# Making RSA–PSS Provably Secure against Non-random Faults

Gilles Barthe, François Dupressoir, Pierre-Alain Fouque, Mehdi Tibouchi, Jean-Christophe Zapalowicz, Benjamin Grégoire

# Making RSA–PSS Provably Secure Against Non-Random Faults

Gilles Barthe[1], François Dupressoir[1], Pierre-Alain Fouque[2], Benjamin Grégoire[4], Mehdi Tibouchi[3], and Jean-Christophe Zapalowicz[4]

[1] IMDEA Software Institute
gilles.barthe@imdea.org,francois.dupressoir@imdea.org
[2] Université de Rennes 1 and Institut universitaire de France
pierre-alain.fouque@ens.fr
[3] NTT Secure Platform Laboratories
tibouchi.mehdi@lab.ntt.co.jp
[4] INRIA
benjamin.gregoire@inria.fr,jean-christophe.zapalowicz@inria.fr

**Abstract.** RSA–CRT is the most widely used implementation for RSA signatures. However, deterministic and many probabilistic RSA signatures based on CRT are vulnerable to fault attacks. Nevertheless, Coron and Mandal (Asiacrypt 2009) show that the randomized PSS padding protects RSA signatures against *random faults*. In contrast, Fouque et al. (CHES 2012) show that PSS padding does not protect against certain *non-random faults* that can be injected in widely used implementations based on the Montgomery modular multiplication.

In this paper, we prove the security of an infective countermeasure against a large class of non-random faults; the proof extends Coron and Mandal's result to a strong model where the adversary can choose the value of the faulty signatures modulo one of the prime factors of the RSA modulus. This fault model is clearly strictly more general than Coron and Mandal's, and it captures most of the non-random faults of Fouque et al. Such non-random faults induce, together with the infective countermeasure, more complex probability distributions than in the original proof; we analyze them using careful estimates of character sums over finite fields. The security proof is formally verified using appropriate extensions of EasyCrypt, and provides the first application of formal verification to provable (i.e. reductionist) security in the context of fault attacks.

**Keywords**: Fault Attacks, PSS, RSA–CRT, Infective countermeasure, Formal Verification, EasyCrypt

## 1 Introduction

Signature schemes are among the most widely used constructions in cryptography. Although there is much interest in signature schemes based on elliptic curves, RSA signatures are still widely used. Moreover, many implementations of RSA, including OpenSSL and implementations for embedded devices such as smartcards,

use the well-known Chinese Remainder Theorem (CRT) technique for computing modular exponentiations more efficiently: exponentiations using the CRT can be expected to be 4 times faster than those using full-size exponents. However, when unprotected, RSA–CRT is vulnerable to the so-called Bellcore attack, first introduced by Boneh, DeMillo and Lipton [7], and later refined [3,28]. An adversary who knows the padded message and can inject a fault in one of the half exponentiations can efficiently factor the public modulus using a single faulty signature and a GCD computation.

Many countermeasures have been proposed to mitigate this vulnerability, including extra computations and sanity checks of intermediate and final results (see [24]). The simplest such protection is to verify the signature before releasing it. This is reasonably cheap since the public exponent $e$ is usually small. Another approach is to use an extended modulus, as in Shamir's trick [25] and its later refinements which also protect CRT recombination using Garner's formula [6,11,27,12]. Finally, redundant exponentiation algorithms [18,24] such as the Montgomery Ladder can be used. Regardless of the approach, RSA–CRT fault countermeasures tend to be rather costly: for example, Rivain's countermeasure [24,19] has a stated overhead of 10% compared to an unprotected implementation, and is purportedly more efficient than previous works [18,27,19].

Boneh et al.'s original fault attack does not apply to RSA signatures with probabilistic encoding functions, but some extensions of it were proposed to attack randomized ad-hoc padding schemes such as ISO 9796-2 and EMV [13,16]. At Asiacrypt 2009, Coron and Mandal [14] paved the way of provable security against side-channel attack in a practical setting by proving that RSA–PSS is secure against *random* faults in the random oracle model. Injecting a fault on the half-exponentiation modulo the second factor $q$ of $N$ produces a result that can be modeled as uniformly distributed modulo $q$, and the result of such a fault cannot be used to break RSA–PSS signatures. It is tempting to conclude that using RSA–PSS should enable signers to dispense with costly RSA–CRT countermeasures. However, Fouque et al. [17] show that it is possible to break RSA–PSS using certain *non-random* faults if the result is not checked. Indeed, they obtain a key recovery attack with a few faulty signatures on CRT implementations of RSA–PSS that use the state-of-the-art modular multiplication algorithm of Montgomery [21]. Thus, even with PSS, it remains important to check the signature before releasing it.

**Infective countermeasures.** Checking results before release is a simple and practical security measure, but it is not sufficient by itself, since simple tests can be easily bypassed by flipping the outcome of a comparison [2,26]. Infective countermeasures are an alternate approach in which results are released all the time, but become gibberish when faulty computations occur: a fault (usually not controlled by the adversary) results in a random value, which consequently makes the faulty signature random. From a security point of view, since faults may not be random, we may not be able to prove that the faulty output is fully random. However, one may ask that the output be independent of secret information even in the presence of non-random faults. Infective countermeasures have been used

before by Canetti and Goldwasser [9] to deal with fault-injecting adversaries when decrypting ciphertexts in a distributed manner. One such countermeasure for RSA–CRT was proposed by Boscher, Handschuh and Trichina [8]. In their technique, the signer computes the signature $S$ and recomputes $y' = S^e \bmod N$ to check the signature against the padded message $y$, before returning $S + y'_p - (y \bmod p) + y'_q - (y \bmod q)$ if $y' = y$, and an error otherwise. Even if the adversary bypasses the verification $y' = y$, the output signature mixes the fault and correct signature in a non-trivial way. Still, this countermeasure was later attacked by Trichina and Korkikyan [26] for deterministic padding schemes. We tackle the problem of masking faulty signatures so as to prevent the exploitation of faults and protect validity checks.

**Our contributions.** In this paper we generalize the fault model from [15] and consider a very powerful adversary able to inject *non-random* faults. More precisely, we let the adversary set the value modulo $q$ of the computed signatures to an arbitrary value of his choice. Clearly, since he could choose that value randomly, the model is strictly more powerful than the one considered by Coron and Mandal. In addition, it captures many other types of faults, such as the "null faults" and "constant faults" introduced by Fouque et al. [17]. If such a signatures is directly returned to the adversary, he can clearly factor the modulus, but we consider a simple countermeasure to avoid that problem. The countermeasure, described in Fig. 1, uses infective techniques, mixing additional randomness into faulty signatures in a *provably secure way*. In practice, we show that our random infection masks faulty signatures enough for us to prove the security of RSA–PSS under the RSA assumption in the random oracle model if enough additional randomness is provided. Concretely, we sample a random value $r'$ and add $r' \cdot (y - y')$ to the signature mod $N$, where $y$ is the original padded message and $y'$ is the padded message recovered from the signature. When the signature is computed correctly, $(y - y')$ is zero and the correct signature is returned. If the signature is faulty, we show that the masked output is statistically close to uniform and hence leaks no secret information. We prove such results in two key lemmas corresponding to [14, Lemmas 1, 2]. Since our faults are non-random, the probability distributions are more complex; we use careful estimates of exponential sums attached to corresponding rational functions to establish their regularity. We only analyze this countermeasure when the validity check is performed in the standard way (by computing the public permutation), but our random infection might also be used to protect other checks such as Rivain's [24,19]. A discussion of the faults we model can be found in Section 2.

The second contribution of the paper is a formal proof of security of the countermeasure using EasyCrypt[5], a computer-aided framework that has previously been used to reason about the security of cryptographic constructions—but was never applied to fault attacks and countermeasures. Our proof is the first application of formal verification to provable security against fault attacks, as other works [10,22,23] applying formal verification to fault attacks are focused on proving the correctness of the countermeasures (that is, that the protected

[5] https://www.easycrypt.info

**Figure 1** Protected signing algorithm

1: **function** SIGN($sk, pk, m$) ▷ $sk = (d_p, d_q, \alpha_p, \alpha_q, N)$, $pk = (e, N)$
2: $r \leftarrow \{0,1\}^{k_0}$ ▷ Start of PSS padding
3: $\omega \leftarrow \mathcal{H}(m, r)$
4: $st \leftarrow \mathcal{G}(\omega) \oplus (r \,||\, 0^{k_g - k_0})$
5: $y \leftarrow \mathsf{os2ip}(0 \,||\, \omega \,||\, st)$
6: $\sigma_p \leftarrow y^{d_p} \bmod p$ ▷ Signature computation
7: $\sigma_q \leftarrow y^{d_q} \bmod q$
8: $\sigma \leftarrow (\alpha_p \cdot \sigma_p + \alpha_q \cdot \sigma_q) \bmod N$ ▷ $\alpha_p = q \cdot (q^{-1} \bmod p)$ and similarly for $\alpha_q$
9: $y' \leftarrow \sigma^e \bmod N$
10: $r' \leftarrow \{0,1\}^{\rho} \backslash \{0\}$ ▷ Infective countermeasure
11: $\sigma' \leftarrow \sigma + r' \cdot (y - y') \bmod N$
12: **return** $\mathsf{i2osp}(\sigma')$

program either returns the same result as the original program, or fails), but do not provide any provable security guarantees. Apart from increasing our confidence in the effectiveness of the countermeasure, our formal proof reveals a glitch in the proof of Coron and Mandal [14], and also paves the way for formally verifying the effectiveness of the countermeasures on standard implementations of PKCS probabilistic signing, in the same way that [1] uses an older prototype of EasyCrypt [5] to prove security of an implementation of PKCS encryption.

**Related work.** Christofi et al. [10] use a combination of program transformation and verification techniques for proving Vigilant's countermeasure for CRT-RSA. They take a source program $p$ and output a program $\hat{p}$ that contains all possible faulty behaviors of $p$. Then, they show that the program $\hat{p}$ either returns a value that matches the value returned by $p$ on the same input, or else returns an error, they conclude that the program is correct for all faults. While it is a natural guarantee to seek, their theorem does not constitute a proof of security in the sense of provable security, but rather a heuristic to validate a countermeasure implementation.

Rauzy and Guilley [23] develop symbolic methods to analyze fault attacks against RSA–CRT implementations. They model arithmetic computations as algebraic expressions, and define a simplification procedure for expressions. Given an expression $e$ (representing the algorithm to be attacked), their tool tests for all possible faulty variants $\hat{e}$ of $e$ if the expression $\gcd(N, e - \hat{e})$ simplifies to a prime factor of the RSA modulus. If some expression $\hat{e}$ is found, then the algorithm is considered insecure. Their tool is useful to find fault attacks on an algorithm, but only provides guarantees of security against a restricted class of attackers. Moreover, it is specialized to deterministic signature schemes and cannot deal with randomized paddings like PSS.

Moro et al. [22] focus on the specific class of instruction skip attacks, in which an adversary forces to skip the execution of a targeted instruction. To protect against skip attacks, they transform a program $p$ into a fault-tolerant program $\hat{p}$, by providing for each instruction a possible replacement for execution in the presence of instruction skip faults. Using a model checker, they establish

the equivalence between executing the instruction without faults and executing the replacement sequence of instructions with instruction skip faults. Their approach is general, and significantly improves resistance against instruction skip attacks. However, it is not suitable for obtaining the strong guarantees required by provable security.

## 2 Our results

Instead of considering the many possible faults an adversary could inject in Fig. 1, we give the adversary access to two distinct oracles (Fig. 2) that compute valid signatures (oracle $\mathcal{S}$) and generalize faulty signatures (oracle $\mathcal{F}$), as justified in Section 2. As discussed, our fault model is independent of the algorithm used to compute modular exponentiation. We therefore use simpler definitions for public and secret key, where a public key $pk$ is composed of a public exponent $e$ and a modulus $N$, and a secret key $sk$ is composed of a private exponent $d$ and a modulus $N$.

Throughout the security proof, we consider a fixed $k$ that serves as the size of the modulus and signatures. In particular, we assume that the modulus is balanced, that is $N = p \cdot q$ is such that $2^{k-1} \leq N < 2^k$ and $2^{k/2-1} \leq p < q < 2^{k/2}$. We also assume that public exponents produced by the key generation algorithm are upper bounded by some constant $e_{max}$ much smaller than $2^k$ (in practice, $2^{16}+1$ is often used). PSS padding is computed using two hash functions $\mathcal{H}$, outputting bitstrings of length $k_h$, and $\mathcal{G}$, producing bitstrings of length $k_g$, where $k_h + k_g + 1 = k$. In addition, the padding scheme uses a random salt of length $k_0 < k_g$. For simplicity, we model $\mathcal{H}$ as a function from $\{0,1\}^* \times \{0,1\}^{k_0}$ to

**Figure 2** Oracles in our fault model

**oracle** $\mathcal{S}(m)$
- $r \leftarrow \{0,1\}^{k_0}$
- $\omega \leftarrow \mathcal{H}(m, r)$
- $st \leftarrow \mathcal{G}(\omega) \oplus (r \,||\, 0^{k_g - k_0})$
- $y \leftarrow \mathsf{os2ip}(0 \,||\, \omega \,||\, st)$
- $\sigma \leftarrow y^d \bmod N$
- **return** $\mathsf{i2osp}(\sigma)$

**oracle** $\mathcal{V}(m, \sigma)$
- $r \leftarrow \perp$
- $s \leftarrow \mathsf{os2ip}(\sigma)$
- **if** $0 < s < N$ **then**
  - $y \leftarrow s^e \bmod N$
  - $b \,||\, \omega \,||\, st \leftarrow \mathsf{i2osp}(y)$
  - $r \,||\, \gamma \leftarrow st \oplus \mathcal{G}(\omega)$
  - $\omega' \leftarrow \mathcal{H}(m, r)$
  - $r = b = 0 \wedge \omega = \omega' \wedge \gamma = 0^{k_g - k_0}$
- **return** $r$

**oracle** $\mathcal{F}(m, a)$
- $r \leftarrow \{0,1\}^{k_0}$
- $\omega \leftarrow \mathcal{H}(m, r)$
- $st \leftarrow \mathcal{G}(\omega) \oplus (r \,||\, 0^{k_g - k_0})$
- $y \leftarrow \mathsf{os2ip}(0 \,||\, \omega \,||\, st)$
- $\sigma \leftarrow y^d \bmod N$
- $r' \leftarrow \{0,1\}^{\rho} \backslash \{0\}$
- $\sigma' \leftarrow y^d \cdot \alpha_p + (a + r' \cdot (y - a^e)) \cdot \alpha_q$
- **return** $\mathsf{i2osp}(\sigma')$

$\{0,1\}^{k_h}$, and $\mathcal{G}$ as a function from $\{0,1\}^{k_h}$ to $\{0,1\}^{k_g}$. This is done without loss of generality. In algorithm and game descriptions, we denote with i2osp and os2ip the conversions between integers and their binary representations. For simplicity, i2osp always produces a bitstring of length $k$.

We reduce the $\mathcal{UF}$-$\mathcal{CMA}$ security of the faulty signature scheme presented in Fig. 2, when the adversary is given access to the faulty signature oracle along with the valid signature oracle and the random oracles $\mathcal{H}$ and $\mathcal{G}$, to the one-way security of RSA. We consider a forgery valid even if it was produced by the faulty signature oracle. In the rest of this paper, we use $\mathcal{S}$ to denote the valid signature oracle, $\mathcal{F}$ to denote the faulty signature oracle, $\mathcal{K}$ to denote the RSA key generation algorithm, and $\mathcal{V}$ for the PSS verification algorithm. Subscripts identify the game in which a particular oracle appears. We denote with $\mathcal{Q}^{\mathcal{X}}$ the set of query-response pairs for queries made to oracle $\mathcal{X}$ so far.

**Figure 3** Initial and Final Games

| | |
|---|---|
| 1: **game** $\mathcal{UF}$-$\mathcal{CMA}$ | 1: **game** $\mathcal{OW}$-$\mathcal{RSA}$ |
| 2: $(e, d, N) \leftarrow \mathcal{K}()$ | 2: $(e, d, N) \leftarrow \mathcal{K}()$ |
| 3: $(m, s) \leftarrow \mathcal{A}^{\mathcal{S},\mathcal{F},\mathcal{H},\mathcal{G}}(e, N)$ | 3: $x^* \leftarrow [0..N)$ |
| 4: $b \leftarrow \mathcal{V}(m, s)$ | 4: $y^* \leftarrow x^{*e} \bmod N$ |
| 5: $win \leftarrow b \wedge (m, s) \notin \mathcal{Q}^{\mathcal{S}}$ | 5: $x \leftarrow \mathcal{I}(e, N, y^*)$ |
| 6: **return** $win$ | 6: **return** $x = x^*$ |

**Theorem 1 ($\mathcal{UF}$-$\mathcal{CMA}$ security of protected PSS in the presence of faults).** *Given a CMA adversary $\mathcal{A}$ against the faulty signature scheme $(\mathcal{K}, \mathcal{S}, \mathcal{F}, \mathcal{V})$ that makes at most $q_{\mathcal{H}}$ queries to $\mathcal{H}$, $q_{\mathcal{G}}$ queries to $\mathcal{G}$, $q_{\mathcal{S}}$ queries to $\mathcal{S}$ and $q_{\mathcal{F}}$ queries to $\mathcal{F}$, we build a one-way inverter $\mathcal{I}$ such that*

$$\Pr[\mathcal{UF}\text{-}\mathcal{CMA} : win] \leq \Pr[\mathcal{OW}\text{-}\mathcal{RSA} : x = x^*] + \epsilon_0$$

*with*

$$\epsilon_0 = \frac{(q_{\mathcal{H}} + q_{\mathcal{S}} + q_{\mathcal{F}}) \cdot (q_{\mathcal{H}} + q_{\mathcal{G}} + q_{\mathcal{S}} + q_{\mathcal{F}}) + q_{\mathcal{G}} \cdot q_{\mathcal{F}} \cdot 3 + 1}{2^{k_h}} + \frac{(q_{\mathcal{S}} + q_{\mathcal{F}}) \cdot (2 \cdot q_{\mathcal{H}} + q_{\mathcal{S}} + q_{\mathcal{F}}) + q_{\mathcal{H}} + q_{\mathcal{S}}}{2^{k_0}} + \frac{1}{2^{\frac{k}{2}-1}} + q_{\mathcal{F}} \cdot \frac{(4e_{max}^2 + 1) \cdot k^2}{2^{\rho - k/2}}$$

*Remark 1.* This theorem allows us to conclude directly with a security claim for PSS when $e_{max}$ is reasonably small (typically $2^{16} + 1$), $\rho \geq k/2 + 200$, and the modulus is not too large (see Remark 2).

**Fault model justification.** In this section, we justify our fault model, described by oracle $\mathcal{F}(m, a)$ in Fig. 2. Our faulty signature oracle computes the correct padded message $y$, samples $r'$ and returns $\sigma' = y^d \cdot \alpha_p + (a + r'(y - a^e)) \cdot \alpha_q$ with $a \in \mathbb{Z}/q\mathbb{Z}$ chosen by the adversary.

We allow multiple faults to be injected, but only during the RSA–CRT computation (lines 6-7 of the protected signing Fig. 1). More precisely, we consider a scenario where the computation modulo $p$ is correct whereas those modulo $q$ is faulted to result in a constant $a$ chosen by the adversary, *i.e.* $\sigma_f = (y^d \bmod p, a \bmod q) \in \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$. Then, using our countermeasure we obtain:

$$\begin{aligned}\sigma' =& \sigma_f + r'(y - \sigma_f^e)\\ =& y^d\alpha_p + \alpha_q a + r'(y - (y^d\alpha_p + \alpha_q a)^e)\\ =& y^d\alpha_p + (a + r'(y - a^e))\alpha_q.\end{aligned}$$

Our fault model leverages the results of Coron and Mandal in [14] who treated the case of random faults against PSS scheme, and those of Fouque et al. [17] who proposed various faults: the Null faults (forcing a small register to 0), the Constant faults (forcing a small register to a constant) and the Zero High-Order Bits faults (forcing part of a small register to 0). When applied during the RSA–CRT computations to a precise small register, these faults may allow the adversary to factor the RSA modulus. They apply to any padding scheme, including randomized padding schemes such as PSS.

With our formalization, we take into account the Null faults model, which results in nullifying the signature modulo $q$, by considering $a = 0$. Moreover, we cover a more powerful model than the random faults one by giving to the adversary the choice of the value $a$.

## 3 Statistical Lemmas

We need several results on the regularity of the probability distributions related to the infective countermeasure. Recall that the *statistical distance* between a random variable $X$ on a finite set $S$ and the uniform distribution is defined as:

$$\Delta_1(X) = \frac{1}{2} \cdot \sum_{s \in S} \left| \Pr[X = s] - \frac{1}{|S|} \right|.$$

We say that $X$ is $\delta$-*statistically close to uniform* when $\Delta_1(X) \leq \delta$.

Our proofs rely on character sums over $\mathbb{Z}/q\mathbb{Z}$. We refer to [20] or the full version of this paper [4] for basic properties of Dirichlet characters and character sums. The main statistical result can be stated as follows.

**Lemma 1.** *Consider integer intervals $\mathcal{X} = [1, X]$, $\mathcal{W} = [w_0, w_0 + W)$ whose lengths $X, W$ satisfy $X, W < q$, and for all $t \in \mathbb{Z}/q\mathbb{Z}$, denote by $T(\mathcal{X}, \mathcal{W}; t) = \frac{XW}{q} \cdot \big(1 + V(\mathcal{X}, \mathcal{W}; t)\big)$ the number of solutions $(x, w) \in \mathcal{X} \times \mathcal{W}$ of the congruence $xw \equiv t \pmod q$. Assuming that the Generalized Riemann Hypothesis holds, then for all $\delta > 0$, there exists a constant $\kappa_\delta > 0$ depending only on $\delta$ (and not $q, \mathcal{X}, \mathcal{W}$) such that:*

$$\sum_{t \in \mathbb{Z}/q\mathbb{Z}} \big|V(\mathcal{X}, \mathcal{W}; t)\big| \leq \frac{\kappa_\delta q^{3/2+\delta}}{\sqrt{XW}}.$$

*In particular, the distribution of the products $xw \bmod q$ is statistically close to uniform in $\mathbb{Z}/q\mathbb{Z}$ whenever $XW \gg q^{1+3\delta}$.*

*Proof.* Note first that all elements of $\mathcal{X}$ are invertible modulo $q$, whereas at most one element of $\mathcal{W}$ is divisible by $q$. Denote by $W^*$ the number of elements of $\mathcal{W}$ which are invertible modulo $q$, which is thus equal to $W$ or $W-1$. We then have:

$$T(\mathcal{X},\mathcal{W};0) = X\cdot(W-W^*) \le X \quad \text{and hence} \quad \big|V(\mathcal{X},\mathcal{W};0)\big| \le \frac{q}{W}.$$

On the other hand, for $t \neq 0$, we can express $T(\mathcal{X},\mathcal{W};t)$ as a sum over the multiplicative characters modulo $q$. Indeed, the orthogonality of characters ensures that, for all $x, w$, we have $\sum_\chi \chi(xw)\overline{\chi(t)} = q-1$ if $xw \equiv t \pmod q$ and 0 otherwise. Hence:

$$\begin{aligned} T(\mathcal{X},\mathcal{W};t) &= \frac{1}{q-1}\sum_{\chi}\sum_{(x,w)\in\mathcal{X}\times\mathcal{W}} \chi(xw)\overline{\chi(t)} \\ &= \frac{XW^*}{q-1} + \frac{1}{q-1}\sum_{\chi\neq\chi_0}\sum_{(x,w)\in\mathcal{X}\times\mathcal{W}} \chi(xw)\overline{\chi(t)}, \end{aligned}$$

by putting aside the contribution of the trivial character $\chi_0$. Write that equality as $T(\mathcal{X},\mathcal{W};t) = \frac{XW^*}{q-1}\cdot\big(1+V^*(t)\big)$. We then have:

$$V^*(t) = \frac{1}{XW^*}\sum_{\chi\neq\chi_0}\sum_{(x,w)\in\mathcal{X}\times\mathcal{W}} \chi(xw)\overline{\chi(t)},$$

and we can express the sum of the squared deviations $\big|V^*(t)\big|^2$ as:

$$\sum_{t\neq 0}\big|V^*(t)\big|^2 = \frac{1}{(XW^*)^2}\sum_{\chi,\chi'\neq\chi_0}\sum_{x,w,x',w'} \chi(xw)\overline{\chi'(x'w')}\sum_{t\neq 0}(\chi\overline{\chi'})(t).$$

The sum over $t$ on the right-hand side is equal to $q-1$ if $\chi=\chi'$ and vanishes otherwise, so that:

$$\sum_{t\neq 0}\big|V^*(t)\big|^2 = \frac{q-1}{(XW^*)^2}\sum_{\chi\neq\chi_0}\sum_{x,w,x',w'} \chi(xw)\overline{\chi'(xw)} = \frac{q-1}{(XW^*)^2}\sum_{\chi\neq\chi_0}\big|S(\chi)\big|^2,$$

where $S(\chi) = \sum_{x\in\mathcal{X}}\chi(x)\sum_{w\in\mathcal{W}}\chi(w)$. Now since $\mathcal{X}$ is an interval of the form $[1,X]$, it is classical that GRH implies, for any $\delta>0$, $\big|\sum_{x\in\mathcal{X}}\chi(x)\big| \le c_\delta X^{1/2}q^\delta$ for some constant $c_\delta>0$ (see e.g. [20, Eq. (13.2)]). Hence:

$$\sum_{t\neq 0}\big|V^*(t)\big|^2 \le \frac{q-1}{(XW^*)^2}\cdot c_\delta^2 X q^{2\delta}\cdot\sum_{\chi}\sum_{(w,w')\in\mathcal{W}^2}\chi(w)\overline{\chi(w')} \le \frac{c_\delta^2 q^{2\delta}(q-1)^2}{XW^*}$$

by using orthogonality again. Then, the Cauchy–Schwarz inequlity yields:

$$\sum_{t\neq 0}\big|V^*(t)\big| \le \sqrt{\frac{c_\delta^2 q^{2+2\delta}}{XW^*}}\cdot\sqrt{q-1} \le \frac{c_\delta q^\delta (q-1)^{3/2}}{\sqrt{XW}}.$$

Finally, observe that for $t \neq 0$, we have:

$$\begin{aligned} V(\mathcal{X},\mathcal{W};t) &= \frac{q}{XW}T(\mathcal{X},\mathcal{W};t) - 1 = \frac{q}{XW}\cdot\frac{XW^*}{q-1}\cdot\big(1+V^*(t)\big) - 1 \\ &= \frac{qW^*}{(q-1)W}V^*(t) - \frac{W-q(W-W^*)}{(q-1)W}. \end{aligned}$$

On the last line, the first term is bounded in absolute value by $\frac{q}{q-1}\big|V^*(t)\big|$, and the second term by $\frac{q}{q-1}W$. As a result, we get:

$$\sum_{t\in\mathbb{Z}/q\mathbb{Z}}\big|V(\mathcal{X},\mathcal{W};t)\big| \leq \frac{q}{q-1}\sum_{t\neq 0}\big|V^*(t)\big| + \frac{q}{W} + \big|V(0)\big| \leq \frac{c_\delta q^{3/2+\delta}}{\sqrt{XW}} + \frac{2q}{W}$$

which yields the stated result for $\kappa_\delta = c_\delta + 2$, say (as a coarse upper bound). □

We now discuss our key statistical lemmas. The first one ensures that the faulty signature $\sigma' = y^d\cdot\alpha_p + \big(a + r'(y-a^e)\big)\cdot\alpha_q$ is indistinguishable from a uniform random element in $\mathbb{Z}/N\mathbb{Z}$ if the nonce $r'$ is large enough. We write $x$ instead of $r'$ in the rest of this section.

**Lemma 2.** *Let $N = pq$ be a $k$-bit balanced RSA modulus and $e$ the public exponent, $0 \leq y < 2^{k-1}$ a random integer and $x$ a random nonzero $\rho$-bit integer. Fix an arbitrary integer $a$. Assuming that the Generalized Riemann Hypothesis holds, the statistical distance between the distribution of $\sigma' = y^d\cdot\alpha_p + \big(a + x(y-a^e)\big)\cdot\alpha_q \bmod N$ and the uniform distribution modulo $N$ is bounded as:*

$$\Delta_1(\sigma') \leq \kappa_\delta q^\delta\sqrt{\frac{N}{XY}} \leq 2k_\delta\cdot 2^{(k\delta-\rho)/2}$$

*for any $\delta > 0$, with $\kappa_\delta$ as in Lemma 1.*

*Proof.* The statistical distance between the distribution of $\sigma'$ and the uniform distribution can be written as:

$$\Delta_1(\sigma') = \frac{1}{2}\sum_{(s,t)\in\mathbb{Z}/p\mathbb{Z}\times\mathbb{Z}/q\mathbb{Z}}\left|\Pr_{(x,y)\in\mathcal{X}\times\mathcal{Y}}\begin{bmatrix}\sigma'\equiv s \pmod p\\ \sigma'\equiv t \pmod q\end{bmatrix} - \frac{1}{N}\right|$$

where $\mathcal{X}$ and $\mathcal{Y}$ are the integer intervals $[1,X]$ and $[0,Y)$ with $X = 2^\rho - 1$ and $Y = 2^{k-1}$ respectively. Let us estimate the probability

$$P(s,t) = \Pr_{(x,y)\in\mathcal{X}\times\mathcal{Y}}\begin{bmatrix}\sigma'\equiv s \pmod p\\ \sigma'\equiv t \pmod q\end{bmatrix}$$

appearing in that equation for some fixed $(s,t)\in\mathbb{Z}/p\mathbb{Z}\times\mathbb{Z}/q\mathbb{Z}$.

We have $\sigma'\equiv s \pmod p$ if and only if $y^d \equiv s \mod p$, i.e. $y \equiv s^e \mod p$. Hence, the solutions of the first congruence are of the form $y = (s^e \bmod p) + pw$ for $w$ in the integer interval $[0, W_s)$, $W_s = \lceil\frac{Y-(s^e \bmod p)}{p}\rceil$. Then, the second

equation, which is equivalent to $a + x(y - a^e) \equiv t \pmod q$, becomes $x(pw + (s^e \bmod p) - a^e) \equiv t - a \pmod q$. This can be written in the form $x(w + w_0) = t_0 \pmod q$, with $w_0 = \frac{(s^e \bmod p) - a^e}{p} \bmod q$ and $t_0 = \frac{t-a}{p} \bmod q$. The number of solutions $(x, w)$ is thus $T(\mathcal{X}, \mathcal{W}_s; t_0)$, with $\mathcal{W}_s = [w_0, w_0 + W_s)$. Hence:

$$P(s,t) = \frac{1}{XY} T\big(\mathcal{X}, \mathcal{W}_s; t_0\big) = \frac{W_s}{qY} + \frac{W_s}{qY} V\big(\mathcal{X}, \mathcal{W}_s; t_0\big).$$

Note that $\mathcal{W}_s$ depends only on $s$ (not on $t$), and that $t \mapsto t_0$ is a permutation of $\mathbb{Z}/q\mathbb{Z}$. Thus, for fixed $s$, we can sum the previous equation over $t \in \mathbb{Z}/q\mathbb{Z}$, which gives:

$$\sum_{t \in \mathbb{Z}/q\mathbb{Z}} \left| P(s,t) - \frac{1}{N} \right| \le q \cdot \left| \frac{W_s}{qY} - \frac{1}{N} \right| + \frac{W_s}{qY} \sum_{t \in \mathbb{Z}/q\mathbb{Z}} \big| V\big(\mathcal{X}, \mathcal{W}_s; t\big) \big|.$$

Now $Y/p - 1 \le W_s \le Y/p + 1$, so that the first term on the right-hand side is bounded by $1/Y$. Thus, Lemma 1 yields:

$$\sum_{t \in \mathbb{Z}/q\mathbb{Z}} \left| P(s,t) - \frac{1}{N} \right| \le \frac{1}{Y} + \frac{W_s}{qY} \cdot \frac{\kappa_\delta q^{3/2+\delta}}{\sqrt{XW_s}} = \frac{\kappa_\delta q^{1/2+\delta}}{\sqrt{XY \cdot p/2}}$$

using the coarse upper bound $W_s/Y \le 2/p$. Summing further over $s$, we finally obtain:

$$\sum_{s,t} \left| P(s,t) - \frac{1}{N} \right| \le \frac{p}{Y} + \kappa_\delta q^\delta \sqrt{\frac{2N}{XY}}$$

and hence the desired result, since $p \le \sqrt{N}$ and $Y > X$. □

*Remark 2.* Concretely, this result means that, for large enough $N$, it suffices to take $\rho$ slightly larger than $m$ to obtain a statistical distance of $2^{-m}$.

If we do not want to rely on the Riemann Hypothesis, we can obtain an unconditional bound by replacing the use of GRH in Lemma 1 by the Pólya–Vinogradov inequality (or the Burgess bound). However, statistical indistinguishability from uniform then requires somewhat larger values of $\rho$: at least $k/4 + m + o(1)$ with Pólya–Vinogradov or $k/8 + m + o(1)$ with the Burgess bound.

The security proof requires another statistical lemma which ensures that the adversary has a negligible probability of querying the correct value $\omega \leftarrow \mathcal{H}(M, r)$ given a faulty signature. The proof, which uses Lemma 1 in a very similar way as the proof of Lemma 2 (simply replacing the interval $\mathcal{Y}$ by a subinterval $\mathcal{Y}_\omega$), is given in the full version of this paper [4].

**Lemma 3.** *Let $N, e, a, \delta, \kappa_\delta$ be as in Lemma 2. Assume that $\rho \ge k_h + \delta k + \log_2(4\kappa_\delta)$. For any choice of $\sigma' \in \mathbb{Z}/N\mathbb{Z}$ and any $k_h$-bit value $\omega'$, the probability that a solution $(x, y) \in [1, 2^\rho) \times [0, 2^{k-1})$ of the equation $\sigma' \equiv y^d \cdot \alpha_p + \big(a + x(y - a^e)\big) \cdot \alpha_q \pmod N$ satisfies that the most significant $k_h$ bits $\omega \in [0, 2^{k_h})$ of $y$ coincides with $\omega'$ is bounded as:*

$$\Pr\left[\omega = \omega' | \sigma'\right] \le \frac{3}{2^{k_h}}.$$

*Remark 3.* Concretely, this result means that we have to choose $\rho$ larger than $k_h/2$.

## 4 Security proof

The sequence of games presented in this Section and formal justifications for all transitions between games are formalized in EasyCrypt. However, Lemmas 2 and 3 are stated as axioms of the formalization. Formally proving these lemmas is outside the scope of this work, as it would first require to formalize at least those properties of additive characters used in our proof.

The hash functions $\mathcal{G}$ and $\mathcal{H}$ are modelled as random oracles. For clarity, we display the initial definition of $\mathcal{H}$ on the left in Fig. 4. The initial definition of $\mathcal{G}$ is similar. We assume two global maps $h$ and $g$ are used to build the random oracles. Our proof works mostly by transforming the random oracle $\mathcal{H}$. We therefore display the code for $\mathcal{H}$ for each transition, only displaying other oracles when they suffer non-trivial changes.

*Game 0.* We initially transform both random oracles to keep track of the first caller to make a particular query. It can be either the adversary (Adv), the signature oracle (Sig), or the faulty signature oracle (FSig). Calls made by the experiment when checking the validity of the forgery do not need to tag their query as they are the last queries made to the random oracles and do not need to update its state. We also extend the internal state of $\mathcal{H}$ with an additional field for use later in the proof, and currently set to a default value $\perp$.

**Figure 4** Initial transition: extending state

| | |
|---|---|
| **oracle** $\mathcal{H}(m, r)$ | **oracle** $\mathcal{H}_0(m, r)$ |
| **if** $(m, r) \notin \mathsf{dom}(h)$ **then** | **if** $(m, r) \notin \mathsf{dom}(h)$ **then** |
| $h[m, r] \leftarrow \{0,1\}^{k_h}$ | $\omega \leftarrow \{0,1\}^{k_h}$ |
| **return** $h[m, r]$ | $h[m, r] \leftarrow (\omega, c, \perp)$ |
| | **return** $\pi_1(h[m, r])$ |

$$\Pr[\mathcal{UF}\text{-}\mathcal{CMA}^{\mathcal{A},\mathcal{K},\mathcal{S},\mathcal{F},\mathcal{V}} : win] = \Pr[\mathcal{G}ame0 : win]$$

*Games 1 and 2.* In Game 1, we anticipate a call to $\mathcal{G}$ on the output of $\mathcal{H}$ every time $\mathcal{H}$ is called. When $\mathcal{H}$ is called by either one of the signing oracles, we return the result of that call to $\mathcal{G}$ as well as the result of the current $\mathcal{H}$ query, allowing broad simplifications to the signing oracles. In Game 2, we deal with collisions on $r$ and $\omega$ values in the signing oracles. In later steps of the proof, we will need the control-flow of the faulty signature oracle to be completely independent from both $r$ and $\omega$, and we modify the oracle to allow these later transformations. Fresh

**Figure 5** Games 1 and 2: anticipating calls to $\mathcal{G}$ and removing signing collisions

```
1: oracle H_1(c, m, r)
2:     if (m, r) ∉ dom(h) then
3:         ω ← {0,1}^{k_h}
4:         h[m, r] ← (ω, c, ⊥)
5:         st ← G(c, ω)
6:     else
7:         ω ← π_1(h[m, r])
8:         if c = Adv then
9:             st ← ⊥
10:        else
11:            st ← G(c, ω)
12:    return (ω, st)
```

```
1: oracle H_2(c, m, r)
2:     if (m, r) ∉ dom(h) ∨ c = FSig ∨
       (c = Sig ∧ π_2(h[m, r]) = FSig)
   then
3:         ω ← {0,1}^{k_h}
4:         st ← {0,1}^{k_g}
5:         if c ≠ FSig ∨ (m, r) ∉ dom(h)
   then
6:             h[m, r] ← (ω, c, ⊥)
7:         if c ≠ FSig ∨ ω ∉ dom(g) then
8:             g[ω] ← (st ⊕ (r || 0^{k_g − k_0}), c)
9:     else
10:        ω ← π_1(h[m, r])
11:        if c = Adv then
12:            st ← ⊥
13:        else
14:            (ω, st) ← ⊥
15:    return (ω, st)
```

$$\Pr[\mathit{Game0} : \mathit{win}] \leq \Pr[\mathit{Game2} : \mathit{win}] + (q_{\mathcal{H}} + q_{\mathcal{S}} + q_{\mathcal{F}}) \cdot \left( \frac{q_{\mathcal{S}} + q_{\mathcal{F}}}{2^{k_0}} + \frac{q_{\mathcal{G}} + q_{\mathcal{H}} + q_{\mathcal{S}} + q_{\mathcal{F}}}{2^{k_h}} \right)$$

queries are treated normally. Non-fresh queries made by the signing oracles are resampled as fresh if the previous query had been made by the faulty signature oracle. Non-fresh queries made by the faulty signature oracle are resampled, but not stored into the state. Game 1 is perfectly indistinguishable from Game 0, and Game 2 can be distinguished from Game 1 if either i. (lines 2, 5 and 6) the fresh $r$ used in $\mathcal{H}$-queries made by the signing oracles collides with a previously used $r$ (with probability at most $(q_{\mathcal{S}}+q_{\mathcal{F}})\cdot(q_{\mathcal{H}}+q_{\mathcal{S}}+q_{\mathcal{F}})\cdot 2^{-k_0}$); ii. (lines 4, 7 and 8) or the fresh $\omega$ used in $\mathcal{G}$-queries made by the signing oracles collides with a previously used $\omega$ (with probability at most $(q_{\mathcal{H}} + q_{\mathcal{S}} + q_{\mathcal{F}}) \cdot (q_{\mathcal{G}} + q_{\mathcal{H}} + q_{\mathcal{S}} + q_{\mathcal{F}}) \cdot 2^{-k_h}$). Note that the value stored in $g[\omega]$ at line 8 in $\mathcal{H}_2$ is uniformly distributed since $st$ is.

*Game 3.* Given that $\mathcal{H}$ now samples both bitstrings that compose the final padded message, we compute the entire signature in $\mathcal{H}$ when called by either one of the signing oracles. We transform the experiment to sample an integer $x^*$ and compute $y^* = x^{*\,e} \bmod N$ to serve as one-way challenge. We embed it in the state when replying to $\mathcal{H}$ queries made by the adversary. Everything up to this point has been set up so that the signing oracles can simply use $\pi_3(h[m, r])$ as the padded message for $m$ with salt $r$. Game 3 includes this simplification. We introduce additional notation for clarity in the rest of the proof. Consider the

function:

$$f_{(e,N),y^*,c} : \sigma \mapsto \begin{cases} y^* \cdot \sigma^e \bmod N \text{ if } c = \mathsf{Adv} \\ \sigma^e \bmod N \qquad \text{otherwise} \end{cases}$$

For a set $X \subseteq \mathbb{Z}/N\mathbb{Z}$, we denote by $\mathsf{pim}_{(e,N),y^*,c}(X)$ the uniform distribution on the set $S = \left\{\sigma \in \mathbb{Z}/N\mathbb{Z} \mid f_{(e,N),y^*,c}(\sigma) \in X\right\}$.

**Figure 6** Games 3 and 4: Embedding one-way challenge and oracle queries in $\mathcal{F}$

**oracle** $\mathcal{H}_3(c, m, r)$
  **if** $(m, r) \notin \mathsf{dom}(h) \vee c = \mathsf{FSig} \vee (c = \mathsf{Sig} \wedge \pi_2(h[m, r]) = \mathsf{FSig})$ **then**
    $\sigma \leftarrow \mathsf{pim}_{(e,N),y^*,c}\left([0..2^{k-1})\right)$
    $y \leftarrow f_{(e,N),y^*,c}(\sigma)$
    $b \,||\, \omega \,||\, st \leftarrow \mathsf{i2osp}(y)$
    **if** $c \neq \mathsf{FSig} \vee (m, r) \notin \mathsf{dom}(h)$ **then**
      $h[m, r] \leftarrow (\omega, c, \sigma)$
    **if** $c \neq \mathsf{FSig} \vee \omega \notin \mathsf{dom}(g)$ **then**
      $g[\omega] \leftarrow (st \oplus (r \,||\, 0^{k_g - k_0}), c)$
  **else**
    $\omega \leftarrow \pi_1(h[m, r])$
    **if** $c = \mathsf{Adv}$ **then**
      $st \leftarrow \bot$
    **else**
      $(\omega, st) \leftarrow \bot$
  **return** $(\omega, st)$

**oracle** $\mathcal{H}_4(c, m, r)$
  **if** $(m, r) \notin \mathsf{dom}(h) \vee c = \mathsf{FSig}$ **then**
    $\sigma \leftarrow \mathsf{pim}_{(e,N),y^*,c}\left([0..2^{k-1})\right)$
    $y \leftarrow f_{(e,N),y^*,c}(\sigma)$
    $b \,||\, \omega \,||\, st \leftarrow \mathsf{i2osp}(y)$
    **if** $c \neq \mathsf{FSig}$ **then**
      $h[m, r] \leftarrow (\omega, c, \sigma)$
      $g[\omega] \leftarrow (st \oplus (r \,||\, 0^{k_g - k_0}), c)$
  **else**
    $\omega \leftarrow \pi_1(h[m, r])$
    **if** $c = \mathsf{Adv}$ **then**
      $st \leftarrow \bot$
    **else**
      $(\omega, st) \leftarrow \bot$
  **return** $(\omega, st)$

$$\Pr[\mathit{Game2} : \mathit{win}] \leq \Pr[\mathit{Game3} : \mathit{win}] + 2^{-\frac{k}{2}+2} \qquad \Pr[\mathit{Game3} : \mathit{win}] \leq \Pr[\mathit{Game4} : \mathit{win}] + \frac{q_{\mathcal{H}} \cdot q_S}{2^{k_0}} + \frac{q_{\mathcal{G}} \cdot q_{\mathcal{F}} \cdot 3}{2^{k_h}}$$

Game 3 is indistinguishable from Game 2 exactly when $x^*$ is invertible. Therefore, the probability that the adversary distinguishes the two games is exactly $\frac{p+q-1}{p \cdot q}$. We have $p + q - 1 \leq 2^{\frac{k}{2}+1}$ and $2^{k-1} \leq p \cdot q$ and we can therefore bound the probability of this simulation failing by $2^{-\frac{k}{2}+2}$. Since the invertibility of $x^*$ is important in some later steps, we in fact let $\mathcal{H}$ compute a response only when $x^*$ is invertible. In the inverter, since $x^*$ is not public, we instead check the invertibility of $y^*$, which is equivalent. For simplicity, we omit discussions regarding this detail in the rest of this section.

*Game 4.* In this game, we stop keeping track of the random oracle queries made by the faulty signature oracle. This is an important step towards being able to apply Lemma 2, which only discusses the statistical distance between two

distributions on $\sigma'$, rather than $(\omega, \sigma')$. Note that, in Coron and Mandal's proof, Lemma 2 is applied before this transition, in a context in which its premises are not fulfilled. By removing data about random oracle queries, we introduce observable changes in the game's behaviour whenever the adversary queries $\mathcal{H}$ with an $r$ that was used previously in a faulty signature query, or whenever the adversary queries $\mathcal{G}$ with an $\omega$ that was used previously in a faulty signature query. We bound the probability of the adversary guessing an $\omega$ value using Lemma 3. Since the view of the adversary does not depend on $r$ values sampled by the faulty signature oracle (see Fig. 7), the probability of the adversary guessing an $r$ value used in generating a faulty signature is easily bounded.

*Game 5.* Our main goal at this stage is to show that faulty signatures are in fact indistinguishable from uniform randomness and can be simulated without using the random oracles. Once this is done, we will be able to resume the proof of security following more standard PSS proofs.

We now use Lemma 2 to completely simulate faulty signature oracle queries. We focus on the faulty signature oracle, inlining and simplifying $\mathcal{H}$ knowing that $c = \mathsf{FSig}$. On the left, we display the simplified faulty signature oracle from Game 4 for reference. We make use of elementary properties of the statistical

**Figure 7** Game 5: sampling faulty signatures

| | |
|---|---|
| 1: **oracle** $\mathcal{F}_4(m, \epsilon, a)$ | 1: **oracle** $\mathcal{F}_5(m, \epsilon, a)$ |
| 2: $\quad r \leftarrow \{0,1\}^{k_0}$ | 2: $\quad r \leftarrow \{0,1\}^{k_0}$ |
| 3: $\quad \sigma \leftarrow \mathsf{pim}_{(e,N),y^*,c}\left(\left[0..2^{k-1}\right)\right)$ | 3: $\quad \sigma' \leftarrow [0..N)$ |
| 4: $\quad y \leftarrow \sigma^e \bmod N$ | 4: $\quad$ **return** $\mathsf{i2osp}(\sigma')$ |
| 5: $\quad r' \leftarrow \{0,1\}^{\rho} \setminus 0$ | |
| 6: $\quad \sigma' \leftarrow y^d * \alpha_p + (a + (y - a^e)) * \alpha_q$ | |
| 7: $\quad$ **return** $\mathsf{i2osp}(\sigma')$ | |

$$\Pr[\mathit{Game4} : \mathit{win}] \leq \Pr[\mathit{Game5} : \mathit{win}] + \frac{q_{\mathcal{F}} \cdot (4e_{max}^2 + 1) \cdot k^2}{2^{\rho - k/2}}$$

distance and Lemma 2 to bound the probability of distinguishing Games 5 and 6. Note that sampling $\sigma$ in $\mathsf{pim}_{(e,N),y^*,c}\left(\left[0..2^{k-1}\right)\right)$ and applying the public RSA permutation to obtain $y$ is perfectly equivalent to sampling $y$ in $\left[0..2^{k-1}\right)$.

*Game 6.* With the faulty signature oracle simplified away, we can now focus on simulating the signature oracle. From now on, the $c$ argument to $\mathcal{H}$ can no longer be $\mathsf{FSig}$. More generally, it is impossible for any entry in $h$ or $g$ to be tagged with $\mathsf{FSig}$. The signature oracle we have defined at this point is not a valid simulator as it does not run in polynomial time. To ensure that it does, we replace the sampling operation at line 3 in Fig. 6 (right) with the loop displayed on the left of Fig. 8 to sample $\sigma$. The adversary can distinguish the two games whenever the loop finishes in a state where $y$ does not start with a 0 bit. At each iteration

**Figure 8** Game 6 and inverter: sampling $\sigma$ in polynomial time

| | |
|---|---|
| 1: **while** $(!0 \leq y < 2^{k-1}) \wedge i < k_0$ **do** | 1: **oracle** $\mathcal{I}(e, N, y^*)$ |
| 2: $\quad \sigma \leftarrow [0..N)$ | 2: $\quad (m, s) \leftarrow \mathcal{A}^{\mathcal{H}_7, \mathcal{G}_7, \mathcal{S}_7, \mathcal{J}_7}(e, N)$ |
| 3: $\quad y \leftarrow f_{(e,N),y^*,c}(\sigma)$ | 3: $\quad \sigma \leftarrow \mathsf{os2ip}(s)$ |
| 4: $\quad i \leftarrow i + 1$ | 4: $\quad y \leftarrow \sigma^e \bmod N$ |
| | 5: $\quad b \,\|\|\, \omega \,\|\|\, st \leftarrow \mathsf{i2osp}(y)$ |
| | 6: $\quad r \,\|\|\, \gamma \leftarrow st \oplus g[\omega]$ |
| | 7: $\quad (\omega', \mathsf{Adv}, u) \leftarrow h[m, r]$ |
| | 8: $\quad$ **return** $\sigma \cdot u^{-1}$ |

$$\Pr[\mathit{Game5} : \mathit{win}] \leq \Pr[\mathit{Game6} : \mathit{win}] + \frac{q_{\mathcal{H}} + q_S}{2^{k_0}} \qquad \Pr[\mathit{Game6} : \mathit{win}] \leq \Pr[\mathcal{OW}\text{-}\mathcal{RSA}^{\mathcal{I}} : x = x^*] + \frac{1}{2^{k_h}} + \frac{q_{\mathcal{H}}}{2^{\frac{k}{2}-1}}$$

of the loop, the $\sigma$ sampled is invalid with probability at most $\frac{1}{2}$. The probability that *all* iterations produce an invalid $\sigma$ is therefore bounded by $\frac{1}{2^{k_0}}$, since all samples are independent. $\mathcal{H}_7$ may now be queried $q_{\mathcal{H}} + q_S$ times, allowing us to conclude.

*Reduction* All the oracles are simulated without using any secret data. We now focus on building an inverter. The adversary can win in two disjoint cases:

- either the $\mathcal{H}$-query made by the verification algorithm is fresh (this occurs with probability at most $2^{-k_h}$),
- or the $\mathcal{H}$-query made by the verification algorithm was previously made by the adversary. If the query was made by the signature oracle, the forgery cannot be fresh and the adversary cannot win.

In the latter case, the one-way challenge can then be recovered by the inverter shown on the right of Fig. 8. The key observation is that, in case of a successful forgery, we have $y = \sigma^e \bmod N$ (line 4) and $y = y^* \cdot u^e \bmod N$ (by invariant on $h$). By definition of $y^*$ and the morphism and injectivity properties of RSA, we therefore have $\sigma = x^* \cdot u$. We need to also consider the case where a value $u$ stored in the $h$ map by the adversary is not invertible, which occurs with probability at most $q_{\mathcal{H}} \cdot 2^{-k/2+1}$.

The final bound is obtained by transitively using the individual transition bounds.

## 5 Conclusion

In this paper we have proven the securrity of an infective countermeasure against the null fault model used in [17] and against more powerfull faults than random faults in the sense that we let the adversary chose the value of the result. It would be interesting in the future to take into account faults at each step of the protected algorithm instead of just the CRT-computation step. Moreover it would be a great addition to consider more fault models.

## References


1. J. B. Almeida, M. Barbosa, G. Barthe, and F. Dupressoir. Certified computer-aided cryptography: efficient provably secure machine code from high-level implementations. In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM CCS 13*, pages 1217–1230, Berlin, Germany, Nov. 4–8, 2013. ACM Press.
2. R. J. Anderson and M. G. Kuhn. Low cost attacks on tamper resistant devices. In B. Christianson, B. Crispo, T. M. A. Lomas, and M. Roe, editors, *Security Protocols Workshop*, volume 1361 of *Lecture Notes in Computer Science*, pages 125–136. Springer, 1997.
3. C. Aumüller, P. Bier, W. Fischer, P. Hofreiter, and J.-P. Seifert. Fault attacks on RSA with CRT: Concrete results and practical countermeasures. In B. S. Kaliski Jr., Çetin Kaya. Koç, and C. Paar, editors, *CHES 2002*, volume 2523 of *LNCS*, pages 260–275, Redwood Shores, California, USA, Aug. 13–15, 2002. Springer, Berlin, Germany.
4. G. Barthe, F. Dupressoir, P.-A. Fouque, B. Grégoire, M. Tibouchi, and J.-C. Zapalowicz. Making RSA-PSS provably secure against non-random faults. Cryptology ePrint Archive, Report 2014/252, 2014. http://eprint.iacr.org/.
5. G. Barthe, B. Grégoire, S. Heraud, and S. Z. Béguelin. Computer-aided security proofs for the working cryptographer. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 71–90, Santa Barbara, CA, USA, Aug. 14–18, 2011. Springer, Berlin, Germany.
6. J. Blömer, M. Otto, and J.-P. Seifert. A new CRT-RSA algorithm secure against Bellcore attacks. In *ACM Conference on Computer and Communications Security*, pages 311–320, 2003.
7. D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of eliminating errors in cryptographic computations. *Journal of Cryptology*, 14(2):101–119, 2001.
8. A. Boscher, H. Handschuh, and E. Trichina. Fault resistant RSA signatures: Chinese remaindering in both directions. *IACR Cryptology ePrint Archive*, 2010:38, 2010.
9. R. Canetti and S. Goldwasser. An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 90–106, Prague, Czech Republic, May 2–6, 1999. Springer, Berlin, Germany.
10. M. Christofi, B. Chetali, L. Goubin, and D. Vigilant. Formal verification of a CRT-RSA implementation against fault attacks. *J. Cryptographic Engineering*, 3(3):157–167, 2013.
11. M. Ciet and M. Joye. Practical fault countermeasures for Chinese remaindering based cryptosystems. In L. Breveglieri and I. Koren, editors, *FDTC*, pages 124–131, 2005.
12. J.-S. Coron, C. Giraud, N. Morin, G. Piret, and D. Vigilant. Fault attacks and countermeasures on Vigilant's RSA-CRT algorithm. In *FDTC*, pages 89–96, 2010.
13. J.-S. Coron, A. Joux, I. Kizhvatov, D. Naccache, and P. Paillier. Fault attacks on RSA signatures with partially unknown messages. In *CHES*, pages 444–456, 2009.
14. J.-S. Coron and A. Mandal. PSS is secure against random fault attacks. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 653–666, Tokyo, Japan, Dec. 6–10, 2009. Springer, Berlin, Germany.
15. J.-S. Coron and A. Mandal. PSS is secure against random fault attacks. In *ASIACRYPT*, pages 653–666, 2009.
16. J.-S. Coron, D. Naccache, and M. Tibouchi. Fault attacks against EMV signatures. In *CT-RSA*, pages 208–220, 2010.

17. P.-A. Fouque, N. Guillermin, D. Leresteux, M. Tibouchi, and J.-C. Zapalowicz. Attacking RSA-CRT signatures with faults on Montgomery multiplication. *J. Cryptographic Engineering*, 3(1):59–72, 2013.
18. C. Giraud. An RSA implementation resistant to fault attacks and to simple power analysis. *IEEE Trans. Computers*, 55(9):1116–1120, 2006.
19. D.-P. Le, M. Rivain, and C. H. Tan. On double exponentiation for securing RSA against fault analysis. In J. Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 152–168, San Francisco, CA, USA, Feb. 25–28, 2014. Springer, Berlin, Germany.
20. H. L. Montgomery. *Topics in multiplicative number theory*. Springer-Verlag, 1971.
21. P. L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44:519–521, 1985.
22. N. Moro, K. Heydemann, E. Encrenaz, and B. Robisson. Formal verification of a software countermeasure against instruction skip attacks. *Journal of Cryptographic Engineering*, pages 1–12, 2014.
23. P. Rauzy and S. Guilley. A formal proof of countermeasures against fault injection attacks on CRT-RSA. *Journal of Cryptographic Engineering*, pages 1–13, 2013.
24. M. Rivain. Securing RSA against fault analysis by double addition chain exponentiation. In M. Fischlin, editor, *CT-RSA 2009*, volume 5473 of *LNCS*, pages 459–480, San Francisco, CA, USA, Apr. 20–24, 2009. Springer, Berlin, Germany.
25. A. Shamir. Improved method and apparatus for protecting public key schemes from timing and fault attacks. Patent Application, 1998. WO 1998/052319 A1.
26. E. Trichina and R. Korkikyan. Multi fault laser attacks on protected CRT-RSA. In L. Breveglieri, M. Joye, I. Koren, D. Naccache, and I. Verbauwhede, editors, *FDTC*, pages 75–86. IEEE Computer Society, 2010.
27. D. Vigilant. RSA with CRT: A new cost-effective solution to thwart fault attacks. In E. Oswald and P. Rohatgi, editors, *CHES 2008*, volume 5154 of *LNCS*, pages 130–145, Washington, D.C., USA, Aug. 10–13, 2008. Springer, Berlin, Germany.
28. S.-M. Yen, S.-J. Moon, and J. Ha. Permanent fault attack on the parameters of RSA with CRT. In R. Safavi-Naini and J. Seberry, editors, *ACISP 03*, volume 2727 of *LNCS*, pages 285–296, Wollongong, NSW, Australia, July 9–11, 2003. Springer, Berlin, Germany.