RSA–OAEP is Still Alive!

Eiichiro Fujisaki[1], Tatsuaki Okamoto[1], David Pointcheval[2], and Jacques Stern[2]

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

December 1, 2000

Abstract. Very recently Victor Shoup showed that the security result (security against adaptive chosen ciphertext attack) of OAEP, claimed by Bellare and Rogaway, is wrong. The flaw in their technique occurs when they simulate the decryption oracle. This paper presents a complete proof of the security of OAEP, and proves that OAEP is (semantically) secure (against adaptive chosen ciphertext attack) in the random oracle model, under the *partial* one-wayness of the underlying permutation, which is stronger than the originally claimed assumption, one-wayness of the underlying permutation. Since the partial one-wayness of the RSA function is equivalent to its one-wayness, as a result, although the security reduction is not tight, the security of RSA–OAEP can be proven under the RSA assumption.

1 Introduction

The OAEP conversion was introduced by Bellare and Rogaway in 1994 [3] and it was believed to provide semantically secure schemes against adaptive chosen ciphertext attacks [5, 6], relative to the one-wayness of a trapdoor permutation, using the (corrected) definition of plaintext-awareness [1].

Very recently, however, Victor Shoup [9] claimed that OAEP cannot be proven secure under the proof technique introduced by [3, 1]. He indicated that their proof technique, based on the (corrected) plaintext-awareness [1] cannot possibly yield a proof of the security, especially the non-malleability, of OAEP. He also proposed a modified version of OAEP, called OAEP+, which can be proven to be secure.

Does his result mean that OAEP is no more secure, or is it impossible to prove the security of OAEP? No! His result only implies that there is no hope of proving the plaintext-awareness of OAEP. In other words, his result does not deny the possibility of proving the security of OAEP by using another type of reduction technique.

This paper employs a different reduction technique: in our reduction, a computational assumption is introduced to prove the existence of a simulator of the decryption oracle. Based on this technique, we prove that OAEP is semantically secure against adaptive chosen ciphertext attack in the random oracle model [3], under the *partial* one-wayness of the underlying permutation, which is stronger than the originally claimed assumption, one-wayness of the underlying permutation. Since the partial one-wayness of the RSA function [7] is equivalent to its one-wayness, the security of RSA-OAEP can be proven under the one-wayness of the RSA function.

2 Review of OAEP

2.1 The Underlying Problems

Let us consider the 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 $k = n + k_0 + k_1$. In the OAEP description [3], it is only required to be a trapdoor one-way permutation. However, in the following, we consider two more kinds of problems related to the one-wayness of f: the partial one-wayness of f and the set partial one-wayness:

- (τ, ε)-One-Wayness of f, means that for any adversary $\mathcal{A}$ whose running time is bounded by τ, its 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-One-Wayness of f, means that for any adversary $\mathcal{A}$ whose running time is bounded by τ, its success probability $\mathsf{Succ}^{\mathsf{p-ow}}(\mathcal{A})$ is upper-bounded by ε, where
$$\mathsf{Succ}^{\mathsf{p-ow}}(\mathcal{A}) = \Pr_{s,t}[\mathcal{A}(f(s,t)) = s].$$
- (k, τ, ε)-Set Partial-One-Wayness of f, means that for any adversary $\mathcal{A}$ which outputs a set of k elements within a time bound τ, its success probability $\mathsf{Succ}^{\mathsf{sp-ow}}(\mathcal{A})$ is upper-bounded by ε, where
$$\mathsf{Succ}^{\mathsf{sp-ow}}(\mathcal{A}) = \Pr_{s,t}[s \in \mathcal{A}(f(s,t)].$$

We denote by $\mathsf{Succ}^{\mathsf{ow}}(\tau)$, (resp. $\mathsf{Succ}^{\mathsf{p-ow}}(\tau)$ and $\mathsf{Succ}^{\mathsf{sp-ow}}(k,\tau)$) the maximal success probability $\mathsf{Succ}^{\mathsf{ow}}(\mathcal{A})$ (resp. $\mathsf{Succ}^{\mathsf{p-ow}}(\mathcal{A})$ and $\mathsf{Succ}^{\mathsf{sp-ow}}(\mathcal{A})$) over all the adversaries whose running times are bounded by τ, and which output k-long sets in the latter case.

However, by randomly selecting an element in the k-long set returned by an adversary the Set Partial One-Wayness, one breaks the Partial One-Wayness with probability $\mathsf{Succ}^{\mathsf{sp-ow}}(\mathcal{A})/k$, and thus

$$\frac{1}{k} \times \mathsf{Succ}^{\mathsf{sp-ow}}(k,\tau) \leq \mathsf{Succ}^{\mathsf{p-ow}}(\tau) \leq \mathsf{Succ}^{\mathsf{ow}}(\tau).$$

But in some particular cases, more efficient reductions exists. Furthermore, in some cases, all these problems are polynomially equivalent (which is the case of the RSA permutation [7], hence the result in section 5).

2.2 The OAEP Cryptosystem

Let us now consider the OAEP conversion $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ obtained from this permutation f, whose inverse is denoted by g. We need G and H, two hash functions:

$$G : \{0,1\}^{k_0} \longrightarrow \{0,1\}^{k-k_0} \text{ and } H : \{0,1\}^{k-k_0} \longrightarrow \{0,1\}^{k_0}.$$

Then,

- $\mathcal{K}(1^k)$: specifies such a function f, and its inverse g. The public key pk is therefore the function f and the secret key sk is the function g.
- $\mathcal{E}_{\mathsf{pk}}(m)$: on any message $m \in \{0,1\}^n$, and a random value $r \stackrel{R}{\leftarrow} \{0,1\}^{k_0}$, one computes
$$s = m \| 0^{k_1} \oplus G(r) \text{ and } t = r \oplus H(s),$$
then the ciphertext is $c = f(s,t)$.
- $\mathcal{D}_{\mathsf{sk}}(c)$: thanks to the secret key, one extracts $(s,t) = g(c)$, then
$$r = t \oplus H(s) \text{ and } M = s \oplus G(r),$$
eventually, if $[M]_{k_1} = 0^{k_1}$, then it returns $[M]^n$, otherwise it returns "Reject".

In 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 Security Result

Let us turn to the security analysis. Indeed, we want to prove that this scheme is IND-CCA in the random oracle model [2], relative to the (partial) one-wayness of the function f. More precisely, one can claim the following exact security result.

Theorem 1. *For any CCA–adversary $\mathcal{A}$ against the "semantic security" of the OAEP conversion $(\mathcal{K}, \mathcal{E}, \mathcal{D})$, within a time bounded by t, after q_D, q_G and q_H queries to the decryption oracle, and the hash functions G and H respectively, his advantage ε is upper-bounded by*

$$2 \times \frac{\mathsf{Succ}^{\mathsf{ow}}(t') + q_H \times \mathsf{Succ}^{\mathsf{p-ow}}(t') + q_D \times \left(\frac{1}{2^{k_1-1}} + \frac{2q_G+1}{2^{k_0}} \right) + \frac{q_G}{2^{k-1}}}{1 - \frac{q_G}{2^{k_0}} - \frac{q_H}{2^{k-k_0}}},$$

where

$$t' \leq t + q_G \cdot q_H \cdot (T_f + \mathcal{O}(1)),$$

and T_f denotes the time complexity of the function f.

Instead of proving this general theorem relative to the partial one-wayness of the permutation, one can state it relative to the set partial one-wayness. Then, above theorem directly comes from the previous inequality, and the lemma claimed below.

Lemma 1. *For any CCA–adversary $\mathcal{A}$ against the "semantic security" of the OAEP conversion $(\mathcal{K}, \mathcal{E}, \mathcal{D})$, within a time bounded by t, after q_D, q_G and q_H queries to the decryption oracle, and the hash functions G and H respectively, his advantage ε is upper-bounded by*

$$2 \times \frac{\mathsf{Succ}^{\mathsf{ow}}(t') + \mathsf{Succ}^{\mathsf{sp-ow}}(t') + q_D \times \left(\frac{1}{2^{k_1 - 1}} + \frac{2q_G + 1}{2^{k_0}} \right) + \frac{q_G}{2^{k-1}}}{1 - \frac{q_G}{2^{k_0}} - \frac{q_H}{2^{k-k_0}}},$$

where

$$t' \leq t + q_G \cdot q_H \cdot (T_f + \mathcal{O}(1)),$$

and T_f denotes the time complexity of the function f.

4 Proof of the Lemma 1

For proving this lemma, we will proceed in two stages. The first one considers the classical semantic security of OAEP against chosen-plaintext attacks, under the one-wayness of the permutation f. We then provide a simulator for the decryption oracle, which is correct with an overwhelming probability under the partial one-wayness of permutation f. The latter part differs from the original proof [3], and corrects the flaw recently spotted [9].

4.1 Semantic Security

Thus, let us first claim the classical result about the IND-CPA of OAEP.

Lemma 2. *Let us consider a CPA–adversary $\mathcal{A}$ against the "semantic security" of $(\mathcal{K}, \mathcal{E}, \mathcal{D})$, within a time bounded by t, with advantage ε, after q_G and q_H queries to the hash functions G and H respectively. Then there exists an adversary $\mathcal{B}$ against the one-wayness of the permutation f, with a success probability ε', within a time bound t' where*

$$\varepsilon' \geq \frac{\varepsilon}{2} \times \left(1 - \frac{q_G}{2^{k_0}} - \frac{q_H}{2^{k-k_0}}\right) - \frac{q_G}{2^{k-1}},$$
$$t' \leq t + q_G \cdot q_H \cdot (T_f + \mathcal{O}(1)).$$

Proof. The semantic security of this scheme intuitively comes from the fact that for any adversary, in order to have any information about the encrypted message m, she must have asked the query r to G. But in order to have any information about the correct r, she must have asked s to H. Then the r and the s can be found in the queries asked to G and H, then $t = r \oplus H(s)$. The pair (s, t) provides the pre-image of the ciphertext c by the function f (hence the inversion of f.) This part is similar to the original one [3], then we postpone the complete proof in appendix A. □

4.2 Simulation of the Decryption.

If one wants to furthermore consider $\mathcal{A}$ as a chosen-ciphertext adversary, $\mathcal{B}$ has to be able to simulate the decryption oracle. Let us describe the behavior of a decryption simulator $\mathcal{DS}$. Then, we analyze the success probability of such a simulator.

Description of the Decryption Simulation. On a query $c = f(s,t)$ to the decryption oracle, $\mathcal{DS}$ looks into all the query-answer $(\gamma, G_\gamma) \in \mathsf{G\text{-}List}$ and $(\delta, H_\delta) \in \mathsf{H\text{-}List}$. For each pair of such query-answer, it defines

$$\sigma = \delta, \tau = \gamma \oplus H_\delta, \mu = G_\gamma \oplus \delta,$$

and checks whether for some pair

$$c = f(\sigma, \tau) \text{ and } [\mu]_{k_1} = 0^{k_1}.$$

If, for one of them, both equalities hold, then it outputs $[\mu]^n$. Otherwise, "Reject" is returned.

Security Claim. About this simulation, we can claim the following result, which repairs the previous proof [3], thanks to a computational assumption. Indeed, we show that the new cases to consider, because of the new definition of plaintext-awareness [1], are very unlikely under the partial one-wayness of the permutation f:

Lemma 3. *The decryption simulation $\mathcal{DS}$, on a ciphertext c, while at most one ciphertext $c^\star$ has been directly obtained from the encryption oracle, can correctly decrypt c with probability greater than ε', within a time bound t', where*

$$\varepsilon' \geq 1 - \left(\mathsf{Succ}^{\mathsf{sp-ow}}(t') + \frac{1}{2^{k_1 - 1}} + \frac{2q_G + 1}{2^{k_0}} \right)$$
$$t' \leq q_G \cdot q_H \cdot (T_f + \mathcal{O}(1)) .$$

Before any analysis, let us recall that the plaintext-extractor is given the ciphertext c to be decrypted, as well as the ciphertext $c^\star$ obtained from the decryption oracle and both lists G-List and H-List based on the interactions with the random oracles G and H. Let us first see that this simulation uniquely defines a possible plaintext, and thus can output the first one it finds. Indeed, with above definition, many pairs could make the equalities to be satisfied. But since the function f is one-to-one, the value of $\sigma = s$ is uniquely defined, and thus δ and H_δ. The same way, $\tau = t$ is uniquely defined, and thus γ and G_γ: at most one μ may be selected. Then either $[\mu]_{k_1} = 0^{k_1}$ or not.

For above remark, eventually note that the lists G-List and H-List correspond to input-output of functions G and H. And thus at most one output is related to a given input.

Furthermore, if the ciphertext has been correctly built by the adversary (r has been asked to G and s to H), the simulation will output the correct answer.

However, it will output "Reject" in any other situation, whereas the adversary may have built a valid ciphertext without asking both queries to the random oracles G and H.

More accurately, we can prove the following lemma which will be useful for evaluating the failure probability after many queries to the decryption simulator.

Lemma 4. *The decryption simulation $\mathcal{DS}$, on a ciphertext c, while at most one ciphertext $c^\star = f(s^\star, t^\star)$ has been directly obtained from the encryption oracle, but $s^\star$ has not been asked to H, can correctly decrypt c with probability greater than ε', within a time bound t', where*

$$\varepsilon' \geq 1 - \left(\frac{1}{2^{k_1 - 1}} + \frac{2q_G + 1}{2^{k_0}} \right)$$
$$t' \leq q_G \cdot q_H \cdot (T_f + \mathcal{O}(1)) \,.$$

Notations. In order to proceed to the analysis of the success probability of above plaintext-extractor, one needs to fix the notations. First, as done in the previous part, we denote with a * all the variables related to the challenge ciphertext $c^\star$, obtained from the encryption oracle. Indeed, this ciphertext, of either m_0 or m_1, implicitly defines some hash values, but maybe without appearing in the G nor H lists. All the other variables refer to the decryption query c, asked by the adversary to the decryption oracle, and thus to be decrypted by this simulation.

We also have to consider some events:

- as above, we denote by AskH the event that $s^\star$ has been asked to H;
- SBad denotes the event that $s = s^\star$;
- RBad denotes the event that $r = r^\star$, which means that
$$H(s) \oplus t = H(s^\star) \oplus t^\star;$$
- Bad denotes the union of these bad events, $\mathsf{Bad} = \mathsf{RBad} \vee \mathsf{SBad}$;
- AskR denotes the event that r has been asked to G;
- AskS denotes the event that s has been asked to H;
- AskRS denotes the intersection of these events, $\mathsf{AskRS} = \mathsf{AskR} \wedge \mathsf{AskS}$, and thus that both r and s have been asked to G and H respectively;
- Fail denotes the event that the above plaintext-extractor outputs a wrong decryption answer.

Remark that the Fail event is limited to the situation that the plaintext-extractor rejects a ciphertext whereas it would be accepted by the decryption oracle. Indeed, as soon as it accepts, it means that the ciphertext is really valid, with the output plaintext.

Success Probability. Clearly, we are interested in the probability of the event Fail, while $\neg\mathsf{AskH}$ occurred, which may be split due to other events:

$$\begin{aligned}
\mathsf{pr}[\mathsf{Fail} \,|\, \neg\mathsf{AskH}] &= \mathsf{pr}[\mathsf{Fail} \wedge \mathsf{Bad} \,|\, \neg\mathsf{AskH}] + \mathsf{pr}[\mathsf{Fail} \wedge \neg\mathsf{Bad} \,|\, \neg\mathsf{AskH}] \\
&= \mathsf{pr}[\mathsf{Fail} \wedge \mathsf{Bad} \,|\, \neg\mathsf{AskH}] + \mathsf{pr}[\mathsf{Fail} \wedge \neg\mathsf{Bad} \wedge \neg\mathsf{AskRS} \,|\, \neg\mathsf{AskH}] \\
&\quad + \mathsf{pr}[\mathsf{Fail} \wedge \neg\mathsf{Bad} \wedge \mathsf{AskRS} \,|\, \neg\mathsf{AskH}]
\end{aligned}$$

We can already simplify this expression, since the latter case cannot occur. About the previous one, if one note than

$$\mathsf{pr}[\neg\mathsf{AskRS}] = \mathsf{pr}[\neg\mathsf{AskR} \vee \neg\mathsf{AskS}] = \mathsf{pr}[\neg\mathsf{AskR}] + \mathsf{pr}[\neg\mathsf{AskS} \wedge \mathsf{AskR}],$$

and

$$\mathsf{pr}[\neg\mathsf{Bad}] = \mathsf{pr}[\neg\mathsf{RBad} \wedge \neg\mathsf{SBad}] \leq \min\{\mathsf{pr}[\neg\mathsf{RBad}], \mathsf{pr}[\neg\mathsf{SBad}]\}.$$

then,

$$\begin{aligned}
\mathsf{pr}[\mathsf{Fail} \wedge \neg\mathsf{Bad} \wedge \neg\mathsf{AskRS}] &= \mathsf{pr}[\mathsf{Fail} \wedge \neg\mathsf{Bad} \wedge \neg\mathsf{AskR}] \\
&\quad + \mathsf{pr}[\mathsf{Fail} \wedge \neg\mathsf{Bad} \wedge (\mathsf{AskR} \wedge \neg\mathsf{AskS})] \\
&\leq \mathsf{pr}[\mathsf{Fail} \wedge \neg\mathsf{RBad} \wedge \neg\mathsf{AskR}] \\
&\quad + \mathsf{pr}[\mathsf{Fail} \wedge \neg\mathsf{SBad} \wedge (\mathsf{AskR} \wedge \neg\mathsf{AskS})] \\
&\leq \mathsf{pr}[\mathsf{Fail} \mid \neg\mathsf{AskR} \wedge \neg\mathsf{RBad}] \times \mathsf{pr}[\neg\mathsf{AskR} \wedge \neg\mathsf{RBad}] \\
&\quad + \mathsf{pr}[\mathsf{Fail} \mid \mathsf{AskR} \wedge \neg\mathsf{AskS} \wedge \neg\mathsf{SBad}] \\
&\qquad \times \mathsf{pr}[\mathsf{AskR} \wedge \neg\mathsf{AskS} \wedge \neg\mathsf{SBad}] \\
&\leq \mathsf{pr}[\mathsf{Fail} \mid \neg\mathsf{AskR} \wedge \neg\mathsf{RBad}] \\
&\qquad + \mathsf{pr}[\mathsf{AskR} \wedge \neg\mathsf{AskS} \wedge \neg\mathsf{SBad}] \\
&\leq \mathsf{pr}[\mathsf{Fail} \mid \neg\mathsf{AskR} \wedge \neg\mathsf{RBad}] \\
&\quad + \mathsf{pr}[\mathsf{AskR} \mid \neg\mathsf{AskS} \wedge \neg\mathsf{SBad}] \times \mathsf{pr}[\neg\mathsf{AskS} \wedge \neg\mathsf{SBad}] \\
&\leq \mathsf{pr}[\mathsf{Fail} \mid \neg\mathsf{AskR} \wedge \neg\mathsf{RBad}] \\
&\qquad + \mathsf{pr}[\mathsf{AskR} \mid \neg\mathsf{AskS} \wedge \neg\mathsf{SBad}].
\end{aligned}$$

But without having asked r to G, and furthermore $\neg\mathsf{RBad}$, $G(r)$ is unpredictable, and thus the probability that $[s \oplus G(r)]_{k_1} = 0^{k_1}$ is less than 2^{-k_1}. On the other hand, the probability of having asked r to G, without any information about $H(s)$ and thus about r ($H(s)$ not asked, and $s \neq s^\star$), is less than $q_G \cdot 2^{-k_0}$. Therefore

$$\mathsf{pr}[\mathsf{Fail} \wedge \neg\mathsf{Bad} \wedge \neg\mathsf{AskRS}] \leq 2^{-k_1} + q_G \cdot 2^{-k_0}.$$

Furthermore, this event is independent of AskH, and thus

$$\mathsf{pr}[\mathsf{Fail} \mid \neg\mathsf{AskH}] \leq \mathsf{pr}[\mathsf{Fail} \wedge \mathsf{Bad} \mid \neg\mathsf{AskH}] + 2^{-k_1} + q_G \cdot 2^{-k_0}.$$

Now, let us focus on the first term, $\mathsf{Fail} \wedge \mathsf{Bad}$, while $\neg\mathsf{AskH}$, which was missing in the original proof [3]. It can be split due to the disjoint sub-cases of $\mathsf{Bad} = \mathsf{SBad} \vee (\neg\mathsf{SBad} \wedge \mathsf{RBad})$:

$$\begin{aligned}
\mathsf{pr}[\mathsf{Fail} \wedge \mathsf{Bad} \mid \neg\mathsf{AskH}] &= \mathsf{pr}[\mathsf{Fail} \wedge \mathsf{SBad} \mid \neg\mathsf{AskH}] \\
&\quad + \mathsf{pr}[\mathsf{Fail} \wedge \mathsf{RBad} \wedge \neg\mathsf{SBad} \mid \neg\mathsf{AskH}] \\
&\leq \mathsf{pr}[\mathsf{Fail} \mid \mathsf{SBad} \wedge \neg\mathsf{AskH}] + \mathsf{pr}[\mathsf{RBad} \mid \neg\mathsf{SBad} \wedge \neg\mathsf{AskH}].
\end{aligned}$$

The latter event means that RBad occurs provided that $s \neq s^\star$ and the adversary has not asked for $s^\star$ to H. When $s^\star$ is not asked to H and $s \neq s^\star$, $H(s^\star)$ is unpredictable and independent from $H(s)$ as well as t and $t^\star$. Then, event RBad,

$H(s^\star) = H(s) \oplus t \oplus t^\star$, occurs with probability at most 2^{-k_0}. The former event can be furthermore split due to AskR, and then is equal to

$$\begin{aligned}
&\mathsf{pr}[\mathsf{Fail} \,|\, \mathsf{AskR} \wedge \mathsf{SBad} \wedge \neg\mathsf{AskH}] \times \mathsf{pr}[\mathsf{AskR} \,|\, \mathsf{SBad} \wedge \neg\mathsf{AskH}] \\
&+\mathsf{pr}[\mathsf{Fail} \,|\, \neg\mathsf{AskR} \wedge \mathsf{SBad} \wedge \neg\mathsf{AskH}] \times \mathsf{pr}[\neg\mathsf{AskR} \,|\, \mathsf{SBad} \wedge \neg\mathsf{AskH}] \\
\leq\ &\mathsf{pr}[\mathsf{AskR} \,|\, \mathsf{SBad} \wedge \neg\mathsf{AskH}] + \mathsf{pr}[\mathsf{Fail} \,|\, \neg\mathsf{AskR} \wedge \mathsf{SBad} \wedge \neg\mathsf{AskH}].
\end{aligned}$$

The former event means that r is asked to G whereas $s = s^\star$ and $H(s^\star)$ is unpredictable, thus $H(s)$ is unpredictable. Since r is unpredictable, the probability of this event is at most $q_G \cdot 2^{-k_0}$ (the probability of asking r to G). On the other hand, the latter event means that the simulator rejects the valid ciphertext c whereas $H(s)$ is unpredictable and r is not asked to G. This probability is simply 2^{-k_1}. To sum up, $\mathsf{pr}[\mathsf{Fail} \,|\, \mathsf{SBad} \wedge \neg\mathsf{AskH}] \leq 2^{-k_1} + q_G \cdot 2^{-k_0}$, thus $\mathsf{pr}[\mathsf{Fail} \wedge \mathsf{Bad} \,|\, \neg\mathsf{AskH}] \leq 2^{-k_1} + (q_G + 1) \cdot 2^{-k_0}$. As a consequence,

$$\mathsf{pr}[\mathsf{Fail} \,|\, \neg\mathsf{AskH}] \leq \frac{1}{2^{k_1-1}} + \frac{2q_G + 1}{2^{k_0}}.$$

Thus, the decryption is correctly simulated with probability greater than

$$1 - \left(\frac{1}{2^{k_1-1}} + \frac{2q_G + 1}{2^{k_0}} \right),$$

provided that $\neg\mathsf{AskH}$ happened. The running time of this simulator just includes the computation of $f(\sigma, \tau)$ for all the possible pairs and thus is bounded by $q_G \cdot q_H \cdot (T_f + \mathcal{O}(1))$.

4.3 Combining Semantic Security with the Plaintext-Extractor

After q_D queries to the decryption oracle, all the decryptions will be correctly simulated with probability greater than

$$\left(1 - \left(\frac{1}{2^{k_1-1}} + \frac{2q_G + 1}{2^{k_0}} \right) \right)^{q_D},$$

provided that $\neg\mathsf{AskH}$ happened, which is upper-bounded by

$$1 - q_D \times \left(\frac{1}{2^{k_1-1}} + \frac{2q_G + 1}{2^{k_0}} \right).$$

Therefore, the probability of failure in at least one decryption simulation is

$$\begin{aligned}
\mathsf{pr}[\text{one } \mathsf{Fail}] &= \mathsf{pr}[\text{one } \mathsf{Fail} \,|\, \neg\mathsf{AskH}] \times \mathsf{pr}[\neg\mathsf{AskH}] + \mathsf{pr}[\text{one } \mathsf{Fail} \,|\, \mathsf{AskH}] \times \mathsf{pr}[\mathsf{AskH}] \\
&\leq \mathsf{pr}[\text{one } \mathsf{Fail} \,|\, \neg\mathsf{AskH}] + \mathsf{pr}[\mathsf{AskH}] \\
&\leq q_D \times \left(\frac{1}{2^{k_1-1}} + \frac{2q_G + 1}{2^{k_0}} \right) + \mathsf{pr}[\mathsf{AskH}]
\end{aligned}$$

The latter part is upper bounded by $\mathsf{Succ}^{\mathsf{sp-ow}}(q_H, t')$. Indeed, at the end of the game, or after a time bound t' fixed in advance, depending on the expected

running time of the reduction, $s^\star$ is one element amongst the q_H queries asked to H, and this set can be returned to break the set partial one-wayness of the function f.

As a consequence, our simulator can make q_D queries to this plaintext extractor, and output the whole pre-image of $c^\star$ with probability greater than

$$\frac{\varepsilon}{2} \times \left(1 - \frac{q_G}{2^{k_0}} - \frac{q_H}{2^{k-k_0}}\right) - \frac{q_G}{2^{k-1}} - q_D \times \left(\frac{1}{2^{k_1-1}} + \frac{2q_G+1}{2^{k_0}}\right) - \mathsf{Succ}^{\mathsf{sp-ow}}(q_H, t').$$

The running time of the overall simulation is still

$$t' \leq t + q_G \cdot q_H \cdot (T_f + \mathcal{O}(1))\,,$$

because all f computations can be done just once for each new hash query. However, the success probability of our simulator is upper-bounded by $\mathsf{Succ}^{\mathsf{ow}}(t')$, and thus

$$\varepsilon \leq 2 \times \frac{\mathsf{Succ}^{\mathsf{ow}}(t') + \mathsf{Succ}^{\mathsf{sp-ow}}(q_H, t') + q_D \times \left(\frac{1}{2^{k_1-1}} + \frac{2q_G+1}{2^{k_0}}\right) + \frac{q_G}{2^{k-1}}}{1 - \frac{q_G}{2^{k_0}} - \frac{q_H}{2^{k-k_0}}},$$

where

$$t' \leq t + q_G \cdot q_H \cdot (T_f + \mathcal{O}(1))\,.$$

hence the theorem.

5 Application to RSA–OAEP

The main application of OAEP is certainly the famous RSA-OAEP, which has been used to update the PKCS #1 standard [8]. Therefore, in his paper [9], Shoup tried to repair the security result for a small exponent, $e = 3$, using the Coppersmith's algorithm [4]. However, our result can be applied for any exponent for repairing RSA–OAEP. Indeed, thanks to random self-reducibility of RSA, the partial one-wayness of RSA is equivalent to that of the whole RSA problem, as soon as a constant fraction of the most significant bits of the pre-image can be recovered.

One may furthermore remark that the following argument can be applied to any random (multiplicatively) self-reducible problem, such as the Rabin function. Before presenting the global reduction, let us consider the problem of finding small solutions for a linear modular equation.

Lemma 5. *Let us be given an equation*

$$t + \alpha u = c \bmod N$$

for which we are looking for small solutions for t and u. For all the α, excepted a small fraction $2^{2k+6}/N$ of them, one can find t and u, smaller than 2^k, within a time bounded by $\mathcal{O}((\log N)^3)$.

Proof. Let us consider the lattice

$$L(\alpha) = \{(x, y) \in \mathbb{Z}^2 \,|\, x - \alpha y = 0 \bmod N\}.$$

Depending on α, some are good, others are bad. We say that $L(\alpha)$ is an ℓ-good lattice (and thus that α is an ℓ-good value) if there is no non-zero vector of length at most ℓ (using the Euclidean Norm). The other are called ℓ-bad lattices (and ℓ-bad values respectively). It is clear that there are approximately less than $\pi\ell^2$ ℓ-bad lattices, which we bound by $4\ell^2$. Indeed, each bad value for α corresponds to a point with integer coordinates in the disk of radius ℓ. Then the proportion of bad values for α is less than $4\ell^2/N$.

Now, let us assume that we have an ℓ-good lattice, on which one applies the Gaussian reduction. One then gets within time $\mathcal{O}((\log N)^3)$ a basis of $L(\alpha)$ consisting of two non-zero vectors U and V such that

$$\|U\| \leq \|V\| \text{ and } |(U, V)| \leq \frac{1}{2}\|U\|.$$

Let us consider a vector $T = \binom{t}{u}$, where (t, u) is a solution of the equation $t + \alpha u = c \bmod N$, with both t and u less than 2^k:

$$T = \lambda U + \mu V, \text{ for some real } \lambda, \mu.$$

$$\begin{aligned}
\|T\|^2 &= \lambda^2\|U\| + \mu^2\|V\| + 2\lambda\mu(U, V) \\
&\geq (\lambda^2 + \mu^2 - \lambda\mu) \times \|U\| \\
&\geq \left((\lambda - \mu/2)^2 + 3\mu^2/4\right) \times \|U\| \\
&\geq 3\mu^2/4 \times \|U\| \geq \frac{3}{4}\mu^2\ell^2.
\end{aligned}$$

Since we furthermore have $\|T\|^2 \leq 2 \times 2^{2k}$,

$$|\mu| \leq \frac{2\sqrt{2} \cdot 2^k}{\sqrt{3} \cdot \ell} \text{ and } |\lambda| \leq \frac{2\sqrt{2} \cdot 2^k}{\sqrt{3} \cdot \ell} \text{ by symmetry.}$$

If one has considered $\ell = 2^{k+2} > 2^{k+2}\sqrt{2/3}$ from the beginning, then

$$-\frac{1}{2} < \lambda, \mu < \frac{1}{2}.$$

Let us now choose any integer solution $T_0 = \binom{t_0}{u_0}$ of the considered equation, and thus by simply choosing a random integer u_0 and then $t_0 = c - \alpha u_0 \bmod N$. Let us write it in the basis (U, V): $T_0 = \rho U + \sigma V$ with real numbers ρ and σ. These coordinates can be found, and then, $T - T_0$ is a solution to the homogeneous equation, and thus is a lattice point: $T - T_0 = aU + bV$, with unknown integers a and b. But,

$$T = T_0 + aU + bV = (a + \rho)U + (b + \sigma)V = \lambda U + \mu V,$$

with $-1/2 \leq \lambda, \mu \leq 1/2$. As a conclusion, a and b are the closest integers to $-\rho$ and $-\sigma$ respectively. With a, b, ρ and σ, one can easily recover λ and μ and thus t and u, which are necessarily unique. □

Lemma 6. *Let us consider an algorithm $\mathcal{A}$ which outputs a q-set which contains half of the most significant bits of the e-th root (partial RSA, for any $N < 2^{2k_0}$), within a time bound t, with probability ε. Then there exists an algorithm $\mathcal{B}$ which solves the RSA problem (N, e) with a success probability ε', within a time bound t' where*

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

Proof. Thanks to the random self-reducibility of RSA, with half the bits of the e-th root of $X = (x \cdot 2^{k_0} + r)^e \bmod N$, and the e-th root of $Y = X\alpha^e = (y \cdot 2^{k_0} + s)^e \bmod N$, for a randomly chosen α, one gets both x and y. Thus,

$$(y \cdot 2^{k_0} + s) = \alpha \times (x \cdot 2^{k_0} + r) \bmod N$$
$$\alpha r - s = (y - x\alpha) \times 2^{k_0} \bmod N$$

which is a linear modular equation with two unknowns r and s which are known to have small solutions. It can be solved using lemma 5.

The algorithm $\mathcal{B}$ just runs twice $\mathcal{A}$, and then runs the Gaussian reduction on all the q^2 pairs of elements in both sets. If the partial pre-images are in the sets, they will be found, unless the random α is a bad one (*cf.* the Gaussian reduction in lemma 5.) □

Remark 1. Above lemma can be extended to the case where a constant fraction Θ of the most significant bits of the e-th root is found. The reduction runs $\mathcal{A}$ $1/\Theta$ times, and the success probability decreases to approximately $\varepsilon^{1/\Theta}$.

Theorem 2. *Let us consider a CCA–adversary $\mathcal{A}$ against the "semantic security" of RSA–OAEP, 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 the RSA problem can be solved with probability ε' greater than*

$$\frac{\varepsilon^2}{16} \times \left(1 - \frac{q_G}{2^{k_0-1}} - \frac{q_H}{2^{k-k_0-1}}\right) - \varepsilon \times \left(\frac{q_D}{2^{k_1+1}} + \frac{q_D(2q_G+1)}{2^{k_0+2}} + \frac{q_G}{2^{k+1}} + \frac{1}{2^{k-2k_0-4}}\right)$$

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

Proof. In any case, we have $\mathsf{Succ}^{\mathsf{ow}}(t'') \leq \mathsf{Succ}^{\mathsf{p-ow}}(t'') \leq \mathsf{Succ}^{\mathsf{sp-ow}}(t'')$, for any t'', and thus the success probability ε of an adversary against RSA–OAEP within a time bound t is less than

$$\frac{2}{1 - \frac{q_G}{2^{k_0}} - \frac{q_H}{2^{k-k_0}}} \times \left(2 \times \mathsf{Succ}^{\mathsf{sp-ow}}(t'') + \frac{q_D}{2^{k_1-1}} + \frac{q_D(2q_G+1)}{2^{k_0}} + \frac{q_G}{2^{k-1}}\right).$$

Therefore,

$$\mathsf{Succ}^{\mathsf{sp-ow}}(t'') \geq \frac{\varepsilon}{4} \times \left(1 - \frac{q_G}{2^{k_0}} - \frac{q_H}{2^{k-k_0}}\right) - \frac{q_D}{2^{k_1}} - \frac{q_D(2q_G+1)}{2^{k_0+1}} - \frac{q_G}{2^k},$$

with

$$t'' \leq t + q_G \cdot q_H \cdot (T_f + \mathcal{O}(1)).$$

With previous relation between q_H-set partial-RSA and RSA, we have

$$\begin{aligned}\mathsf{Succ}^{\mathsf{ow}}(t') &\geq \mathsf{Succ}^{\mathsf{sp-ow}}(t'') \times (\mathsf{Succ}^{\mathsf{sp-ow}}(t'') - 2^{2k_0-k+6}) \\ &\geq \frac{\varepsilon^2}{16} \times \left(1 - \frac{q_G}{2^{k_0-1}} - \frac{q_H}{2^{k-k_0-1}}\right) \\ &\quad -\varepsilon \times \left(\frac{q_D}{2^{k_1+1}} + \frac{q_D(2q_G+1)}{2^{k_0+2}} + \frac{q_G}{2^{k+1}} + \frac{1}{2^{k-2k_0-4}}\right)\end{aligned}$$

with

$$t' \leq 2t'' + q_H^2 \times \mathcal{O}(k^3) \leq 2t + q_H \cdot (q_H + q_G) \times \mathcal{O}(k^3).$$

□

6 Conclusion

Our conclusion is that one can still trust the security of RSA–OAEP, but the reduction is more costly than the original one. For other OAEP applications, however, more care is needed, since the security does not actually rely on the one-wayness of the permutation, only on the partial one-wayness.

Acknowledgments

We thank Victor Shoup, Don Coppersmith and Dan Boneh for fruitful comments.

References

1. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, Berlin, 1998.
2. 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.
3. 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.
4. D. Coppersmith. Finding a Small Root of a Univariate Modular Equation. In *Eurocrypt '96*, LNCS 1070, pages 155–165. Springer-Verlag, Berlin, 1996.
5. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
6. 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.
7. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
8. RSA Data Security, Inc. Public Key Cryptography Standards – PKCS. Available from `http://www.rsa.com/rsalabs/pubs/PKCS/`.
9. V. Shoup. OAEP Reconsidered. Cryptology ePrint Archive 2000/060.

A Proof of Lemma 2

In this proof, we first recall how works the simulator, and then we analyze the probability with which it extracts the whole pre-image of a given element.

A.1 Formalization.

We thus describe a simulator $\mathcal{B}$ which provides all the view of the adversary as if she were in a real attack. Let us consider $\mathcal{A} = (A_1, A_2)$, an adversary against the semantic security of $(\mathcal{K}, \mathcal{E}, \mathcal{D})$, using a chosen-plaintext attack. Within time bound t, she asks q_G and q_H queries to the hash functions G and H respectively, and distinguishes the right plaintext with an advantage greater than ε. Let us describe the simulator $\mathcal{B}$:

1. $\mathcal{B}$ first runs $\mathcal{K}(1^k)$ to get a function f (defined by the public key);
2. $\mathcal{B}$ is then given $c^\star \leftarrow f(s^\star, t^\star)$, for $(s^\star, t^\star) \stackrel{R}{\leftarrow} \{0,1\}^{k-k_0} \times \{0,1\}^{k_0}$. $\mathcal{B}$ is aimed to recover the pre-image $(s^\star, t^\star)$.
3. $\mathcal{B}$ runs A_1 on the public data, and gets a pair of messages $\{m_0, m_1\}$ as well as a state information st. It chooses a random bit b, and then outputs $c^\star$, as the ciphertext of m_b.
4. $\mathcal{B}$ runs $A_2(c^\star, st)$ and finally gets an answer b'. Then $\mathcal{B}$ outputs the pre-image of $c^\star$, if one has been found among the queries asked to G and H (see below), or Fail.

Of course, during all this simulation, the simulator also has to simulate the random oracle answers, managing query/answer lists G-List and H-List for the oracles G and H respectively, both initially set to empty lists:

- for a new query γ to G, for any query δ asked to H with answer H_δ (or for any $(\delta, H_\delta) \in$ H-List), one builds $z = \gamma \oplus H_\delta$, and checks whether $c^\star = f(\delta, z)$. If for some δ that relation holds, then we have inverted the function f, and we can still correctly simulate G, by answering $G_\gamma = \delta \oplus m_b \| 0^{k_1}$. Remark that G_γ is then a uniformly distributed value since $\delta = s^\star$, and the latter is uniformly distributed, by definition of x. Otherwise, one outputs a random value G_γ. In both cases, the pair (γ, G_γ) is concatenated to the G-List.
- for a new query δ to H, one outputs a random value H_δ, and the pair (δ, H_δ) is concatenated to the H-List. Note that once again, for any $(\gamma, G_\gamma) \in$ G-List, one may build $z = \gamma \oplus H_\delta$, and check whether $c^\star = f(\delta, z)$. If for some γ that relation holds, then we have inverted the function f.

When we have found the pre-image of $c^\star$, and thus inverted f, one could output it and stop the game. But for the analysis, we assume the game follows and $\mathcal{B}$ only outputs the answer (or Fail, if no pre-image has been found) when A_2 answers b'.

All this simulation is perfect, namely from the random oracle point of view. Indeed, as we have seen, a new uniformly distributed value is returned for any new query. However, the function property of the random oracles may fail: if two distinct answers are given for a same query. Even if no answer is explicitly specified, excepted by a random value for new queries, some are implicitly defined. Indeed, $c^\star$ is defined to be a ciphertext of m_b with a random tape $r^\star$:

$$r^\star \leftarrow H(s^\star) \oplus t^\star \text{ and } G(r^\star) \leftarrow s^\star \oplus m_b \| 0^{k_1}.$$

Since $H(s^\star)$ is randomly defined, $r^\star$ can be seen as a random variable. Let us denote by AskG the event that the query $r^\star$ has been asked to G, and by AskH

the event that the query $s^\star$ has been asked to H, and more specifically by FAskH the event that the query $s^\star$ has been asked to H during the find-stage (by A_1). Let us furthermore denote

- by FBad the event that $r^\star$ has been queried to G in the find-stage (by A_1), but answered by something else than $s^\star \oplus m_0 \| 0^{k_1}$ or $s^\star \oplus m_1 \| 0^{k_1}$;
- and by GBad the event that the query $r^\star$ has been asked to G in the guess-stage (by A_2), but answered by something else than $s^\star \oplus m_0 \| 0^{k_1}$ or $s^\star \oplus m_1 \| 0^{k_1}$. Furthermore when this query has been asked, $H(s^\star)$ has not been asked before (otherwise the simulation of G is correct, because of the control during the G-simulation presented above: $G_\gamma \leftarrow \delta \oplus m_b \| 0^{k_1} = s^\star \oplus m_b \| 0^{k_1}$.)

Note that each event FBad, or GBad, implies AskG. As seen above, FBad and GBad are the only events which make the simulation not to be perfect, then let us denote

$$\mathsf{Bad} = \mathsf{FBad} \vee \mathsf{GBad}.$$

A.2 Probabilities Analysis.

We denote by $\Pr[\cdot]$ the probabilities in the real attack, and by $\mathsf{pr}[\cdot]$ the probabilities in the simulated game.

As remarked in the first intuition, because of the randomness of G, and of the way the message is masked by $G(r^\star)$, the adversary cannot gain any advantage, in the real game, without having asked $r^\star$ to G. Indeed the simulation in this case is perfect, since $\neg\mathsf{AskG}$ implies $\neg(\mathsf{FBad} \vee \mathsf{GBad})$, and it is clearly independent of b: m_b only appears when r is remarked to be asked to G. Thus the probability of correctly guessing b is exactly of one half:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A}) &= 2 \times \Pr[A = b \,|\, \mathsf{AskG} \wedge \mathsf{AskH}] \times \Pr[\mathsf{AskG} \wedge \mathsf{AskH}] \\
&\quad +2 \times \Pr[A = b \,|\, \mathsf{AskG} \wedge \neg\mathsf{AskH}] \times \Pr[\mathsf{AskG} \wedge \neg\mathsf{AskH}] \\
&\quad +2 \times \Pr[A = b \,|\, \neg\mathsf{AskG}] \times \Pr[\neg\mathsf{AskG}] - 1 \\
&\leq 2 \times \Pr[\mathsf{AskG} \wedge \mathsf{AskH}] + 2 \times \Pr[\mathsf{AskG} \wedge \neg\mathsf{AskH}] \\
&\quad +(2 \times \Pr[A = b \,|\, \neg\mathsf{AskG}] - 1) \\
&\leq 2 \times \Pr[\mathsf{AskG} \wedge \mathsf{AskH}] + 2 \times \Pr[\mathsf{AskG} \wedge \neg\mathsf{AskH}] + 0.
\end{aligned}$$

However, in the following, we are interested in the probabilities in the simulated game, which is perfect unless FBad or GBad happens:

$$\varepsilon \leq 2 \times (\mathsf{pr}[(\mathsf{AskG} \wedge \mathsf{AskH}) \,|\, \neg\mathsf{Bad}] + \mathsf{pr}[(\mathsf{AskG} \wedge \neg\mathsf{AskH}) \,|\, \neg\mathsf{Bad}]) .$$

But, one may remark that

$$(\mathsf{AskG} \wedge \neg\mathsf{AskH}) \wedge \neg\mathsf{Bad} = (\mathsf{AskG} \wedge \neg\mathsf{AskH}) \wedge \neg(\mathsf{FBad} \vee \mathsf{GBad})$$

is exactly the event that $r^\star$ has been asked to G, without having asked $s^\star$ to H. But since we furthermore want $\neg(\mathsf{FBad} \vee \mathsf{GBad})$, the answer $G(r^\star)$ has been either $s^\star \oplus m_0 \| 0^{k_1}$ or $s^\star \oplus m_1 \| 0^{k_1}$. The probability of such an event is less than

$$q_G \cdot 2^{-k_0} \times 2 \cdot 2^{-k+k_0} = 2q_G \cdot 2^{-k} = q_G \cdot 2^{-k+1}.$$

Therefore,

$$\mathsf{pr}[(\mathsf{AskG} \wedge \mathsf{AskH}) \mid \neg\mathsf{Bad}] \geq \varepsilon/2 - q_G \cdot 2^{-k+1}/\mathsf{pr}[\neg\mathsf{Bad}],$$

and thus,

$$\begin{aligned}\mathsf{pr}[\mathsf{AskG} \wedge \mathsf{AskH}] &\geq \mathsf{pr}[(\mathsf{AskG} \wedge \mathsf{AskH}) \wedge \neg\mathsf{Bad}] \\ &\geq \mathsf{pr}[(\mathsf{AskG} \wedge \mathsf{AskH}) \mid \neg\mathsf{Bad}] \times \mathsf{pr}[\neg\mathsf{Bad}] \\ &\geq \varepsilon/2 \times \mathsf{pr}[\neg\mathsf{Bad}] - q_G \cdot 2^{-k+1}.\end{aligned}$$

Therefore, we just have to evaluate the probability $\mathsf{pr}[\mathsf{Bad}] = \mathsf{pr}[\mathsf{FBad} \vee \mathsf{GBad}]$:

$$\begin{aligned}\mathsf{pr}[\mathsf{Bad}] &= \mathsf{pr}[\mathsf{Bad} \mid \neg\mathsf{FAskH}] \times \mathsf{pr}[\neg\mathsf{FAskH}] + \mathsf{pr}[\mathsf{Bad} \mid \mathsf{FAskH}] \times \mathsf{pr}[\mathsf{FAskH}] \\ &\leq \mathsf{pr}[\mathsf{FBad} \vee \mathsf{GBad} \mid \neg\mathsf{FAskH}] + \mathsf{pr}[\mathsf{FAskH}].\end{aligned}$$

First, because of the randomness of $s^\star$ (which is uniformly distributed in $\{0,1\}^{k-k_0}$, but not used yet, and thus no information can be revealed about it), the probability of having asked it to H, in the first stage, is less than $q_H/2^{k-k_0}$: $\mathsf{pr}[\mathsf{FAskH}] \leq q_H \cdot 2^{-k+k_0}$.

Furthermore, one has to note that the events FBad or GBad, knowing that $\neg\mathsf{FAskH}$, mean that $\mathcal{A}$ asks $r^\star$ to G without having asked $s^\star$ to H yet. Therefore the random variable $r^\star$ is undefined at the moment of the actual query. Thus FBad or GBad may be set to be true if, later, $H(s^\star)$ is set to a value v such that $v \oplus t^\star$ has been asked to G: $\mathsf{pr}[\mathsf{FBad} \vee \mathsf{GBad} \mid \neg\mathsf{FAskH}] \leq q_G \cdot 2^{-k_0}$. Thus,

$$\mathsf{pr}[\mathsf{Bad}] \leq q_G \cdot 2^{-k_0} + q_H \cdot 2^{-k+k_0}.$$

As a conclusion, the probability that $\mathcal{B}$ outputs the pre-image is greater than

$$\varepsilon/2 \times (1 - q_G \cdot 2^{-k_0} - q_H \cdot 2^{-k+k_0}) - q_G \cdot 2^{-k+1}.$$

A.3 Complexity Analysis.

Note that during the running time of $\mathcal{B}$, for any new G-query γ, it has to look into all the query-answer $(\delta, H_\delta) \in \mathsf{H\text{-}List}$, and to compute

$$s = \delta, t = \gamma \oplus H_\delta, f(s,t).$$

Furthermore, it needs to do that again to extract the pre-image if AskG and AskH occur. But this can be done only once for each pair, when the query is asked to the hash function. Thus, the time complexity of the overall reduction is

$$t + q_G \cdot q_H \cdot (T_f + \mathcal{O}(1)),$$

where T_f denotes the time complexity for evaluating the function f.