On the Security of OAEP

Alexandra Boldyreva[1] and Marc Fischlin[2]

[1] College of Computing, Georgia Institute of Technology, USA
sasha@gatech.edu www.cc.gatech.edu/~aboldyre

[2] Darmstadt University of Technology, Germany
marc.fischlin@gmail.com www.fischlin.de

Abstract. Currently, the best and only evidence of the security of the OAEP encryption scheme is a proof in the contentious random oracle model. Here we give further arguments in support of the security of OAEP. We first show that partial instantiations, where one of the two random oracles used in OAEP is instantiated by a function family, can be provably secure (still in the random oracle model). For various security statements about OAEP we specify sufficient conditions for the instantiating function families that, in some cases, are realizable through standard cryptographic primitives and, in other cases, may currently not be known to be achievable but appear moderate and plausible. Furthermore, we give the first non-trivial security result about *fully* instantiated OAEP *in the standard model*, where both oracles are instantiated simultaneously. Namely, we show that instantiating both random oracles in OAEP by modest functions implies non-malleability under chosen plaintext attacks for random messages. We also discuss the implications, especially of the full instantiation result, to the usage of OAEP for secure hybird encryption (as required in SSL/TLS, for example).

1 Introduction

OAEP is one of the most known and widely deployed asymmetric encryption schemes. It was designed by Bellare and Rogaway [5] as a scheme based on a trapdoor permutation such as RSA. 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. The encryption algorithm of $\text{OAEP}^{G,H}[F]$ takes a public key f, which is an instance of a trapdoor permutation family F, and a message M, picks r at random and computes the ciphertext $C = f(s||t)$ for $s = G(r) \oplus M||0^{k_1}$ and $t = H(s) \oplus r$, where G and H are some hash functions. Despite its importance the only security results for OAEP are a proof of IND-CPA security assuming F is a one-way trapdoor permutation family [5] and a proof of IND-CCA2 security assuming F is partial one-way [16], both in the random oracle (RO) model, i.e., where G and H are idealized and modeled as random oracles [4]. However, such proofs merely provide heuristic evidence that breaking the scheme may be hard in reality (when the random oracles are instantiated with real functions).

A growing number of papers raised concerns regarding soundness of the controversial random oracle model [12, 19, 20, 17, 1, 14, 9, 21]. Moreover, most of the recent results question security of the practical schemes known to be secure in the RO model. For example, Dodis et al. [14] showed some evidence that the RSA Full Domain Hash signature scheme may not be secure in the standard model. Boldyreva and Fischlin [9] showed that even presumably strong candidates like perfectly one-way hash functions (POWHFs) [11, 13] are insufficient to prove security of partial instantiations of OAEP (when only one of the two random oracles is instantiated with an instance of a POWHF).

The motivation of this work is to gather evidence of soundness of the OAEP design. Like the aforementioned works our goal is to go beyond the classical RO heuristic and study security of the scheme when one or all of its ROs are instantiated. Positive results in the direction of partial instantiations would give further evidence that breaking OAEP for good instantiations is hard, because breaking the scheme would then require to exploit interdependent weaknesses between the instantiations or the family F. Given the negative results of [9] it is unlikely to expect that the properties needed from the instantiating function families are weak or even easily realizable, even if one accepts weaker security stipulations than chosen-ciphertext security for partial or full instantiations. For example, although it seems plausible, it is currently not even known whether OAEP can be proven IND-CPA secure in the standard model assuming any reasonable properties of the instantiating functions.

Here we show that security proofs for instantiations of OAEP are indeed possible. For various security statements about OAEP we specify sufficient conditions on G and H that are certainly weaker than assuming that the functions behave as random oracles, yielding "positive" security statements regarding partially instantiated OAEP. Furthermore, we give the first non-trivial security results about fully instantiated OAEP in the standard model, where both oracles G and H are instantiated simultaneously. We next discuss these results in more detail.

The OAEP Framework. For better comprehension of our technical results we first reconsider the OAEP encryption scheme from a more abstract viewpoint. Let f be a random instance of a partial one-way trapdoor permutation family F, and the encryption algorithm computes a ciphertext as $C = f(s||t)$. Partial one-wayness [16] requires that it is hard to find the leading part of the pre-image $s||t$ under f and to output, say, s only. If we consider now for example a family $F_{\mathsf{t\text{-}clear}}$ where each function is defined as $f \equiv g||\mathrm{ID}$ such that $f(s||t) = g(s)||t$ for a trapdoor permutation g, then this family $F_{\mathsf{t\text{-}clear}}$ is clearly partial one-way (and also a trapdoor permutation). Hence, this example describes a special case $\mathrm{OAEP}^{G,H}[F_{\mathsf{t\text{-}clear}}]$ for the partial one-way trapdoor permutation family $F_{\mathsf{t\text{-}clear}}$ where each function outputs the t-part in clear. In particular, the security proof in the random oracle model for OAEP and general partial one-way families (including RSA as a special case) [16] carries over, but we outdo this by giving positive results of partial instantiation for such families $F_{\mathsf{t\text{-}clear}}$.

Towards the standard-model security results for fully instantiated OAEP we take the above viewpoint one step further and look at $\mathrm{OAEP}^{G,H}[F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}]$ for families $F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}$ where each function f outputs the k_1 least significant bits of $s = G(r) \oplus M||0^{k_1}$ (which equal those bits of $G(r)$) and t in clear. Since each function in $F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}$ is also a member in $F_{\mathsf{t\text{-}clear}}$ the partial instantiation results above remain true for $\mathrm{OAEP}^{G,H}[F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}]$.

We note that security of partial instantiations of $\mathrm{OAEP}^{G,H}[F_{\mathsf{t\text{-}clear}}]$ and of $\mathrm{OAEP}^{G,H}[F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}]$, although for qualified partial one-way trapdoor families, also have implications for the popular $\mathrm{OAEP}^{G,H}[\mathrm{RSA}]$ case. They show that any successful attacks on instantiations for RSA would have to take advantage of specific properties of the RSA function. Generic attacks which would also work for $F_{\mathsf{t\text{-}clear}}$ or $F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}$ are then ruled out.

Partial Instantiation Results. Positive results about partial instantiations were first shown in [9] for the PSS-E encryption scheme. There it was also shown, however, that perfectly one-way hash functions cannot be securely used to instantiate either one of the ROs in OAEP. These negative results about partial instantiation through POWHFs hold for $\mathrm{OAEP}^{G,H}[F_{\mathsf{t\text{-}clear}}]$ as well. Yet we show that partial instantiations are possible by switching to other primitives.

To instantiate the G-oracle in $\mathrm{OAEP}^{G,H}[F_{\mathsf{t\text{-}clear}}]$ while preserving IND-CCA2 security (in the random oracle model), we introduce the notion of a near-collision resistant pseudorandom generator. For such a generator G it is infeasible to find different seeds $r \neq r'$ such that predetermined parts of the generator's outputs $\mathsf{G}(r)$, $\mathsf{G}(r')$ match (they may differ on other parts). To be more precise for $\mathrm{OAEP}^{G,H}[F_{\mathsf{t\text{-}clear}}]$ the generator G is not allowed to coincide on the k_1 least significant bits, bequeathing this property to the values $s = G(r) \oplus M||0^{k_1}$ and $s' = G(r') \oplus M||0^{k_1}$ in the encryption process. We discuss that such pseudorandom generators can be derived from any one-way permutation.

Instantiating the H oracle in OAEP turns out to be more challenging. To this end we consider non-malleable pseudorandom generators, where a given image of a seed r should not help significantly to produce an image of a related seed r'. Instantiating H through such a non-malleable pseudorandom generator the resulting scheme achieves NM-CPA security, where it is infeasible to convert a given ciphertext into one of a related message. Although this security notion for encryption schemes is not as strong as IND-CCA, it yet exceeds the classical IND-CPA security. That is, Bellare et al. [3] show that NM-CPA implies IND-CPA and is incomparable to IND-CCA1 security. Hence, NM-CPA security of schemes lies somewhere in between IND-CPA and IND-CCA2.[1]

We also show that it is possible to extend the above result and to instantiate the H-oracle in $\mathrm{OAEP}^{G,H}[F_{\mathsf{t\text{-}clear}}]$ without even sacrificing IND-CCA2 security (again, for random oracle G). This however requires the very strong assumption for the pseudorandom generators which then must be non-malleable under

[1] We mitigate the notion of NM-CPA such that the relation specifying related messages and the distribution over the messages must be fixed at the outset. This mildly affects the relationship to the IND notions, but we omit technical details in the introduction.

chosen-image attacks. For such a generator non-malleability should even hold if the adversary can learn seeds of chosen images, and such generators resemble chosen-ciphertext secure encryption schemes already. Hence, we see this partial instantiation as a mere plausibility result that one can presumably instantiate oracle H and still have IND-CCA2 security. This is contrast to the results in [12] for example, showing that there are encryption schemes secure in the random oracle model but which cannot be securely realized for any primitive, not even for a secure encryption scheme itself.

As for the existence of non-malleable pseudorandom generators, we are not aware if they can be derived from standard cryptographic assumptions, and we leave this as an interesting open problem. We also remark that, while non-malleability under chosen-image attacks seems to be a rather synthetic property, plain non-malleability as required in the NM-CPA result appears to be a modest and plausible assumption for typical instantiation candidates like hash functions. For instance, it should not be easy to flip bits in given hash value, affecting bits in the pre-image in a reasonable way.

Full Instantiation Result. Our main result is a standard-model security proof for a fully instantiated OAEP. It is not very reasonable to expect a proof of IND-CCA2 security of OAEP in the standard model, even assuming very strong properties of instantiating functions (although we all would like to see such result). As we mentioned above, we are not aware if one can even show IND-CPA security of fully instantiated OAEP.

Nevertheless we show that OAEP in the standard model can be proven to satisfy a rather strong notion of security notion, namely \$NM-CPA. It is slightly weaker than the standard non-malleability notion NM-CPA in that there is a restriction that an unknown random message is encrypted in the challenge ciphertext. A bit more formally this security notion \$NM-CPA requires that given a public key and a ciphertext of a challenge message chosen uniformly at random from a large message space it is hard to compute a valid ciphertext of a message non-trivially related to the challenge message. Note that this is consistent with how asymmetric schemes are typically used to build hybrid encryption schemes, where the key of the symmetric scheme is derived from a random string encrypted with the public-key scheme. To appreciate the power of the \$NM-CPA definition we note that it implies for example the notion of OW-CPA and, moreover, Bleichenbacher's attack [7] on PKCS #1 v1.5 is not possible for \$NM-CPA secure schemes.[2] Thus our result provides better evidence that OAEP resists such attacks, and specifies what properties of the instantiating functions are sufficient for this.

[2] Bleichenbacher's attack works by generating a sequence of ciphertexts from a given ciphertext and verifying validity of the derived ciphertexts by querying the decryption oracle. While requiring *adaptive* queries to recover the entire message, one can view the message in first derived ciphertext in such an attack as having a small (but not negligible) probability of being non-trivially related to the original (possibly random) message.

For our full instantiation proof we consider $\mathrm{OAEP}^{G,H}[F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}]$ where the t-part and the least significant bits of the s-part are output in clear. To achieve the \$NM-CPA security notion under full instantiation of both oracles G and H in $\mathrm{OAEP}^{G,H}[F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}]$ we need to augment the near-collision resistant generator G by a trapdoor property, allowing to invert images efficiently given the trapdoor information; such generators exist if trapdoor permutations exist. We again use a non-malleable pseudorandom generator H for instantiating H. Assuming that the generators above exist we show that $\mathrm{OAEP}^{G,H}[F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}]$ is \$NM-CPA.[3]

To give further evidence of the usefulness of the \$NM-CPA notion we finally show that we can derive a hybrid encryption scheme that is NM-CPA in the random oracle model from an asymmetric scheme secure in the sense of \$NM-CPA. For this, one encrypts a random string r with the asymmetric scheme and then runs r through an idealized key derivation process to obtain $K = G(r)$, modeled through a random oracle G. The actual message is then encrypted with a symmetric scheme for key K. The construction of such hybrid encryption schemes resembles the encryption method in SSL/TLS [18]. There, simply speaking, the client encrypts a random string under the server's public key and then both parties derive the actual symmetric key K by hashing the random string iteratively. If one considers this hashing step as an idealized process then our results provide a security guarantee for this technique. Observe that this result is still cast in the random oracle model; yet it separates the security of the key derivation process from the security of the asymmetric encryption scheme and can be seen as a partial instantiation for the random oracles in the encryption algorithm.

PROSPECT. The random oracle model should provide confidence that the design of a cryptographic scheme is sound, even if a security proof in the standard model for this scheme is missing. The heuristic argument is that "good" instantiations of random oracles then give evidence that no "clever" attacks against a scheme work. But the well-known negative results about the random oracle principle have raised some doubts how much confidence this security heuristic really gives.

The approach we take here towards challenging the doubts is to trade security goals against partial or full instantiations of random oracles. Our "test case" OAEP shows that this is a viable way and gives more insights in "how clever" attacks against the instantiations would have to be. And while this still does not rule out the possibility of extraordinary attacks we see this as an important supplement to the random oracle heuristic and to the question how instantiating candidates should be selected, hopefully inciting other results along this direction.

2 Preliminaries

If S is a set then $x \xleftarrow{\$} S$ means that the value x is chosen uniformly at random from S. If $\mathcal{A}$ is a deterministic (resp. randomized algorithm) with a single output

[3] Very recently, Brown [2] has shown that RSA-OAEP cannot be proven OW-CPA under certain security reductions. Our approach here does not fall under this kind of reductions and does not contradict his result. We provide more details in Section 3.2.

then $x \leftarrow \mathcal{A}(y, z, \dots)$ (resp. $x \xleftarrow{\$} \mathcal{A}(y, z, \dots)$) means that the value x is assigned the output of $\mathcal{A}$ for input $(y, z, \dots)$. An algorithm is called efficient if it runs in polynomial time in the input length (which, in our case, usually refers to polynomial time in the security parameter).

A function family $F = \bigcup_k F(1^k)$ consists of sets of functions $F(1^k) = \{f : \{0,1\}^{m(k)} \to \{0,1\}^{n(k)}\}$. It is called a family of trapdoor permutations if for each $f \in F(1^k)$ there exists f^{-1} such that $f(f^{-1}) \equiv \mathrm{ID}$. We usually identify the functions f and f^{-1} simply with their descriptions, and write $(f, f^{-1}) \xleftarrow{\$} F(1^k)$ for the random choice of f (specifying also f^{-1}) from the family $F(1^k)$. Unless stated differently the minimal assumption about a function family in this paper is that it is one-way, and that it is efficiently computable.

2.1 The OAEP Framework

The OAEP encryption framework [5] is parameterized by integers k, k_0 and k_1 (where k_0, k_1 are linear functions of k) and makes use of a trapdoor permutation family F with domain and range $\{0,1\}^k$ and two random oracles

$$G\colon \{0,1\}^{k_0} \to \{0,1\}^{k-k_0} \quad \text{and} \quad H\colon \{0,1\}^{k-k_0} \to \{0,1\}^{k_0} \ .$$

The message space is $\{0,1\}^{k-k_0-k_1}$. The scheme $\mathrm{OAEP}^{G,H}[F] = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined as follows:

- The key generation algorithm $\mathcal{K}(1^k)$ picks a pair $(f, f^{-1}) \leftarrow F(1^k)$ at random. Let pk specify f and let sk specify f^{-1}.
- The encryption algorithm $\mathcal{E}(pk, M)$ picks $r \xleftarrow{\$} \{0,1\}^{k_0}$, and computes $s \leftarrow G(r) \oplus (M \| 0^{k_1})$ and $t \leftarrow H(s) \oplus r$. It finally outputs $C \leftarrow f(s||t)$.
- The decryption algorithm $\mathcal{D}(sk, C)$ computes $s\|t \leftarrow f^{-1}(C)$, $r \leftarrow t \oplus H(s)$ and $M \leftarrow s \oplus G(r)$. If the last k_1 bits of M are zeros, then it returns the first $k - k_0 - k_1$ bits of M, else it returns $\perp$.

The encryption scheme $\mathrm{OAEP}^{G,H}[F]$ is IND-CCA2 secure in the RO model if the underlying trapdoor permutation family F is partial one-way [16].

As a side effect of the partial one-wayness result for OAEP [16] we can immediately conclude security of a particular OAEP variant, where we use partial one-way trapdoor permutation family $F_{\text{t-clear}}$ based on a trapdoor permutation function family F. Namely, each function $f_{\text{t-clear}} : \{0,1\}^k \to \{0,1\}^k$ in $F_{\text{t-clear}}$ is described by $f_{\text{t-clear}}(s||t) \equiv f(s)||\mathrm{ID}(t) = f(s)||t$ for a one-way permutation $f : \{0,1\}^{k-k_0} \to \{0,1\}^{k-k_0}$, i.e., the t-part is output in clear. A random instance $(f_{\text{t-clear}}, f^{-1}_{\text{t-clear}}) \leftarrow F_{\text{t-clear}}(1^k)$ is sampled by picking $(f, f^{-1}) \leftarrow F(1^k)$ and setting $f_{\text{t-clear}}$ as above (the inverse $f^{-1}_{\text{t-clear}}$ is straightforwardly defined). Then $F_{\text{t-clear}}$ is clearly partial one-way and thus $\mathrm{OAEP}^{G,H}[F_{\text{t-clear}}]$ IND-CCA2 secure in the random oracle model.

Analogously, we consider another important variant of OAEP where we also output the k_1 least significant bits $\mathrm{lsb}_{k_1}(s)$ of s in clear and merely apply the trapdoor function f to the leading $k - k_0 - k_1$ bits of s. That is, a random function $f_{\mathsf{lsb}||\mathsf{t\text{-}clear}} : \{0,1\}^k \to \{0,1\}^k$ in $F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}(1^k)$ is described by a random trapdoor permutation $f : \{0,1\}^{k-k_0-k_1} \to \{0,1\}^{k-k_0-k_1}$ and $f_{\mathsf{lsb}||\mathsf{t\text{-}clear}}(s||t) = f(s_{1\ldots k-k_0-k_1})||\mathrm{lsb}_{k_1}(s)||t$. Note that since $s = G(r) \oplus M||0^{k_1}$ this means that we output the least significant bits $\mathrm{lsb}_{k_1}(G(r))$ of $G(r)$ and t in clear. For this reason we sometimes write $s||\gamma$ instead of s and denote by γ the k_1 bits $\mathrm{lsb}_{k_1}(G(r))$ such that $f_{\mathsf{lsb}||\mathsf{t\text{-}clear}}(s||\gamma||t) = f(s)||\gamma||t$. $F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}$ is clearly partial one-way and $\mathrm{OAEP}^{G,H}[F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}]$ is IND-CCA2 secure in the random oracle model.

In both cases we often identify $F_{\mathsf{t\text{-}clear}}$ resp. $F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}$ simply with the underlying family F and vice versa. In particular we often denote a random function from $F_{\mathsf{t\text{-}clear}}$ or $F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}$ simply by f. We call *$F_{\mathsf{t\text{-}clear}}$ resp. $F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}$ the induced family of F.*

Random Oracle Instantiations. For an instantiation of the random oracle G in $\mathrm{OAEP}^{G,H}[F]$ we consider a pair of efficient algorithms $\mathcal{G} = (\mathsf{KGenG}, \mathsf{G})$ where KGenG on input 1^k returns a random key K and the deterministic algorithm[4] G maps this key K and input $r \in \{0,1\}^{k_0}$ to an output string $\mathsf{G}(K,r) = \mathsf{G}_K(r)$ of $k - k_0$ bits. Then we write $\mathrm{OAEP}^{\mathcal{G},H}[F]$ for the encryption scheme which works as defined above, but where the key pair (sk, pk) is now given by $sk = (f^{-1}, K)$ and $pk = (f, K)$ and where each evaluation of $G(r)$ is replaced by $\mathsf{G}_K(r)$. We say that *$OAEP^{\mathcal{G},H}[F]$ is a partial G-instantiation of OAEP through $\mathcal{G}$.*

A *partial H-instantiation $OAEP^{G,\mathcal{H}}[F]$ of OAEP through $\mathcal{H}$* and partial instantiations of the aforementioned OAEP variations are defined accordingly. If we instantiate both oracles G, H simultaneously then we speak of a *full instantiation $OAEP^{\mathcal{G},\mathcal{H}}[F]$ of OAEP through $\mathcal{G}$ and $\mathcal{H}$.*

2.2 Security of Encryption Schemes

In this section we review the relevant security notions for asymmetric encryption schemes $\mathsf{AS} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. In addition to indistinguishability under chosen-plaintext and chosen-ciphertext attacks (IND-CPA, IND-CCA1, IND-CCA2) — see for instance [3] for formal definitions— we occasionally also rely on the notions of non-malleability. This notion was introduced and formalized in [15, 3]. The most basic version is called NM-CPA and says that a ciphertext of a message M^* should not help to find a ciphertext of a related message M, where the distribution of message M^* is defined by an efficient distribution $\mathcal{M}$ and related messages are specified by an efficient relation R, both chosen by the adversary.

Definition 1 (NM-CPA). *Let AS be an asymmetric encryption scheme. Then AS is called secure in the sense of NM-CPA if for for every efficient algorithm $\mathcal{A}$ the following random variables $\mathbf{Exp}^{\text{nm-cpa-1}}_{\mathsf{AS},\mathcal{A}}(k)$, $\mathbf{Exp}^{\text{nm-cpa-0}}_{\mathsf{AS},\mathcal{A}}(k)$ are computationally indistinguishable:*

[4] In general, the instantiating functions can be randomized. This requires some care with the decryption algorithms and possibly introduces new attacks. Since our results all hold with respect to deterministic algorithms this is beyond our scope here; see [9] for more details.

$$
\begin{array}{l|l}
\textbf{\textit{Experiment}}\ \mathbf{Exp}^{\text{nm-cpa-1}}_{\mathsf{AS},\mathcal{A}}(k) & \textbf{\textit{Experiment}}\ \mathbf{Exp}^{\text{nm-cpa-0}}_{\mathsf{AS},\mathcal{A}}(k) \\
\quad (pk, sk) \xleftarrow{\$} \mathcal{K}(1^k) & \quad (pk, sk) \xleftarrow{\$} \mathcal{K}(1^k) \\
\quad (\mathcal{M}, \mathsf{state}) \xleftarrow{\$} \mathcal{A}(pk) & \quad (\mathcal{M}, \mathsf{state}) \xleftarrow{\$} \mathcal{A}(pk) \\
\quad M^* \xleftarrow{\$} \mathcal{M} & \quad M^* \xleftarrow{\$} \mathcal{M}\ ;\ M' \xleftarrow{\$} \mathcal{M} \\
\quad C^* \xleftarrow{\$} \mathcal{E}_{pk}(M^*) & \quad C' \xleftarrow{\$} \mathcal{E}_{pk}(M') \\
\quad (R, C) \xleftarrow{\$} \mathcal{A}(\mathsf{state}, C^*) & \quad (R, C) \xleftarrow{\$} \mathcal{A}(\mathsf{state}, C') \\
\quad M \leftarrow \mathcal{D}_{sk}(C) & \quad M \leftarrow \mathcal{D}_{sk}(C) \\
\quad \textit{Return 1 iff} & \quad \textit{Return 1 iff} \\
\qquad (C \neq C^*) \wedge R(M^*, M) & \qquad (C \neq C') \wedge R(M^*, M)
\end{array}
$$

It is assumed that the messages in the support of $\mathcal{M}$ have equal length.

We note that the original definition of NM-CPA in [3] actually allows the adversary to output a vector of ciphertexts. Our results for OAEP merely hold with respect to binary relations and therefore we restrict the definition here to such relations. We remark that the aforementioned relationships of NM-CPA to the indistinguishability notions, e.g., that this notion is strictly stronger than the one of IND-CPA, hold for relations of arity two as well.

We define a weaker security notion is that of \$NM-CPA where the adversary does not have the ability to choose a distribution over the messages, but where a random message is encrypted and the adversary tries to find a ciphertext of a related message.

Definition 2 (\$NM-CPA). *Let $\mathsf{AS} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an asymmetric encryption scheme and let $\mathcal{M}$ for input 1^k describe the uniform distribution over all $\ell(k)$ bit strings for some polynomial ℓ. Then AS is called secure in the sense of \$NM-CPA if for for every efficient algorithm $\mathcal{A}$ and for every efficient relation R the following random variables $\mathbf{Exp}^{\$\text{nm-cpa-1}}_{\mathsf{AS},\mathcal{A},\mathcal{M},R}(k)$, $\mathbf{Exp}^{\$\text{nm-cpa-1}}_{\mathsf{AS},\mathcal{A},\mathcal{M},R}(k)$ are computationally indistinguishable:*

$$
\begin{array}{l|l}
\textbf{\textit{Experiment}}\ \mathbf{Exp}^{\$\text{nm-cpa-1}}_{\mathsf{AS},\mathcal{A},\mathcal{M},R}(k) & \textbf{\textit{Experiment}}\ \mathbf{Exp}^{\$\text{nm-cpa-0}}_{\mathsf{AS},A,\mathcal{M},R}(k) \\
\quad (pk, sk) \xleftarrow{\$} \mathcal{K}(1^k) & \quad (pk, sk) \xleftarrow{\$} \mathcal{K}(1^k) \\
\quad M^* \xleftarrow{\$} \mathcal{M}(1^k) & \quad M^* \xleftarrow{\$} \mathcal{M}(1^k)\ ;\ M' \xleftarrow{\$} \mathcal{M}(1^k) \\
\quad C^* \xleftarrow{\$} \mathcal{E}_{pk}(M^*) & \quad C' \xleftarrow{\$} \mathcal{E}_{pk}(M') \\
\quad C \xleftarrow{\$} \mathcal{A}(pk, C^*, \langle R \rangle) & \quad C \xleftarrow{\$} \mathcal{A}(pk, C', \langle R \rangle) \\
\quad M \leftarrow \mathcal{D}_{sk}(C) & \quad M \leftarrow \mathcal{D}_{sk}(C) \\
\quad \textit{Return 1 iff} & \quad \textit{Return 1 iff} \\
\qquad (C \neq C^*) \wedge R(M^*, M) & \qquad (C \neq C') \wedge R(M^*, M)
\end{array}
$$

While the notion of \$NM-CPA is weaker than the one of NM-CPA —in addition to the restriction to uniformly distributed messages the relation is now fixed in advance— it yet suffices for example to show security in the sense of OW-CPA (where the adversary's goal is to recover a random message in a given ciphertext) and it also covers Bleichenbacher's attack on PKCS #1 v1.5. In Section 5 we also show that the notion of \$NM-CPA is enough to derive NM-CPA security under an idealized key derivation function. Namely, one encrypts a random string r

under the \$NM-CPA public-key encryption scheme and then pipes r through a random oracle G to derive a key $K = G(r)$ for the symmetric scheme. In fact, one can view the SSL encryption method where the client sends an encrypted random key to the server and both parties derive a symmetric key through a complicated hash function operation as a special case of this method. Then this result about lifting \$NM-CPA to NM-CPA security, together with the \$NM-CPA security proof for the full instantiation of $\text{OAEP}_{\mathsf{lsb||t\text{-}clear}}$, provides an interesting security heuristic (as long as the key derivation process behaves in an ideal way).

2.3 Pseudorandom Generators

Typically, the minimal expected requirement when instantiating a random oracle is that the instantiating function describes a pseudorandom generator, consisting of the key generation algorithm KGen producing a public key K and the evaluation algorithm G mapping a random seed r with key K to the pseudorandom output. Usually the output of this generator should still look random when some side information $\mathsf{hint}(r)$ about the seed r is given. This probabilistic function hint must be of course uninvertible, a weaker notion than one-wayness (cf. [11]).

We also incorporate into the definition the possibility that the key generation algorithm outputs some secret trapdoor information K^{-1} in addition to K. Given this information K^{-1} one can efficiently invert images. If this trapdoor property is not required we can assume that $K^{-1} = \perp$ and often omit K^{-1} in the key generator's output.

Definition 3 ((Trapdoor) Pseudorandom Generator). *Let* KGen *be an efficient key-generation algorithm that takes as input* 1^k *for* $k \in \mathbb{N}$ *and outputs a key* K*; let* G *be an efficient deterministic evaluation algorithm that, on input* K *and a string* $r \in \{0,1\}^k$ *returns a string of length* $\ell(k)$*. Then* $\mathcal{G} = (\mathsf{KGen}, \mathsf{G})$ *is called a pseudorandom generator (with respect to* hint*) if the following random variables are computationally indistinguishable:*

- *Let* $K \leftarrow \mathsf{KGen}(1^k)$*,* $r \xleftarrow{\$} \{0,1\}^k$*,* $h \leftarrow \mathsf{hint}(r)$*, output* $(K, \mathsf{G}(K,r), h)$*.*
- *Let* $K \leftarrow \mathsf{KGen}(1^k)$*,* $r \xleftarrow{\$} \{0,1\}^k$*,* $h \leftarrow \mathsf{hint}(r)$*,* $u \leftarrow \{0,1\}^{\ell(n)}$*, output* (K, u, h)*.*

Furthermore, if there is an efficient algorithm TdG *such that for any* $k \in \mathbb{N}$*, any* $(K, K^{-1}) \leftarrow \mathsf{KGen}(1^k)$*, any* $r \in \{0,1\}^k$ *we have* $\mathsf{G}(K, \mathsf{TdG}(K^{-1}, \mathsf{G}(K,r))) = \mathsf{G}(K,r)$ *then* $(\mathsf{KGen}, \mathsf{G}, \mathsf{TdG})$ *is called a trapdoor pseudorandom generator.*

For our results about OAEP we often need further properties from the pseudorandom generator, including near-collision resistance and non-malleability. The former means that given a seed r it is hard to find a different seed r' such that $\mathsf{G}(K,r)$ and $\mathsf{G}(K,r')$ coincide on a predetermined set of bits (even if they are allowed to differ on the other bits). Non-malleability refers to generators where the generator's output for a seed should not help to produce an image of a related seed. We give precise definitions and details concerning existential questions on site.

3 Partial Instantiations for OAEP

In this section we prove security of partial instantiations of OAEP. Our results show that one can replace either one of the random oracle in OAEP by reasonable primitives and still maintain security (in the random oracle model).

3.1 Instantiating the G-Oracle for IND-CCA2 security

We first show how to construct a pseudorandom generator with a special form of collision-resistance. This property says that finding an input r' to a random input r, such that $\mathsf{G}(K,r)$ and $\mathsf{G}(K,r')$ coincide on the k least significant bits $\mathrm{lsb}_k(\mathsf{G}(K,r)), \mathrm{lsb}_k(\mathsf{G}(K,r'))$, is infeasible. According to comparable collision types for hash functions [6] we call this *near-collision resistance.*

Definition 4 (Near-collision Resistant Pseudorandom Generator). *A pseudorandom generator $\mathcal{G} = (\mathsf{KGen}, \mathsf{G})$ is called near-collision resistant (for the least significant k bits) if for any efficient algorithm $\mathcal{C}$ the following holds: Let $K \leftarrow \mathsf{KGen}(1^k)$, $r \leftarrow \{0,1\}^k$, $r' \leftarrow \mathcal{C}(K,r)$. Then the probability that $r \neq r'$ but $lsb_k(\mathsf{G}(K,r)) = lsb_k(\mathsf{G}(K,r'))$ is negligible.*

Near-collision resistant generators can be built, for example, from one-way permutations via the well-known Yao-Blum-Micali construction [22, 8]. In that case, given a family G of one-way permutations the key generation algorithm $\mathsf{KGen}_{\mathrm{YBM}}(1^k)$ of this generator simply picks a random instance $g : \{0,1\}^k \to \{0,1\}^k$ of $G(1^k)$, and $\mathsf{G}_{\mathrm{YBM}}(g,r) = (\mathrm{hb}(r), \mathrm{hb}(g(r)), \ldots, \mathrm{hb}(g^{n-1}(r)), g^n(r))$ is defined through the hardcore bits hb of g. Since g is a permutation different inputs $r \neq r'$ yield different output parts $g^n(r) \neq g^n(r')$.

Given a near-collision resistant pseudorandom generator we show how to instantiate the G-oracle in $\mathrm{OAEP}^{G,H}[F_{\mathsf{t\text{-}clear}}]$ for the family $F_{\mathsf{t\text{-}clear}}$ which is induced by a trapdoor permutation family F (i.e., where a member $f : \{0,1\}^{k-k_0} \to \{0,1\}^{k-k_0}$ of F is applied to the k-bit inputs such that the lower k_0 bits are output in clear).

Theorem 1. *Let $\mathcal{G} = (\mathsf{KGenG}, \mathsf{G})$ be a pseudorandom generator which is near-collision resistant (for the k_1 least significant bits). Let F be trapdoor permutation family and let $F_{\mathsf{t\text{-}clear}}$ be the induced partial one-way trapdoor permutation family defined in Section 2.1. Then the partial G-instantiation $OAEP^{\mathcal{G},H}[F_{\mathsf{t\text{-}clear}}]$ of OAEP through $\mathcal{G}$ is IND-CCA2 in the random oracle model.*

The full proof appears in the full version [3]. The idea is to gradually change the way the challenge ciphertext (encrypting one of two adversarially chosen messages, the hidden choice made at random) is computed in a sequence of games. We show that each of these steps does not change an adversary's success probability of predicting the secret choice noticeably:

- Initially, in Game^0 the challenge ciphertext $f(s^*)||t^*$ for message M^* is computed as in the scheme's description by $s^* = \mathsf{G}(K,r^*) \oplus M^*||0^{k_1}$ for the near-collision resistant generator G and $t^* = H(s^*) \oplus r^*$ for random oracle H.

– In Game^1 the ciphertext is now computed by setting $s^* = \mathsf{G}(K, r^*) \oplus M^*||0^{k_1}$ as before, but letting $t^* = \omega \oplus r^*$ for a random ω which is independent of $H(s^*)$. Because H is a random oracle this will not affect the adversary's success probability, except for the rare case that the adversary queries H about s^*.
– In Game^2, in a rather cosmetic change, we further substitute $t^* = \omega \oplus r^*$ simply for $t^* = \omega$, making the t-part independent of the generator's pre-image r^*.
– in Game^3 we use the pseudorandomness of generator G to replace $s^* = \mathsf{G}(K, r^*) \oplus M^*||0^{k_1}$ by $s^* = u \oplus M^*||0^{k_1}$ for a random u.

Since ciphertexts in the last game are distributed independently of the actual message security of the original scheme follows, after a careful analysis that decryption queries do not help; this is the step where we exploit that H is still a random oracle and that $\mathcal{G}$ is near-collision resistant. Namely, the near-collision resistance prevents an adversary from transforming the challenge ciphertext for values r^*, s^* into a valid one for the same s^* but a different r; otherwise the least significant bits of $s^* = \mathsf{G}(K, r^*) \oplus M^*||0^{k_1} = \mathsf{G}(K, r) \oplus M||0^{k_1}$ would not coincide and the derived ciphertext would be invalid with high probability. Given this, the adversary must always use a "fresh" value s when submitting a ciphertext to the decryption oracle, and must have queried the random oracle H about s before (or else the ciphertext is most likely invalid). But then the adversary already "knows" $r = t \oplus H(s)$ —recall that for $F_{\mathsf{t\text{-}clear}}$ the t-part is included in clear in ciphertexts— and therefore "knows" the (padded) message $M||z = s \oplus \mathsf{G}(K, r)$ encapsulated in the ciphertext.

3.2 Instantiating the *H*-Oracle

To instantiate the H-oracle we introduce the notion of a non-malleable pseudorandom generator. For such a pseudorandom generator it should be infeasible to find for a given image $y^* = \mathsf{H}_K(s^*)$ of a random s^* a different image $y = \mathsf{H}_K(s)$ of a related value s, where the corresponding efficient relation $R(s^*, s)$ must be determined *before* seeing K and y^*.[5] More precisely, we formalize non-malleability of a pseudorandom generator by the indistinguishability of two experiments. For any adversary $\mathcal{B}$ it should not matter whether $\mathcal{B}$ is given $f(s^*)$, $y^* = \mathsf{H}_K(s^*)$ or $f(s^*)$, $y' = \mathsf{H}_K(s')$ for an independent s' instead: the probability that $\mathcal{B}$ outputs $f(s)$ and $y = \mathsf{H}_K(s)$ such that s is related to s^* via relation R should be roughly the same in both cases.[6]

[5] We are thankful to the people from the Ecrypt network for pointing out that a possibly stronger definition for adaptively chosen relations allows trivial relations over the images and cannot be satisfied.

[6] Adding the image under the trapdoor permutation uniquely determines the pre-image of the pseudorandom generator's output and enables us to specify $R(s^*, s)$ via *the* pre-images. Since this also bundles the security of the trapdoor permutation and the generator, Brown's recent impossibility result about security reductions for OAEP [2] does not apply.

Definition 5 (Non-Malleable Pseudorandom Generator). *Assume* $\mathcal{H} = (\mathsf{KGenH}, \mathsf{H})$ *is a pseudorandom generator (which is pseudorandom with respect to* $\mathsf{hint}(x) = (f, f(x))$ *for* $(f, f^{-1}) \leftarrow F(1^k)$ *from the trapdoor function family* F*). Then* $\mathcal{H}$ *is called non-malleable with respect to* hint *if for any efficient algorithm* $\mathcal{B}$ *and any efficient relation* R *the following random variables* $\mathbf{Exp}^{\textit{nm-cma-1}}_{\mathcal{H},\mathcal{B},F,R}(k)$, $\mathbf{Exp}^{\textit{nm-cma-0}}_{\mathcal{H},\mathcal{B},F,R}(k)$ *are computationally indistinguishable, where the experiments are defined as follows.*

Experiment $\mathbf{Exp}^{\textit{nm-cpa-1}}_{\mathcal{G},\mathcal{B},F,R}(k)$	***Experiment*** $\mathbf{Exp}^{\textit{nm-cpa-0}}_{\mathcal{G},\mathcal{B},F,R}(k)$
$K \xleftarrow{\$} \mathsf{KGenH}(1^k)$	$K \xleftarrow{\$} \mathsf{KGenH}(1^k)$
$(f, f^{-1}) \xleftarrow{\$} F$	$(f, f^{-1}) \xleftarrow{\$} F$
$s^* \xleftarrow{\$} \{0,1\}^k$	$s^* \xleftarrow{\$} \{0,1\}^k$; $s' \xleftarrow{\$} \{0,1\}^k$
$y^* \xleftarrow{\$} \mathsf{H}_K(s^*)$	$y' \xleftarrow{\$} \mathsf{H}_K(s')$
$(z, y) \xleftarrow{\$} \mathcal{B}(K, f, f(s^*), y^*)$	$(z, y) \xleftarrow{\$} \mathcal{B}(K, f, f(s^*), y')$
$s \leftarrow f^{-1}(z)$	$s \leftarrow f^{-1}(z)$
Return 1 *iff*	*Return 1 iff*
$R(s^*, s) \wedge \mathsf{H}_K(s) = y \wedge s^* \neq s$	$R(s^*, s) \wedge \mathsf{H}_K(s) = y \wedge s^* \neq s$

Given a non-malleable pseudorandom generator we can prove NM-CPA security of the partial H-instantiation of OAEP, under the restriction that the adversarial chosen message distribution and relation are defined at the beginning of the attack via $(\mathcal{M}, R, \mathsf{state}) \leftarrow \mathcal{A}(1^k)$ and thus depend only the security parameter. This relaxed notion still implies for example IND-CPA security (but for messages picked independently of the public key), is still incomparable to IND-CCA1 security, and also thwarts Bleichenbacher's attack. We call such schemes *NM-CPA for pre-defined message distributions and relations.*

Theorem 2. *Let* F *be a trapdoor permutation family and let* $F_{\textsf{t-clear}}$ *be the induced partial one-way trapdoor permutation family. Let* $\mathcal{H} = (\mathsf{KGenH}, \mathsf{H})$ *be a pseudorandom generator (with respect to* $\mathsf{hint}(x) = (f, f(x))$ *for* $(f, f^{-1}) \leftarrow F(1^k)$*). Assume further that* $\mathcal{H}$ *is non-malleable with respect to* hint*. Then the partial* H*-instantiation* $\textit{OAEP}^{G,\mathcal{H}}[F_{\textsf{t-clear}}]$ *through* $\mathcal{H}$ *is NM-CPA for pre-defined message distributions and relations in the random oracle model.*

The proof idea is as follows. Assume that an attacker, given a ciphertext for some values r^*, s^* (which uniquely define the message in a ciphertext), tries to prepare a related ciphertext for some value $r \neq r^*$, without having queried random oracle G about r before. Then such a ciphertext is most likely invalid because with overwhelming probability the least significant bits of $s \oplus G(r)$ are not zero. Else, if $r = r^*$, then we must have $f(s) \neq f(s^*)$ and $s \neq s^*$, since the adversarial ciphertext must be different for a successful attack. But then the values $\mathsf{H}(K, s^*)$ and $\mathsf{H}(K, s)$ for different pre-images must be related via the ciphertext's relation, contradicting the non-malleability of the generator H. In any other case, if $r \neq r^*$ and r is among the queries to G, the random value $G(r^*)$ is independent of $G(r)$. So must be the messages $M^*||0^{k_1} = s^* \oplus G(r^*)$ and $M||0^{k_1} = s \oplus G(r)$, as required for non-malleability. Details can be found in the full version [3].

Replacing the H-oracle without violating IND-CCA2 security is more ambitious and we require a very strong assumption on the pseudorandom generator, called non-malleability under chosen-image attacks (where the adversary can also make inversion queries to the trapdoor pseudorandom generator). Since any pseudorandom generator with this property is already close to a chosen-ciphertext secure encryption scheme, we rather see this as an indication that a partial instantiation might be possible and that separation results as [12, 19, 20, 1, 17, 21, 9, 14] seem to be hard to find. The formal treatment of the following and the proof appear in the full version [10].

Theorem 3. *Let F be trapdoor permutation family and let $F_{\textsf{t-clear}}$ be the induced partial one-way trapdoor permutation family defined in Section 2.1. Let $\mathcal{H} = (\mathsf{KGenH}, \mathsf{H}, \mathsf{TdH})$ be a trapdoor pseudorandom generator which is non-malleable under chosen-image attacks (with respect to $\mathsf{hint}(x) = (f, f(x))$ for $(f, f^{-1}) \leftarrow F_{\textsf{t-clear}}(1^k)$). Then the partial H-instantiation $OAEP^{G,\mathcal{H}}[F_{\textsf{t-clear}}]$ through $\mathcal{H}$ is IND-CCA2 in the random oracle model.*

4 Full Instantiation for OAEP

In this section we prove that there exists a full instantiation of $\mathrm{OAEP}_{\textsf{lsb}||\textsf{t-clear}}$ which is secure in the sense of \$NM-CPA in the standard model, implying for example that the scheme is OW-CPA. Recall that in $\mathrm{OAEP}_{\textsf{lsb}||\textsf{t-clear}}$ we write $s||\gamma = G(s) \oplus M||0^{k_1}$ instead of s to name the least significant bits explicitly.

To prove our result we need a near-collision resistant *trapdoor* pseudorandom generator, i.e., which combines near-collision resistance with the trapdoor property. Such generators can be easily built by using again the Blum-Micali-Yao generator, but this time by deploying a trapdoor permutation g instead of a one-way permutation, i.e., the generator's output for random r is given by $\mathsf{G}_{\mathrm{YBM}}(g, r) = (\mathrm{hb}(r), \mathrm{hb}(g(r)), \ldots, \mathrm{hb}(g^{n-1}(r)), g^n(r))$. Letting K^{-1} contain the trapdoor information g^{-1} algorithm TdG can easily invert the k_1 least significant bits y of the output to recover a pre-image r.

To be precise we make use of two additional, specific properties of the Blum-Micali-Yao generator. First, we assume that recovering a pre-image is possible given the k_1 least significant bits only, i.e., without seeing the remaining part of the image. To simplify the proof we furthermore presume that the k_1 least significant bits of the generator's output are statistically close to uniform (over the choice of the seed).[7] We simply refer to generators with the above properties as a *near-collision resistant trapdoor pseudorandom generator (for the least significant k bits).*

Theorem 4. *Let F be trapdoor permutation family and let $F_{\textsf{lsb}||\textsf{t-clear}}$ be the induced partial one-way trapdoor permutation family. Let $\mathcal{G} = (\mathsf{KGenG}, \mathsf{G})$ be a*

[7] It is easy to adapt the proof to the more general case of arbitrary distributions of the least significant bits, as long as they support extraction. But this would also require to change the definition of the non-malleable pseudorandom generator $\mathsf{G}_{\mathrm{KG}}(s||\gamma)$ to support arbitrary distributions on the γ-part.

near-collision resistant trapdoor pseudorandom generator (for the k_1 least significant bits). Let $\mathcal{H} = (\mathsf{KGenH}, \mathsf{H})$ be a generator which is pseudorandom and non-malleable with respect to $\mathsf{hint}(s||\gamma) = (f, f(s)||\gamma)$ for $(f, f^{-1}) \leftarrow F(1^k)$. Then the full instantiation OAEP$^{\mathcal{G},\mathcal{H}}[F_{\mathsf{lsb}||\mathsf{t\text{-}clear}}]$ through $\mathcal{G}$ and $\mathcal{H}$ is \$NM-CPA.

The proof appears in the full version [10]. The basic idea is similar to the one of NM-CPA security for the partial H-instantiation. The important difference is that the randomness of the encrypted message M in a ciphertext $f(s)||\gamma||t$ for $s||\gamma = \mathsf{G}_K(r) \oplus M||0^{k_1}$ helps to overcome otherwise existing "circular" dependencies between $\mathcal{G}$ and $\mathcal{H}$ in the computations of ciphertexts (which, in the partial instantiation case, do not occur due to the fact that G is a random oracle).

5 Hybrid Encryption from \$NM-CPA Schemes

We show that a public-key scheme which is secure in the sense of \$NM-CPA (i.e., for pre-defined relations), together with an IND-CCA2 secure symmetric scheme suffices to build a NM-CPA secure hybrid scheme in the random oracle model (i.e., even for adaptively chosen message distributions and relations).

Construction 1. *Let $\mathsf{AS} = (\mathcal{EK}_{asym}, \mathcal{E}_{asym}, \mathcal{D}_{asym})$ be an asymmetric encryption scheme and let $\mathsf{SS} = (\mathcal{EK}_{sym}, \mathcal{E}_{sym}, \mathcal{D}_{sym})$ be a symmetric encryption scheme. Let G be a hash function mapping k-bit strings into the key space of the symmetric scheme. Then the hybrid encryption scheme $\mathsf{AS}' = (\mathcal{EK}'_{asym}, \mathcal{E}'_{asym}, \mathcal{D}'_{asym})$ is defined as follows.*

- *The key generation algorithm $\mathcal{EK}'_{asym}(1^k)$ outputs a key pair $(\mathrm{sk}, \mathrm{pk}) \stackrel{\$}{\leftarrow} \mathcal{EK}_{asym}(1^k)$.*
- *The encryption algorithm $\mathcal{E}'_{asym}$ on input pk, M picks $r \stackrel{\$}{\leftarrow} \{0,1\}^k$, computes $C_{asym} \stackrel{\$}{\leftarrow} \mathcal{E}_{asym}(\mathrm{pk}, r)$, $C_{sym} \stackrel{\$}{\leftarrow} \mathcal{E}_{sym}(G(r), M)$ and returns (C_{asym}, C_{sym}).*
- *The decryption algorithm $\mathcal{D}'_{asym}$ on input (C_{asym}, C_{sym}) and sk computes $r \leftarrow \mathcal{D}_{asym}(\mathrm{sk}, C_{asym})$, $M \leftarrow \mathcal{D}_{sym}(G(r), C_{sym})$ and returns M.*

Theorem 5. *Let $\mathsf{AS} = (\mathcal{EK}_{asym}, \mathcal{E}_{asym}, \mathcal{D}_{asym})$ be an asymmetric encryption scheme which is \$NM-CPA. Let $\mathsf{SS} = (\mathcal{EK}_{sym}, \mathcal{E}_{sym}, \mathcal{D}_{sym})$ be an IND-CCA2 symmetric encryption scheme. Let G be a hash function and assume $\mathsf{AS}' = (\mathcal{EK}'_{asym}, \mathcal{E}'_{asym}, \mathcal{D}'_{asym})$ is the hybrid encryption scheme defined according to Construction 1. Then AS' is NM-CPA secure in the random oracle model.*

The proof is in the full version [10] and actually shows that the scheme is NM-CPA with respect to the stronger notion where the adversary outputs a sequence $\mathbf{C} = (C_1, \ldots, C_m)$ of ciphertexts and the success is measured according to $R(M^*, \mathbf{M})$ for $\mathbf{M} = (M_1, \ldots, M_m)$.

Acknowledgments

We thank the anonymous reviewers for comments. Part of the work done while both authors were visiting Centre de Recerca Matematica (CRM) and Technical

University of Catalonia (UPC), Barcelona, Spain, whose support is highly appreciated. The second author was also supported by the Emmy Noether Program Fi 940/2-1 of the German Research Foundation (DFG).

References

1. M. Bellare, A. Boldyreva and A. Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In *Eurocrypt 2004*, Volume 3027 of *LNCS*, pp. 171–188. Springer-Verlag, 2004.
2. D. R. L. Brown. Unprovable Security of RSA-OAEP in the Standard Model. *Cryptology ePrint Archive, Report 2006/223*, 2006.
3. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In *CRYPTO '98*, Volume 1462 of *LNCS*, pp. 26–45. Springer-Verlag, 1998.
4. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS '93*, pp. 62–73. ACM, 1993.
5. M. Bellare and P. Rogaway. Optimal asymmetric encryption – how to encrypt with RSA. In *Eurocrypt '94*, Volume 950 of of *LNCS*, pp. 92–111. Springer-Verlag, 1995.
6. E. Biham and R. Chen. Near-Collisions of SHA-0. In *CRYPTO' 2004*, Volume 3152 of *LNCS*, pp. 290–305. Springer-Verlag, 2004.
7. D. Bleichenbacher. Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1. In *CRYPTO '98*, Volume 1462 of *LNCS*, pp. 1–12. Springer-Verlag, 1998.
8. M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *Journal on Computing*, Volume 13, pp. 850–864, SIAM, 1984.
9. A. Boldyreva and M. Fischlin. Analysis of random-oracle instantiation scenarios for OAEP and other practical schemes. In *CRYPTO 2005*, Volume 3621 of *LNCS*, pp. 412–429. Springer-Verlag, 2005.
10. A. Boldyreva and M. Fischlin. On the Security of OAEP. Full version of this paper, available from the authors' homepages. 2006.
11. R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *CRYPTO '97*, Volume 1294 of *LNCS*. pp. 455–469. Springer-Verlag, 1997.
12. R. Canetti, O. Goldreich and S. Halevi. The random oracle methodology, revisited. In *STOC '98*, pp. 209–218. ACM, 1998.
13. R. Canetti, D. Micciancio and O. Reingold. Perfectly one-way probabilistic hash functions. In *STOC '98*, pp. 131–140. ACM, 1998.
14. Y. Dodis, R. Oliveira, and K. Pietrzak. On the generic insecurity of full-domain hash. In *CRYPTO 2005*, Volume 3621 of *LNCS*, pp. 449–466. Springer-Verlag, 2005.
15. D. Dolev, C. Dwork and M. Naor. Non-malleable cryptography. *Journal on Computing*, Vol. 30(2), pp. 391–437. SIAM, 2000.
16. E. Fujisaki, T. Okamoto, D. Pointcheval and J. Stern. RSA-OAEP is secure under the RSA assumption. In *CRYPTO 2001*, volume 2139 of *LNCS*, pp. 260–274. Springer-Verlag, 2001.
17. S. Goldwasser and Y. T. Kalai. On the (in)security of the Fiat-Shamir paradigm. In *FOCS 2003*. IEEE, 2003.

18. IETF-TLS Working Group. Transport Layer Security. `http://www.ietf.org/html.charters/tls-charter.html`, November 2005.
19. U. Maurer, R. Renner and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In *TCC 2004*, volume 2951 of *LNCS*, pp. 21–39. Springer-Verlag, 2004.
20. J. Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *CRYPTO 2002*, volume 2442 of *LNCS*, pp. 111–126. Springer-Verlag, 2002.
21. P. Paillier and D. Vergnaud. Discrete-Log-Based Signatures May Not Be Equivalent to Discrete Log. In *Asiacrypt 2005*, volume 3788 of *LNCS*, pp. 1–20. Springer-Verlag, 2005.
22. A. Yao. Theory and applications of trapdoor functions. In *FOCS '82*, pp. 80–91. IEEE, 1982.