denoted CCA1) and the adaptive ones (denoted CCA2.) In both cases, the adversary has access to a decryption oracle. In the former case, this access is limited until the challenge ciphertext is known, while the latter case allows an unlimited access (with the natural restriction not to ask the challenge ciphertext.)

In this paper, we consider more concrete cases by introducing the (i, j)-IND security level, in which an adversary can ask at most i (j resp.) queries before (after resp.) receiving the challenge ciphertext. The reason for such a more precise notation, than just IND-CCA1 thus captured by $(\mathrm{poly}(\cdot), 0)$-IND and IND-CCA2 captured by $(\mathrm{poly}(\cdot), \mathrm{poly}(\cdot))$-IND, is that we can prove that no decryption query before receiving the challenge can be replaced by queries (whatever the number is) after having received the challenge, and vice-versa. Indeed, excepted the trivial implications, all the other relations between the (i, j)-IND security levels are strict gaps, with no polynomial reduction (under the basic assumption that IND-CCA2 secure encryption schemes exist.)

As an application, we introduce a new kind of attack, we call the *post-challenge chosen-ciphertext attack*, denoted CCAO2 (for chosen-ciphertext attacks in the 2nd stage only.) This new scenario completes the above picture with the $(0, \mathrm{poly}(\cdot))$-IND security notion. Furthermore, from a practical point of view, it models very realistic situations since it limits the control the adversary may have on the "a priori" distribution of the plaintexts, but it also encompasses situations where the adversary starts the attack when it becomes aware of the importance of a specific ciphertext (after the latter is generated and sent.)

Even if it seems clear that the CCA1 security model has been introduced because the authors [12] failed at achieving the CCA2 level [14], it is still studied, and considered as a goal to be achieved. However, it seems more realistic to consider scenarios where the adversary has not so much control on the challenge plaintexts: they could just be chosen right after having received the identity and the public-key of the target decryptor. Therefore, the messages m_0 and m_1 should be chosen before having access to any oracle.

1.1 Related Work

In the early 80s, people formally defined the security notions for cryptographic primitives (namely, for signature [10, 11], and for encryption [8] with the notions of polynomial security, or indistinguishability denoted IND.) While these notions did not evolve so much for signatures since adaptive chosen-message attacks were introduced, stronger notions appeared later for encryption, namely after the zero-knowledge concept [9].

Indistinguishability was indeed defined in the basic scenario only, where the adversary has just access to the public information, and can thus encrypt any plaintext of its choice, hence the name of chosen-plaintext attacks (denoted CPA.) Naor and Yung [12] introduced the notion of chosen-ciphertext attacks. However, their solution based on non-interactive zero-knowledge proofs of membership, without the recent non-malleable NIZK or simulation-soundness [15] notions. Therefore, they could not simulate correctly the decryption oracle after the adversary had received the challenge ciphertext. As a consequence, they restricted the chosen-ciphertext attacks to be non-adaptive, in the sense that the decryption queries could not depend on the challenge ciphertext (*a.k.a. lunchtime attacks*, denoted CCA1.) Rackoff and Simon [14] extended this notion, with an unlimited access to the decryption oracle (excepted on the challenge ciphertext), denoted CCA2, and provided a candidate granted the non-interactive zero-knowledge proofs of knowledge.

The above improvements were about the attack model, but then also appeared a relaxed goal for the adversary: the non-malleability [5, 6]. In [2], Bellare *et al.* provided comparisons between all the resulting security notions, but just between the large classes IND/NM combined with CPA, CCA1 or CCA2.

1.2 Contributions

Adaptive chosen-ciphertext attacks (CCA2) are clearly the strongest scenario in the framework of the complexity theory, using perfect oracles and polynomial reductions, or even exact reductions. However, this notion can be considered as a very strong notion. In the real life, which motivated the exact/concrete security [3, 1, 13] (vs. asymptotic or polynomial framework), the adversary may be limited in the number of queries it can ask to the decryption oracle, and then the scheme can be designed

to resist such a specified number of queries. Therefore, it's worth considering the exact/concrete security notions. We thus introduce two classes of security notions: (i, j)-IND and (i, j)-NM, or even more precisely (t, i, j)-IND and (t, i, j)-NM secure schemes, which resist (in the indistinguishability sense or non-malleability sense) to adversaries which can make exactly i (j resp.) decryption queries before (after resp.) receiving the challenge within time t.

First, we consider the relations inside each class of security. At a first glance, one could think that a query in the second stage is much more important than a query in the first stage (since then, queries may depend on the challenge ciphertext, and this would justify the consideration of CCA1 and CCA2 only in chosen-ciphertext scenarios.) Surprisingly, we show that no query before receiving the challenge can be replaced by queries (whatever the number is) after having received the challenge, and vice-versa: a query before, helps to correctly choose the messages m_0 and m_1.) This remark leads to a strict and more complex hierarchy for security notions in the public-key setting: excepted the trivial implications, all the other relations are strict gaps, with no polynomial reduction.

As an illustration, we introduce post-challenge chosen-ciphertext attacks (denoted CCAO2.) In this scenario, the adversary has access to the decryption oracle, but after the challenge ciphertext is known only. From the above result, we show that any security notion (IND or NM) under these attacks (CCA1 and CCAO2) are independent. Furthermore, we show that CCA1 + CCAO2 does not necessarily yield CCA2.

2 Security Model

Let us review the main security notions for public-key encryption, but also more theoretical notions, which will be useful for exhibiting gaps, such as the pseudo-random function families.

2.1 Public-Key Encryption

A public-key encryption scheme π is defined by the three following algorithms:

- The *key generation algorithm* $\mathcal{G}$. On input 1^k, where k is the security parameter, the algorithm $\mathcal{G}$ produces a pair $(\mathsf{pk}, \mathsf{sk})$ of matching public and private keys.
- The *encryption algorithm* $\mathcal{E}$. Given a message m (in the space of plaintexts $\mathcal{M}$) and a public key pk, $\mathcal{E}_{\mathsf{pk}}(m)$ produces a ciphertext c (in the space of ciphertexts $\mathcal{C}$) of m. This algorithm may be probabilistic (involving random coins $r \in \mathcal{R}$) it is then denoted $\mathcal{E}_{\mathsf{pk}}(m; r)$.
- The *decryption algorithm* $\mathcal{D}$. Given a ciphertext $c \in \mathcal{C}$ and the secret key sk, $\mathcal{D}_{\mathsf{sk}}(c)$ gives back the plaintext $m \in \mathcal{M}$.

2.2 Security Notions

As already noted, the fundamental security notions are the indistinguishability and the non-malleability.

Definition 1 (Indistinguishability). Let $\pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. Let us consider a two-stage probabilistic adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ whose running time

is bounded by t. We define the advantage of $\mathcal{A}$ against the indistinguishability of π as follows:

$$\mathsf{Adv}^{\mathsf{ind}}_{\pi}(\mathcal{A}) \stackrel{\mathsf{def}}{=} \left| 2 \times \Pr_{b,r}\left[\begin{array}{c}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{G}(1^k), (m_0, m_1, s) \leftarrow \mathcal{A}_1(\mathsf{pk}),\\ c = \mathcal{E}_{\mathsf{pk}}(m_b, r), b' = \mathcal{A}_2(m_0, m_1, s, c) : b' = b\end{array}\right] - 1 \right|.$$

We insist above on that $\mathcal{A}_1$ outputs two messages m_0 and m_1 such that $|m_0| = |m_1|$. As usual, we define by $\mathsf{Adv}^{\mathsf{ind}}_{\pi}(t)$ the maximum advantage over all the adversaries $\mathcal{A}$ whose running time is bounded by t. Then we say that π is (t, ε)-IND secure if $\mathsf{Adv}^{\mathsf{ind}}_{\pi}(t)$ is less than ε.

Definition 2 (Non-malleability). Let $\pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. Let us consider a two-stage probabilistic adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ whose running time is bounded by t. We define the advantage of $\mathcal{A}$ against the non-malleability of π by:

$$\mathsf{Adv}^{\mathsf{nm}}_{\pi}(\mathcal{A}) \stackrel{\mathsf{def}}{=} \mathsf{Succ}^{\mathsf{nm}}_{\pi}(\mathcal{A}) - \mathsf{Succ}^{\mathsf{nm},\$}_{\pi}(\mathcal{A}),$$

where the two successes use the same probability distribution, for a distribution of plaintexts M and a binary relation R, generated by

$$\begin{array}{c}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{G}(1^k), (M, s) \leftarrow \mathcal{A}_1(\mathsf{pk});\\ m, \tilde{m} \leftarrow M; c \leftarrow \mathcal{E}_{\mathsf{pk}}(m, r); (R, y) \leftarrow \mathcal{A}_2(M, s, c); x \leftarrow \mathcal{D}_{\mathsf{sk}}(y)\end{array}$$

and

$$\begin{array}{rl}\mathsf{Succ}^{\mathsf{nm}}_{\pi}(\mathcal{A}) & \stackrel{\mathsf{def}}{=} \Pr[y \neq c \wedge x \neq \perp \wedge R(x, m)]\\ \mathsf{Succ}^{\mathsf{nm},\$}_{\pi}(\mathcal{A}) & \stackrel{\mathsf{def}}{=} \Pr[y \neq c \wedge x \neq \perp \wedge R(x, \tilde{m})].\end{array}$$

We also define by $\mathsf{Adv}^{\mathsf{nm}}_{\pi}(t)$ the maximum advantage over all the adversaries $\mathcal{A}$ whose running time is bounded by t. Then we say that π is (t, ε)-NM secure if $\mathsf{Adv}^{\mathsf{nm}}_{\pi}(t)$ is bounded by ε.

This definition models the above intuition about non-malleability (the adversary cannot output a second ciphertext so that the corresponding plaintexts are meaningfully related.) This is a particular case of the general definition used in [2, 4], and denoted $\mathsf{CNM}^{(k)}$, in which the adversary could output a vector of ciphertexts $(y_1, \ldots, y_k)$ of the plaintexts $(x_1, \ldots, x_k)$ and a relation R so that $R(x_1, \ldots, x_k, m)$ holds more often than $R(x_1, \ldots, x_k, \tilde{m})$. A discussion is provided in Section 3.3.

2.3 Attack Models

For a public-key encryption, the adversary has access, as anybody, to the encryption key. It can thus encrypt any plaintext of its choice. Hence the basic attack is called "Chosen Plaintext Attack", or in short CPA. But the adversary may also have access to more information, and namely some decryptions. This is modeled by an access to the decryption oracle.

Definition 3 (Lunchtime Attacks). An adversary is called a *non-adaptive chosen-ciphertext adversary*, (or a lunchtime adversary, denoted by $\mathsf{CCA1}$-adversary) if it can access the oracle before the challenge ciphertext is known only.

Definition 4 (Adaptive Attacks). An adversary is called an *adaptive chosen-ciphertext adversary* (denoted by CCA2-adversary) if it can access the oracle whenever it wants, that is before and after the challenge ciphertext is known, with the sole restriction not to use it on the challenge itself.

These two attack models are the classical ones, but for historical reasons. For more generality, we introduce a more precise definition with a (i, j)-CCA adversary which can ask at most i queries (resp. j queries) before the challenge ciphertext is known (after resp.)

Definition 5 (Chosen-Ciphertext Attack). An adversary is called an (i, j) *chosen-ciphertext adversary* (denoted by (i, j)-CCA adversary) if it can access the oracle, up to i times before the challenge ciphertext is known, and up to j times after, still with the restriction not to use it on the challenge itself.

Notation. An encryption scheme $\pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is said to be (t, ε)-XXX-YYY secure if for any YYY-adversary $\mathcal{A}$ against the security XXX within running time t, where XXX can be either IND or NM, and YYY can be either CPA, CCA1, CCA2, or (i, j)-CCA, the advantage of $\mathcal{A}$ is bounded by ε. In the latter case, in short, we say that π is (t, ε, i, j)-IND secure (resp. (t, ε, i, j)-NM secure) if for any (i, j)-CCA adversary $\mathcal{A}$ whose running time is bounded by t, $\mathsf{Adv}^{\mathsf{ind}}_{\pi}(\mathcal{A}) \leq \varepsilon$ (resp. $\mathsf{Adv}^{\mathsf{nm}}_{\pi}(\mathcal{A}) \leq \varepsilon$.)

2.4 Trapdoor One-Way Permutations

Some constructions below will need the existence of a trapdoor one-way permutation. Informally, for such a permutation which can be inverted granted the trapdoor, it should be hard to invert without the latter:

Definition 6 (One-Way Permutation). Let $f : \{0,1\}^{\ell} \to \{0,1\}^{\ell}$ be a permutation, and let us consider the adversary $\mathcal{A}$ against the one-wayness. We define the success probability of $\mathcal{A}$ for inverting f by: $\mathsf{Succ}^{\mathsf{ow}}_{f}(\mathcal{A}) \stackrel{\mathsf{def}}{=} \Pr_x[\mathcal{A}(f(x)) = x]$. As above, we also denote by $\mathsf{Succ}^{\mathsf{ow}}_{f}(t)$ the maximal success over all the adversaries whose running time is bounded by t. Therefore, we say that f is (t, ε)-OW if $\mathsf{Succ}^{\mathsf{ow}}_{f}(t)$ is bounded by ε.

2.5 Pseudo-Random Functions

The notion of pseudo-random functions [7] requires that any adversary, accessing an oracle $\mathcal{O}_b$, which is either a truly random function F (in case $b = 0$) or a random instance F_K in the family $\mathcal{F} = (F_K)$ (in case $b = 1$), cannot guess the actual bit b. The advantage of such an adversary is defined by:

Definition 7 (Pseudo-Random Functions).

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{A}) = 2 \times \Pr_{b,F,K}[\mathcal{O}_0 = F, \mathcal{O}_1 = F_K, \mathcal{A}^{\mathcal{O}_b} = b] - 1.$$

We also denote by $\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(t, n)$ the maximal advantage over all the adversaries whose running time is bounded by t, which makes less than n queries to the oracle. Finally, we say that a family $\mathcal{F}$ is a (ε, t, n)-PRF if $\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(t, n)$ is bounded by ε.

3 Concrete Security

In this section, we show some non-intuitive gaps in the (i, j)-IND class: a decryption query in the first stage cannot be postponed to the second stage, and reversely. As a consequence, we are interested by a possible comparison of the importance of queries in the first stage and in the second stage. In the following, we formally prove that allowing one more query in the first stage gives a different strength to an adversary than allowing it as many queries as it wants in the second stage. We do the same for an additional query in the second stage, which cannot be compared with even many queries in the first stage.

3.1 Preliminaries

To this aim, we need a new intractability problem, which can hopefully be related to a classical PRF one. Furthermore, we denote below by PRP the analogous notion as PRF, when the functions are permutations. Similarly, we denote by $\mathsf{Adv}^{\mathsf{prp}}_{\mathcal{G}}(\mathcal{A})$ the advantage with which an adversary can distinguish a permutation, randomly drawn from the pseudo-random permutation family, and a truly random permutation. Note that the inverse is not available (*i.e.*, we do not consider the super pseudo-randomness.)

Definition 8. For any function (or permutation) G and any two-stage adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we denote by $\mathsf{Succ}^{m,n}_{G}(\mathcal{A})$ the success probability for $\mathcal{A}_2(v, s)$ to output $G^n(v)$, for a given random value v, and a working tape s transmitted by $\mathcal{A}_1$, when $\mathcal{A}_1$ was limited to m queries to G, and $\mathcal{A}_2$ is limited to $n-1$ queries.

$$\mathsf{Succ}^{m,n}_{G}(\mathcal{A}) = \Pr[v \xleftarrow{R} \mathcal{M}; s \leftarrow \mathcal{A}_1^G : \mathcal{A}_2^G(v, s) = G^n(v)].$$

As before, we denote by $\mathsf{Succ}^{m,n}_{G}(t)$ the maximal success probability over all the adversaries whose running time is bounded by t.

Proposition 9. *For any function/permutation G randomly drawn from a pseudo-random function/permutation family $\mathcal{G}$ into a set of cardinality larger than $\{0,1\}^\ell$, we have:*

$$\mathsf{Succ}^{m,n}_{G}(t) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{G}}(m + 2n - 1, t) + \frac{mn+1}{2^\ell}.$$

Proof. We prove that for any adversary $\mathcal{A}$ against the above "one-more evaluation", we can design a PRF-adversary $\mathcal{B}$ such that $\mathsf{Succ}^{m,n}_{G}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{G}}(\mathcal{B})$. Our adversary $\mathcal{B}$ simulates $\mathcal{A}$'s view as follows: whenever $\mathcal{A}$ queries G, $\mathcal{B}$ asks the same query to $\mathcal{O}_b$ and forwards the answer to $\mathcal{A}$ (at most $m + n - 1$ queries.) Eventually, $\mathcal{A}$ outputs x. $\mathcal{B}$ successively queries the oracle $\mathcal{O}_b$ to get $y = \mathcal{O}_b^n(v)$. If $x = y$, $\mathcal{B}$ outputs its guess $b' = 1$, otherwise $\mathcal{B}$ outputs $b' = 0$.

- when $b = 1$, $\mathcal{B}$ actually accesses in fact G and therefore $y = \mathcal{O}_b^n(v) = G^n(v)$, whenever $\mathcal{A}$ outputs the correct value $G^n(v)$. $\mathcal{B}$ always wins the game when $\mathcal{A}$ wins. Since $b' = 1$ means $x = y$:

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{G}}(\mathcal{B} \,|\, b = 1) = 2\Pr[x = y \,|\, b = 1] - 1 = 2\mathsf{Succ}^{m,n}_{G}(\mathcal{A}) - 1.$$

- when $b = 1$, the value $y = \mathcal{O}_b^n(v)$ that $\mathcal{B}$ computes is perfectly random and independent of the view of $\mathcal{A}$ unless $\mathcal{A}_1$ has asked one of the values $\mathcal{O}_b^i(v)$ (for $0 \leq i < n$) to the oracle. We therefore have

$$\mathsf{Adv}^{\mathsf{prp}}_{\mathcal{G}}(\mathcal{B} \,|\, b = 0) = 2\Pr[x = y \,|\, b = 0] - 1 \leq 2 \times \left(\frac{mn}{2^\ell} + \frac{1}{2^\ell}\right) - 1.$$

Combining the two cases, with a random bit b, we get the result. □

The following simple proposition will be used several times in the future.

Definition 10. Let $\pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let f be a permutation onto $\mathcal{M}$ modeled by the two oracles f and f^{-1}. We define the new encryption scheme $\pi^{(f)} = (\mathcal{G}^{(f)}, \mathcal{E}^{(f)}, \mathcal{D}^{(f)})$ by

$$\mathcal{M}^{(f)} = \mathcal{M} \qquad \mathcal{R}^{(f)} = \mathcal{R} \qquad \mathcal{C}^{(f)} = \mathcal{C}$$

Algorithm $\mathcal{G}^{(f)}(1^k)$	Algorithm $\mathcal{E}^{(f)}_{\mathsf{pk}^{(f)}}(m, r)$	Algorithm $\mathcal{D}^{(f)}_{\mathsf{sk}^{(f)}}(c)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{G}(1^k)$	$\mathsf{pk}\|\|f\|\|f^{-1} \stackrel{\mathsf{def}}{=} \mathsf{pk}^{(f)}$	$\mathsf{sk} \stackrel{\mathsf{def}}{=} \mathsf{sk}^{(f)}$
$\mathsf{pk}^{(f)} \leftarrow \mathsf{pk}\|\|f\|\|f^{-1}$	return $\mathcal{E}_{\mathsf{pk}}(f(m), r)$	return $f^{-1}(\mathcal{D}_{\mathsf{sk}}(c))$
$\mathsf{sk}^{(f)} \leftarrow \mathsf{sk}$		
return $(\mathsf{pk}^{(f)}, \mathsf{sk}^{(f)})$		

Proposition 11. *For any encryption scheme $\pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ and for any permutation f (so that f and f^{-1} are efficient), π and $\pi^{(f)}$ have a similar indistinguishability level whatever the kind of attack:*

$$\mathsf{Adv}^{\mathsf{ind\text{-}yyy}}_{\pi}(t) \leq \mathsf{Adv}^{\mathsf{ind\text{-}yyy}}_{\pi^{(f)}}(t + 2T_f + q_d T_{f^{-1}}) \leq \mathsf{Adv}^{\mathsf{ind\text{-}yyy}}_{\pi}(t + (2 + q_d)(T_f + T_{f^{-1}})),$$

where T_f (and $T_{f^{-1}}$ resp.) is an upper-bound of the time required to evaluate f (and f^{-1} resp.)

Proof. We first prove that if $\pi^{(f)}$ is secure then π is secure too. We insist here that both f and f^{-1} are efficiently computable and are included in the *public* key. Let us consider an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against π, we build an adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ against $\pi^{(f)}$: whenever $\mathcal{A}$ makes a decryption query c to the oracle $\mathcal{D}_{\mathsf{sk}}$, $\mathcal{B}$ makes the same query to the decryption oracle $\mathcal{D}^{(f)}_{\mathsf{sk}^{(f)}}$. $\mathcal{B}$ receives the answer m and forwards $f(m)$ to $\mathcal{A}$. When $\mathcal{A}_1$ outputs two candidates m_0 and m_1, $\mathcal{B}_1$ computes $f^{-1}(m_0)$ and $f^{-1}(m_1)$. Finally, when $\mathcal{A}$ outputs its guess b', $\mathcal{B}$ forwards this value. It is clear that the advantage of $\mathcal{B}$ is exactly the advantage of $\mathcal{A}$, while its running time needs extra time for two evaluations of f and q_d evaluations of f^{-1}, where q_d is the number of decryption queries:

$$\mathsf{Adv}^{\mathsf{ind\text{-}yyy}}_{\pi}(t) \leq \mathsf{Adv}^{\mathsf{ind\text{-}yyy}}_{\pi^{(f)}}(t + 2T_f + q_d T_{f^{-1}}).$$

Since $\pi = \pi^{(f)(f^{-1})}$, and both f and f^{-1} are public and efficient, one easily concludes. □

3.2 Each Query is Important

In this section, we show that each query, before receiving the challenge or after having received it, has its own role. This means that no query before receiving the challenge can be replaced by queries (whatever the number is) after having received the challenge, and vice-versa.

Theorem 12. *For any pair of integers (m, n), there is an encryption scheme that is (m, N)-IND secure and (M, n)-IND secure, but not $(m+1, n+1)$-IND secure, whatever M and N are.*

Proof. We first assume that there exists an IND-CCA2 secure encryption scheme $\pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$, which is thus (i,j)-IND for any pair (i,j). We also need a trapdoor one-way permutation f onto $\mathcal{M}$. The encryption scheme $\pi^{(f)}$ is therefore IND-CCA2 secure, when the trapdoor for computing f^{-1} is included in the public key. We modify $\pi^{(f)}$ into a new encryption scheme $\pi' = (\mathcal{G}', \mathcal{E}', \mathcal{D}')$ which is not $(m+1, n+1)$-IND secure anymore, but still both (m, N)-IND secure and (M, n)-IND secure. Note that a main difference comes from the fact that the trapdoor for f^{-1} is now in the private key only. The scheme $\pi' = (\mathcal{G}', \mathcal{E}', \mathcal{D}')$ works as follows:

- We denote by I_M a specific element of $\mathcal{M}$ and we note $p_M = f^{-1}(I_M)$.
- We fix two families, a pseudo-random function family $\mathcal{F} = \{F_K : K \in \{0,1\}^k\}$ and a pseudo-random permutation family $\mathcal{G} = \{G_K : K \in \{0,1\}^k\}$, from the set $\mathcal{C}$ into $\mathcal{C}$. We furthermore assume that the cardinality of $\mathcal{C}$ is larger than 2^ℓ.
  For sake of simplicity, we use the same key sets, domain and range sets for $\mathcal{F}$ and $\mathcal{G}$, but this is not necessary.

Then, the intuition behind the construction is that $m+1$ decryption queries in the first stage will help to determine a specific plaintext μ. It has the specificity that, in the second stage, it will be possible to check after $n+1$ decryption queries whether a given ciphertext actually encrypts μ or not.

Algorithm $\mathcal{G}'(1^k)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{G}(1^k)$
$I_M \overset{R}{\leftarrow} \mathcal{M}, K_f, K_g \overset{R}{\leftarrow} \{0,1\}^k$
$\mathsf{pk}' \leftarrow \mathsf{pk}||f||I_M||m||n$
$\mathsf{sk}' \leftarrow \mathsf{sk}||f^{-1}||K_f||K_g$
return $(\mathsf{pk}', \mathsf{sk}')$

Algorithm $\mathcal{E}'_{\mathsf{pk}'}(\mu, r)$
$\mathsf{pk}||f||I_M||m||n \overset{\mathsf{def}}{=} \mathsf{pk}'$
$\varphi \leftarrow f(\mu)$
return $0||\mathcal{E}_{\mathsf{pk}}(\varphi, r)||\epsilon$

Algorithm $\mathcal{D}'_{\mathsf{sk}'}(b||c||z)$
$\mathsf{sk}||f^{-1}||K_f||K_g \overset{\mathsf{def}}{=} \mathsf{sk}'$
1. if $(b = 0 \wedge z = \epsilon)$ return $f^{-1}(\mathcal{D}_{\mathsf{sk}}(c))$
2. if $(b = 1 \wedge z = \epsilon)$ return $F_{K_f}(c)$
3. if $(b = 2 \wedge z = \epsilon)$ return $G_{K_g}(c)$
4. if $(b = 1 \wedge z = F^n_{K_f}(c) \wedge \mathcal{D}(c) = G^m_{K_g}(I_M))$ return $f^{-1}(G^m_{K_g}(I_M))$

otherwise, return $\perp$

In the above scheme, $\mu = f^{-1}(G^m_{K_g}(I_M))$ is the crucial plaintext the adversary should send as a challenge, because with a ciphertext $0\|c\|\epsilon$ of this plaintext μ, and the knowledge of $F^n_{K_f}(c)$, one can derive a second ciphertext of μ (and thus break both the non-malleability and the IND-CCA2 security level), using the fourth case in the decryption oracle.

Lemma 13. *π' is not $(m+1, n+1)$-IND secure.*

Proof. The following $(m+1, n+1)$-IND adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ can successfully attack π':

- In the first stage, $\mathcal{A}_1$ asks $2||I_M||\epsilon$ to $\mathcal{D}'_{\mathsf{sk}'}$ and gets $G_{K_g}(I_M)$. Then for $i = 1$ to $m-1$, $\mathcal{A}_1$ asks $2||G^i_{K_g}(I_M)||\epsilon$ to $\mathcal{D}'_{\mathsf{sk}'}$ and finally gets $G^m_{K_g}(I_M)$, after m decryption queries. It then computes by itself $c = \mathcal{E}_{\mathsf{pk}}(G^m_{K_g}(I_M), r)$ (since pk is part of pk') and asks $0||c||\epsilon$ to $\mathcal{D}'_{\mathsf{sk}'}$ to get $m_0 = f^{-1}(G^m_{K_g}(I_M))$. It randomly chooses a second different candidate $m_1 \neq m_0$, and outputs (m_0, m_1), after exactly $m+1$ decryption queries.
- In the second stage, $\mathcal{A}_2$ receives the challenge ciphertext $y = 0||c^\star||\epsilon$, where $c^\star = \mathcal{E}(m_b, r)$. A_2 asks $1||c^\star||\epsilon$ to $\mathcal{D}'_{\mathsf{sk}'}$ and gets $F_{K_f}(c^\star)$. Then, for $i = 1$ to $n-1$, the adversary asks $1||F^i_{K_f}(c^\star)||\epsilon$ and finally gets $F^n_{K_f}(c^\star)$ after n decryption queries. As a last query, it asks $1||c^\star||F^n_{K_f}(c^\star)$ to $\mathcal{D}'_{\mathsf{sk}'}$. If the answer is m_0, the adversary returns 0, otherwise (in which case the answer is $\perp$), the adversary returns 1.

It is easy to see that the value returned by the adversary is always equal to b. □

Lemma 14. *π' is (m, N)-IND secure: for $t' \leq t + 2T_f$ and $q_d \leq m + N$,*

$$\mathsf{Adv}^{(m,N)\text{-}\mathsf{ind}}_{\pi'}(t) \leq \mathsf{Adv}^{\mathsf{ind\text{-}cca2}}_{\pi}(t' + q_d T_f + 2T_{f^{-1}}) + 2 \times \begin{pmatrix} \mathsf{Succ}^{\mathsf{ow}}_{f}(t') + (m+2) \times \mathsf{Adv}^{\mathsf{prp}}_{\mathcal{G}}(2m-1, t') \\ +\mathsf{Adv}^{\mathsf{prp}}_{\mathcal{G}}(q_d, t') + \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(q_d, t') + (m+2) \times 2^{-\ell} \end{pmatrix}.$$

Proof. Since π is $\mathsf{IND\text{-}CCA2}$ secure, it is also the case for $\pi^{(f)}$. We then prove that an (m, N)-IND adversary $\mathcal{A}$ against π' can be used by an adversary $\mathcal{B}$ to break the $\mathsf{IND\text{-}CCA2}$ security level of $\pi^{(f)}$ with a similar advantage.

Before presenting this adversary, let us claim the following proposition, which proof is straightforward.

Proposition 15. *Providing F_{K_f}, G_{K_g}, f, f^{-1} and the decryption oracle $\mathcal{D}^{(f)}_{\mathsf{sk}^{(f)}}$ of $\pi^{(f)}$, one can perfectly simulate the decryption $\mathcal{D}'_{\mathsf{sk}'}$ of π'.*

Game $\mathbf{G}_0$: In this game, our adversary $\mathcal{B}$ is provided the decryption oracle $\mathcal{D}'_{\mathsf{sk}'}$. It is thus not a $\pi^{(f)}$ adversary yet. Anyway, it can easily simulate the view of the adversary $\mathcal{A}$, granted the oracle access to $\mathcal{D}'_{\mathsf{sk}'}$. When $\mathcal{A}_1$ outputs the candidates (m_0, m_1), $\mathcal{B}_1$ forwards them, as its own output. On the challenge ciphertext c, $\mathcal{B}_2$ runs $\mathcal{A}_2(0||c||\epsilon)$. When $\mathcal{A}_2$ outputs its guess b' for the bit b involved in the challenge, $\mathcal{B}_2$ forwards it as its own guess. We denote by S_0 the event $b' = b$. We clearly have: $\Pr[\mathsf{S}_0] = \Pr[\mathcal{B} = b] = \Pr[\mathcal{A} = b]$.

Game $\mathbf{G}_1$: We modify a little bit $\mathcal{B}_1$, so that it aborts if a bad case occurs. We define $g_m = G^m_{K_g}(I_M)$. When $\mathcal{A}_1$ outputs (m_0, m_1), $\mathcal{B}_1$ computes by itself $f_0 = f(m_0)$ and $f_1 = f(m_1)$. If g_m is one of f_0 or f_1, or appears in a decryption query of the form $1\|c\|z$ (*i.e.*, $g_m = \mathcal{D}(c)$, see case 4), then $\mathcal{B}$ aborts the game, outputting a random guess, otherwise, it continues as in the previous game. We denote by $\mathsf{EventGM}$ the above bad event that $f_0 = g_m$, $f_1 = g_m$ or g_m appears in a decryption query: $|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_0]| \leq \Pr[\mathsf{EventGM}]$.
Let us evaluate the probability of this event. To this aim, we consider two situations, since $\mathcal{A}_1$ is allowed to ask at most m decryption queries:

- $\mathcal{A}_1$ asks m queries of the form $2\|c\|\epsilon$, which are answered by $G_{K_g}(c)$ (see case 3.) Then $\mathcal{B}_1$ does not use any query to f^{-1} to simulate the answers of the decryption queries of $\mathcal{A}_1$. We can thus build an invertor for f: we give every private information to $\mathcal{B}$, except the trapdoor for inverting f. When event $\mathsf{EventGM}$ happens, $\mathcal{B}$

has inverted f on the random element g_m (since I_M is random and G_{K_g} is a public permutation, and thus $G^m_{K_g}$) without making any query to f^{-1}. Indeed, using the private informations, one can compute g_m, and then one can check which one of m_0 or m_1 is the pre-image of g_m by f.
– $\mathcal{A}_1$ asks at most than $m-1$ queries of the form $2\|c\|\epsilon$. Event $\mathsf{EventGM}$ means that g_m is one of f_0 or f_1, or appears in a decryption query of the form $1\|c\|z$. This time, we can build an adversary against the PRP property of the family $\mathcal{G}$: we give every private information to $\mathcal{B}$, except K_g, but an oracle access to G_{K_g}. When event $\mathsf{EventGM}$ happens, g_m is one of f_0 or f_1, or appears in a decryption query of the form $1\|c\|z$ (note that $g_m = \mathcal{D}(c)$, which can be computed since now $\mathcal{B}$ knows sk.) By randomly outputting f_0, f_1 or $\mathcal{D}(c)$ from one of the m decryption queries of the form $1\|c\|z$, after at most $m-1$ queries to G_{K_g}, with probability of $1/(m+2)$, we get $g_m = G^m_{K_g}(I_M)$, for a random input I_M.

Regrouping these two cases, we have:

$$\begin{aligned}\Pr[\mathsf{EventGM}] &\leq \mathsf{Succ}^{\mathsf{ow}}_f(t+2T_f) + (m+2)\times \mathsf{Succ}^{0,m}_{G_{K_g}}(t+2T_f)\\ &\leq \mathsf{Succ}^{\mathsf{ow}}_f(t+2T_f) + (m+2)\times \mathsf{Adv}^{\mathsf{prp}}_{\mathcal{G}}(2m-1,t+2T_f) + \frac{m+2}{2^\ell}.\end{aligned}$$

Game $\mathbf{G}_2$: In this game, we still exclude event $\mathsf{EventGM}$, and thus $\mathcal{B}$ does not need to check g_m, and thus to compute it either. $\mathcal{B}$ is no longer provided with $\mathcal{D}'_{\mathsf{sk}'}$, but $\mathcal{D}^{(f)}_{\mathsf{sk}^{(f)}}$ only. By the Proposition 15, $\mathcal{B}$ can use this decryption oracle $\mathcal{D}^{(f)}_{\mathsf{sk}^{(f)}}$ to perfectly simulate $\mathcal{D}'_{\mathsf{sk}'}$, thanks to the access to F_{K_f}, G_{K_g}, f and f^{-1}. The only situation that $\mathcal{B}$ cannot simulate is when $\mathcal{A}_2$ asks for $1||c||z$ because $\mathcal{B}$ cannot ask the decryption oracle on its challenge c. Fortunately, in such a case, $\mathcal{B}$ can safely answer $\perp$ (since we excluded event $\mathsf{EventGM}$): $\Pr[\mathsf{S}_2] = \Pr[\mathsf{S}_1]$.

Game $\mathbf{G}_3$: In this game, we replace the permutation G_{K_g} by a truly random permutation. Whenever $\mathcal{B}$ needs to use G_{K_g} (for simulating decryptions), it uses G: $|\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_2]| \leq \mathsf{Adv}^{\mathsf{prp}}_{\mathcal{G}}(q_d, t+2\times T_f)$.

Game $\mathbf{G}_4$: We now replace the function F_{K_f} by a truly random function F. Whenever $\mathcal{B}$ needs to use F_{K_f}, it uses F:

$$|\Pr[\mathsf{S}_4] - \Pr[\mathsf{S}_3]| \leq \mathsf{Adv}^{\mathsf{prp}}_{\mathcal{G}}(q_d, t+2\times T_f).$$

In this last game, with an access to f and f^{-1}, $\mathcal{B}$ is an actual $\mathsf{IND\text{-}CCA2}$ adversary against $\pi^{(f)}$. Since $\mathcal{A}$ is an (m,N)-IND adversary against π', $q_d \leq m+N$, hence the result. □

Lemma 16. *π' is (M,n)-IND secure:*

$$\begin{aligned}\mathsf{Adv}^{(M,n)\text{-}\mathsf{ind}}_{\pi'}(t) \leq\ & \mathsf{Adv}^{\mathsf{ind\text{-}cca2}}_{\pi'}(t+(M+n)T_f+2T_{f^{-1}})\\ &+n\times\begin{pmatrix}2\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(M+2n-1,t)+(Mn^2+n+M)\times 2^{-\ell}\\ +\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(M+n,t)+\mathsf{Adv}^{\mathsf{prp}}_{\mathcal{G}}(M+n,t)\end{pmatrix}.\end{aligned}$$

Proof. As above, we start from an (M,n)-ind adversary $\mathcal{A}$ against π'. We prove that $\mathsf{Adv}^{(M,n)\text{-}\mathsf{ind}}_{\pi'}(\mathcal{A})$ is small by exhibiting an $\mathsf{IND\text{-}CCA2}$ adversary $\mathcal{B}$ against $\pi^{(f)}$ with a similar advantage.

Game $\mathbf{G}_0$: In this first game, as above, $\mathcal{B}$ is provided with $\mathcal{D}'_{\mathsf{sk}'}$, and plays exactly the same way:

$$\Pr[\mathsf{S}_0] = \Pr[\mathcal{B} = b] = \Pr[\mathcal{A} = b].$$

Game $\mathbf{G}_1$: We provide $\mathcal{B}$ with $\mathcal{D}^{(f)}_{\mathsf{sk}^{(f)}}$ instead of $\mathcal{D}'_{\mathsf{sk}'}$, together with oracle access to $F_{\mathcal{K}_f}$, G_{K_g}, but also the trapdoor to compute f^{-1}. By the Proposition 15, $\mathcal{B}$ can use this decryption oracle to perfectly simulate $\mathcal{D}'_{\mathsf{sk}'}$, excepted on a query $1\|c\|z$, where c is the challenge ciphertext for $\mathcal{B}$. Fortunately, in this case, $\mathcal{B}$ can safely output $\perp$. Indeed, it would be a mistake only if $z = F^n_{K_f}(c)$. Note that c is not known to $\mathcal{A}_1$, and thus such a case can appear in the first stage only by chance (less than $M/2^\ell$.) If this happens in the second stage, by randomly outputting a z from a $1\|c\|z$ decryption query, one would break the PRF property of $\mathcal{F}$ with probability of $1/n$, since one would output $F^n_{K_f}(c)$ after only $n-1$ queries (since this critical decryption query is one of the n possible queries of $\mathcal{A}_2$.)

$$|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_0]| \leq \frac{M}{2^\ell} + n \cdot \mathsf{Succ}^{M,n}_{F_{K_f}}(t) \leq n \times \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(M + 2n - 1, t) + \frac{Mn^2 + n + M}{2^\ell}.$$

Game $\mathbf{G}_2$: In this game, we replace the function F_{K_f} by a truly random function F. Similarly, we replace the permutation G_{K_g} by a truly random permutation G. With the same argument as in the proof of the Lemma 14, in the games $\mathbf{G}_3$ and $\mathbf{G}_4$, we have:

$$|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_1]| \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(q_d, t) + \mathsf{Adv}^{\mathsf{prp}}_{\mathcal{G}}(q_d, t).$$

In this last game, $\mathcal{B}$ is an actual IND-CCA2 adversary against $\pi^{(f)}$, hence the result. □

From the Lemmas 13, 14 and 16, one completes the proof of the Theorem 12. □

3.3 Discussion about Non-Malleability

We now briefly discuss on the general notion of non-malleability (denoted by $\mathsf{CNM}^{(k)}$) in which the adversary finally outputs a ciphertext vector of size k, instead of a single ciphertext. In [4], Bellare and Sahai introduced the notion of parallel attacks, denoted PA (or more precisely $\mathsf{PA}^{(k)}$ by us), where the adversary can ask a ciphertext vector of size k to the decryption oracle just after the last normal single decryption query (derived in three ways, as usual, with PA0, PA1 and PA2, according to the access of the decryption oracle for single ciphertext queries.) They proved that IND-PAX is equivalent to $\mathsf{CNM}^{(k)}$-CCAX, where CCA0 is indeed CPA. Their result can be translated within our formalism under the following theorem, which proof can be found in the Appendix.

Theorem 17. *The two notions (m,n)-IND-$\mathsf{PA}^{(k)}$ and (m,n)-$\mathsf{CNM}^{(k)}$ are equivalent. In other words, for any encryption scheme $\pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$:*

$$\frac{1}{2} \times \mathsf{Adv}^{(m,n)\text{-ind-pa}^{(k)}}_{\pi}(t) \leq \mathsf{Adv}^{(m,n)\text{-ind-cnm}^{(k)}}_{\pi}(t) \leq \mathsf{Adv}^{(m,n)\text{-ind-pa}^{(k)}}_{\pi}(t + T_R),$$

where T_R is an upper-bound on the time to evaluate the relation R.

Granted the following identifications,

$$(m,n)\text{-IND-PA}^{(1)} = (m, n+1)\text{-IND} \qquad (m,n)\text{-CNM}^{(1)} = (m,n)\text{-NM},$$

one gets $(m, n+1)$-IND $= (m,n)$-NM.

4 A New Attack Model: CCAO2

Definition 18 (Post-Challenge Attacks). An adversary is called a *post-challenge chosen-ciphertext adversary* (denoted by CCAO2-adversary) if it can access the oracle after the challenge ciphertext is known only still with the restriction not to use it on the challenge itself.

Given this new attack model of post-challenge chosen-ciphertext adversaries, combined with the classical goals, one gets the two security notions: IND-CCAO2 and NM-CCAO2. These notions are independent with the previous ones, excepted the trivial implications. First, it is clear that for any XXX, XXX-CCA2 implies both XXX-CCA1 and XXX-CCAO2. But from the above result, we show that the opposite is not true. In fact, we clearly have the following corollaries:

Corollary 19. IND-CCAO2 *and* IND-CCA1 *are independent notions. In other words, there is a scheme which is* IND-CCA1 *secure but not* IND-CCAO2 *secure and, there is a scheme that is* IND-CCAO2 *secure but not* IND-CCA1 *secure.*

Corollary 20. IND-CCA1 *and* IND-CCAO2 *do not imply, even together,* IND-CCA2. *In other words, there is a scheme which is both* IND-CCA1 *secure and* IND-CCAO2 *secure but not* IND-CCA2 *secure.*

Another discussion. Since parallel attacks [4] do not give more power to a CCAO2 adversary, we still have equivalence between the two notions of IND and NM under this new kind of attack, as shown in [2] under CCA2.

Acknowledgement. The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document reflects only the authors' views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

References

1. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation. In *Proc. of the 38th FOCS*. IEEE, New York, 1997.
2. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-key Encryption Schemes. In *Adv. in Cryptology – Proceedings of Crypto '98*, volume LNCS 1462, pages 26–45, Berlin, 1998. Springer-Verlag.
3. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures – How to Sign with RSA and Rabin. In *Eurocrypt '96*, LNCS 1070, pages 399–416. Springer-Verlag, Berlin, 1996.
4. M. Bellare and A. Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In *Adv. in Cryptology – Proceedings of Crypto '99*, volume LNCS 1666, pages 519–536, Berlin, 1999. Springer-Verlag.
5. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *Proc. of the 23rd STOC*. ACM Press, New York, 1991.
6. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
7. O. Goldreich, Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):210–217, 1986.
8. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

9. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. In *Proc. of the 17th STOC*, pages 291–304. ACM Press, New York, 1985.
10. S. Goldwasser, S. Micali, and R. Rivest. A "Paradoxical" Solution to the Signature Problem. In *Proc. of the 25th FOCS*, pages 441–448. IEEE, New York, 1984.
11. S. Goldwasser, S. Micali, and R. Rivest. A Digital Signature Scheme Secure Against Adaptative Chosen-Message Attacks. *SIAM Journal of Computing*, 17(2):281–308, April 1988.
12. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, New York, 1990.
13. K. Ohta and T. Okamoto. On Concrete Security Treatment of Signatures Derived from Identification. In *Crypto '98*, LNCS 1462, pages 354–369. Springer-Verlag, Berlin, 1998.
14. C. Rackoff and D. R. Simon. Non-interactive Zero-knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Proc. of CRYPTO '91*, volume LNCS 576, pages 433–444, Berlin, 1992. Springer-Verlag.
15. A. Sahai. Non-Malleable Non-Interactive Zero-Knowledge and Chosen-Ciphertext Security. In *Proc. of the 40th FOCS*. IEEE, New York, 1999.

A Proof of Theorem 17:

First, let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}'_2)$ be an (m,n)-$\mathsf{IND\text{-}PA}^{(k)}$ adversary against π. Such an adversary is a classical IND adversary in two stages. But the second stage is split into $\mathcal{A}_2$ which may be allowed to ask decryption queries and output a ciphertext vector. $\mathcal{A}'_2$ receives the plaintext vector and is not allowed any more to query the decryption oracle before outputting its guess. We define the following (m,n)-$\mathsf{CNM}^{(k)}$ adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$:

- $\mathcal{B}_1$ runs $\mathcal{A}_1$, and forwards all the query-answers to/from the decryption oracle (the same number of queries are thus asked.) When $\mathcal{A}_1$ outputs two plaintexts (m_0, m_1) and s, one defines and outputs the distribution $\mathcal{M} = \{m_0, m_1\}$ (the uniform distribution among the two messages) together with the state information s;
- The challenger randomly chooses m according to $\mathcal{M}$, which is equivalent to choose $b \stackrel{R}{\leftarrow} \{0,1\}$ and $m = m_b$, then computes $c = \mathcal{E}_{\mathsf{pk}}(m, r)$ for random coins r;
- $\mathcal{B}_2$ is given the state information s and the ciphertext c, which it forwards to $\mathcal{A}_2$. It also forwards all the query-answers to/from the decryption oracle (the same number of queries are thus asked.) When $\mathcal{A}_2$ outputs its ciphertext vector $\mathbf{y}$ and a state information s', $\mathcal{B}_2$ outputs $(R, \mathbf{y})$ where the relation R is defined as follows: $R(\mathbf{x}, m)$ returns $(m = m_0) \oplus A'_2(\mathbf{x}, s')$.

By definition, if we consider the distribution $\mathcal{M}$ for $\tilde{m}$, that is equivalent to the random choice of a bit d independent to b and b', and $\tilde{m} = m_d$, then $\mathsf{Adv}_{\pi}^{(m,n)\text{-}\mathsf{ind\text{-}cnm}^{(k)}}(\mathcal{A})$ is equal to

$$
\begin{aligned}
&\Pr[R(\mathbf{x}, m)] - \Pr[R(\mathbf{x}, \tilde{m})] = \Pr[R(\mathbf{x}, m_b)] - \Pr[R(\mathbf{x}, m_d)] \\
&= \frac{1}{2} \times \left(\Pr[R(\mathbf{x}, m_b)] - \Pr[R(\mathbf{x}, m_{\bar{b}})] \right) \\
&= \frac{1}{2} \times \left(\Pr[(m_0 = m_b) \oplus A'_2(\mathbf{x}, s')] - \Pr[(m_0 = m_{\bar{b}}) \oplus A'_2(\mathbf{x}, s')] \right) \\
&= \frac{1}{2} \times \left(\Pr[(b = 0) \oplus (b' = 1)] - \Pr[(b = 0) \oplus (b' = 1)] \right) \\
&= \frac{1}{2} \times \left(\Pr[b = b'] - \Pr[b \neq b'] \right) = \frac{1}{2} \times \mathsf{Adv}_{\pi}^{(m,n)\text{-}\mathsf{ind\text{-}pa}^{(k)}}(\mathcal{B}).
\end{aligned}
$$

Note that the running time of $\mathcal{B}$ is exactly the same as of $\mathcal{A}$.

Let us turn to the second part of the relation. Let $\mathcal{A}$ be an (m, n)-$\mathsf{CNM}^{(k)}$ adversary. We define the following (m, n)-$\mathsf{IND\text{-}PA}^{(k)}$ adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}'_2)$:

- $\mathcal{B}_1$ runs $\mathcal{A}_1$, and forwards all the query-answers to/from the decryption oracle (the same number of queries are thus asked.) When $\mathcal{A}_1$ outputs a distribution $\mathcal{M}$ and s, one draws independently two plaintexts (m_0, m_1) according to $\mathcal{M}$, and outputs them together with the state information s;
- The challenger randomly chooses $b \stackrel{R}{\leftarrow} \{0, 1\}$, and random coins r, and computes $c = \mathcal{E}_{\mathsf{pk}}(m_b, r)$;
- $\mathcal{B}_2$ is given the state information s and the ciphertext c, which it forwards to $\mathcal{A}_2$. It also forwards all the query-answers to/from the decryption oracle (the same number of queries are thus asked.) When $\mathcal{A}_2$ outputs $(R, \mathbf{y})$. $\mathcal{B}_2$ outputs the ciphertext vector $\mathbf{y}$. $\mathcal{B}'_2$ is then given the corresponding plaintext vector $\mathbf{x}$. It checks whether $R(\mathbf{x}, m_0)$ holds. If it is true, one outputs $b' = 0$, otherwise $b' = 1$.

$$\begin{aligned}
\mathsf{Adv}_{\pi}^{(m,n)\text{-}\mathsf{ind\text{-}pa}^{(k)}}(\mathcal{B}) &= \Pr[b' = 0 \,|\, b = 0] - \Pr[b' = 0 \,|\, b = b_1] \\
&= \Pr[R(\mathbf{x}, m_0) \,|\, b = 0] - \Pr[R(\mathbf{x}, m_0) \,|\, b = 1] \\
&= \Pr[R(\mathbf{x}, m_0) \,|\, b = 0] - \Pr[R(\mathbf{x}, m_1) \,|\, b = 0] \\
&= \mathsf{Adv}_{\pi}^{(m,n)\text{-}\mathsf{ind\text{-}cnm}^{(k)}}(\mathcal{A}).
\end{aligned}$$

Note that the running time of $\mathcal{B}$ is exactly the same as of $\mathcal{A}$ plus one evaluation of R. □