Second NESSIE Workshop (september 12 – 13, 2001, Egham, UK).

RSA–REACT: An Alternative to RSA–OAEP

Tatsuaki Okamoto[1] and David Pointcheval[2]

[1] NTT Labs, 1-1 Hikarino-oka, Yokosuka-shi, 239-0847 Japan
E-mail: okamoto@isl.ntt.co.jp
[2] Dépt d'Informatique, ENS – CNRS, 45 rue d'Ulm, 75230 Paris Cedex 05, France
E-mail: David.Pointcheval@ens.fr – URL: http://www.di.ens.fr/users/pointche

Abstract. The last few months, several new results appeared about the OAEP construction, and namely the RSA–OAEP cryptosystem. Whereas OAEP was believed to provide the highest security level (IND-CCA2), with an efficient exact security level, the effective security result had been showed to be incomplete. Nevertheless, the particular instantiation with RSA (which is anyway almost the sole application) had been eventually proven secure, but the security reduction appears to be quite inefficient. Therefore, with respect to the provable security result, RSA–OAEP with a 1024-bit modulus just provides a 2^{40} security level.
Several alternatives have been recently proposed, but most of them face the same problem with a quadratic time security reduction. Excepted the recent generic conversion, called REACT, which admits a linear time reduction. Consequently, RSA–REACT appears to be the best alternative to RSA–OAEP, granted the high security level, even with real world parameters. RSA–REACT with a 1024-bit modulus indeed guarantees a 2^{80} security level (IND-CCA2 under the RSA assumption).
Furthermore, the full construction is already proven secure when integrating symmetric encryption, which guarantees the security of the overall communication.

1 Introduction

The OAEP conversion method [5] was introduced by Bellare and Rogaway in 1994 and was believed to provide semantic security against adaptive chosen-ciphertext attacks [17, 27], based on the one-wayness of a trapdoor permutation. Therefore, when Bleichenbacher published his attack on RSA–PKCS #1 v1.5 [28, 7], OAEP was the only efficient and "provably secure" construction. RSA–OAEP thus became the natural successor, the RSA–PKCS #1 v2.0.

Unfortunately, Shoup [30] recently showed that the security result was incomplete. More precisely, he gave a strong argument against the chosen-ciphertext security under the sole one-wayness of the permutation.

Did Shoup's result mean that OAEP (and consequently RSA–PKCS #1 v2.0) was insecure or that it was impossible to prove the security of OAEP? Fortunately not: Shoup's result did not preclude the possibility of proving the security of OAEP from stronger assumptions. And thus, Fujisaki, Stern and the authors [16] introduced a stronger assumption, the *partial-domain* one-wayness of the permutation, to prove that OAEP is semantically secure against adaptive chosen-ciphertext attack in the random oracle model. With the further result that the partial-domain one-wayness of the RSA function is equivalent to the (full-domain) one-wayness, we provided a complete argument for the security of RSA–OAEP under the sole one-wayness of the RSA function.

However, our proof reduction is not efficient. It is indeed a quadratic time reduction (as the original one [5]). Thus, it does not provide any guarantee for real world parameters. For example, RSA–OAEP with a 1024-bit modulus achieves a security level of 2^{40} only!

Several alternatives to OAEP have been recently proposed, such as OAEP+, by Shoup [30] himself, and SAEP/SAEP+ by Boneh [8]. But either they provided an efficient linear time reduction for small exponents only or inefficient quadratic time reduction for larger exponents. Since people do not trust RSA with small exponents [11, 9], these alternatives do not provide a better security level than RSA–OAEP, because of the bad reductions.

Only one construction provides guarantees, even for real world parameters, the Rapid Enhanced-security Asymmetric Cryptosystem Transform [22], which applies to most of the trapdoor one-way functions. This is a generalization of the construction suggested by Bellare and Rogaway for trapdoor permutations [4]. For this construction, we gave an efficient linear reduction which guarantees the security of RSA–REACT even for 1024-bit moduli.

2 Public-Key Encryption

The aim of public-key encryption is to allow anybody who knows the public key of Alice to send her a message that only she will be able to recover it through her private key.

2.1 Definitions

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

- The *key generation algorithm* $\mathcal{K}$. On input 1^k, where k is the security parameter, the algorithm $\mathcal{K}$ produces a pair $(\mathsf{pk}, \mathsf{sk})$ of matching public and secret keys. Algorithm $\mathcal{K}$ is probabilistic.
- The *encryption algorithm* $\mathcal{E}$. Given a message m and a public key pk, $\mathcal{E}$ produces a ciphertext c of m. This algorithm may be probabilistic.
- The *decryption algorithm* $\mathcal{D}$. Given a ciphertext c and the secret key sk, $\mathcal{D}$ returns the plaintext m.

2.2 Security Notions

The first security notion that one would like for an encryption scheme is *one-wayness*: starting with just public data, an attacker cannot recover the complete plaintext of a given ciphertext. More formally, this means that for any adversary $\mathcal{A}$, her success in inverting $\mathcal{E}$ without the secret key should be negligible over the probability space $\mathcal{M} \times \Omega$, where $\mathcal{M}$ is the message space and Ω is the space of the random coins r used for the encryption scheme, and the internal random coins of the adversary:

$$\mathsf{Succ}^{\mathsf{ow}}(\mathcal{A}) = \Pr_{m,r}[(\mathsf{pk}, sk) \leftarrow \mathcal{K}(1^k) : \mathcal{A}(\mathsf{pk}, \mathcal{E}_{\mathsf{pk}}(m; r)) = m].$$

However, many applications require more from an encryption scheme, namely *semantic security* (*a.k.a. polynomial security* or *indistinguishability of encryptions* [17], denoted IND): if the attacker has some information about the plaintext, for example that it is either "yes" or "no" to a crucial query, no adversary

should learn more with the view of the ciphertext. This security notion requires computational impossibility to distinguish between two messages, chosen by the adversary, one of which has been encrypted, with a probability significantly better than one half: her advantage $\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A})$, where the adversary $\mathcal{A}$ is seen as a 2-stage Turing machine $(\mathcal{A}_1, \mathcal{A}_2)$, should be negligible, where $\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A})$ is formally defined as.

$$2 \times \Pr_{b,r}\left[\begin{array}{l}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k), (m_0, m_1, s) \leftarrow \mathcal{A}_1(\mathsf{pk}),\\ c = \mathcal{E}_{\mathsf{pk}}(m_b; r) : \mathcal{A}_2(m_0, m_1, s, c) = b\end{array}\right] - 1.$$

Another notion was defined thereafter, the so-called *non-malleability* [12], in which the adversary tries to produce a new ciphertext such that the plaintexts are meaningfully related. This notion is stronger than the above one, but it is equivalent to semantic security in the most interesting scenario [3, 6].

On the other hand, an attacker can use many kinds of attacks: since we are considering asymmetric encryption, the adversary can encrypt any plaintext of her choice with the public key, hence *chosen-plaintext attack*. She may, furthermore, have access to more information, modeled by partial or full access to some oracles:

- a plaintext-checking oracle which, on input of a pair (m, c), answers whether c encrypts the message m. This attack has been named the *Plaintext-Checking Attack* (PCA) [22];
- a validity-checking oracle which, on input of a ciphertext c, just answers whether it is a valid ciphertext. This weak oracle (involved in the reaction attacks [18]) had been enough to break some famous encryption schemes [7, 20], namely PKCS #1 v1.5;
- or the decryption oracle itself, which on the input of any ciphertext, except the challenge ciphertext, responds with the corresponding plaintext (*non-adaptive/adaptive chosen-ciphertext attacks* [21, 27]).

The latter, the adaptive chosen-ciphertext attack denoted CCA2, is clearly the strongest one.

A general study of these security notions and attacks was given in [3], we therefore refer the reader to this paper for more details. However, the by now expected security level for public-key encryption schemes is semantic security against adaptive chosen-ciphertext attacks (IND-CCA2) – where the adversary just wants to distinguish which plaintext, between two messages of her choice, had been encrypted; she can ask any query she wants to a decryption oracle (except the challenge ciphertext). This is the strongest scenario one can define.

3 Review of OAEP

3.1 Description

We briefly describe the f-based OAEP cryptosystem $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ obtained from any permutation $f : \{0,1\}^k \longrightarrow \{0,1\}^k$, which can also be seen as

$$f : \{0,1\}^{n+k_1} \times \{0,1\}^{k_0} \longrightarrow \{0,1\}^{n+k_1} \times \{0,1\}^{k_0},$$

with $k = n + k_0 + k_1$, whose inverse is denoted by g. We need two hash functions G and H:

$$G : \{0,1\}^{k_0} \longrightarrow \{0,1\}^{n+k_1} \text{ and } H : \{0,1\}^{n+k_1} \longrightarrow \{0,1\}^{k_0}.$$

Then,

- $\mathcal{K}(1^k)$: specifies an instance of the function f, and of its inverse g. The public key pk is therefore f and the secret key sk is g.
- $\mathcal{E}_{\mathsf{pk}}(m; r)$: given a message $m \in \{0,1\}^n$, and a random value $r \in_R \{0,1\}^{k_0}$, the encryption algorithm $\mathcal{E}_{\mathsf{pk}}$ computes

  $$s = (m \| 0^{k_1}) \oplus G(r) \text{ and } t = r \oplus H(s),$$

  and outputs the ciphertext $c = f(s,t)$.
- $\mathcal{D}_{\mathsf{sk}}(c)$: thanks to the secret key, the decryption algorithm $\mathcal{D}_{\mathsf{sk}}$ extracts

  $$(s,t) = g(c), \text{ and next } r = t \oplus H(s) \text{ and } M = s \oplus G(r).$$

  If $[M]_{k_1} = 0^{k_1}$, the algorithm returns $[M]^n$, otherwise it returns "Reject".

In the above description, $[M]_{k_1}$ denotes the k_1 least significant bits of M, while $[M]^n$ denotes the n most significant bits of M.

3.2 Security Analysis

The Underlying Problems. In the original analysis of OAEP from [5], it is only required that f is a trapdoor one-way permutation. However, since [16], we have to consider additional related problems: the partial-domain one-wayness and the set partial-domain one-wayness of permutation f:

- (τ, ε)-One-Wayness of f, means that for any adversary $\mathcal{A}$ whose running time is bounded by τ, the success probability $\mathsf{Succ}^{\mathsf{ow}}(\mathcal{A})$ is upper-bounded by ε, where

  $$\mathsf{Succ}^{\mathsf{ow}}(\mathcal{A}) = \Pr_{s,t}[\mathcal{A}(f(s,t)) = (s,t)];$$

- (τ, ε)-Partial-Domain One-Wayness of f, means that for any adversary $\mathcal{A}$ whose running time is bounded by τ, the success probability $\mathsf{Succ}^{\mathsf{pd-ow}}(\mathcal{A})$ is upper-bounded by ε, where

  $$\mathsf{Succ}^{\mathsf{pd-ow}}(\mathcal{A}) = \Pr_{s,t}[\mathcal{A}(f(s,t)) = s];$$

- $(\ell, \tau, \varepsilon)$-Set Partial-Domain One-Wayness of f, means that for any adversary $\mathcal{A}$ that outputs a set of ℓ elements within time bound τ, the success probability $\mathsf{Succ}^{\mathsf{s-pd-ow}}(\mathcal{A})$ is upper-bounded by ε, where

  $$\mathsf{Succ}^{\mathsf{s-pd-ow}}(\mathcal{A}) = \Pr_{s,t}[s \in \mathcal{A}(f(s,t))].$$

We denote by $\mathsf{Succ}^{\mathsf{ow}}(\tau)$, (resp. $\mathsf{Succ}^{\mathsf{pd-ow}}(\tau)$ and $\mathsf{Succ}^{\mathsf{s-pd-ow}}(\ell, \tau)$) the maximal success probability $\mathsf{Succ}^{\mathsf{ow}}(\mathcal{A})$ (resp. $\mathsf{Succ}^{\mathsf{pd-ow}}(\mathcal{A})$ and $\mathsf{Succ}^{\mathsf{s-pd-ow}}(\mathcal{A})$). The maximum ranges over all adversaries whose running time is bounded by τ. In the third case, there is an obvious additional restriction on this range from the fact that $\mathcal{A}$ outputs sets with ℓ elements.

Security Results. In their paper [5], Bellare and Rogaway provided a security analysis, which proved that the OAEP construction together with any trapdoor one-way permutation is semantically security and *weakly* plaintext-aware. Unfortunately, this just proves semantic security against non-adaptive chosen-ciphertext attacks (*a.k.a.* lunchtime attacks [21] or IND-CCA1). Even if the achieved security was believed to be stronger (namely IND-CCA2), it had never been proven. Actually, Shoup [30] recently showed that it is quite unlikely that such a security proof exists, for any trapdoor one-way permutation, but maybe under a stronger assumption on the permutation.

In [16], we introduced such a stronger assumption to provide the plaintext-awareness, and thus to prove the following result.

Theorem 1. *Let $\mathcal{A}$ be a CCA2–adversary against the "semantic security" of the OAEP conversion $(\mathcal{K}, \mathcal{E}, \mathcal{D})$, with advantage ε and running time t, making q_D, q_G and q_H queries to the decryption oracle, and the hash functions G and H respectively. Then, $\mathsf{Succ}^{\mathsf{s-pd-ow}}(q_H, t')$ is greater than*

$$\frac{\varepsilon}{2} - \frac{2q_Dq_G + q_D + q_G}{2^{k_0}} - \frac{2q_D}{2^{k_1}},$$

where $t' \leq t + q_G \cdot q_H \cdot (T_f + \mathcal{O}(1))$, and T_f denotes the time complexity of function f.

4 RSA–OAEP

The main application of OAEP is certainly the famous RSA–OAEP, which has been used to update the PKCS #1 standard [28], granted the believed security result.

4.1 Description: the RSAES–OAEP

The description of RSA–OAEP seems straightforward, but one problem had to be dealt with, since the RSA function does not map any $\{0,1\}^k$ into $\{0,1\}^k$. The RSAES–OAEP proposal (in the PKCS #1 v2.0 standard, and in the NESSIE submission [19]) uses a padding on 1 byte less than the size of the modulus.

Furthermore, in the RSAES–OAEP encoding, the padding differs a little bit from the original proposition, since the redundancy 0^{k_1} is replaced by another kind of redundancy (a hash value), and the order of s and t is inverted before applying the RSA function: the value $\mathsf{maskedSeed} \| \mathsf{maskedDB}$ is the input to f, the RSA function, where maskedSeed plays the role of t, and maskedDB the role of s. But these latter modifications do not affect the security analysis.

However, the difference between the size of the padding and the length of the modulus affects it a lot: let us consider an ℓ-bit modulus N. Define $k = 8(|N|_{\mathsf{byte}} - 1)$ and $k = n + k_0 + k_1$, where k_0 and k_1 are security parameters, and n is the size of the messages to be encrypted. The padding of size k is still the same (see figure 1), involving two hash functions G and H:

$$G : \{0,1\}^{k_0} \longrightarrow \{0,1\}^{n+k_1} \text{ and } H : \{0,1\}^{n+k_1} \longrightarrow \{0,1\}^{k_0}.$$

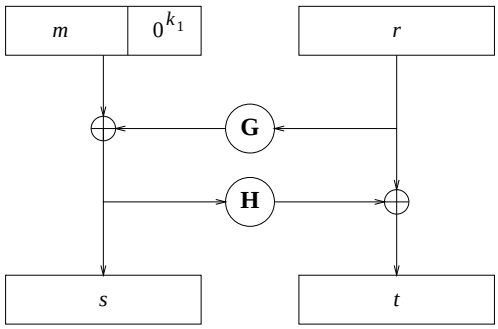

Fig. 1. OAEP Padding

But the function f

$$f : \{0,1\}^k \longrightarrow \mathbb{Z}_N^\star$$
$$x \longmapsto x^e \bmod N$$

is no longer a permutation, since the e-th roots of elements in $\mathbb{Z}_N^\star$ are not necessarily smaller than 2^k. Indeed, this only happens with average probability between 2^{-7} and 2^{-8}. However, a full security analysis can be driven, but it leads to a more inefficient reduction [31].

4.2 Security Result

In his paper [30], Shoup was able to repair the security result for a small exponent, $e = 3$, using Coppersmith's algorithm from [10]. However, the above result can be applied to repair RSA–OAEP, regardless of the exponent. Thanks to the multiplicative random self-reducibility of RSA, the partial-domain one-wayness of RSA is indeed equivalent to that of the whole RSA problem, as soon as a constant fraction of the most significant bits (or the least significant bits) of the pre-image can be recovered:

Lemma 2 (see [16]). *Let $\mathcal{A}$ be an algorithm that outputs a q-set containing $\ell - k_0$ of the most significant bits of the e-th root of its input (partial-domain RSA, for any $2^{\ell-1} < N < 2^\ell$, with $\ell > 2k_0$), within time bound t, with probability ε. There exists an algorithm $\mathcal{B}$ that solves the RSA problem (N, e) with success probability ε', within time bound t' where*

$$\varepsilon' \geq \varepsilon \times (\varepsilon - 2^{2k_0-\ell+6}),$$
$$t' \leq 2t + q^2 \times \mathcal{O}(\ell^3).$$

Thanks to this lemma, in [16], we immediately derived a security theorem for RSA–OAEP, which is not completely true for the RSAES–OAEP, because of the above remark. In the reduction, when one is given, or gets from a random self-reduction, an element $y \in \mathbb{Z}_N^\star$ to be used as the challenge ciphertext, it is a correct one (with an e-th root smaller than 2^k) only with probability greater than $1/256$. Therefore, one can only state the following security result:

Theorem 3. *Let $\mathcal{A}$ be a CCA2–adversary against the "semantic security" of RSAES–OAEP (with a ℓ-bit long modulus, with $\ell > 2k_0$), with running time bounded by t and advantage ε, making q_D, q_G and q_H queries to the decryption oracle, and the hash functions G and H respectively. Then, the RSA problem can be solved with probability ε' greater than*

$$\frac{\varepsilon^2}{4} - \varepsilon \cdot \left(\frac{2q_D q_G + q_D + q_G}{2^{k_0}} + \frac{2q_D}{2^{k_1}} + \frac{32}{2^{k-2k_0}} \right)$$

within time bound $t' \leq 2^{17}t + q_H \cdot (q_H + 2q_G) \times \mathcal{O}(\ell^3)$.

Which is a totally inefficient security reduction, since it is quadratic in the number of queries to the random oracles, and it furthermore has to run thousands of time the adversary to invert the RSA function.

5 Some OAEP Alternatives

5.1 The OAEP+ Padding

In his paper [30], Shoup also proposed a formal security proof of RSA–OAEP with a much more efficient security reduction, but in the particular case where the encryption exponent e is equal to 3. However many people think that RSA with exponent 3 may be weaker than with greater exponents [11, 9]. Therefore, he also proposed a slightly modified version of OAEP, called OAEP+, which can be proven secure, under the sole one-wayness of the permutation, whatever the exponent is.

5.2 The SAEP+ Padding

Boneh [8] recently proposed two new paddings to be used with the Rabin primitive [26] or RSA. They are simpler than OAEP, hence the name Simplified Asymmetric Encryption Padding. Indeed, whereas OAEP is a two-round Feistel network [13], SAEP is just a single-round. Unfortunately, as it was with the improved security reduction for OAEP proposed by Shoup, the SAEP conversion only works with low exponents ($e = 2$, the Rabin primitive, or $e = 3$). Nevertheless, the SAEP+ padding works with any exponent, of any size.

5.3 Security Reductions

Even though OAEP and SAEP with small exponents admit efficient reductions (linear in the number of oracle queries), OAEP+ and SAEP+ both have only been provided with more expensive reductions (quadratic in the number of oracle queries, as in the recent RSA–OAEP result), with possibly a further loss in the success probability, which does not guarantee anything for practical parameters.

6 Rapid Enhanced-security Asymmetric Cryptosystem Transform

In [22], the authors proposed a new generic conversion, we called REACT. It is an efficient conversion, which admits a very efficient security reduction.

6.1 Description of REACT

The Basic Conversion. Let us describe this generic conversion [22] on any asymmetric encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$

$$\mathcal{E}_{\mathsf{pk}} : \mathcal{M}_{\mathsf{pk}} \times \Omega_{\mathsf{pk}} \to \mathcal{C}_{\mathsf{pk}} \qquad \mathcal{D}_{\mathsf{sk}} : \mathcal{C}_{\mathsf{pk}} \to \mathcal{M}_{\mathsf{pk}},$$

where $\mathcal{M}_{\mathsf{pk}}$ is the messages space, $\mathcal{C}_{\mathsf{pk}}$ is the ciphertexts space and Ω_{pk} is the random coins space, which may depend on the public key pk. We also need two hash functions G and H,

$$G : \mathcal{M}_{\mathsf{pk}} \to \{0,1\}^{k_1}, H : \mathcal{M}_{\mathsf{pk}} \times \{0,1\}^{k_1} \times \mathcal{C}_{\mathsf{pk}} \times \{0,1\}^{k_1} \to \{0,1\}^{k_2},$$

where k_1 and k_2 are security parameters. The REACT conversion is depicted on Figure 2.

$\mathcal{K}'$**: Key Generation**
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k)$ $\rightarrow (\mathsf{pk}, \mathsf{sk})$
$\mathcal{E}'$**: Encryption of** $m \in \mathcal{M}' = \{0,1\}^{k_1} \rightarrow (a, b, c)$
$R \in \mathcal{M}_{\mathsf{pk}}$ and $r \in \Omega_{\mathsf{pk}}$ are randomly chosen $c_1 = \mathcal{E}_{\mathsf{pk}}(R; r) \qquad c_2 = m \oplus G(R) \qquad c_3 = H(R, m, a, b)$ $\rightarrow (c_1, c_2, c_3)$ is the ciphertext
$\mathcal{D}'$**: Decryption of** (c_1, c_2, c_3)
Given $c_1 \in \mathcal{C}_{\mathsf{pk}}$, $c_2 \in \{0,1\}^{k_1}$ and $c_3 \in \{0,1\}^{k_2}$ $R = \mathcal{D}_{\mathsf{sk}}(c_1) \qquad m = c_2 \oplus G(R)$ if $c_3 = H(R, m, c_1, c_2)$ and $R \in \mathcal{M}_{\mathsf{pk}} \rightarrow m$ is the plaintext (otherwise, "Reject: invalid ciphertext")

Fig. 2. Rapid Enhanced-security Asymmetric Cryptosystem Transform

Then, the new scheme $(\mathcal{K}', \mathcal{E}', \mathcal{D}')$ works as follows:

- $\mathcal{K}'(1^k)$: it simply runs $\mathcal{K}(1^k)$ to get a pair of keys $(\mathsf{sk}, \mathsf{pk})$, and outputs it.
- $\mathcal{E}'_{\mathsf{pk}}(m; R, r)$: for any k_1-bit message m and random values $R \in \mathcal{M}_{\mathsf{pk}}$ and $r \in \Omega_{\mathsf{pk}}$, it gets $c_1 = \mathcal{E}_{\mathsf{pk}}(R; r)$, then it computes the session key $K = G(R)$, $c_2 = K \oplus m$ as well as $c_3 = H(R, m, c_1, c_2)$. The ciphertext consists of the triple $C = (c_1, c_2, c_3)$.
- $\mathcal{D}'_{\mathsf{sk}}(c_1, c_2, c_3)$: it first extracts R from c_1 by decrypting it, $R = \mathcal{D}_{\mathsf{sk}}(c_1)$. It verifies whether $R \in \mathcal{M}_{\mathsf{pk}}$. It can therefore recover the session key $K = G(R)$ and $m = K \oplus c_2$ which is returned if and only if $c_3 = H(R, m, c_1, c_2)$ and $R \in \mathcal{M}_{\mathsf{pk}}$. Otherwise, it outputs "Reject".

The overload is minimal. Actually, if we consider the encryption phase, it just adds the computation of two hash values and one XOR. Concerning the decryption phase, which had been made heavy in some previous conversions [14, 15, 25] with a re-encryption to check the validity, we also just add the computation of two hash values and one XOR, as in the encryption process.

The Hybrid Conversion. As it has already been done with some previous encryption schemes [14, 15, 23–25], the "one-time pad" encryption can be generalized to any symmetric encryption scheme which is not perfectly secure, but semantically secure against passive attacks.

Let us consider two encryption schemes, $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ is a $\mathsf{OW\text{-}PCA}$–secure asymmetric scheme and $(\mathbf{SymE}, \mathbf{SymD})$ is a IND–secure symmetric scheme on λ-bit long messages, which uses k_1-bit long keys, as well as two hash functions G and H which output k_1-bit strings and k_2-bit strings respectively. Then, the hybrid scheme $(\mathcal{K}^{\mathsf{hyb}}, \mathcal{E}^{\mathsf{hyb}}, \mathcal{D}^{\mathsf{hyb}})$ works as follows:

- $\mathcal{K}^{\mathsf{hyb}}(1^k)$: exactly has above, for $\mathcal{K}(1^k)$.
- $\mathcal{E}^{\mathsf{hyb}}_{\mathsf{pk}}(m; R, r)$: for any λ-bit message m and random values $R \in \mathcal{M}_{\mathsf{pk}}$ and $r \in \Omega_{\mathsf{pk}}$, it gets $c_1 = \mathcal{E}_{\mathsf{pk}}(R; r)$ and a random session key $K = G(R)$. It computes $c_2 = \mathbf{SymE}_K(m)$ as well as the checking part $c_3 = H(R, m, c_1, c_2)$. The ciphertext consists of $C = (c_1, c_2, c_3)$.
- $\mathcal{D}^{\mathsf{hyb}}_{\mathsf{sk}}(c_1, c_2, c_3)$: it first extracts R from c_1 by decrypting it, $R = \mathcal{D}_{\mathsf{sk}}(c_1)$. It verifies whether $R \in \mathcal{M}_{\mathsf{pk}}$ or not. It can therefore recover the session key $K = G(R)$ as well as the plaintext $m = \mathbf{SymD}_K(c_2)$ which is returned if and only if $c_3 = H(R, m, c_1, c_2)$ and $R \in \mathcal{M}_{\mathsf{pk}}$. Otherwise, it outputs "Reject".

The overload is similar to the previous conversion one, but then, the plaintext can be longer. Furthermore, the required property for the symmetric encryption is very weak: it is just required to be semantically secure in the basic scenario (no plaintext/ciphertext attacks).

6.2 Security Result

About the (basic) converted scheme, one can claim that if an attacker, against the semantic security in a chosen-ciphertext scenario, can gain an advantage ε after q_D, q_G and q_H queries to the decryption oracle and to the random oracles G and H respectively, within a time t, then one can design an algorithm that outputs, for any given C, the corresponding plaintext, after less than $q_G + q_H$ queries to the Plaintext-Checking Oracle, with probability greater than $\varepsilon/2 - q_D/2^{k_2}$, within a time $t + (q_G + q_H)T_{\mathsf{pca}}$, where T_{pca} denotes the time required by the PCA oracle to answer any query.

This security result, in the random oracle model, comes from two distinct remarks:

- the adversary has necessarily asked either $G(R)$ or $H(R, m_i, c_1, c_2)$ to get any information about the encrypted message m (either m_0 or m_1). Which means that for a given $c_1 = \mathcal{E}_{\mathsf{pk}}(R; r)$, R is in the list of queries asked to G or to H. Simply asking for the $q_G + q_H$ candidates to the Plaintext-Checking Oracle, one can output the right one. Then, with probability $\varepsilon/2$, one inverts $\mathcal{E}_{\mathsf{pk}}$, after $(q_G + q_H)$ queries to the Plaintext-Checking Oracle.
- However, in the chosen-ciphertext scenario, the adversary may ask queries to the decryption oracle. We have to simulate it. To any query (c_1, c_2, c_3) asked by the adversary to the decryption oracle, one looks at all the pairs (R, m) such that (R, m, c_1, c_2) has been asked to the random oracle H. For any such

R, one asks to the Plaintext-Checking Oracle whether c_1 is a ciphertext of R (remark that it does not make more queries to the Plaintext-Checking Oracle, since it has already been taken in account above). Then it computes $K = G(R)$, maybe using a simulation of G if the query R has never been asked. If $c_2 = K \oplus m$ then one outputs m as the plaintext of the triple (c_1, c_2, c_3). Therefore, any correctly computed ciphertext is decrypted by the simulator. But if the adversary has not asked $H(R, m, c_1, c_2)$ the probability that the ciphertext is valid, and thus the decryption not correctly simulated, is less than $1/2^{k_2}$.

For the hybrid construction, the proof is a bit more intricate, because of the symmetric encryption, but one can claim [22]:

Theorem 4. *Let us consider a* CCA2*–adversary* $\mathcal{A}^{\mathsf{cca2}}$ *against the "semantic security" of the conversion* $(\mathcal{K}^{\mathsf{hyb}}, \mathcal{E}^{\mathsf{hyb}}, \mathcal{D}^{\mathsf{hyb}})$*, on* λ*-bit long messages, within a time bounded by* t*, with advantage* ε*, after* q_D*,* q_G *and* q_H *queries to the decryption oracle, and the hash functions* G *and* H *respectively. Then for any* $0 < \nu < \varepsilon$*, and*

$$t' \leq t + q_G T_{\mathsf{sym}} + (q_H + q_G) T_{\mathsf{pca}}$$

(T_{sym} *and* T_{pca} *are the time complexity of* $\boldsymbol{SymE}_K$ *and the* PCA *oracle respectively), there either exists*

- *an adversary* $\mathcal{B}^{\mathsf{pca}}$ *against the* (t', φ)*-*OW-PCA*-security of the asymmetric encryption scheme* $(\mathcal{K}, \mathcal{E}, \mathcal{D})$*, after less than* $q_G + q_H$ *queries to the Plaintext-Checking Oracle, where*
$$\varphi = \frac{\varepsilon - \nu}{2} - \frac{q_D}{2^{k_2}}.$$
- *or an adversary* $\mathcal{B}$ *against the* (t', ν)*-*IND*–security of the symmetric encryption scheme* $(\boldsymbol{SymE}, \boldsymbol{SymD})$.

7 RSA–REACT

7.1 Description

Contrarily to OAEP, the specific instantiation of RSA–REACT is straightforward (see figure 3), since it works with any functions

$$\mathcal{E}_{\mathsf{pk}} : \mathcal{M}_{\mathsf{pk}} \times \Omega_{\mathsf{pk}} \to \mathcal{C}_{\mathsf{pk}} \qquad \mathcal{D}_{\mathsf{sk}} : \mathcal{C}_{\mathsf{pk}} \to \mathcal{M}_{\mathsf{pk}},$$

the encryption and the decryption algorithms, where $\mathcal{M}_{\mathsf{pk}}$ is the messages space, $\mathcal{C}_{\mathsf{pk}}$ is the ciphertexts space and Ω_{pk} is the random coins space. One has just to remark that for the RSA function, $\mathcal{M}_{\mathsf{pk}} = \mathcal{C}_{\mathsf{pk}} = \mathbb{Z}_N$. Since it is furthermore deterministic, Ω_{pk} is an empty set for any pk. Then, RSA–REACT works as follows:

- $\mathcal{K}(1^k)$: randomly choose a k-bit RSA modulus N, a public exponent e, relatively prime to $\varphi(N)$. Then, define $d = e^{-1} \bmod \varphi(N)$.

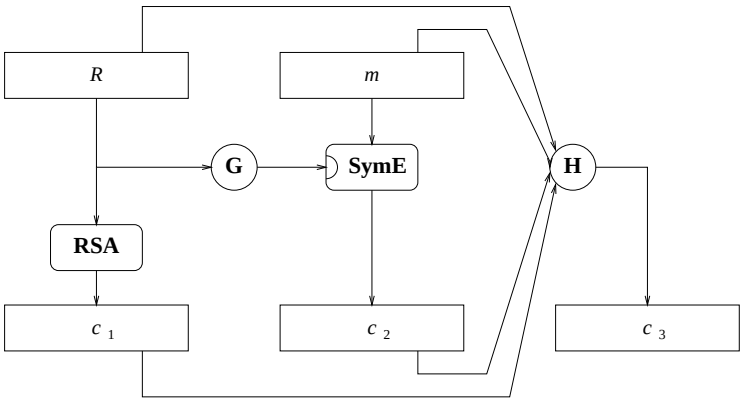

Fig. 3. RSA–REACT

- $\mathcal{E}_{\mathsf{pk}}(m; R)$: for any λ-bit message m and a random value $R \in \mathbb{Z}_N$, it gets $c_1 = R^e \bmod N$ and a random session key $K = G(R)$. It computes $c_2 = \mathbf{SymE}_K(m)$ as well as the checking part $c_3 = H(R, m, c_1, c_2)$. The ciphertext consists of $C = (c_1, c_2, c_3)$.
- $\mathcal{D}_{\mathsf{sk}}(c_1, c_2, c_3)$: it first extracts R from c_1 by decrypting it, $R = c_1^d \bmod N$. (Then, by construction, R necessarily lies in $\mathbb{Z}_N$.) It can therefore recover the session key $K = G(R)$ as well as the plaintext $m = \mathbf{SymD}_K(c_2)$ which is returned if and only if $c_3 = H(R, m, c_1, c_2)$. Otherwise, it outputs "Reject".

7.2 Security Result

One can thus simply apply the Theorem 4, with the particular RSA instantiation. But for a complete security result, one needs to know what OW-PCA means for RSA. Actually, since RSA is a deterministic function, a Plaintext-Checking Oracle on (R, c_1) simply consists in computing $R^e \bmod N$, and checking the result with the given challenge ciphertext c_1. Therefore, OW-PCA and OW-CPA are equivalent notions for RSA, and thus intractable for moduli large enough under the RSA assumption. Furthermore, the time required by an execution of the Plaintext-Checking Algorithm, in the RSA situation, is just that of one exponentiation to the power e. Thus:

Theorem 5. *Let us consider a CCA2–adversary $\mathcal{A}^{\mathsf{cca2}}$ against the "semantic security" of RSA–REACT, on λ-bit long messages, within a time bounded by t, with advantage ε, after q_D, q_G and q_H queries to the decryption oracle, and the hash functions G and H respectively. Then for any $0 < \nu < \varepsilon$, and*

$$t' \leq t + q_G T_{\mathsf{sym}} + (q_H + q_G) T_{\mathsf{rsa(e)}}$$

(T_{sym} is the time complexity of $\mathbf{SymE}_K$, and $T_{\mathsf{rsa(e)}}$ is the time complexity for an N-modular exponentiation to the power e, and thus is in $\mathcal{O}(k^3)$), there either exists

- *an algorithm $\mathcal{B}$ that inverts RSA, within a time bound t', with success probability greater than*

$$\frac{\varepsilon - \nu}{2} - \frac{q_D}{2^{k_2}}.$$

– *or an adversary* $\mathcal{B}$ *against the* (t', ν)*-IND–security of the symmetric encryption scheme* $(\boldsymbol{SymE}, \boldsymbol{SymD})$*, on* λ*-bit messages.*

7.3 Discussion

This latter theorem shows how efficient is the reduction. It is indeed in linear time, without any loss in the success probability, if the symmetric encryption is secure enough. Consequently, it guarantees the perfect equivalence with the RSA inversion, for moduli which require just a bit more than 2^{70} to be factored. This is achieved with 1024 bit-long moduli, which is anyway the minimal size currently advised.

In comparison to other proposals (OAEP, OAEP+, SAEP, SAEP+), REACT is a full scheme and not just a pure padding applied to the message before the RSA function. Consequently, the ciphertext is a bit longer. However, even if it can be used for key transport, it furthermore allows integration of a symmetric encryption scheme to achieve very high rates, as shown in the hybrid construction.

In an ISO report [29], Shoup suggested a possible OAEP-alternative, based on ideas from Bellare and Rogaway [4]. In that paper, Bellare and Rogaway proposed a generic construction from any trapdoor one-way permutation. Actually, it is a particular case of the above REACT construction. And thus, the suggested "simple RSA" is nothing else than a slight variant of RSA–REACT. However, it requires a stronger symmetric encryption scheme.

As Shoup remarked, thanks to the random self-reducibility of RSA, a high security level is still guaranteed even when encrypting many cleartexts [1, 2], in the "simple RSA", but also in the RSA–REACT construction. Unfortunately, the random self-reducibility of RSA is not preserved in the OAEP and SAEP variants, as we have already seen, because of the padding which only outputs uniformly distributed k-bit strings, but not uniformly distributed in $\mathbb{Z}_N$, or in $\mathbb{Z}_N^\star$.

8 Conclusion

Granted the efficient construction, the efficient linear time security reduction, and the above comments, RSA–REACT appears like the best alternative to RSA–OAEP. Even if the ciphertext is a bit longer, it does not make any difference with the classical use of asymmetric encryption, which is only used for the key transport, and then combined with a symmetric encryption scheme.

References

1. O. Baudron, D. Pointcheval, and J. Stern. Extended Notions of Security for Multicast Public Key Cryptosystems. In *Proc. of the 27th ICALP*, LNCS 1853, pages 499–511. Springer-Verlag, Berlin, 2000.
2. M. Bellare, A. Boldyreva, and S. Micali. Public-key Encryption in a Multi-User Setting: Security Proofs and Improvements. In *Eurocrypt '2000*, LNCS 1807, pages 259–274. Springer-Verlag, Berlin, 2000.

3. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, Berlin, 1998.
4. M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCS*, pages 62–73. ACM Press, New York, 1993.
5. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, Berlin, 1995.
6. M. Bellare and A. Sahai. Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. In *Crypto '99*, LNCS 1666, pages 519–536. Springer-Verlag, Berlin, 1999.
7. D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS #1. In *Crypto '98*, LNCS 1462, pages 1–12. Springer-Verlag, Berlin, 1998.
8. D. Boneh. Simplified OAEP for the RSA and Rabin Functions. In *Crypto '2001*, LNCS 2139, pages 275–291. Springer-Verlag, Berlin, 2001.
9. D. Boneh, G. Durfee, and M. Franklin. An Attack on RSA Given a Small Fraction of the Private Key Bits. In *Asiacrypt '98*, LNCS 1514, pages 25–34. Springer-Verlag, Berlin, 1998.
10. D. Coppersmith. Finding a Small Root of a Univariate Modular Equation. In *Eurocrypt '96*, LNCS 1070, pages 155–165. Springer-Verlag, Berlin, 1996.
11. D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10:233–260, 1997.
12. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
13. H. Feistel. Cryptography and Computer Privacy. *Scientific American*, 228(5):15–23, May 1973.
14. E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In *PKC '99*, LNCS 1560, pages 53–68. Springer-Verlag, Berlin, 1999.
15. E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Crypto '99*, LNCS 1666, pages 537–554. Springer-Verlag, Berlin, 1999.
16. E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA–OAEP is Secure under the RSA Assumption. In *Crypto '2001*, LNCS 2139, pages 260–274. Springer-Verlag, Berlin, 2001.
17. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
18. C. Hall, I. Goldberg, and B. Schneier. Reaction Attacks Against Several Public-Key Cryptosystems. In *Proc. of ICICS'99*, LNCS, pages 2–12. Springer-Verlag, 1999.
19. J. Jonsson and B. Kaliski. RSA–OAEP Encryption Scheme. Submission to NESSIE. September 2000.
20. M. Joye, J. J. Quisquater, and M. Yung. On the Power of Misbehaving Adversaries and Security Analysis of the Original EPOC. In *CT – RSA '2001*, LNCS 2020, pages 208–222. Springer-Verlag, Berlin, 2001.
21. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, New York, 1990.
22. T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In *CT – RSA '2001*, LNCS 2020, pages 159–175. Springer-Verlag, Berlin, 2001.
23. T. Okamoto, S. Uchiyama, and E. Fujisaki. EPOC: Efficient Probabilistic Public-Key Encryption. Submission to IEEE P1363a. November 1998.
24. D. Pointcheval. HD–RSA: Hybrid Dependent RSA – a New Public-Key Encryption Scheme. Submission to IEEE P1363a. October 1999.
25. D. Pointcheval. Chosen-Ciphertext Security for any One-Way Cryptosystem. In *PKC '2000*, LNCS 1751, pages 129–146. Springer-Verlag, Berlin, 2000.
26. M. O. Rabin. Digitalized Signatures. In R. Lipton and R. De Millo, editors, *Foundations of Secure Computation*, pages 155–166. Academic Press, New York, 1978.
27. C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, Berlin, 1992.
28. RSA Data Security, Inc. Public Key Cryptography Standards – PKCS.
29. V. Shoup. Editor's Contribution on Public Key Encryption. ISO/IEC JTC 1/SC27. February 13, 2001.
30. V. Shoup. OAEP Reconsidered. In *Crypto '2001*, LNCS 2139, pages 239–259. Springer-Verlag, Berlin, 2001.
31. Y. Verhoeven. Sécurité Prouvée des Schémas de Chiffrement. Technical report, LIENS, June 2001.