Trading One-Wayness against Chosen-Ciphertext Security in Factoring-Based Encryption

Pascal Paillier[1] and Jorge L. Villar[2]

[1] Cryptography Group, Security Labs, Gemalto
pascal.paillier@gemalto.com
[2] Departament de Matemàtica Aplicada, Universitat Politècnica de Catalunya
jvillar@ma4.upc.edu

Abstract. We revisit a long-lived folklore impossibility result for factoring-based encryption and properly establish that reaching maximally secure one-wayness (*i.e.* equivalent to factoring) and resisting chosen-ciphertext attacks (CCA) are incompatible goals for single-key cryptosystems. We pinpoint two tradeoffs between security notions in the standard model that have always remained unnoticed in the Random Oracle (RO) model. These imply that simple RO-model schemes such as Rabin/RW-SAEP[+]/OAEP[+][+], EPOC-2, etc. admit *no* instantiation in the standard model which CCA security is equivalent to factoring via a key-preserving reduction. We extend this impossibility to *arbitrary* reductions assuming non-malleable key generation, a property capturing the intuition that factoring a modulus n should not be any easier when given a factoring oracle for moduli $n' \neq n$. The only known countermeasures against our impossibility results, besides malleable key generation, are the inclusion of an additional random string in the public key, or encryption twinning as in Naor-Yung or Dolev-Dwork-Naor constructions.

1 Introduction

The Paradox. When a proof is given that some cryptosystem is semantically secure under chosen ciphertext attack (IND-CCA) under some complexity assumption, one generally checks whether one-wayness can be guaranteed under a weaker assumption. In the case of simple cryptosystems based on factoring large integers however, an inevitable tradeoff seems to exist between one-wayness and chosen ciphertext security. This incompatibility, which was observed for factoring-based signature schemes as well [20,22,13], is folklore knowledge and dates back to the late eighties. Despite early reasonings and attempts (later shown to be wrong) by a number of authors to formally prove it, this so-called "paradox" [13, Section 4] has remained essentially unexplored in a formal manner and, surprisingly enough, overlooked by contributors.

It is well known that the one-wayness of Rabin encryption and variants thereof [22,4,8,5] is equivalent to factoring (FACT), meaning that any efficient algorithm inverting encryption provides an efficient way to factor the modulus. It turns out that the same algorithm can be used to totally break the cryptosystem (*i.e.* factor the modulus) under a trivial chosen ciphertext attack. This kind of attack has never been reported for RSA. But the one-wayness of RSA has not been shown to be equivalent to FACT. In fact, there is a separation result by Boneh and Venkatesan [6] which roughly tells that if a reduction from FACT to low-exponent RSA existed, then an efficient factoring algorithm could be constructed. Simultaneously, RSA-based cryptosystems such as OAEP [3] seem to resist chosen-ciphertext attacks convincingly well in practice. This provides the intuition that some sort of incompatibility must exist between achieving one-wayness under the weakest possible assumption (factoring) and achieving chosen ciphertext security at all.

In an early attempt to capture this intuition, Williams [22] makes the following (over)statement[3]: *if the one-wayness of a factoring-based cryptosystem $\mathcal{E}$ is equivalent to factoring then $\mathcal{E}$ can be totally broken under chosen-ciphertext attack.* A simple proof for this claim was later shown to be incorrect by Goldwasser, Micali and Rivest [13], and the first public-key encryption scheme fully IND-CCA-secure under the factoring assumption was then discovered by Dolev, Dwork and Naor a few years later [10]. However, the incompatibility seems to persist for factoring-based encryption for which the public key consists of a single modulus.

Our Contributions. Our goal in this paper is to revisit [20,22,13] completely and clarify the conditions for such security incompatibilities to exist. We find that when properly formulated, certain security reductions for one-wayness and chosen-ciphertext security are indeed incompatible when considering *single-key* factoring-based encryption *i.e.* where the public key is just made of one hard-to-factor modulus. We reformulate the paradox observed by Williams in terms of *key-preserving* black-box reductions *i.e.* reductions which always call the adversarial oracle with the public-key they were given as input. We strengthen the original observation to show that if one can provide a key-preserving reduction from factoring to the (chosen-plaintext) semantic security of $\mathcal{E}$, then $\mathcal{E}$ cannot fulfil *plaintext-checking* security. Plaintext-checking attacks, introduced in [18], assume that the attacker is given oracle access to a distinguishing oracle that

[3] The paradox appearing in [20,22,13] is discussed in the context of factoring-based signatures. This is a straightforward reformulation for factoring-based encryption.

tells whether a given ciphertext encrypts a given plaintext. It follows from combining these results that a wide class of factoring-based cryptosystems admit *no* key-preserving black-box reduction from factoring to breaking the security notions IND-CCA, OW-CCA and IND-PCA in the standard model. This provides black-box separations with well-known security proofs standing in the RO model [2] such as the one of Rabin-SAEP [5]. We provide later an explanation as to why these incompatibilities are avoided in the case of Naor-Yung [17] and Dolev-Dwork-Naor [10] constructions where public keys are composed of two or more independent moduli, as well as in the RO model.

Finally, we define the notion of non-malleable key generators, which formally captures the property that the factorizations of two public moduli n, n' where $n \neq n'$ are somehow "computationally independent" from one another. Similar notions of non-malleability for discrete logarithms recently appeared in [14,16]. Using non-malleability, we extend the scope of the previous impossibility results to *arbitrary* black-box reductions. Our refined results state that simple and innocuous-looking RO-secure factoring-based encryption schemes (*e.g.* Rabin-SAEP), when combined with non-malleable key generation, black-box separate the RO model from the standard model in a very strong sense: IND-CCA security is equivalent to FACT in the RO model while *no instantiation* of these schemes preserves such equivalence in the standard model.

We note that all impossibility results stated in this paper are easily transposed (mutatis mutandis) to factoring-based signature schemes. We do not treat the case of signatures here due to lack of space.

Roadmap. The paper is structured as follows. Section 2 gives preliminary facts about black-box reductions, single-key factoring-based encryption schemes and related security notions. Section 3 formally establishes the tradeoff between one-wayness and chosen ciphertext security. We also put forward a second tradeoff between semantic security against passive adversaries and plaintext-checking security. In Section 4, we give a formal definition of non-malleable instance generators and provide extended impossibility results. Section 5 discusses possible countermeasures such as encryption twinning to overcome these tradeoffs. We finally conclude on directions for further research in Section 6.

2 Preliminaries

Instance Generators. We define FACT as the problem of computing the list of all prime factors $\mathsf{factors}(n) = (p_1, \ldots, p_t)$ of a randomly chosen positive integer n. In cryptographic applications, one generally focuses on a specifically chosen distribution of hard instances by defining an instance generator Gen. Given a security parameter k, $\mathsf{Gen}(1^k)$ generates a hard-to-factor modulus n, as well as the side information $\mathsf{factors}(n)$. A probabilistic algorithm $\mathcal{A}$ is said to (ε, τ)-break $\mathsf{FACT}\,[\mathsf{Gen}]$ when

$$\Pr\left[(n, \mathsf{factors}(n)) \leftarrow \mathsf{Gen}(1^k) \ : \ \mathcal{A}(n) = \mathsf{factors}(n)\right] \geq \varepsilon \ ,$$

where the probability is taken over the random coins of $\mathcal{A}$ and Gen and $\mathcal{A}$ halts after τ steps. $\mathsf{FACT}\,[\mathsf{Gen}]$ is commonly referred to as the "factoring problem" when Gen is specified implicitly. For readability reasons, we may equivalently write $(n, \mathsf{factors}(n)) \leftarrow \mathsf{Gen}(1^k)$ or $n \leftarrow \mathsf{Gen}(1^k)$ to state that n is drawn according to the distribution induced by $\mathsf{Gen}(1^k)$. We note $\mathcal{PK}_k$ the range of n *i.e.* the set of integers n such that $\Pr\left[n \leftarrow \mathsf{Gen}(1^k)\right] > 0$ and $\mathcal{SK}_k = \mathsf{factors}(\mathcal{PK}_k)$. Finally $\mathcal{PK} = \cup_k \mathcal{PK}_k$ and $\mathcal{SK} = \cup_k \mathcal{SK}_k$. Here are some instance generators commonly used in factoring-based encryption:

Rabin-Williams. Given 1^k, select uniformly at random two $\lceil k/2 \rceil$-bit primes p and q such that $p = 3 \bmod 8$ and $q = 7 \bmod 8$. Set $n = pq$ and output $(n, (p, q))$.

OU. Given 1^k, randomly select two $\lceil k/3 \rceil$-bit primes p and q. Set $n = p^2 q$ and output $(n, (p, q))$.

RSA-e. Given a small integer e and 1^k, randomly select two $\lceil k/2 \rceil$-bit primes p and q such that $\gcd(p-1, e) = \gcd(q-1, e) = 1$. Set $n = pq$ and output $(n, (p, q))$.

Sophie-Germain. Given 1^k, randomly select two $(\lceil k/2 \rceil - 1)$-bit primes p' and q' such that $p = 2p' + 1$ and $q = 2q' + 1$ are also primes. Set $n = pq$ and output $(n, (p, q))$.

Single-Key Factoring-Based Encryption. A single-key factoring-based encryption scheme $\mathcal{E}$ with security parameter k can be described as the combination of an instance generator Gen with a family of trapdoor functions on Gen, namely a pair $(\mathsf{Enc}, \mathsf{Dec})$ such that for any $n \in \mathcal{PK}$, $\mathsf{Enc}(n, \cdot, \cdot)$ and $\mathsf{Dec}(\mathsf{factors}(n), \cdot)$ are integer-valued functions

$$\mathsf{Enc}(\,n, \cdot, \cdot\,) \ : \ \mathsf{M}_n \times \mathsf{R}_n \to \mathsf{C}_n \ , \quad \mathsf{Dec}\big(\,\mathsf{factors}(n), \cdot\,\big) \ : \ \mathsf{C}_n \to \mathsf{M}_n$$

where M_n, R_n and C_n denote respectively the plaintext, random and ciphertext spaces[4]. We impose that for any $n \in \mathcal{PK}$, $m \in \mathsf{M}_n$ and $r \in \mathsf{R}_n$, $\mathsf{Dec}(\mathsf{factors}(n), \mathsf{Enc}(n, m, r)) = m$. When $\mathsf{Enc}(n, \mathsf{M}_n, \mathsf{R}_n) \subsetneq \mathsf{C}_n$,

[4] R_n is the empty set when encryption is deterministic.

some elements of C_n are not proper ciphertexts. When $c \notin \mathsf{Enc}(n, \mathsf{M}_n, \mathsf{R}_n)$, $\mathsf{Dec}(\mathsf{factors}(n), c)$ returns a failure symbol $\perp \in \mathsf{M}_n$. We impose that $\mathsf{Enc}(n, \cdot, \cdot)$ and $\mathsf{Dec}(n, \cdot, \cdot)$ be efficiently computable for any arguments *i.e.* can be evaluated in time at most $\mathsf{poly}(k)$ for $n \in \mathcal{PK}_k$. We identify $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ to the three following probabilistic procedures:

$\mathcal{E}.\mathsf{keygen}$: Run $\mathsf{Gen}(1^k)$ to get $(n, \mathsf{factors}(n))$. The secret key is $\mathsf{factors}(n)$ while the public key is n.

$\mathcal{E}.\mathsf{encrypt}$: Given a public key n and a message $m \in \mathsf{M}_n$, select $r \leftarrow \mathsf{R}_n$ uniformly at random and compute $c = \mathsf{Enc}(n, m, r)$. The output ciphertext is $c \in \mathsf{C}_n$.

$\mathcal{E}.\mathsf{decrypt}$: Given the secret key $\mathsf{factors}(n)$ and a ciphertext $c \in \mathsf{C}_n$, output $m = \mathsf{Dec}(\mathsf{factors}(n), c)$.

Examples of single-key factoring-based cryptosystems as defined above are countless: RSA[5] and its numerous variants OAEP [3], REACT-RSA [18], PKCS#1 v1.5 [21], Rabin and related systems (Rabin-Williams [22], Blum-Goldwasser [4], Chor-Goldreich [8], Rabin-SAEP [5]), Naccache-Stern, Okamoto-Uchiyama and the EPOC family [12,11], Paillier [19] and variants. Many elliptic-curve-based cryptosystems such as KMOV [15], Vanstone-Zuccherato or Demytko [9] also fall into this category. We refer the reader to the extensive literature on factoring and its applications to cryptography for more detail.

Black-Box Reductions. Black-box reductions constitute a natural tool to relate computational problems and capture the way most security proofs are constructed. Given two computational problems P_1 and P_2, a black-box reduction from P_1 to P_2 is a probabilistic algorithm $\mathcal{R}$ which solves P_1 with the help of an oracle solving instances of P_2. $\mathcal{R}$ interacts with the oracle strictly as defined by the specification of P_2 and in particular has no view on the internal tapes of the oracle. The (extra) time of $\mathcal{R}$ is the number of elementary steps required by $\mathcal{R}$ to complete given that oracle calls count for one step by convention. A black-box reduction is polynomial when it runs in polynomial extra time (in a security parameter). It is crucial to remind that $\mathcal{R}$ can be polynomial even when no polynomial-time algorithm solving P_2 is known to exist. We denote by $P_1 \Leftarrow P_2$ the fact that P_1 is polynomially black-box reducible to P_2. We write $P_1 \Leftarrow_{\mathcal{R}} P_2$ when $\mathcal{R}$ is known to reduce P_1 to P_2. Polynomial equivalence is denoted by $P_1 \equiv P_2$. $\mathsf{Succ}(P, \tau)$ stands for the maximal

[5] If the public exponent e is fixed (as usually done in practice), RSA decryption can be performed given the factors of n only.

success probability of probabilistic algorithms solving P in no more than τ elementary steps. Similarly, $\mathsf{Succ}\,(P_1 \Leftarrow P_2, \tau, \varepsilon, \ell)$ stands for the maximal success probability of probabilistic algorithms solving P_1 in no more than τ elementary steps and at most ℓ calls to an oracle solving P_2 with probability ε. All the reductions considered in this paper are black-box.

Security Notions for Factoring-Based Encryption. Security notions for encryption schemes are obtained by combining an adversarial goal with an attack model. **(Goals)** We say that an encryption scheme is *unbreakable* (UBK) when one cannot extract the secret key matching a prescribed public key. The scheme is said to be *one-way* (OW) when no adversary can recover a plaintext given its encryption. *Indistinguishability* (IND, a.k.a. *semantic security*) relates to the hardness of deciding whether a given ciphertext encrypts a given plaintext. **(Attacks)** We consider three attack models in this paper. In a *chosen-plaintext attack* (CPA), the adversary is given nothing more than the public key as input. In a *plaintext-checking attack* (PCA), the adversary is given access to a plaintext-checking oracle that tells whether a given ciphertext encrypts a given plaintext [18]. In a *chosen-ciphertext attack* (CCA), the adversary has access to a decryption oracle. Oracle access in $\mathsf{OW\text{-}CCA}$, $\mathsf{IND\text{-}PCA}$ and $\mathsf{IND\text{-}CCA}$ games is limited in the sense that the adversary is not allowed to call the oracle on the challenge ciphertext itself. These definitions are classical. We refer to [1,18] for more detail on security notions for encryption schemes.

For convenience, we denote security notions in a positive fashion *e.g.* $\mathsf{OW\text{-}PCA}\,[\mathcal{E}]$ denotes the problem of breaking the one-wayness of $\mathcal{E}$ under plaintext-checking attack. This convention allows one to easily describe hierarchies between security notions using reductions. When the focus is on an adaptive attack (*i.e.* either PCA or CCA), we denote by ℓ-$\mathsf{GOAL\text{-}ATK}[\mathcal{E}]$ the problem of breaking GOAL in no more than ℓ calls to the resource defined by ATK. Thus, breaking ℓ-$\mathsf{IND\text{-}CCA}\,[\mathcal{E}]$ authorizes at most ℓ calls to the decryption oracle to break IND. We recall that $\mathsf{GOAL\text{-}CCA}\,[\mathcal{E}] \Leftarrow \mathsf{GOAL\text{-}PCA}\,[\mathcal{E}] \Leftarrow \mathsf{GOAL\text{-}CPA}\,[\mathcal{E}]$ for any factoring-based encryption scheme $\mathcal{E}$ and adversarial goal $\mathsf{GOAL} \in \{\mathsf{UBK}, \mathsf{OW}, \mathsf{IND}\}$. We also have $\mathsf{UBK\text{-}CPA}\,[\mathcal{E}] \equiv \mathsf{FACT}\,[\mathcal{E}.\mathsf{keygen}]$. We plot on Fig. 1 the map of security levels needed for the sake of this work.

3 Impossibility Results for Key-Preserving Reductions

In this section we focus on the standard-model security of single-key factoring-based encryption schemes. All black-box reductions known for

$$
\begin{array}{ccccccc}
\mathsf{UBK\text{-}CCA}\,[\mathcal{E}] & \Leftarrow & \mathsf{UBK\text{-}PCA}\,[\mathcal{E}] & \Leftarrow & \mathsf{UBK\text{-}CPA}\,[\mathcal{E}] & \equiv & \mathsf{FACT}\,[\mathcal{E}.\mathsf{keygen}] \\
\Downarrow & & \Downarrow & & \Downarrow & & \\
\mathsf{OW\text{-}CCA}\,[\mathcal{E}] & \Leftarrow & \mathsf{OW\text{-}PCA}\,[\mathcal{E}] & \Leftarrow & \mathsf{OW\text{-}CPA}\,[\mathcal{E}] & & \\
\Downarrow & & \Downarrow & & \Downarrow & & \\
\mathsf{IND\text{-}CCA}\,[\mathcal{E}] & \Leftarrow & \mathsf{IND\text{-}PCA}\,[\mathcal{E}] & \Leftarrow & \mathsf{IND\text{-}CPA}\,[\mathcal{E}] & &
\end{array}
$$

Fig. 1. Relations among security notions for single-key factoring-based encryption.

such schemes are *key-preserving*, meaning informally that they make oracle calls to the adversary with the same key that they are given as input. We properly formalize this particular class of reductions in our setting[6].

3.1 Key-Preserving Black-Box Reductions

Definition. We define key preservation for arbitrary security games related to a single-key factoring-based encryption scheme $\mathcal{E}$. Assume that $P_1[\mathcal{E}]$ and $P_2[\mathcal{E}]$ are two computational problems (view P_1 and P_2 as security notions) associated to $\mathcal{E}$. Consider a black-box reduction algorithm $\mathcal{R}$ such that $P_1[\mathcal{E}] \Leftarrow_{\mathcal{R}} P_2[\mathcal{E}]$, meaning that $\mathcal{R}$ makes oracle calls to an algorithm $\mathcal{A}$ breaking $P_2[\mathcal{E}]$ to break $P_1[\mathcal{E}]$. Let $\mathsf{Keys}(n, \mathsf{aux}, \varpi)$ be the list $(n_1, \ldots, n_\ell)$ of public keys given by $\mathcal{R}$ as input to $\mathcal{A}$ where (n, aux) is the modulus and auxiliary input for which $\mathcal{R}$ has to break $P_1[\mathcal{E}]$ and $\varpi \in \{0,1\}^{\mathsf{poly}(k)}$ denotes the random tape of $\mathcal{R}$. Here the auxiliary input aux depends on the specification of P_1. Note that the number ℓ of oracle calls is a deterministic function of n, aux and ϖ. $\mathcal{R}$ is said to be key-preserving when for any aux, ϖ and $n \in \mathcal{PK}_k$, either $\ell = 0$ or $n_i = n$ for $i \in [1, \ell]$.

Key-preservation is transitive. It is obvious that if $P_1[\mathcal{E}] \Leftarrow_{\mathcal{R}_1} P_2[\mathcal{E}]$ and $P_2[\mathcal{E}] \Leftarrow_{\mathcal{R}_2} P_3[\mathcal{E}]$ such that $\mathcal{R}_1$ and $\mathcal{R}_2$ are both key-preserving, then there is a key-preserving reduction $\mathcal{R}_3$ such that $P_1[\mathcal{E}] \Leftarrow_{\mathcal{R}_3} P_3[\mathcal{E}]$.

Reductions among security notions are key-preserving. We use later the property that all the straightforward black-box reductions between the classical security notions for $\mathcal{E}$ such as $\mathsf{IND\text{-}CCA}[\mathcal{E}] \Leftarrow \mathsf{IND\text{-}PCA}[\mathcal{E}]$ and $\mathsf{IND\text{-}CPA}[\mathcal{E}] \Leftarrow \mathsf{OW\text{-}CPA}[\mathcal{E}]$ and so forth [1], are key-preserving.

[6] A similar class of reductions for RSA encryption called *simple reductions* was recently considered by Brown [7].

3.2 One-Wayness versus Chosen-Ciphertext Security

The following reformulates the observation made by Williams [22].

Theorem 1. *Let $\mathcal{E}$ be a single-key factoring-based encryption scheme. If there exists a polynomial key-preserving black-box reduction $\mathcal{R}$ such that* $\mathsf{FACT}[\mathcal{E}.\mathsf{keygen}] \Leftarrow_{\mathcal{R}} \mathsf{OW\text{-}CPA}[\mathcal{E}]$*, then* $\mathsf{UBK\text{-}CCA}[\mathcal{E}]$ *is polynomial.*

Proof. The main idea of the proof is basically a one-line statement and follows the reasoning of [22,13]. Let $\mathcal{R}$ be such a key-preserving reduction algorithm, *i.e.* an algorithm that factors a modulus n randomly selected by $\mathcal{E}.\mathsf{keygen}$ with non-negligible probability $\varepsilon_{\mathcal{R}}$ and extra time τ given black-box access to an adversary $\mathcal{A}$ breaking $\mathsf{OW\text{-}CPA}[\mathcal{E}]$ with probability at least ε. We construct an adversary $\mathcal{M}$ against $\mathsf{UBK\text{-}CCA}[\mathcal{E}]$.

Upon reception of the public key n in the $\mathsf{UBK\text{-}CCA}$ game, $\mathcal{M}$ runs $\mathcal{R}$ on input n and uses the decryption oracle to simulate the $\mathsf{OW\text{-}CPA}$ adversary. Since by definition the decryption oracle decrypts any ciphertext with probability $1 \geq \varepsilon$ in one elementary step, the simulation of $\mathcal{A}$ is perfect for any $\varepsilon \in (0,1)$. The simulation complies to the definition of $\mathcal{R}$ because $\mathcal{R}$ is key-preserving. It is therefore crucial that this property holds otherwise $\mathcal{M}$ can by no means satisfy the queries $\mathcal{R}$ makes to $\mathcal{A}$.

$\mathcal{R}$ eventually returns the factorization of n with probability $\varepsilon_{\mathcal{R}}$ which $\mathcal{M}$ then returns as output value. $\mathsf{UBK\text{-}CCA}[\mathcal{E}]$ can therefore be broken with probability at least $\varepsilon_{\mathcal{R}}$ in extra time at most τ. $\square$

3.3 Indistinguishability versus Plaintext-Checking Security

Let us now consider $\mathsf{IND\text{-}CPA}[\mathcal{E}]$. We know that there is a key-preserving reduction $\mathsf{IND\text{-}CPA}[\mathcal{E}] \Leftarrow \mathsf{OW\text{-}CPA}[\mathcal{E}]$ and also that key-preservation is transitive. Therefore Theorem 1 implies that there is no key-preserving reduction $\mathsf{FACT}[\mathcal{E}.\mathsf{keygen}] \Leftarrow \mathsf{IND\text{-}CPA}[\mathcal{E}]$ unless $\mathsf{UBK\text{-}CCA}[\mathcal{E}]$ is polynomial. But precisely because $\mathsf{IND\text{-}CPA}[\mathcal{E}]$ is weaker than $\mathsf{OW\text{-}CPA}[\mathcal{E}]$, a stronger incompatibility result can be found. We state:

Theorem 2. *Let $\mathcal{E}$ be a single-key factoring-based encryption scheme. If there exists a polynomial key-preserving black-box reduction $\mathcal{R}$ such that* $\mathsf{FACT}[\mathcal{E}.\mathsf{keygen}] \Leftarrow_{\mathcal{R}} \mathsf{IND\text{-}CPA}[\mathcal{E}]$*, then* $\mathsf{UBK\text{-}PCA}[\mathcal{E}]$ *is polynomial.*

Proof. Let us first describe in more detail the game played by a key-preserving reduction $\mathcal{R}$ such that $\mathsf{FACT}[\mathcal{E}.\mathsf{keygen}] \Leftarrow_{\mathcal{R}} \mathsf{IND\text{-}CPA}[\mathcal{E}]$. Given a modulus n, $\mathcal{R}$ calls the adversarial oracle $\mathcal{A}$ breaking $\mathsf{IND\text{-}CPA}[\mathcal{E}]$ as follows. When $\mathcal{R}$ calls $\mathcal{A}(\mathsf{find}, n)$, $\mathcal{A}$ outputs two plaintexts $m_0, m_1 \in \mathsf{M}_n$

of equal length. $\mathcal{R}$ then encrypts m_b for $b \leftarrow \{0,1\}$ as c_b and calls $\mathcal{A}(\mathsf{guess}, c_b)$. $\mathcal{A}$ then returns its guess $\hat{b} \in \{0,1\}$ to $\mathcal{R}$ and $\Pr[\hat{b} = b] \geq \varepsilon$. We may assume w.l.o.g. that $\mathcal{R}$ never calls $\mathcal{A}(\mathsf{guess}, c_b)$ before calling $\mathcal{A}(\mathsf{find}, n)$ first and always calls $\mathcal{A}(\mathsf{guess}, c_b)$ immediately after $\mathcal{A}(\mathsf{find}, n)$, and that c_b is always a proper encryption of m_0 or m_1. Let 2ℓ be the total number of calls to $\mathcal{A}$. Overall $\mathcal{R}$ returns $\mathsf{factors}(n)$ with probability $\varepsilon_{\mathcal{R}}$ and extra time τ.

We construct a trivial meta-reduction $\mathcal{M}$ which converts the key-preserving black-box reduction $\mathcal{R}$ into an adversary against $\mathsf{UBK\text{-}PCA}[\mathcal{E}]$ and works with identical success probability in similar time. $\mathcal{M}$ works as follows. Given a public key $n \leftarrow \mathcal{E}.\mathsf{keygen}$, $\mathcal{M}$ runs $\mathcal{R}$ on input n and simulates the distinguisher $\mathcal{A}$ using the plaintext-checking oracle of the $\mathsf{UBK\text{-}PCA}$ game. When $\mathcal{R}$ calls $\mathcal{A}(\mathsf{find}, n)$, $\mathcal{M}$ returns two randomly selected plaintexts $m_0, m_1 \leftarrow \mathsf{M}_n$ of equal length. When $\mathcal{R}$ calls $\mathcal{A}(\mathsf{guess}, c_b)$, $\mathcal{M}$ sends (m_1, c_b) to the plaintext-checking oracle and sends its output back to $\mathcal{R}$ (recall that given $(m, c) \in \mathsf{M}_n \times \mathsf{C}_n$, the plaintext-checking oracle returns 1 if c encrypts m and 0 otherwise). Eventually $\mathcal{R}$ stops and $\mathcal{M}$ forwards the output of $\mathcal{R}$. By definition, the plaintext-checking oracle distinguishes plaintext-ciphertext pairs with probability one and $\mathcal{M}$ therefore provides a perfect simulation of $\mathcal{A}$ to $\mathcal{R}$ for any $\varepsilon \in (0,1)$. Hence $\mathcal{M}$ outputs the factors of n with identical probability $\varepsilon_{\mathcal{R}}$ in time $\tau + 2\ell\rho(k)$ where $\rho(k) = \mathsf{poly}(k)$ is the time needed to perform a random selection in M_n. $\square$

3.4 Separation Results

Corollary 1. *Let $\mathcal{E}$ be a single-key factoring-based encryption scheme. Unless* $\mathsf{FACT}[\mathcal{E}.\mathsf{keygen}]$ *is polynomial, there is no polynomial key-preserving black-box reduction* $\mathsf{FACT}[\mathcal{E}.\mathsf{keygen}] \Leftarrow \mathsf{IND\text{-}CCA}[\mathcal{E}]$.

Proof. Assume that $\mathsf{FACT}[\mathcal{E}.\mathsf{keygen}] \Leftarrow_{\mathcal{R}_1} \mathsf{IND\text{-}CCA}[\mathcal{E}]$ for some polynomial key-preserving black-box (PKPBB) reduction $\mathcal{R}_1$. Since there exists a PKPBB reduction $\mathcal{R}_2$ such that $\mathsf{IND\text{-}CCA}[\mathcal{E}] \Leftarrow_{\mathcal{R}_2} \mathsf{OW\text{-}CPA}[\mathcal{E}]$, there must be a PKPBB reduction $\mathcal{R}_3$ such that $\mathsf{FACT}[\mathcal{E}.\mathsf{keygen}] \Leftarrow_{\mathcal{R}_3} \mathsf{OW\text{-}CPA}[\mathcal{E}]$ by transitivity, resulting in that $\mathsf{UBK\text{-}CCA}[\mathcal{E}]$ is polynomial by Theorem 1. Moreover since $\mathsf{IND\text{-}CCA}[\mathcal{E}] \Leftarrow \mathsf{UBK\text{-}CCA}[\mathcal{E}]$, one gets that $\mathsf{IND\text{-}CCA}[\mathcal{E}]$ is polynomial and therefore that $\mathsf{FACT}[\mathcal{E}.\mathsf{keygen}]$ is polynomial as well. $\square$

Similar impossibility results are found for other security notions such as $\mathsf{OW\text{-}CCA}[\mathcal{E}]$ and $\mathsf{IND\text{-}PCA}[\mathcal{E}]$ using Theorem 2.

The Typical Example of Rabin-SAEP. We illustrate the importance of Corollary 1 by deducing a uninstantiability result for Rabin-SAEP. We first recall the definition of Rabin-SAEP [5]. Let s_m, s_0, s_1 be security parameters and $k = s_m + s_0 + s_1$. H denotes a fixed-size hash function $H : \{0,1\}^{s_1} \to \{0,1\}^{s_m+s_0}$. Here k plays the role of security parameter and the security proofs in [5] view s_m, s_0, s_1 as polynomial functions of k.

Rabin-SAEP.keygen : Given 1^k, generate a $(k+2)$-bit RSA modulus $n = pq$, $|p| = |q| = \lceil k/2 \rceil + 1$, $p = q = 3 \bmod 4$ and $n \in [2^{k+1}, 2^{k+1} + 2^k)$. The secret key is $\mathsf{factors}(n) = (p, q)$ while the public key is n.

Rabin-SAEP.$\mathsf{encrypt}$: Given a public key n, the message space is $\mathsf{M}_n = \{0,1\}^{s_m}$ and the random space is $\mathsf{R}_n = \{0,1\}^{s_1}$. For $(m, r) \in \mathsf{M}_n \times \mathsf{R}_n$, $\mathsf{Enc}(n, m, r)$ is defined as $(((m \,\|\, 0^{s_0}) \oplus H(r)) \,\|\, r)^2 \bmod n$. The ciphertext space is $\mathsf{C}_n = \mathbb{Z}_n$.

Rabin-SAEP.$\mathsf{decrypt}$: Given $c \in \mathsf{C}_n$ and (p, q), compute $z_p = c^{(p+1)/4} \bmod p$ and $z_q = c^{(q+1)/4} \bmod q$. Output $\perp$ if $z_p^2 \neq c \bmod p$ or $z_q^2 \neq c \bmod q$. Among the four values $\mathsf{CRT}(\pm z_p, \pm z_q)$, select the only one y such that $y < n/2$ and y can be parsed as $((m \,\|\, 0^{s_0}) \oplus H(r)) \,\|\, r$ for some $(m, r) \in \mathsf{M}_n \times \mathsf{R}_n$. If this fails or can be done for more than one value for y, output $\perp$. Otherwise output m.

It is easily seen that Rabin-SAEP is a single-key factoring-based encryption scheme as per the definition of Section 2. We refer to [5, Section 4] for a proof that Rabin-SAEP is chosen-ciphertext secure under the factoring assumption in the RO model:

Theorem 3 (RO-model security of Rabin-SAEP [5]). *Let us view H as a random oracle. There exists a PKPBB reduction $\mathcal{R}$ such that* $\mathsf{FACT}\,[\text{Rabin-SAEP}.\mathsf{keygen}] \Leftarrow_{\mathcal{R}} \mathsf{IND\text{-}CCA}\left[\text{Rabin-SAEP}^H\right]$.

We now state that for any instantiation of H, Rabin-SAEP does *not* admit a standard model counterpart of Theorem 3. This impossibility result comes as a direct application of Corollary 1.

Theorem 4 (Standard-model security of Rabin-SAEP). *Assuming* $\mathsf{FACT}\,[\text{Rabin-SAEP}.\mathsf{keygen}]$ *is intractable, there exists no PKPBB reduction* $\mathsf{FACT}\,[\text{Rabin-SAEP}.\mathsf{keygen}] \Leftarrow \mathsf{IND\text{-}CCA}\,[\text{Rabin-SAEP}]$.

Similar separations can be obtained for a wide range of factoring-based encryptions which chosen-ciphertext security is shown to be equivalent to factoring through key-preserving reductions in the RO model such as Rabin/RW-SAEP[+]/OAEP[+][+]/REACT, EPOC-2 [11], etc.

What Goes Wrong in the RO Model. Consider the meta-reduction $\mathcal{M}$ in the proof of Theorem 1. $\mathcal{M}$ cannot make any appropriate use of a key-preserving reduction $\mathcal{R}$ standing in the RO model. In a typical random-oracle-based reduction, the random oracles of $\mathcal{E}$ are simulated by $\mathcal{R}$. This additional power is beneficial to $\mathcal{R}$ which introduces some form of correlation between its own variables and the responses of the simulated oracles. In a sense, $\mathcal{R}$ is not totally black-box *i.e.* does not only rely on the input-output behavior of the $\mathsf{OW\text{-}CPA}$ adversary because $\mathcal{R}$ controls the interactions between the adversary and the random oracles to increase its success probability.

In the chosen-ciphertext security game, however, the decryption oracle makes implicit calls (*i.e.* not controllable by any simulator) to the random oracles. Therefore, the meta-reduction cannot influence the decryption procedure by mimicking $\mathcal{R}$ and consequently, can by no means correlate the internal variables of the decryption oracle to its own variables the same way $\mathcal{R}$ does with the $\mathsf{OW\text{-}CPA}$ adversary. This explains why the RO model is unaware of incompatibilities in a general sense.

4 Extended Results for Non-Malleable Key Generation

What we are after in this section is a way to strengthen the previous impossibility results. Recall we had to restrict the scope of Theorems 1 and 2 to key-preserving security reductions because the meta-reduction $\mathcal{M}$ was unable to simulate the adversary $\mathcal{A}$ when $\mathcal{R}$ makes oracle calls to $\mathcal{A}$ with arbitrary moduli. Our approach is to explicitly assume, as a property of the key generation of $\mathcal{E}$, that calling $\mathcal{A}$ with $n' \neq n$ is essentially of no help to $\mathcal{R}$ anyways. It appears that one faces definitional options when capturing this in a formal way: what we provide hereafter is the simplest definition that is strong enough for our purposes. This in turn allows us to consider *arbitrary* black-box reductions at the expense of making a complexity assumption on the key generation of $\mathcal{E}$.

4.1 Defining Non-Malleable Generators

Intuition. An instance generator Gen is said to be malleable if factoring a randomly selected instance $n \leftarrow \mathsf{Gen}(1^k)$ becomes substantially easier when given repeated access to an oracle which factors other instances $n' \neq n$ for $n' \in \mathcal{PK}_k$. A typical example of malleability is when $\mathcal{PK}_k$ contains integers of variable size and number of prime factors. It is indeed trivial to factor n given an oracle that factors $n' = \alpha n$ if it happens that both

n and n' are proper elements of $\mathcal{PK}_k$. We observe that most factoring-based cryptosystems define instance generators which precisely tend to avoid this malleability property by construction (see Section 2). What we need for our purposes is to define non-malleability in a strong sense.

Definition. To properly capture non-malleability, we define two games in which a probabilistic algorithm $\mathcal{R}$ attempts to factor $n \leftarrow \mathsf{Gen}(1^k)$ given access to an oracle $\mathcal{A}(n, \mathsf{aux})$ solving with probability one some computational problem reducible to $\mathsf{FACT}\,[\mathsf{Gen}]$. Here, $\mathcal{A}$ models the computational resources $\mathcal{R}$ has access to and aux stands for any auxiliary input given to the oracle $\mathcal{A}$ depending on how $\mathcal{A}$ is specified. We may write $\mathcal{A}(n, \cdot)$ instead of $\mathcal{A}(n, \mathsf{aux})$ to notify that aux is chosen freely and arbitrarily by $\mathcal{R}$ when $\mathcal{A}$ is called. Since we impose that oracle $\mathcal{A}$ be perfect, we can abuse notations and identify $\mathcal{A}$ to the problem solved by $\mathcal{A}$. A typical example of computational resources modelled by $\mathcal{A}$ is when $\mathcal{A}$ is polynomial (in which case $\mathcal{R}$ is given no extra power), but one may consider problems reducible to $\mathsf{FACT}\,[\mathsf{Gen}]$ that do confer a computational advantage to $\mathcal{R}$, such as distinguishing quadratic residues modulo n, extracting e-th roots for $\gcd(e, \phi(n)) = 1$ and so forth. In any case, we require $\mathcal{A}$ to be perfectly reducible to $\mathsf{FACT}\,[\mathsf{Gen}]$ in polynomial time, that is, for any $n \in \mathcal{PK}_k$ and any admissible value for aux, $\mathcal{A}(n, \mathsf{aux})$ must be solvable with probability one in time $t_{\mathcal{A}} = \mathsf{poly}\,(k)$ given $\mathsf{factors}(n)$. **In Game 0**, the success probability of $\mathcal{R}$ is defined as

$$\mathsf{Succ}_{\mathsf{Gen}}^{\text{Game } 0}\,(\mathcal{R}, \mathcal{A}, \tau, \ell) = \Pr\left[n \leftarrow \mathsf{Gen}(1^k)\ :\ \mathcal{R}^{\mathcal{A}(n,\cdot)}(n) = \mathsf{factors}(n)\right]$$

where the probability is taken over the random tapes of $\mathcal{R}$ and $\mathcal{A}$, $\mathcal{R}$ runs in extra time at most τ and makes at most ℓ queries to $\mathcal{A}(n, \cdot)$. We further define

$$\mathsf{Succ}_{\mathsf{Gen}}^{\text{Game } 0}\,(\mathcal{A}, \tau, \ell) = \max_{\mathcal{R}} \mathsf{Succ}_{\mathsf{Gen}}^{\text{Game } 0}\,(\mathcal{R}, \mathcal{A}, \tau, \ell)$$

where the maximum is taken over all probabilistic algorithms $\mathcal{R}$ playing Game 0. This can be interpreted as the success probability of the best reduction that makes use of $\mathcal{A}(n, \mathsf{aux})$ to factor n for the given reduction parameters (τ, ℓ). **In Game 1**, the reduction $\mathcal{R}$ is given, in addition to $\mathcal{A}$, access to an auxiliary oracle $\mathsf{FACT}(\cdot)$ that factors integers $n' \in \mathcal{PK}_k \setminus \{n\}$ with probability one. Its success probability $\mathsf{Succ}_{\mathsf{Gen}}^{\text{Game } 1}\,(\mathcal{R}, \mathcal{A}, \tau, \ell)$ is then

$$\Pr\left[n \leftarrow \mathsf{Gen}(1^k)\ :\ \mathcal{R}^{\mathcal{A}(n,\cdot), \mathsf{FACT}(\cdot)}(n) = \mathsf{factors}(n)\right]$$

where the probability is taken over the random tapes of $\mathcal{R}$ and $\mathcal{A}$, $\mathcal{R}$ runs in extra time at most τ, makes $\ell_{\mathcal{A}}$ calls to $\mathcal{A}(n, \cdot)$ and ℓ_{FACT} calls of the type $\mathsf{FACT}(n')$ with $n' \in \mathcal{PK}_k \setminus \{n\}$ such that $\ell_{\mathcal{A}} + \ell_{\mathsf{FACT}} \leq \ell$. Let us define

$$\mathsf{Succ}_{\mathsf{Gen}}^{\text{Game 1}}(\mathcal{A}, \tau, \ell) = \max_{\mathcal{R}} \mathsf{Succ}_{\mathsf{Gen}}^{\text{Game 1}}(\mathcal{R}, \mathcal{A}, \tau, \ell)$$

where the maximum is taken over all probabilistic algorithms $\mathcal{R}$ playing Game 1. This measures the success probability of the best reduction that uses simultaneously oracles $\mathcal{A}(n, \cdot)$ and $\mathsf{FACT}(\cdot)$ to factor n in time τ and totalling no more than ℓ oracle calls. We finally define the *malleability* of Gen as

$$\varDelta_{\mathsf{Gen}}(\tau, \ell) = \max_{\mathcal{A} \Leftarrow \mathsf{FACT}[\mathsf{Gen}]} \left| \mathsf{Succ}_{\mathsf{Gen}}^{\text{Game 1}}(\mathcal{A}, \tau, \ell) - \mathsf{Succ}_{\mathsf{Gen}}^{\text{Game 0}}(\mathcal{A}, \tau, \ell) \right| ,$$

where the maximum is now taken over all computational problems $\mathcal{A}$ perfectly reducible to $\mathsf{FACT}[\mathsf{Gen}]$ in polynomial time.

Remark 1. It is easily seen that $\varDelta_{\mathsf{Gen}}(\tau, 0) = 0$ for any $\tau \geq 0$.

Definition 1 (Non-Malleable Instance Generators). *We say that an instance generator* Gen *is non-malleable when* $\varDelta_{\mathsf{Gen}}(\tau, \ell)$ *remains polynomially negligible in* k *when* $\tau = \mathsf{poly}(k)$ *and* $\ell = \mathsf{poly}(k)$.

Remark 2. The purpose of Game 0 is to include all key-preserving reductions $\mathcal{R}$ such that $\mathsf{FACT}[\mathsf{Gen}] \Leftarrow_{\mathcal{R}} \mathcal{A}$. Since the success probability ε of the adversarial oracle plays no role in the proofs of Theorems 1 and 2, these can be reformulated as follows. For any positive integers τ, ℓ:

Th. 1: $\mathsf{Succ}_{\mathcal{E}.\mathsf{keygen}}^{\text{Game 0}}(\mathsf{OW\text{-}CPA}[\mathcal{E}], \tau, \ell) \leq \mathsf{Succ}(\ell\text{-}\mathsf{UBK\text{-}CCA}[\mathcal{E}], \tau)$

Th. 2: $\mathsf{Succ}_{\mathcal{E}.\mathsf{keygen}}^{\text{Game 0}}(\mathsf{IND\text{-}CPA}[\mathcal{E}], \tau, \ell) \leq \mathsf{Succ}(\ell\text{-}\mathsf{UBK\text{-}PCA}[\mathcal{E}], \tau + 2\ell\rho(k))$

4.2 A Fundamental Lemma

We now come back to our earlier discussion about extending the scope of Theorem 1 and dealing with $\mathcal{R}$ calling $\mathcal{A}$ with arbitrary moduli $n' \neq n$. The oracle calls $\mathcal{R}$ makes to $\mathcal{A}$ are now of two types: calls with the same modulus n (key-preserving calls) and calls with $n' \neq n$ (non-key-preserving calls). Our definition of non-malleability allows us to limit the computational advantage conferred to $\mathcal{R}$ by its non-key-preserving calls.

Lemma 1. *Let* Gen *be an instance generator and let* $\mathcal{A}$ *be a computational problem perfectly reducible to* $\mathsf{FACT}[\mathsf{Gen}]$ *in time* $t_{\mathcal{A}}$. *Then for any positive integers* τ, ℓ *and any* $\varepsilon \in (0, 1)$,

$$\mathsf{Succ}(\mathsf{FACT}[\mathsf{Gen}] \Leftarrow \mathcal{A}, \tau, \varepsilon, \ell) \leq \mathsf{Succ}_{\mathsf{Gen}}^{\text{Game 1}}(\mathcal{A}, \tau + \ell \cdot t_{\mathcal{A}}, \ell) \ .$$

Proof. Recall that $\mathcal{A}$ denotes a computational problem here. Assume $\mathcal{R}$ $(\tau, \varepsilon, \ell)$-solves $\mathsf{FACT}\,[\mathsf{Gen}] \Leftarrow \mathcal{A}$ *i.e.* factors $n \leftarrow \mathsf{Gen}(1^k)$ in extra time τ with no more than ℓ calls to an oracle $\mathcal{A}_\mathcal{R}$ solving $\mathcal{A}$ with probability ε. Let $\varepsilon_\mathcal{R}$ be the success probability of $\mathcal{R}$. We construct an algorithm $\mathcal{M}$ which plays Game 1 with respect to a perfect oracle $\mathcal{A}_\mathcal{M}$ for $\mathcal{A}$ and succeeds with identical probability and similar running time. Algorithm $\mathcal{M}$ works as follows. Given a randomly selected modulus $n \leftarrow \mathsf{Gen}(1^k)$, $\mathcal{M}$ runs $\mathcal{R}$ on input n. Now when $\mathcal{R}$ calls $\mathcal{A}_\mathcal{R}(n, \mathsf{aux})$, $\mathcal{M}$ calls $\mathcal{A}_\mathcal{M}(n, \mathsf{aux})$ and forwards the output to $\mathcal{R}$. When $\mathcal{R}$ calls $\mathcal{A}_\mathcal{R}(n', \mathsf{aux})$ for $n' \in \mathcal{PK}_k \setminus \{n\}$, $\mathcal{M}$ calls $\mathsf{FACT}(n')$ to get $\mathsf{factors}(n')$ and solves $\mathcal{A}(n', \mathsf{aux})$ in time $t_\mathcal{A}$. $\mathcal{M}$ then returns the result to $\mathcal{R}$. $\mathcal{R}$ eventually stops and $\mathcal{M}$ returns the output of $\mathcal{R}$. The simulation of $\mathcal{A}_\mathcal{R}$ is perfect for any $\varepsilon \in (0, 1)$. $\mathcal{M}$ requires extra time at most $\tau + \ell \cdot t_\mathcal{A}$ and makes at most ℓ calls to oracles $\mathcal{A}_\mathcal{M}$ and $\mathsf{FACT}(\cdot)$ altogether. □

4.3 Extended Separation Results

Theorem 5. *Let $\mathcal{E}$ be a single-key factoring-based encryption scheme and assume $\mathcal{E}.\mathsf{keygen}$ is non-malleable. If $\mathsf{FACT}\,[\mathcal{E}.\mathsf{keygen}] \Leftarrow \mathsf{OW\text{-}CPA}\,[\mathcal{E}]$ then $\mathsf{UBK\text{-}CCA}\,[\mathcal{E}]$ is polynomial.*

Proof. Let us consider $\mathcal{A} = \mathsf{OW\text{-}CPA}\,[\mathcal{E}]$. Obviously $\mathcal{A}$ is perfectly reducible to $\mathsf{FACT}\,[\mathcal{E}.\mathsf{keygen}]$ since given any $n \in \mathcal{PK}_k$, $\mathsf{aux} = c \in \mathsf{C}_n$ and $\mathsf{factors}(n)$, $\mathcal{A}(n, \mathsf{aux})$ is solved by computing $m = \mathsf{Dec}(\mathsf{factors}(n), c)$ in time $t_\mathcal{A} = \mathsf{poly}\,(k)$. Applying Lemma 1, we get for any τ, ℓ and $\varepsilon \in (0, 1)$:

$$\begin{aligned}
&\mathsf{Succ}\left(\mathsf{FACT}\,[\mathcal{E}.\mathsf{keygen}] \Leftarrow \mathsf{OW\text{-}CPA}\,[\mathcal{E}]\,, \tau, \varepsilon, \ell\right)\\
\le\ &\mathsf{Succ}^{\text{Game 1}}_{\mathcal{E}.\mathsf{keygen}}\left(\mathsf{OW\text{-}CPA}\,[\mathcal{E}]\,, \tau + \ell\cdot\mathsf{poly}\,(k)\,, \ell\right)\\
\le\ &\mathsf{Succ}^{\text{Game 0}}_{\mathcal{E}.\mathsf{keygen}}\left(\mathsf{OW\text{-}CPA}\,[\mathcal{E}]\,, \tau + \ell\cdot\mathsf{poly}\,(k)\,, \ell\right) + \Delta_{\mathsf{Gen}}\left(\tau + \ell\cdot\mathsf{poly}\,(k)\,, \ell\right)\\
\le\ &\mathsf{Succ}\left(\ell\text{-}\mathsf{UBK\text{-}CCA}\,[\mathcal{E}]\,, \tau + \ell\cdot\mathsf{poly}\,(k)\right) + \Delta_{\mathsf{Gen}}\left(\tau + \ell\cdot\mathsf{poly}\,(k)\,, \ell\right)\ .
\end{aligned}$$

We now extend asymptotically the above to $\tau, \ell = \mathsf{poly}\,(k)$. Since $\mathcal{E}.\mathsf{keygen}$ is non-malleable, the malleability term $\Delta_{\mathsf{Gen}}\left(\tau + \ell\cdot\mathsf{poly}\,(k)\,, \ell\right)$ remains negligible. Since $\mathsf{Succ}\left(\mathsf{FACT}\,[\mathcal{E}.\mathsf{keygen}] \Leftarrow \mathsf{OW\text{-}CPA}\,[\mathcal{E}]\,, \tau, \varepsilon, \ell\right)$ is non-negligible by assumption, $\mathsf{Succ}\left(\ell\text{-}\mathsf{UBK\text{-}CCA}\,[\mathcal{E}]\,, \tau + \ell\cdot\mathsf{poly}\,(k)\right)$ must be non-negligible as well, thereby giving the result. □

The same proof technique applies to $\mathsf{IND\text{-}CPA}\,[\mathcal{E}]$ and shows that there exists no reduction $\mathsf{FACT}\,[\mathcal{E}.\mathsf{keygen}] \Leftarrow \mathsf{IND\text{-}CPA}\,[\mathcal{E}]$ unless $\mathsf{UBK\text{-}PCA}\,[\mathcal{E}]$ is polynomial or $\mathcal{E}.\mathsf{keygen}$ is malleable. Based on a reasoning similar to the proof of Corollary 1, we deduce from these incompatibilities that:

Corollary 2. *Let $\mathcal{E}$ be a single-key factoring-based encryption scheme and assume* $\mathcal{E}.\mathsf{keygen}$ *is non-malleable. There is no polynomial black-box reduction* $\mathsf{FACT}\,[\mathcal{E}.\mathsf{keygen}] \Leftarrow \mathsf{IND\text{-}CCA}\,[\mathcal{E}]$ *unless* $\mathsf{FACT}\,[\mathcal{E}.\mathsf{keygen}]$ *is polynomial.*

To exemplify Corollary 2, we provide this extended impossibility result for Rabin-SAEP.

Theorem 6 (Standard-model security of Rabin-SAEP, revisited). *Assume* Rabin-SAEP.keygen *is non-malleable. Then* Rabin-SAEP *admits no instantiation in the standard model which is chosen-ciphertext secure under the factoring assumption* i.e. *for any instantiation of H,*

$$\mathsf{IND\text{-}CCA}\,[\text{Rabin-SAEP}] \not\equiv \mathsf{FACT}\,[\text{Rabin-SAEP}.\mathsf{keygen}] \ .$$

Similar uninstantiability results hold for single-key factoring-based encryption schemes which chosen-ciphertext security is shown to be equivalent to factoring in the RO model. Again, these stronger separations are effective only when the underlying key generation is non-malleable. In other words, either these encryption schemes do separate the RO model from the standard model in a very strong sense, or their key generation must be malleable along the lines of Definition 1.

5 Overcoming Uninstantiability

Keyed Paddings. At first look, including some additional key material such as a random string in the public key seems to invalidate our impossibility results completely. Typically the extra parameter can serve as a function index in a keyed family of hash functions. This seems to be an efficient countermeasure for single-key factoring-based encryption making use of encryption paddings which, unlike SAEP[+]/OAEP[+][+], Fujisaki-Okamoto and REACT, include keyed hash functions.

Encryption Twinning. Naor and Yung [17] and Dolev, Dwork and Naor [10] suggested transformations which when applied to $\mathsf{IND\text{-}CPA}$-secure encryptions such as Blum-Goldwasser [4] or Chor-Goldreich [8] may lead to $\mathsf{IND\text{-}CCA}$-secure schemes under the factoring assumption. The transformed schemes use public keys containing two or more independently generated moduli with respect to the basic scheme. This paradigm makes it possible to generically construct a larger class of factoring-based cryptosystems which $\mathsf{IND\text{-}CCA}$-security can possibly be proven equivalent to factoring, thereby escaping all incompatibility results described earlier.

We comment that the cornerstone of Theorem 1 resides in that the decryption oracle provided in the UBK-CCA game can serve as a factoring algorithm when interfaced with the black-box reduction $\mathcal{R}$. We now see how encryption twinning prohibits such a use of the decryption oracle. The public key in a Naor-Yung-transformed encryption scheme $\mathsf{NY}(\mathcal{E})$ is (n_1, n_2, r) where $n_1, n_2 \leftarrow \mathcal{E}.\mathsf{keygen}$ and r is a random string used to generate NIZK proofs during encryption. An encryption of $m \in \mathsf{M}_{n_1} \cap \mathsf{M}_{n_2}$ is $(c_1 = \mathsf{Enc}(n_1, m, r_1), c_2 = \mathsf{Enc}(n_2, m, r_2), \pi)$ where π is a proof that c_1 and c_2 encrypt the same plaintext. Now assume (as typically the case with single-key factoring-based encryption) there exists an efficient way to generate a random-looking c_1 such that its decryption $\mathsf{Dec}(\mathsf{factors}(n_1), c_1)$ leads to an immediate recovery of $\mathsf{factors}(n_1)$. In a typical reduction $\mathcal{R}$ from $\mathsf{FACT}[\mathcal{E}.\mathsf{keygen}]$ to breaking the OW-CPA security of $\mathsf{NY}(\mathcal{E})$, $\mathcal{R}$ takes as input a modulus $n_1 \leftarrow \mathcal{E}.\mathsf{keygen}(1^k)$ but generates by itself the second key pair $(n_2, \mathsf{factors}(n_2)) \leftarrow \mathcal{E}.\mathsf{keygen}(1^k)$ and r to constitute a public key $\mathsf{pk} = (n_1, n_2, r)$. Since $\mathcal{R}$ fully controls the generation of n_2 and r, $\mathcal{R}$ can use the simulator of the underlying NIZK proof system to create a valid encryption $c = (c_1, c_2, \pi)$ for a random c_1. Calling the OW-CPA adversary will then provide $\mathsf{Dec}(\mathsf{factors}(n_1), c_1)$, thus allowing $\mathcal{R}$ to factor n_1. The meta-reduction $\mathcal{M}$ playing the UBK-CCA game against $\mathsf{NY}(\mathcal{E})$ however, is given some public key $\mathsf{PK} = (N_1, N_2, R)$ and a decryption oracle implicitly parameterized by PK. Since $\mathcal{R}$ takes as input a single modulus and generates by itself the rest of the public key to be given to its adversarial oracle, $\mathcal{M}$ cannot, even if $\mathcal{R}$ is run on input N_1, use the decryption oracle to answer the request(s) $((N_1, n_2, r), (c_1, c_2, \pi))$ made by $\mathcal{R}$ because $\Pr[n_2 \neq N_2 \vee r \neq R]$ is overwhelming.

6 Are Key Generators Non-Malleable?

Our extended impossibility results apply to single-key encryption schemes based on non-malleable key generation. We conjecture that most instance generators are in turn non-malleable and expect to see further research works based on this property in the future. A possible improvement of this work would be to give a formal proof of non-malleability for commonly referred generators such as RSA-3 or Sophie-Germain using computational number theory. Another issue is the design of non-trivial examples of *malleable* key generators.

Acknowledgements. We thank the anonymous referees of Asiacrypt'06 for their numerous comments as well as Mihir Bellare for suggestions that substantially improved the presentation of this paper. This work has been financially supported by the European Commission through the IST Program under Contract IST-2002-507932 ECRYPT.

References

1. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO'98*, LNCS 1462, pp. 26–46. Springer Verlag, 1998.
2. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS'93*, pp. 62–73, 1993.
3. M. Bellare and P. Rogaway. Optimal asymmetric encryption: How to encrypt with RSA. In *EUROCRYPT'94*, LNCS 950, pp. 92–111. Springer Verlag, 1994.
4. M. Blum and S. Goldwasser. An efficient probabilistic public key encryption scheme which hides all partial information. In *CRYPTO'84*, LNCS 196, pp. 289–299, 1985.
5. D. Boneh. Simplified OAEP for the RSA and Rabin functions. In *CRYPTO'01*, LNCS 2139, pp. 275–291. Springer Verlag, 2001.
6. D. Boneh and R. Venkatesan. Breaking RSA may not be equivalent to factoring (extended abstract). In *EUROCRYPT'98*, LNCS 1403, pp. 59–71, 1998.
7. D. R. L. Brown. Unprovable security of RSA-OAEP in the standard model, 2006. http://eprint/iacr.org/2006/223.
8. B. Chor and O. Goldreich. RSA/Rabin least significant bits are $\frac{1}{2} + \frac{1}{poly(\log N)}$ secure. In *CRYPTO'84*, LNCS 196, pp. 303–313. Springer Verlag, 1985.
9. N. Demytko. A new elliptic curve based analogue of RSA. In *EUROCRYPT'93*, LNCS 765, pp. 40–49. Springer Verlag, 1994.
10. D. Dolev, C. Dwork and M. Naor. Non-malleable cryptography. In *ACM STOC'91*, pp. 542–552, 1991.
11. E. Fujisaki. Chosen-chipertext security of EPOC-2. Technical report, NTT Corporation, 2001.
12. E. Fujisaki, T. Kobayashi, H. Morita, H. Oguro, T. Okamoto, S. Okazaki, D. Pointcheval and S. Uchiyama. EPOC: Efficient probabilistic public-key encryption. Submitted to ISO and NESSIE.
13. S. Goldwasser, S. Micali and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. on Comp.*, 17(2):281–308, April 1988.
14. D. Jao, S. Miller and R. Venkatesan. Do all elliptic curves of the same order have the same difficulty of disc. log? In *ASIACRYPT'05*, LNCS 3788, pp. 21–40, 2005.
15. K. Koyama, U. Maurer, T. Okamoto and S. Vanstone. New public-key schemes based on curves over the ring $\mathbb{Z}_n$. In *CRYPTO'91*, LNCS 576, pp. 252–266, 1992.
16. T. Malkin, R. Moriarty and N. Yakovenko. Generalized environmental security from number theoretic assumptions. In *TCC'06*, LNCS 3876, pp. 343–359, 2006.
17. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen-ciphertext attacks. In *22nd ACM Symposium on Theory of Computing*, 1990.
18. T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In *CT-RSA'01*, LNCS 2020, pp. 159–175, 2001.
19. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT'99*, LNCS 1592, pp. 223–238. Springer Verlag, 1999.
20. M. O. Rabin. Digital signatures and public key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Jan. 1979.
21. RSA Data Security. PKCS #1: RSA encryption standard, Nov. 1993. Version 1.5.
22. H. C. Williams. A modification of the RSA public-key encryption procedure. *IEEE Transactions on Information Theory*, IT-26(6):726–729, 1980.