On the Security of Padding-Based Encryption Schemes
– or –
Why We Cannot Prove OAEP Secure in the Standard Model

Eike Kiltz⋆ and Krzysztof Pietrzak

Cryptology & Information Security Group
CWI Amsterdam, The Netherlands
{pietrzak,kiltz}@cwi.nl

Abstract. We investigate the security of "padding-based" encryption schemes in the standard model. This class contains all public-key encryption schemes where the encryption algorithm first applies some invertible public transformation to the message (the "padding"), followed by a trapdoor permutation. In particular, this class contains OAEP and its variants.

Our main result is a black-box impossibility result showing that one cannot prove any such padding-based scheme chosen-ciphertext secure even assuming the existence of ideal trapdoor permutations. The latter is a strong ideal abstraction of trapdoor permutations which inherits all security properties of uniform random permutations.

Keywords: Padding-based encryption, OAEP, black-box, ideal trapdoor permutations.

1 Introduction

1.1 Padding Schemes for Encryption

Optimal Asymmetric Encryption Padding (OAEP) is one of the most known and widely deployed asymmetric encryption schemes. It was designed by Bellare and Rogaway [3] as a scheme based on a trapdoor permutation (TDP). OAEP is standardized in RSA's PKCS #1 V2.1 and is part of the ANSI X9.44, IEEE P1363, ISO 18033-2 and SET standards. After the proposal of OAEP, several variants were proposed, such as Shoup's OAEP+ [41], Boneh's Simplified OAEP [8] (SAEP), and many others (e.g., [1,8,12,13,16,30,29,34,36,37]). All the aforementioned schemes can be classified as "padding-based encryption

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-642-01001-9_35

⋆ Supported by the research program Sentinels Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

A. Joux (Ed.): EUROCRYPT 2009, LNCS 5479, pp. 389–406, 2009.

schemes": the encryption algorithm first applies a public injective transformation π to message m and randomness r, and then a trapdoor permutation f to the result, i.e., $\mathsf{Enc}(m;r) = f(\pi(m,r))$. Decryption inverts the trapdoor permutation and then applies an inverse transformation $\hat{\pi}$ to reconstruct the message (or output a special rejection symbol), i.e., $\mathsf{Dec}(c) = \hat{\pi}(f^{-1}(c))$. The two public transformations $\mathit{\Pi} = (\pi, \hat{\pi})$ (with the consistency requirement $\hat{\pi}(\pi(m,r)) = m$, for all m, r) are called a padding scheme. For example, in the case of OAEP the padding $\mathit{\Pi}$ consists of a two round Feistel network, involving two hash functions.

Despite their practical importance, the only known security results for all known padding-based encryption schemes are in the random oracle model [2], where one assumes the existence of "ideal hash functions." For example, in the random oracle model, OAEP+ is secure against chosen-ciphertext attack (IND-CCA secure [38]) under the assumption that the TDP is one-way [41]; OAEP is IND-CCA secure under the (stronger) assumption that the TDP is partial-domain one-way [18]. However, such proofs merely provide heuristic evidence that breaking the scheme may be hard in reality, where the random oracles must be instantiated with some efficient hash functions. Moreover, a growing number of papers raised theoretical concerns regarding the soundness of the random oracle model. (See, e.g., [10,24].) This leaves the question whether or not padding-based encryption schemes can be securely instantiated in the standard model based on reasonable assumptions to the underlying trapdoor permutation. Or, the same question applied to OAEP: can we securely instantiate OAEP's random oracles assuming the TDP fulfils some strong security assumption beyond (partial-domain) one-wayness?

Ideal Trapdoor Permutations. We consider keyed trapdoor permutations TDP (which formally consist of key-generation, evaluation, and inversion algorithms). Extending Dodis et. al. [17], we propose the notion of an *ideal trapdoor permutation.* Informally, an ideal trapdoor permutation is a TDP that inherits *all* security properties of a uniformly random permutation. More concretely, TDP is an ideal trapdoor permutation if it satisfies all game-based security properties which are satisfied by random permutations. We stress that our basic definition only covers games where the challenger is not given the trapdoor to invert the TDP.[1]

Ideal TDPs are very powerful cryptographic primitives: by definition their security properties include, for example, (exponentially-hard) one-wayness, partial-domain one-wayness, claw-freeness, pseudo-randomness, and many other notions; from an ideal TDP we can build (in a black-box way) most cryptographic primitives, including collision-resistant hashing, pseudorandom generators (PRGs) and functions (PRFs), digital signatures, perfectly one-way hash functions (POWHFs), and, in particular, IND-CCA secure public-key encryption.

Let us remark that conceptually ideal TDPs are quite different from other idealized models like the random oracle model or the ideal cipher model. Informally,

[1] Note that allowing the challenger to access the trapdoor to invert the TDP would make it possible to model the IND-CCA security experiment itself as such a game, and thus security of schemes like OAEP would trivially follow from the fact that they are secure in the random oracle model.

the latter two models refer to particular objects (e.g., the random oracle model assumes that all parties have access to an oracle realizing a uniformly random function), whereas an ideal TDP is defined by its security properties. Although we also realize an ideal TDP via an oracle (basically, an ideal cipher with some extra functionalities modelling the trapdoors), we can add other inefficient oracles (in our case an oracle breaking all padding based encryption schemes) and –if this oracle does not affect the security properties of an ideal TDP– still are in the "ideal TDP" model.

Black-box reductions. Usually, when one constructs a cryptographic primitive A (e.g., a PRG) out of another cryptographic primitive B (e.g., a OWF), A uses B as a subroutine, independent of the particular implementation of B. The security proof for A constructs an adversary for B using any adversary for A as a subroutine. This is known as a "black-box reduction from primitive A to B" [27,26]. Black-box reductions play an important role in cryptography since almost all reductions are black-box. A black-box separation result means that there exists no black-box reduction from A to B. The common interpretation of these results is that there are inherent limitations in building primitive A from B, and that these impossibility results can be overcome only by explicitly using the code of primitive B in the construction. Although there are quite a few cryptographic constructions which are not black box —in our context, the most notable is the Naor-Yung paradigm to construct IND-CCA secure cryptosystem from any enhanced trapdoor permutation [33,31]— such constructions are usually prohibitively inefficient (e.g., the Naor-Yung paradigm relies on non-interactive zero knowledge proofs), and thus mostly of theoretical interest.

1.2 Results

Main Result. Our main result is a negative one. We show that there is no instantiation of a padding-based encryption scheme that allows a *black-box reduction* from ideal trapdoor permutations to its IND-CCA security. That is, we consider all possible padding-based encryption schemes where the padding scheme $\Pi = \Pi^{\mathsf{TDP}}$ is an oracle circuit having arbitrary access to the underlying trapdoor permutation TDP (and Π may even arbitrarily depend on the trapdoor-key of the final permutation). None of these constructions can be proved IND-CCA secure based on the security properties of the ideal trapdoor permutation TDP, using a black-box reduction.[2] As already discussed before, this hints some inherent limitations in the design concept of padding-based encryption schemes in general, and of OAEP, in particular.

Let us stress that *efficient* ideal trapdoor permutations do not exist, thus showing a black-box construction of some primitive from ideal TDPs would have limited practical relevance. (Somewhat similar to, say, a proof in the random oracle model.) But keep in mind, that we prove an impossibility result, and showing that no padding-based encryption scheme can be proved secure from ideal TDPs, immediately implies that no such scheme can be proven secure assuming a TDP which

[2] Since we require that $\hat{\pi}$ from $\Pi = (\pi, \hat{\pi})$ is publicly invertible, we avoid that the padding itself is already an IND-CCA secure encryption scheme.

has some subset of the security properties of ideal TDPs (where the subset is such that it potentially could be satisfied by some efficient construction).

Technical overview. To obtain our separation result, we describe two oracles, T and B, such that T implements an ideal trapdoor permutation. Furthermore, given access to oracle B, an adversary can (trivially) break the IND-CCA security of any padding-based encryption scheme. Yet, B does not help an adversary to break the security properties of the ideal trapdoor permutation. Now our main result can be derived using the "two-oracle separation technique" by Hsiao and Reyzin [26]. (Informally, since a black-box security proof would also have to be valid relative to the two oracles T and B, such a proof cannot exists.)

Impact on OAEP and Related Schemes. One direct application of our general theorem is that OAEP is unprovable, even assuming that (i) the TDP (used for the final encryption) has any security property satisfied by ideal TDPs; (ii) one makes any computational assumption on the hash functions in the Feistel network, such that hash functions with this security properties can be constructed from ideal TDPs.

This in particular rules out the instantiability of OAEP from most cryptographic primitives like (partial-domain) one-way trapdoor permutations, collision-resistant hash functions, PRGs, PRFs, POWHFs, and more. (Since all these primitives are black-box implied by ideal TDPs.) Our interpretation of this result is that, in order to prove IND-CCA security of OAEP in the standard model, one has to rely on a security property of the underlying TDP that is not fulfilled by ideal TDPs[3]; or, one has to make use of special non black-box properties of the employed trapdoor permutation (e.g., if one uses the RSA permutation one may try to use the fact that it is homomorphic and random self-reducible).

We stress that, even though our results indicate certain limitations in the design of OAEP and related schemes, we do not show that they are insecure, nor that one of the security claims from [41,18] are incorrect. In particular, OAEP's *random oracle model* security proof [18] can still be viewed as a valid argument supporting OAEP's security in practice. In particular, there is no "generic attack" on OAEP which treats the hash functions like random oracles.

Extensions. For the sake of simplicity we first prove our basic impossibility result as outlined above. In the full version of this paper we will then discuss how this result can be strengthened in several ways, in particular:

- We observe that our impossibility proof does not actually need the full power of IND-CCA attacks, and thus we already exclude the possibility of proving security under a significantly weaker notion. Although this notion is somewhat artificial, it contains (and thus we rule out) natural notions such as security in the sense of IND-CCA1 (lunchtime) and NM-CPA (non-malleability).
- Following [17], we extend our results to *ideal trapdoor permutations with bounded inversions.* The latter is like an ideal TDP, but in the game defining

[3] This could either be security properties where the challenger has access to the trapdoor and thus can invert the permutation or properties *not* satisfied by random permutations.

the security property, the challenger is additionally allowed to invert f on an a-priori bounded number of points. We remark that ideal TDPs with bounded inversion black-box imply the important cryptographic primitive of verifiable random functions [32] (VRFs).

- A permutation $f(\cdot)$ is homomorphic, if from $f(x), f(y)$ one can efficiently compute $f(x \circ y)$ (for some group operation $\circ$). The homomorphic property (of, e.g., RSA) has proved very useful and was exploited in numerous security proofs. As ideal TDPs are *not* homomorphic, our main result does not rule out the possibility of basing the security of some padding-based encryption scheme on the homomorphic property of the underlying TDP. Unfortunately, we show that this is not the case, as our impossibility result still holds if we add some additional oracle which imposes a homomorphic structure on the ideal TDP.

1.3 Related Work

Black-box Separations. After Impagliazzo and Rudich [27] showed that there are no black-box constructions of key-agreement protocols from one-way permutations, substantial additional work in this line followed (see, for example [19,20,22,28,42], and many more). To obtain our separation result, we use the direct "two-oracle separation technique" by Hsiao and Reyzin [26]. Most relevant to our result is the work of Dodis et. al. [17], who consider the security of full-domain hash signatures in the standard model. They showed that there is no instantiation of full-domain hash signatures that can be proven secure based on black-box access to (in our language) ideal trapdoor permutations. Also related is the separation result by Gertner et. al. [21] on the black-box impossibility of constructing IND-CCA from IND-CPA secure public-key encryption without using "re-encryption" (i.e., where decryption of the IND-CCA secure scheme is not allowed to use encryption of the IND-CPA secure scheme).

(In)security of OAEP. Due to its practical importance, a growing number of papers consider the security properties of OAEP. Revisiting the earlier security claims by Bellare and Rogaway [3], Shoup [41] showed that OAEP is black-box unprovable solely based on the one-wayness of the underlying TDP, even in the random oracle model. Later this result got complemented in [18] by showing that, in the random oracle model, one needs to assume the stronger security assumption of *partial-domain* one-wayness to prove OAEP secure.

In a series of two papers [6,7], Boldyreva and Fischlin considered the question of instantiating the random oracles in OAEP (and other scenarios) by specific candidates of standard-model hash functions, such as POWHFs and VRFs. In particular, they showed that POWHFs or VRFs cannot generically instantiate the random oracles in OAEP, no matter which TDP is used [6]. Although it follows immediately from our generic impossibility result that one cannot *prove the security* of OAEP (or any other padding-based scheme) assuming the hash functions are instantiated with such primitives, the result of [6] (for the special case of OAEP) is stronger as they show concrete instantiations which actually *make OAEP insecure.* On the positive side, [7] show that if the hash functions in OAEP are instantiated using non-malleable pseudorandom generators, then the resulting OAEP scheme is proved non-malleable. However, their security definition of non-malleability is actually weaker than what is commonly called NM-CPA security [4] which is the reason why their positive instantiation result does not contradict our separation results.[4]

Brown [9] showed that RSA-OAEP cannot be proven CCA secure under a certain class of security reductions denoted as "key-preserving" black-box reductions, i.e., reductions that are restricted to make oracle calls to the CCA adversary with respect to the *same RSA instance* that they are given as challenge. Similar results (for the class of "single-key factoring-based encryption schemes") were independently obtained by Paillier and Villar [35]. Our impossibility results seem more general since we can exclude any black-box reduction (and not only key-preserving ones) from ideal trapdoor permutations (and not only one-wayness). Furthermore, the results from [9,35] do not allow the scheme's public key to contain any additional information beyond the RSA/factoring instance. In particular, their results do not exclude the possibility to securely instantiate OAEP from standard *keyed* hash functions, such as POWHFs and VRFs.

2 Preliminaries

2.1 Notation

If x is a string, then $|x|$ denotes its length, while if S is a set then $|S|$ denotes its size. If $k \in \mathbb{N}$ then 1^k denotes the string of k ones. If x is a string, then $[x]_\ell$ denote the ℓ left-most bits of x. If S is a set then $s \leftarrow_R S$ denotes the operation of picking an element s of S uniformly at random. We write $\mathsf{A}(x, y, \ldots)$ to indicate that A is an algorithm (i.e., a Turing Machine) with inputs $x, y, \ldots$ and by $z \leftarrow_R \mathsf{A}(x, y, \ldots)$ we denote the operation of running A with inputs $(x, y, \ldots)$ and letting z be the output. We write $\mathsf{A}^{\mathcal{O}_1, \mathcal{O}_2, \ldots}(x, y, \ldots)$ to indicate that A is an algorithm with inputs $x, y, \ldots$ and access to oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$. With PT we denote polynomial time and with PPT we denote probabilistic polynomial time.

2.2 Public-Key Encryption

A *public key encryption* scheme $\mathsf{PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\{0,1\}^\mu$ (where $\mu = \mu(k)$ is some polynomial in k) consists of three PT algorithms, of which the first two, Kg and Enc, are probabilistic and the last one, Dec, is deterministic. Public/secret keys for security parameter $k \in \mathbb{N}$ are generated using

[4] The non-malleability definitions of [7] only consider relations over one ciphertext and not over polynomial-size ciphertext vectors. Using the characterization of [4], one can actually prove that this is an even weaker notion than IND-CPA security where the adversary is allowed to additionally make one single decryption query (also called 1-bounded IND-CCA-security [14]). Our results show that full NM-CPA security [4] is not black-box achievable. We remark that the security notion of [7] is still meaningful since it prevents Bleichenbacher's attack on PKCS [5].

$(pk, sk) \leftarrow_R \mathsf{Kg}(1^k)$. Given such a key pair, a message $m \in \{0,1\}^\mu$ is encrypted by $c \leftarrow_R \mathsf{Enc}(pk, m)$; a ciphertext is decrypted by $m \leftarrow \mathsf{Dec}(sk, c)$, where possibly Dec outputs a special reject symbol $\perp$ to denote an invalid ciphertext. For correctness, we require that for all $k \in \mathbb{N}$, all messages $m \in \{0,1\}^\mu$, it must hold that $\Pr[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m] = 1$, where the probability is taken over the above randomized algorithms and $(pk, sk) \leftarrow_R \mathsf{Kg}(1^k)$. Sometimes we also associate a randomness space $\{0,1\}^\rho$ to PKE (where $\rho = \rho(k)$ is some polynomial in k), to denote that the randomness used in Enc is picked uniformly from $\{0,1\}^\rho$.

We recall the standard security notion of *chosen ciphertext security* [38] of a PKE scheme which is defined through the following advantage function of an adversary A.

$$\mathbf{Adv}^{\mathsf{cca}}_{\mathsf{PKE}}(\mathsf{A}) \;=\; \left| \Pr\left[b = b' : \begin{array}{c} (pk, sk) \leftarrow_R \mathsf{Kg}(1^k) \\ (m_0, m_1, \mathit{state}) \leftarrow_R \mathsf{A}^{\mathcal{O}(sk,\cdot)}(pk) \\ b \leftarrow_R \{0,1\}\,;\; c^* \leftarrow_R \mathsf{Enc}(pk, m_b) \\ b' \leftarrow_R \mathsf{A}^{\mathcal{O}(sk,\cdot)}(c^*, \mathit{state}) \end{array} \right] - \frac{1}{2} \right| ,$$

where $\mathcal{O}(sk, c) = \mathsf{Dec}(sk, c)$, in the second phase ("guess phase"), A is not allowed to query $\mathcal{O}(sk, \cdot)$ for the challenge ciphertext c^*, and we require that m_0 and m_1 are of the same length. *state* is some arbitrary state information. PKE scheme PKE is said to be chosen ciphertext secure ($\mathsf{IND\text{-}CCA}$ secure) if the advantage function $\mathbf{Adv}^{\mathsf{cca}}_{\mathsf{PKE}}(\mathsf{A})$ is a negligible function in k for all PPT adversaries A.

3 Ideal Trapdoor Permutations

In this section we introduce the notion of an ideal trapdoor permutation. Intuitively, this is a trapdoor permutation that inherits all security properties of a *uniformly random* permutation. This includes (partial-domain) one-wayness, claw-freeness, and other non-standard notions.

3.1 Trapdoor Permutations

Definition 1. *A triple of PPT algorithms* $(\mathsf{Tdg}, \mathsf{F}, \mathsf{F}^{-1})$ *implements a trapdoor permutation if* Tdg *is probabilistic and on input* 1^k *generates an evaluation/trapdoor key-pair* $(ek, td) \leftarrow_R \mathsf{Tdg}(1^k)$*,* $\mathsf{F}(ek, \cdot)$ *implements a permutation* $f_{ek}(\cdot)$ *over* $\{0,1\}^k$ *and* $\mathsf{F}^{-1}(td, \cdot)$ *implements its inverse* $f_{ek}^{-1}(\cdot)$*.*

Note that the above definition is only functional, and does not impose any security property. The most common security property for TDPs is one-wayness, i.e., one requires that it is hard to invert the permutation on random inputs without knowing the trapdoor. We will consider a much broader class of security properties. Using the notion of *δ-hard games* (generalizing a notion previously used in [17]) we can capture any game-based cryptographic hardness experiment involving permutations.

3.2 Hard Games

A *game* is defined by a PPT algorithm G (the challenger). This G can interact with another PPT algorithm A (the adversary) by exchanging messages over a shared communication tape. Eventually, G outputs a decision bit d. We denote one execution of game G with adversary A by $d \leftarrow_R \mathbf{Exp}^{\mathsf{G}}(\mathsf{A})$ and say that A *wins game* G if $d = 1$.

A game G as above defines a *δ-hard game*, where $0 \leq \delta < 1$, if G is a PPT algorithm (in the security parameter $k \in \mathbb{N}$), and further no PPT adversary A can win the game when both, G and A, have oracle access to $t = t(k)$ uniform random permutations $\tau_1, \ldots, \tau_t$ over $\{0,1\}^k$ (here $t(\cdot)$ is implicitly defined by G) with probability significantly better than δ. Formally, we define the advantage function of an adversary A in game G as

$$\mathbf{Adv}^{\mathsf{G}}_{\mathrm{RP}}(\mathsf{A}, k) = \Pr\left[d = 1 \ : \ d \leftarrow_R \mathbf{Exp}^{\mathsf{G}^{\tau_1(\cdot),\ldots,\tau_t(\cdot)}}(\mathsf{A}^{\tau_1(\cdot),\ldots,\tau_t(\cdot)}(1^k))\right] . \quad (1)$$

The RP in $\mathbf{Adv}^{\mathsf{G}}_{\mathrm{RP}}(\mathsf{A}, k)$ stands for "random permutation". Let us stress once more, that in the above game we do not give G (or A) access to the inversion oracles $\tau_1^{-1}(\cdot), \ldots, \tau_t^{-1}(\cdot)$.

Definition 2. *Game* G *is δ-hard for some* $0 \leq \delta \leq 1$*, if for all PPT adversaries* A*,* $\mathbf{Adv}^{\mathsf{G}}_{\mathrm{RP}}(\mathsf{A}, k) - \delta$ *is negligible in k. The hardness of a game* G*, denoted* $\delta(\mathsf{G})$*, is the smallest δ such that* G *is δ-hard.*

Table 1 shows examples of hard games with the corresponding upper bound on the advantage function $\mathbf{Adv}^{\mathsf{G}}_{\mathrm{RP}}(\mathsf{A})$. Typical values for $\delta(\mathsf{G})$ are 0 and 1/2, the latter comes up in games where the adversary just has to distinguish two cases, and thus can trivially win with probability 1/2 by a random guess. The notion of δ-hard games generalizes the hard-games used in [17], which only covered the case $\delta = 0$, but the results (and proofs) from [17] can easily be adapted to cover general δ-hard games.

3.3 Ideal Trapdoor Permutations

A trapdoor permutation $\mathsf{TDP} = (\mathsf{Tdg}, \mathsf{F}, \mathsf{F}^{-1})$ is secure for hard game G if Definition 2 is satisfied even if the random permutations used to define (1) are replaced with instantiations of TDP. Let

$$\mathbf{Adv}^{\mathsf{G}}_{\mathsf{TDP}}(\mathsf{A}, k) := \Pr\left[d = 1 : \begin{array}{l} \text{for } i = 1, \ldots, t : \quad (ek_i, td_i) \leftarrow_R \mathsf{Tdg}(1^k)\ ; \\ d \leftarrow_R \mathbf{Exp}^{\mathsf{G}^{\mathsf{F}(ek_1,\cdot),\ldots,\mathsf{F}(ek_t,\cdot)}}(\mathsf{A}(ek_1, \ldots, ek_t)) \end{array}\right]$$

Definition 3. *A trapdoor permutation* TDP *is* secure for game G *if for all PPT adversaries* A*,* $\mathbf{Adv}^{\mathsf{G}}_{\mathsf{TDP}}(\mathsf{A}, k) - \delta(\mathsf{G})$ *is negligible in k. Furthermore,* TDP *is an* ideal trapdoor permutation *if it is secure for all games.*

Table 1. Examples of hard games. In the last column, q is an upper bound on the number of queries the adversary A makes to each of its oracles $\tau_1(\cdot), \ldots, \tau_t(\cdot)$. Note that if $q = q(k)$ is polynomially bounded, then in all cases $\mathbf{Adv}^{\mathsf{G}}_{\mathrm{RP}}(\mathsf{A}, k) - \delta(\mathsf{G})$ is negligible (even exponentially small), thus the games are δ-hard (with $\delta = 0$ or $\delta = 1/2$ as indicated in the table).

Game	Description of game $\mathsf{G}^{\tau_1,\ldots,\tau_t}$			Advantage
	Experiment	Winning cond.	$\delta(\mathsf{G})$	$\mathbf{Adv}^{\mathsf{G}}_{\mathrm{RP}}(\mathsf{A}, k)$
One-wayness (OW)	$x \leftarrow_R \{0,1\}^k$; $y \leftarrow \tau_1(x)$; $x' \leftarrow_R \mathsf{A}^{\tau_1}(y)$	$x' = x$	0	$\leq (q+1)/2^k$
Partial-domain OW	$x \leftarrow_R \{0,1\}^k$; $y \leftarrow \tau_1(x)$; $x' \leftarrow_R \mathsf{A}^{\tau_1}(y)$	$x' = [x]_\ell$	0	$\leq \frac{q}{2^k} + \frac{2^{k-\ell}}{2^k - q}$
Claw-freeness	$(x_1, x_2) \leftarrow_R \mathsf{A}^{\tau_1,\tau_2}(1^k)$	$\tau_1(x_1) = \tau_2(x_2)$	0	$\leq q^2/2^k$
Pseudorandomness	$x \leftarrow_R \{0,1\}^k$; $y_1 \leftarrow \tau_1(x)$; $b \leftarrow_R \{0,1\}$ $y_{2,0} \leftarrow_R \{0,1\}^k$; $y_{2,1} \leftarrow \tau_2(x)$ $b' \leftarrow_R \mathsf{A}^{\tau_1,\tau_2}(y_1, y_{2,b})$	$b' = b$	$\frac{1}{2}$	$\leq \frac{1}{2} + \frac{q}{2^k}$
$t(k)$-correlated input OW	$x \leftarrow_R \{0,1\}^k$; for $i = 1, \ldots, t : y_i \leftarrow \tau_i(x)$ $x' \leftarrow_R \mathsf{A}^{\tau_1,\ldots,\tau_t}(y_1, \ldots, y_t)$	$x' = x$	0	$\leq \frac{1+t\cdot q}{2^k}$

3.4 On the Power of Ideal Trapdoor Permutations

Ideal trapdoor permutations are a quite powerful primitive as most of the known cryptographic primitives can be efficiently instantiated in a *black-box way* from ideal trapdoor permutations. Examples include: collision-resistant hashing (from claw-freeness [15]); pseudorandom generators and functions (from one-wayness, using the Goldreich-Levin predicate [23]); perfectly one-way probabilistic hash functions (from one-wayness [11]); IND-CCA secure public-key encryption (from $t(k)$-correlated input one-wayness [39] for $t(k) = 2k+1$); bit commitment (from one-wayness), digital signatures, oblivious transfer [25], trapdoor commitments, and many more.

On the other hand, it is easy to see (see [17] for a proof) that ideal trapdoor permutations do not exists. However, keep in mind that we are aiming for an impossibility result: we rule out the existence of padding-based encryption schemes whose security can be black-box reduced to ideal TDPs. This will immediately also rule out this possibility for any TDP which are only hard for a "realistic" subset of all hard games.

4 Padding Schemes for Encryption

In this section we introduce the notion of padding-schemes and padding-based encryption schemes. Many efficient and widely employed encryption schemes, in particular OAEP and its variants, are padding-based.

4.1 Definitions

Let k, μ, ρ be three integers such that $\mu + \rho \leq k$. A *padding scheme Π* consists of two mappings $\pi : \{0,1\}^\mu \times \{0,1\}^\rho \to \{0,1\}^k$ and $\hat{\pi} : \{0,1\}^k \to \{0,1\}^\mu \cup \{\bot\}$ such that π is injective and the following consistency requirement is fulfilled:

$$\forall m \in \{0,1\}^\mu, r \in \{0,1\}^\rho \;:\quad \hat{\pi}(\pi(m \,\|\, r)) = m\,.$$

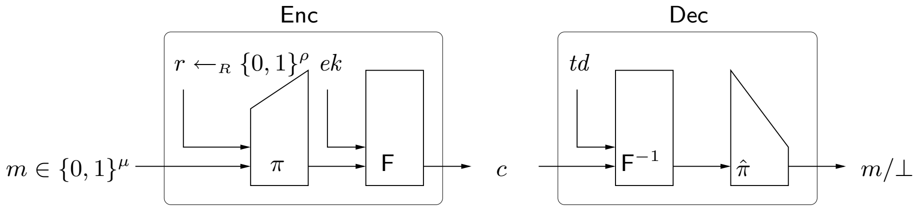

Fig. 1. Padding based encryption scheme from a trapdoor permutation $\mathsf{TDP} = (\mathsf{Tdg}, \mathsf{F}, \mathsf{F}^{-1})$

A pair of oracle circuits $\Pi = (\pi, \hat{\pi})$ is a *TDP-based padding scheme*, if $\pi^{\mathsf{TDP}}, \hat{\pi}^{\mathsf{TDP}}$ is a padding scheme for any trapdoor permutation $\mathsf{TDP} = (\mathsf{Tdg}, \mathsf{F}, \mathsf{F}^{-1})$.

Definition 4. *Let $\mu, \rho : \mathbb{N} \to \mathbb{N}$ be functions (defining a message and randomness space, respectively) where $\mu(k) + \rho(k) \leq k$ for all k. We call a triple of efficient oracle algorithms* $\mathsf{PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ *a* padding-based encryption scheme *with message space μ and randomness space ρ, if for any trapdoor permutation* $\mathsf{TDP} = (\mathsf{Tdg}, \mathsf{F}, \mathsf{F}^{-1})$*:*

- *(Key Generation) Given the security parameter k,* $\mathsf{Kg}^{\mathsf{TDP}}(k)$ *returns a public key $pk = (ek, \Pi)$ and a secret key $sk = (td, \hat{\pi})$, where ek and td are computed by running $(ek, td) \leftarrow_R$* $\mathsf{Tdg}(1^k)$ *and $\Pi = (\pi, \hat{\pi})$ is a TDP based padding scheme.*
- *(Encryption) Given the public-key pk and a message $m \in \{0,1\}^\mu$,* $\mathsf{Enc}^{\mathsf{TDP}}$ *(pk, m) returns a ciphertext computed as $c = f_{ek}(\pi^{\mathsf{TDP}}(m \,\|\, r)) \in \{0,1\}^k$, for $r \leftarrow_R \{0,1\}^\rho$.*
- *(Decryption) Given the secret-key sk and a ciphertext $c \in \{0,1\}^k$,* $\mathsf{Dec}^{\mathsf{TDP}}$ *(sk, c) returns $m = \hat{\pi}^{\mathsf{TDP}}(f_{ek}^{-1}(c)) \in \{0,1\}^\mu \cup \{\bot\}$.*[5]

See Fig. 1 for a graphical illustration of Definition 4. Note that it is not further specified how $\mathsf{Dec}^{\mathsf{TDP}}(sk, \cdot)$ behaves on invalid ciphertexts, i.e., on a c which is not in the range of $\mathsf{Enc}^{\mathsf{TDP}}(pk, \cdot)$. $\mathsf{Dec}^{\mathsf{TDP}}(sk, \cdot)$ may return the special reject symbol $\bot$, or output any message, depending on the definition of $\hat{\pi}$.

Also note that we include $\hat{\pi}$ as part of the public-key, even though $\hat{\pi}$ is not required in the encryption algorithm at all. This apparently minor detail is important as we will explain later in Section 5. Basically, by including $\hat{\pi}$ in pk, we make sure that the padding scheme $\Pi = (\pi, \hat{\pi})$ *itself* is not already an IND-CCA secure encryption scheme.

4.2 Examples

Our definition of padding-based encryption schemes is quite broad and contains many known encryption schemes, including OAEP and its variants. Fig. 2 contains the description of π for the underlying padding scheme Π, for some

[5] Recall that we use $f_{ek}(\cdot) := \mathsf{F}(ek, \cdot)$ and $f_{ek}^{-1}(\cdot) := \mathsf{F}^{-1}(td, \cdot)$, where td is the trapdoor corresponding to ek.

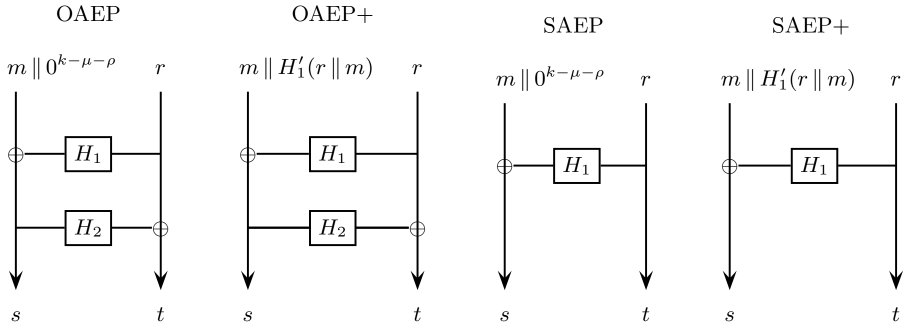
OAEP
$m \,\|\, 0^{k-\mu-\rho}$
r
H_1
H_2
s
t
OAEP+
$m \,\|\, H_1'(r \,\|\, m)$
r
H_1
H_2
s
t
SAEP
$m \,\|\, 0^{k-\mu-\rho}$
r
H_1
s
t
SAEP+
$m \,\|\, H_1'(r \,\|\, m)$
r
H_1
s
t

Fig. 2. Examples of the mapping π for important padding schemes. Here $H_1 : \{0,1\}^\rho \to \{0,1\}^{k-\rho}$, $H_1' : \{0,1\}^{k-\rho} \to \{0,1\}^{k-\mu-\rho}$, $H_2 : \{0,1\}^{k-\rho} \to \{0,1\}^\rho$ are hash functions (whose circuit description are contained in the description of π).

important schemes. The corresponding inverse $\hat{\pi}$ can be easily derived from Π's consistency property. For example, in OAEP, $\hat{\pi}(s \,\|\, t)$ is defined as $\perp$ or m, depending whether $w = 0^{k-\mu-\rho}$, or not, where $m \,\|\, w = s \oplus H_1(H_2(s) \oplus t))$. Other examples of padding-based encryption schemes not contained in Fig. 2 include OAEP++ [29], PSS-E [13], PSP2 S-Pad [16], full-domain permutation encryption [36], 3-round OAEP [36,37], 4-round OAEP [1], and the schemes from [12,30,34].

5 Uninstantiability from any Ideal Trapdoor Permutation

The following main theorem states that there does not exist a padding-based encryption scheme $\mathsf{PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ such that any adversary who breaks the $\mathsf{IND\text{-}CCA}$ security of $\mathsf{PKE}^{\mathsf{TDP}}$ can be used (in a black-box way) to break the security of TDP as an ideal TDP.

Theorem 5. *There is no black-box reduction from an ideal trapdoor permutation to a chosen-ciphertext (*$\mathsf{IND\text{-}CCA}$*) secure padding-based encryption scheme.*

Proof. Fix a padding-based encryption scheme $\mathsf{PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mu(k)$ and randomness space $\rho(k)$. If the message plus the randomness space is too small, i.e., $\mu(k)+\rho(k) \in O(\log k)$ or equivalently $2^{\mu(k)+\rho(k)} \in poly(k)$, then $\mathsf{PKE}^{\mathsf{TDP}}$ is trivially insecure to matter what TDP we use, as an adversary in the $\mathsf{IND\text{-}CCA}$ experiment can trivially decrypt the challenge ciphertext by doing an exhaustive search over all possible plaintext and randomness pairs $(m, r) \in \{0,1\}^{\mu(k)+\rho(k)}$, without even using the decryption oracle. Such an adversary runs in $2^{\mu(k)+\rho(k)} \in poly(k)$ time units and has success probability 1 in breaking the $\mathsf{IND\text{-}CCA}$ security of PKE.

So from now on we can assume (w.l.o.g.) $2^{\mu(k)+\rho(k)} \notin poly(k)$. Following [26, Proposition 1], as to rule out black-box reductions, it is enough to prove that there exist two oracles T and B such that the following holds:

1. T can be used to implement a trapdoor permutation, i.e., there exists a triple of oracle PPT algorithms $\mathsf{TDP} = (\mathsf{Tdg}, \mathsf{F}, \mathsf{F}^{-1})$ such that $\mathsf{TDP}^{\mathsf{T}}$ implements a trapdoor permutation (in the sense of Definition 1).
2. Relative to the oracles T and B, $\mathsf{TDP}^{\mathsf{T}}$ is an *ideal trapdoor permutation.* That is, $\mathsf{TDP}^{\mathsf{T}}$ is an ideal trapdoor permutation as in Definition 3 even if the adversary is given access to the oracles T and B.
3. For any padding-based encryption schemes PKE: relative to the oracles T and B, $\mathsf{PKE}^{\mathsf{TDP}^{\mathsf{T}}}$ is not $\mathsf{IND\text{-}CCA}$ secure.

We first define the oracle T and show that it satisfies point 1 (Lemma 6) and a relaxed version of point 2 where we do not consider the breaking oracle B (Lemma 7). We then define the breaking oracle B, and prove that points 2 and 3 hold (Lemma 10 and Lemma 8, respectively).

Definition of T. (Oracle used to implement the trapdoor permutation.) Let $\mathcal{P}_k$ denote the set of all permutations over $\{0,1\}^k$. For any $k \in \mathbb{N}$, choose $2^k + 1$ permutations $f_{k,0}, \ldots, f_{k,2^k-1}$ and g_k from $\mathcal{P}_k$ uniformly at random. $\mathsf{T} = (\mathsf{T}_1, \mathsf{T}_2, \mathsf{T}_3)$ is defined as follows.

- $\mathsf{T}_1(td)$ returns $g_k(td)$, where $k = |td|$. (Convert trapdoor into public key)
- $\mathsf{T}_2(ek, x)$ with $|ek| = |x|$ returns $f_{k,ek}(x)$, where $k = |x|$. (Evaluation)
- $\mathsf{T}_3(td, y)$ with $|td| = |y|$ returns $f^{-1}_{k,g_k(td)}(y)$, where $k = |y|$. (Inversion)

Lemma 6. *There is a PPT* TDP *such that* $\mathsf{TDP}^{\mathsf{T}}$ *implements a trapdoor permutation.*

Proof. We implement a trapdoor permutation $\mathsf{TDP}^{\mathsf{T}} = (\mathsf{Tdg}, \mathsf{F}, \mathsf{F}^{-1})$ as follows.

- $\mathsf{Tdg}(1^k)$ first picks a random trapdoor $td \leftarrow_R \{0,1\}^k$, then computes the corresponding key $ek = \mathsf{T}_1(td)$, and outputs (ek, td).
- $\mathsf{F}(ek, x)$ returns $\mathsf{T}_2(ek, x)$.
- $\mathsf{F}^{-1}(td, y)$ returns $\mathsf{T}_3(td, y)$.

According to Definition 1 this implements a trapdoor permutation. ▲

Lemma 7. $\mathsf{TDP}^{\mathsf{T}}$ *is an ideal trapdoor-permutation with probability* 1 *(over the choice of* T*).*

Proof. Consider any δ-hard game G (assume for now that $t = 1$, i.e., the game involves just one permutation). Recall that the advantage of an adversary A in winning the game is

$$p_{real} := \Pr\left[d = 1 : (ek, td) \leftarrow_R \mathsf{Tdg}(1^k)\ ;\ d \leftarrow_R \mathbf{Exp}^{\mathsf{G}^{\mathsf{F}(ek,\cdot)}}(\mathsf{A}^{\mathsf{TDP}^{\mathsf{T}}}(ek))\right] \quad (2)$$

Now let T_{ek} denote T, where $f_{k,ek}(\cdot)$ is replaced with a "fresh" random permutation $\tau(\cdot)$. Let

$$p_{rand} := \Pr\left[d = 1 : \begin{array}{c} (ek, td) \leftarrow_R \mathsf{Tdg}(1^k)\ ;\ \tau(\cdot) \leftarrow_R \mathcal{P}_k\ ; \\ d \leftarrow_R \mathbf{Exp}^{\mathsf{G}^{\tau(\cdot)}}(\mathsf{A}^{\mathsf{TDP}^{\mathsf{T}_{ek}}}(ek)) \end{array}\right] \quad (3)$$

By definition of a δ-hard game we have $p_{rand} - \delta \leq negl(k)$. Below we will show that $|p_{real} - p_{rand}| = negl(k)$, and thus $p_{real} - \delta \leq negl(k)$. As this holds for any δ-hard game, $\mathsf{TDP}^{\mathsf{T}}$ is an ideal TDP.

To show that $|p_{real} - p_{rand}| = negl(k)$, we will argue that A cannot distinguish the two games considered in (2) and (3) with non-negligible probability. First, as a random permutation is one way almost certainly (see [19] for a proof), an efficient adversary will find the trapdoor td given $ek = \mathsf{T}_1(td)$ only with exponentially small probability. (In particular, A will almost certainly never query the inversion oracle F^{-1} with trapdoor td.) Second, one can show (we omit the proof) that a permutation chosen at random from a set of exponentially many permutations (in particular, $f_{k,ek}$ for $ek \in \{0,1\}^k$) can be distinguished from a randomly chosen permutation by a polynomial size circuit only with an exponentially small advantage. This two points imply that the distribution of d in experiments (2) and (3) have exponentially small statistical distance, and thus $|p_{real} - p_{rand}|$ is exponentially small. ▲

Definition of B. (Oracle used to break the encryption scheme.) The oracle B is defined below. It takes two types of inputs. On input the description of a padding-based encryption scheme, B outputs a vector of challenge ciphertexts. On input a padding-based encryption scheme with a vector of plaintexts, B checks if those plaintexts correspond to the ciphertexts it would ask on a query of the first type. If this is the case, B outputs the trapdoor of the encryption scheme.

It is quite obvious how the oracle breaks the security of any padding based encryption scheme (with message space $\mu(\cdot)$ and randomness space $\rho(\cdot)$). It is less obvious that this oracle will not be of much use in winning any hard game. In order to prove this we will show that basically the only possibility of making a query of the second type to B containing the correct plaintexts (and thus receiving the trapdoor), is by already knowing the trapdoor. Note that the chosen-ciphertext security game itself cannot be formulated as a hard game since the challenger has to know the trapdoor to answer decryption queries. It is exactly this additional power of the CCA challenger (as compared to the challenger in a hard game) that allows an adversary interacting with the CCA challenger to exploit the answers of the breaking oracle.

We now formally define B. (Like all other oracles we consider, this oracle is stateless, and thus will return the same output when queried twice on an identical input.)

1. B, on an input of the form (k, ek, Π), where $k \in \mathbb{N}$, $ek \in \{0,1\}^k$, and $\Pi = \{\pi, \hat{\pi}\}$ (where $\pi : \{0,1\}^{\mu(k)+\rho(k)} \to \{0,1\}^k, \hat{\pi} : \{0,1\}^k \to \{0,1\}^{\mu(k)}$ is a TDP-based padding scheme), outputs a vector of challenge ciphertexts $[c_1, \ldots, c_{4k}]$, computed as
$$c_i = f_{k,ek}(\pi^{\mathsf{T}}(m_i \,\|\, r_i)), \tag{4}$$
for randomly chosen $m_i \leftarrow_R \{0,1\}^{\mu(k)}$ and $r_i \leftarrow_R \{0,1\}^{\rho(k)}$. Note that if (ek, Π) is the public-key of some padding based encryption scheme, then c_i is a proper ciphertext of message m_i for this public key.
2. B, on input $(k, ek, \Pi, [m'_1, \ldots, m'_{4k}])$, checks if $[m'_1, \ldots, m'_{4k}] = [m_1, \ldots, m_{4k}]$, where the m_i are the plaintext messages chosen by B on input (k, ek, π) as

described above. If the two plaintext vectors are identical, it returns $td := g_k^{-1}(ek)$, the trapdoor of the trapdoor permutation corresponding to ek from the input of B. Otherwise, it outputs $\perp$.

Lemma 8. *There is a PPT* A *such that* $\mathsf{A}^{\mathsf{T},\mathsf{B}}$ *breaks the* $\mathsf{IND\text{-}CCA}$ *security of* $\mathsf{PKE}^{\mathsf{TDP}^{\mathsf{T}}}$*.*

Proof. Adversary A in the $\mathsf{IND\text{-}CCA}$ experiment first obtains a public key which contains ek for the trapdoor permutation and the padding scheme Π. Next, it queries B on input (k, ek, π) to obtain the vector of challenge ciphertexts $[c_1, \ldots, c_{4k}]$. Next, A asks its decryption oracle $\mathcal{O}$ for the plaintexts $m_i' = \mathcal{O}(c_i)$, for $i = 1, \ldots, 4k$. Finally, A makes the query $(k, ek, \pi, [m_1', \ldots, m_{4k}'])$ to B and obtains the trapdoor td which can be used to decrypt the challenge ciphertext c^* (and hence distinguish). We have just shown that A is a PPT algorithm with $\mathbf{Adv}^{\mathsf{cca}}_{\mathsf{PKE}}(\mathsf{A}) = 1$. ▲

Let us stress that it is in the proof above where we exploit the fact (as mentioned in the paragraph before Section 4.2) that any ciphertext can be decrypted given the public key $pk = (ek, \Pi = (\pi, \hat{\pi}))$ and trapdoor td. In particular, this means that $\hat{\pi}$ is efficiently computable.

By the following lemma, the breaking oracle B can be efficiently simulated in some settings. In particular, this will be the case for δ-hard games, and thus —as stated in Lemma 10 below— we are able to generalize Lemma 7 to hold relative to B.

Lemma 9. *For* $i = 1, \ldots, t$ *let* $(ek_i, td_i) \leftarrow_R \mathsf{Tdg}^{\mathsf{T}}(1^k)$*. Consider any oracle circuit* C *where* $\mathsf{C}^{\mathsf{T},\mathsf{B}}(ek_1, \ldots, ek_t)$ *makes at most* $q(k)$ *oracle queries where* $q(k) \leq 2^{\mu(k)+\rho(k)}/2$*. Then there exists an efficient simulator* S *such that the output of* $\mathsf{C}^{\mathsf{T},\mathsf{B}}(ek_1, \ldots, ek_t)$ *and* $\mathsf{C}^{\mathsf{T},\mathsf{S}^{\mathsf{T}}}(ek_1, \ldots, ek_t)$ *is exponentially close (i.e., the statistical distance is* $\leq 2^{-k/2}$*). Here* S *will get to see all* $\mathsf{F}^{-1}(\cdot,\cdot)$ *(and only those) queries made by* C *to the* T *oracle.*

Before we prove this lemma, let us see how we can use it to generalize Lemma 7.

Lemma 10. $\mathsf{TDP}^{\mathsf{T}}$ *is an ideal trapdoor-permutation with probability* 1 *(over the choice of* T*)* even relative to B.

Proof. By Lemma 7 we know that $\mathsf{TDP}^{\mathsf{T}}$ is an ideal TDP relative to T only. Now consider any game G and any adversary A, and let $q(k) \in poly(k)$ denote the total number of oracle queries made by G and A. As we assume $2^{\mu(k)+\rho(k)} \notin poly(k)$ we have $q(k) \leq 2^{\mu(k)+\rho(k)}/2$ for all but finitely many k. Thus, by Lemma 9, we can simulate B efficiently, in particular

$$\mathbf{Adv}^{\mathsf{G}}_{\mathsf{TDP}^{\mathsf{T}}}(\mathsf{A}^{\mathsf{T},\mathsf{B}}, k) = \mathbf{Adv}^{\mathsf{G}}_{\mathsf{TDP}^{\mathsf{T}}}(\mathsf{A}^{\mathsf{T},\mathsf{S}^{\mathsf{T}}}, k) \pm negl(k)\ . \tag{5}$$

As S is efficient, we can let its computation be done by the adversary: let $\hat{\mathsf{A}}$ denote the adversary A, but which simulates the S^{T} queries itself. (Note that this is possible, as G never makes F^{-1} queries, and S needs to see only those.)

$$\mathbf{Adv}^{\mathsf{G}}_{\mathsf{TDP}^{\mathsf{T}}}(\mathsf{A}^{\mathsf{T},\mathsf{S}^{\mathsf{T}}}, k) = \mathbf{Adv}^{\mathsf{G}}_{\mathsf{TDP}^{\mathsf{T}}}(\hat{\mathsf{A}}^{\mathsf{T}}, k) \tag{6}$$

As by Lemma 7, $\mathsf{TDP}^{\mathsf{T}}$ is an ideal TDP relative to T, $\hat{\mathsf{A}}$ cannot win the δ-hard game G with advantage more than

$$\mathbf{Adv}^{\mathsf{G}}_{\mathsf{TDP}^{\mathsf{T}}}(\hat{\mathsf{A}}^{\mathsf{T}}, k) \leq \delta \pm negl(k) \ . \tag{7}$$

By (5)-(7), we obtain

$$\mathbf{Adv}^{\mathsf{G}}_{\mathsf{TDP}^{\mathsf{T}}}(\mathsf{A}^{\mathsf{T},\mathsf{B}}, k) \leq \delta \pm negl(k) \ ,$$

which implies that G is a δ-hard game, even relative to oracle B. ▲

Proof (Lemma 9)

We only consider the case $t = 1$, i.e., where there is only one single key ek. The generalization to $t \geq 1$ is straight forward. The simulator S is defined as follows.

On input of a query (k, ek, Π), S samples $m_1, \ldots, m_{4k}$ and $r_1, \ldots, r_{4k}$ and outputs $[c_1, \ldots, c_{4k}]$ computed as in (4). This query, as well as the values m_i, r_i are stored. (If the query is repeated, the same answer is given). Note that the output of this query was computed exactly the same way as B would.

On input a query $(k, ek, \Pi, [m'_1, \ldots, m'_{4k}])$, S first checks if the query (k, ek, Π) was already made.

- If this is not the case, S outputs $\perp$. Note that this almost perfectly simulates B, as B would also output $\perp$ in this case, except if by chance all the m'_i correspond to the m_i that B would use on input (k, ek, Π). (The probability of this event is $2^{-\mu(k)4k}$ what we ignore.)
- If C made the query (k, ek, Π), let $[m_1, \ldots, m_{4k}]$ denote the message vector used by S to answer this query. If $[m_1, \ldots, m_{4k}] \neq [m'_1, \ldots, m'_{4k}]$ then output $\perp$. Note that this perfectly simulates B.
- Otherwise, if $[m_1, \ldots, m_{4k}] = [m'_1, \ldots, m'_{4k}]$, S checks for all F^{-1} queries (td, x) made by C, if $ek \overset{?}{=} \mathsf{Tdg}(td)$. If this is the case, S outputs this trapdoor td, exactly as B would. If C never used the trapdoor td corresponding to ek in a query, S outputs "fail". Note that this is the only case where the answer from S differs from what B would output.

To prove that S can almost perfectly simulate B, it remains to upper bound the probability that an efficient adversary C can make S^{T} output "fail" in the last item above.

S outputs "fail", if C makes a query (k, ek, Π) to S^{T}, then receives $4k$ ciphertexts $[c_1, \ldots, c_{4k}]$ computed as $c_i = f_{k,ek}(\pi^{\mathsf{T}}(m_i \,\|\, r_i))$ for random m_i, r_i, and then correctly computes (or guesses) all the m_i without ever inverting $f_{k,ek}$ (i.e., never using the F^{-1} oracle with the trapdoor td where $ek = \mathsf{Tdg}(td)$). To analyze the probability that S outputs "fail", consider the following three sets.

- Let $\mathcal{R} = \{\pi^{\mathsf{T}}(m \,\|\, r) \ : \ m \,\|\, r \in \{0,1\}^{\mu(k)+\rho(k)}\}$ denote the set of possible inputs on which one must evaluate $f_{k,ek}$ in order to compute a ciphertext. As π^{T} is injective, $|\mathcal{R}| = 2^{\mu(k)+\rho(k)}$.
- Let $\mathcal{Y} = \{\pi^{\mathsf{T}}(m_i \,\|\, r_i) \ : \ i = 1, \ldots, 4k\}$ denote the set of inputs to $f_{k,ek}$ one must make in order to compute the c_i's. As π^{T} is injective, $|\mathcal{Y}| = 4k$.

- Let $\mathcal{X} \subset \mathcal{R}$ denote all the set of queries that C made to $f_{k,ek}$ (before and after seeing the challenge vector $[c_1, \ldots, c_{4k}]$).

Let *miss* $:= |\mathcal{Y} \setminus \mathcal{X}|$ denote the number of preimages of the c_i's which C did not query. As the preimages of the c_i are uniformly in $\mathcal{R}$, and $f_{k,ek}$ is a random permutation, *miss* is a random variable, which can be sampled as follows: from a set $\mathcal{R}$ of (super-polynomial) size $2^{\mu(k)+\rho(k)}$, sample a random subset $\mathcal{X}$ of (polynomial size) $q(k)$ and a random subset $\mathcal{Y}$ of size $4k$ and let *miss* denote the number of elements in $\mathcal{Y}$ which are not in $\mathcal{X}$. The expected value of *miss* is $(1 - q(k)/2^{\mu(k)+\rho(k)})4k$, which, as $q(k) \leq 2^{\mu(k)+\rho(k)}/2$, is at least $4k/2 = 2k$. Applying a Hoeffding bound, we get that the probability that *miss* $\geq k$ (i.e., that *miss* is not bounded away by more than k from its expectation) is at least $1 - e^{-k/2}$.

Thus, in order to get an answer $\neq \perp$ from B, C will have to guess almost certainly at least k of the m_i's, the probability of that happening is roughly[6] $2^{-\mu(k)\cdot k} \leq 2^{-k}$. ▲

This concludes the proof of Theorem 5. ■

References

1. Abe, M., Kiltz, E., Okamoto, T.: CCA-security with optimal ciphertext overhead. In: ASIACRYPT, pp. 355–371 (2008)
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: ACM CCS 1993, pp. 62–73 (1993)
3. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)
4. Bellare, M., Sahai, A.: Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 519–536. Springer, Heidelberg (1999)
5. Bleichenbacher, D.: Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 1–12. Springer, Heidelberg (1998)
6. Boldyreva, A., Fischlin, M.: Analysis of random oracle instantiation scenarios for OAEP and other practical schemes. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 412–429. Springer, Heidelberg (2005)
7. Boldyreva, A., Fischlin, M.: On the security of OAEP. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 210–225. Springer, Heidelberg (2006)
8. Boneh, D.: Simplified OAEP for the RSA and rabin functions. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 275–291. Springer, Heidelberg (2001)
9. Brown, D.R.L.: What hashes make RSA-OAEP secure? Cryptology ePrint Archive, Report 2006/223 (2006), `http://eprint.iacr.org/`
10. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM 51(4), 557–594 (2004)
11. Canetti, R., Micciancio, D., Reingold, O.: Perfectly one-way probabilistic hash functions (preliminary version). In: STOC, pp. 131–140 (1998)

[6] It is not exactly that, as $f_{k,ek}$ is a random permutation, not a function.

12. Chevallier-Mames, B., Phan, D.H., Pointcheval, D.: Optimal asymmetric encryption and signature paddings. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 254–268. Springer, Heidelberg (2005)
13. Coron, J.-S., Joye, M., Naccache, D., Paillier, P.: Universal padding schemes for RSA. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 226–241. Springer, Heidelberg (2002)
14. Cramer, R., Hanaoka, G., Hofheinz, D., Imai, H., Kiltz, E., Pass, R., Shelat, A., Vaikuntanathan, V.: Bounded CCA2-secure encryption. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 502–518. Springer, Heidelberg (2007)
15. Damgård, I.: Collision free hash functions and public key signature schemes. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 203–216. Springer, Heidelberg (1988)
16. Dodis, Y., Freedman, M.J., Jarecki, S., Walfish, S.: Versatile padding schemes for joint signature and encryption. In: ACM CCS, pp. 344–353 (2004)
17. Dodis, Y., Oliveira, R., Pietrzak, K.: On the generic insecurity of the full domain hash. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 449–466. Springer, Heidelberg (2005)
18. Fujisaki, E., Okamoto, T., Pointcheval, D., Stern, J.: RSA-OAEP is secure under the RSA assumption. Journal of Cryptology 17(2), 81–104 (2004)
19. Gennaro, R., Trevisan, L.: Lower bounds on the efficiency of generic cryptographic constructions. In: FOCS, pp. 305–313 (2000)
20. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: FOCS, pp. 325–335 (2000)
21. Gertner, Y., Malkin, T.G., Myers, S.: Towards a separation of semantic and CCA security for public key encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 434–455. Springer, Heidelberg (2007)
22. Gertner, Y., Malkin, T., Reingold, O.: On the impossibility of basing trapdoor functions on trapdoor predicates. In: FOCS, pp. 126–135 (2001)
23. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: STOC, pp. 25–32 (1989)
24. Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: FOCS, pp. 102–115 (2003)
25. Haitner, I.: Implementing oblivious transfer using collection of dense trapdoor permutations. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 394–409. Springer, Heidelberg (2004)
26. Hsiao, C.-Y., Reyzin, L.: Finding collisions on a public road, or do secure hash functions need secret coins? In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 92–105. Springer, Heidelberg (2004)
27. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: STOC, pp. 44–61 (1989)
28. Kim, J.H., Simon, D.R., Tetali, P.: Limits on the efficiency of one-way permutation-based hash functions. In: FOCS, pp. 535–542 (1999)
29. Kobara, K., Imai, H.: OAEP++: A very simple way to apply OAEP to deterministic OW-CPA primitives. Cryptology ePrint Archive, Report 2002/130 (2002), http://eprint.iacr.org/
30. Komano, Y., Ohta, K.: Efficient universal padding techniques for multiplicative trapdoor one-way permutation. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 366–382. Springer, Heidelberg (2003)
31. Lindell, Y.: A simpler construction of CCA2-secure public-key encryption under general assumptions. Journal of Cryptology 19(3), 359–377 (2006)

32. Micali, S., Rabin, M.O., Vadhan, S.P.: Verifiable random functions. In: FOCS, pp. 120–130 (1999)
33. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC (1990)
34. Okamoto, T., Pointcheval, D.: REACT: Rapid enhanced-security asymmetric cryptosystem transform. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 159–175. Springer, Heidelberg (2001)
35. Paillier, P., Villar, J.L.: Trading one-wayness against chosen-ciphertext security in factoring-based encryption. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 252–266. Springer, Heidelberg (2006)
36. Phan, D.H., Pointcheval, D.: Chosen-ciphertext security without redundancy. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 1–18. Springer, Heidelberg (2003)
37. Phan, D.H., Pointcheval, D.: OAEP 3-round:A generic and secure asymmetric encryption padding. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 63–77. Springer, Heidelberg (2004)
38. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 433–444. Springer, Heidelberg (1993)
39. Rosen, A., Segev, G.: Chosen-ciphertext security via correlated products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)
40. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)
41. Shoup, V.: OAEP reconsidered. Journal of Cryptology 15(4), 223–249 (2002)
42. Simon, D.R.: Findings Collisions on a One-Way Street: Can Secure Hash Functions Be Based on General Assumptions? In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 334–345. Springer, Heidelberg (1998)