Published in A. Kholosha, E. Rosnes, and M. Parker, Eds, *International Workshop on Coding and Cryptography (WCC 2009)*, Preproceedings, pp. 142–157, 2009.

# How (Not) to Design Strong-RSA Signatures

Marc Joye

Thomson R&D France
Technology Group, Corporate Research, Security Laboratory
1 avenue de Belle Fontaine, 35576 Cesson-Sévigné Cedex, France
marc.joye@thomson.net – http://www.geocities.com/MarcJoye/

**Abstract.** This paper considers strong-RSA signature schemes built from the scheme of Cramer and Shoup. We present a basic scheme encompassing the main features of the Cramer-Shoup scheme. We analyze its security in both the random oracle model and the standard model. This helps us to spot potential security flaws. As a result, we show that a seemingly secure signature scheme (*Int. J. of Security and Networks*, 2006) is universally forgeable under a known-message attack. In a second step, we discuss how to turn the basic scheme into a fully secure signature scheme. Doing so, we rediscover several known schemes (or slight variants thereof).

**Key words:** Digital signature, strong RSA assumption, Cramer-Shoup signature scheme, standard model

## 1 Introduction

Since the invention of public-key cryptography by Diffie and Hellman [12], numerous encryption schemes and signature schemes have been proposed. Some of them were shown to be vulnerable.

The quest for schemes being at the same time efficient and secure is central in cryptography. Usually, the security of a cryptographic scheme is assessed through the simulation/reduction paradigm: a security "proof" is a computational reduction between a well-established hard problem and an attack against the scheme. Examples of hard problems used to build cryptographic schemes include the factorization problem, the RSA problem or the strong RSA problem.

The highest security notion for signature schemes is unforgeability against chosen-messages attacks, as introduced by Goldwasser, Micali, and Rivest [19]. In the same paper, they also proposed a signature scheme meeting this security notion (see also [18, 27]). The efficiency of their scheme was subsequently improved by Dwork and Naor [14], by Cramer and Damgård [10], and more recently, by Catalano and Gennaro [7].

Goldwasser-Micali-Rivest signature scheme and its improved versions are based on signing trees. The main drawback of these schemes is that the signature on a message depends on previously signed messages.

To overcome this, *stateless* signature schemes [18] were later proposed. These include the Gennaro-Halevi-Rabin signature scheme (GHR) [17] and the Cramer-Shoup signature scheme (CS) [11], independently introduced in 1999. These two

schemes require the strong-RSA assumption [1, 16]. The GHR scheme further assumes so-called *division-intractable hash functions* (see [17] for a precise definition). In [9], Coron and Naccache pointed out that the easiest way to achieve this additional requirement is to construct hash functions mapping strings to prime numbers. A candidate injective function mapping to prime numbers is proposed in [26]; this paper also presents a twin version of the (basic) GHR scheme built on this function. A weakened version of division-intractability and companion signature schemes are described in [24]. Finally, an efficient variant of GHR, not relying on special-type functions, is presented in [8]. Very recently, Hofheinz and Kiltz designed a signature scheme relying on the strong-RSA assumption from a (bounded) programmable hash function [20].

This paper is interested in signature schemes of the CS family. In contrast to schemes of the GHR family, there are many known variations of the signature scheme originally proposed by Cramer and Shoup. We quote the works of Zhu [35, 36], Camenisch and Lysyanskaya [4], Popescu [29], Tan, Yi, and Siew [33] (shown to be insecure in [21]), Fischlin [15], Tan [31, 32], Cao and Liu [6], and Yu and Tate [34]. A description of these schemes can be found in Appendix A.

We present a basic (signature) scheme encompassing the main features of CS-like signature schemes. Its security proof can be divided in two main cases. While the first case is easy to prove, the second case is more difficult to tackle. We analyze it in both the random oracle model and in the standard model. This helps us to spot potential security flaws. As an application, we show that a seemingly secure signature scheme is actually insecure. For a given valid signature, we will see that it is easy to derive a signature on *any* chosen message. Our analysis also yields conditions that have to be fulfilled to get a secure scheme. We so rediscover most of the known CS-like signature schemes, or slight variants thereof. In certain cases, the obtained schemes are even simpler.

The rest of this paper is organized as follows. In the next section, we review some definitions and introduce some useful notation. Section 3 describes our basic scheme and presents a security analysis thereof. In Section 4, from our analysis, we show that Tan's signature scheme is insecure. We also explain how to adapt the basic scheme so as to obtain secure signature schemes. Finally, we conclude in Section 5.

Appendix A review all CS-like signature schemes proposed so far. Appendix B presents a better security proof of Zhu's signature scheme (or more exactly, of a slightly revised version thereof).

## 2 Preliminaries

We start by introducing some notation. If $x_1$ and $x_2$ are bit-strings then $x_1 \| x_2$ denote a string encoding $x_1$ and $x_2$ from which $x_1$ and $x_2$ are uniquely recovered. For convenience, we often identify an integer with its binary representation: a bit-string $x \in \{0,1\}^\ell$ is also viewed as an integer in $[0, 2^\ell - 1]$. We say that $x$ is an $\ell$-bit integer if $x$ is an integer in the range $[2^{\ell-1}, 2^\ell - 1]$. An (odd) prime $p$ is

a strong prime if $p = 2p' + 1$ with $p'$ prime. An RSA modulus $N = pq$ is said to be *safe* if it is the product of two equal-size strong primes.

The resources of an adversary, including the running time and the number of oracle queries, are measured asymptotically as a function of a security parameter $k$. A function $\nu(k)$ is *negligible* (in $k$) if for all $c \in \mathbb{Z}_{>0}$ there exists an integer $k_c$ such that $\nu(k) < k^{-c}$ for all $k > k_c$.

A *signature scheme* is a tuple of algorithms $\mathsf{Sig} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ with running time polynomial in a security parameter $k$.

**Key Generation** On input security parameter $1^k$, algorithm $\mathsf{KeyGen}$ produces a pair $(\mathsf{pk}, \mathsf{sk})$ of matching public and private keys.

**Signature Generation** Given a message $m$ in a set $\mathcal{M}$ of messages and a private signing key $\mathsf{sk}$ (with corresponding public key $\mathsf{pk}$), $\mathsf{Sign}$ produces a signature $\sigma$. The signing algorithm can be probabilistic.

**Signature Verification** Given a signature $\sigma$, a message $m \in \mathcal{M}$ and a public key $\mathsf{pk}$, $\mathsf{Verify}$ checks whether $\sigma$ is a valid signature on $m$ with respect to $\mathsf{pk}$.

For security we consider the customary notion of *existential unforgeability* ($\mathsf{EUF}$) against *chosen-message attacks* ($\mathsf{CMA}$), as defined by Goldwasser, Micali, and Rivest [19]. An existential forgery is a signature on a new message, valid and generated by the adversary. The verification key is public to anyone, including to the adversary. But more information may also be available. The strongest kind of attack scenario is formalized by *adaptive chosen-message attacks*, where the adversary can ask the signer to sign any message of her choice, in an adaptive way. More formally, a signature scheme $\mathsf{Sig}$ is said *secure* if the success probability

$$\mathsf{Succ}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{Sig}}(\mathcal{A}, q_s) := \Pr\left[\begin{array}{c}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^k), (m_*, \sigma_*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk};\cdot)}(\mathsf{pk}) : \\ \mathsf{Verify}(\mathsf{pk}; m_*, \sigma_*) = \text{true}\end{array}\right]$$

is negligible for any polynomial-time adversary $\mathcal{A}$, making (at most) $q_s$ queries to a signing oracle $\mathsf{Sign}(\mathsf{sk};\cdot)$, and returning a valid signature $\sigma_*$ on a message $m_*$ that was not submitted to the signing oracle.

The security of signature schemes we consider in this paper is conditioned to a certain intractability assumption, namely the *strong RSA assumption* ($\mathsf{sRSA}$), introduced in [1, 16]. This assumption says that it is hard, on input a safe RSA modulus $N$ and a random element $z \in \mathbb{Z}_N^*$, to find a pair $(w, e) \in \mathbb{Z}_N \times \mathbb{Z}_{>1}$ satisfying $z \equiv w^e \pmod N$. More formally, the success probability of any probabilistic polynomial-time adversary $\mathcal{A}$,

$$\Pr\big[N \leftarrow \mathsf{sRSA}(1^k), z \leftarrow \mathbb{Z}_N^*, (w, e) \leftarrow \mathcal{A}(N, z) : w^e \equiv z \pmod N \ \wedge\ e > 1\big]$$

is assumed to be negligible. The difference with the ordinary RSA assumption [30] is that exponent $e$ can be freely chosen in $\mathbb{Z}_{>1}$ and is not a priori fixed. The strong RSA assumption is a potentially stronger assumption but at present time, the only known way to break either assumption is to solve the integer factorization problem.

Recently, Paillier [28] pointed out that the impossibility of constructing a provably secure signature scheme — under a black-box reduction — based on the (ordinary) RSA assumption (see also [13]). It is however possible to prove the security of signature schemes under the sole RSA assumption in the so-called *random oracle model* [2]. The random oracle model assumes that a hash function behaves as a random function. This assumption allows one to design very efficient schemes. Examples include the FDH scheme [2] (see also [22] for a slight variant featuring a tight security reduction) and the PSS scheme [3]. A proof in the random oracle is weaker than a proof in the standard model. In [5], Canetti, Goldreich and Halevi gave a (contrived) example of a signature scheme secure in the random oracle model but insecure under any instantiation of the random oracle. Despite this, the random oracle methodology has proved extremely useful in the analysis of cryptographic schemes. A proof in the random oracle model indicates that a scheme is not essentially flawed in the sense that an attacker, in order to be successful, should use the hash function in a non-generic way.

## 3 Basic Strong-RSA Signature Scheme

### 3.1 Description

We define a basic signature scheme that captures the main properties of CS-like signature schemes. The scheme is parameterized by two parameters, $\ell$ and $\ell'$, with $\ell' > \ell + 1$. Typical values are $\ell = 160$ and $\ell' = 512$. It also requires a collision-resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^{\ell}$.

**Key Generation** Define $N = pq$ for two random strong $\ell'$-bit primes $p = 2p'+1$ and $q = 2q' + 1$ where $p'$ and $q'$ are prime. Let $g$ and $X$ be two random quadratic residues in $\mathbb{Z}_N^*$.

The public key is $\mathsf{pk} = \{N, X, g\}$ and the private key is $\mathsf{sk} = \{p', q'\}$.

**Signature Generation** To sign a message $m \in \{0,1\}^*$ (viewed as an arbitrary bit string), a random $(\ell+1)$-bit prime $e$ is chosen as well as a random integer $u \in \mathbb{Z}_e^*$. Next, using her private key $\mathsf{sk}$, the signer calculates

$$y = (X^u \, g^{H(m)})^{e^{-1} \bmod p'q'} \bmod N \ .$$

The signature on message $m$ is $\sigma = (y, u, e)$.

**Signature Verification** To verify a putative signature $\sigma = (y, u, e)$ on a message $m$, it is checked that $(i)$ $e$ is an odd $(\ell + 1)$-bit integer, $(ii)$ $0 < u < e$, and $(iii)$ $y^e \equiv X^u \, g^{H(m)} \pmod N$. If all verifications succeed, the signature is accepted.

### 3.2 Security Analysis

We analyze the security of the previous scheme in the sense of EUF-CMA (cf. Section 2). As in the original Cramer-Shoup signature scheme, there are essentially two main cases to consider.

We start with the easiest case, namely when attacker $\mathcal{A}$ returns a signature forgery, $\sigma_* = (y_*, u_*, e_*)$, with a value for $e_*$ different from all $e_i$'s returned by the signing oracle. The next lemma shows that the signature scheme is then secure under the strong RSA assumption. The proof in this case is an adaptation of the proof given in [11].

**Lemma 1.** *Given on input public key* $\mathsf{pk} = \{N, g, X\}$ *and* $q_s$ *accesses to a signing oracle returning signatures* $\sigma_i = (y_i, u_i, e_i)$ *on chosen messages* $m_i$ *generated as described in § 3.1, it is infeasible to generate a signature* $\sigma_* = (y_*, u_*, e_*)$ *on a message* $m_*$ *that succeeds the verification step, with*

$$e_* \neq e_i \quad \textit{for all } i \in \{1, \ldots, q_s\}\,,$$

*under the strong RSA assumption.*

We stress that this lemma does not imply the security of the signature scheme. It only shows how to break the strong-RSA assumption from a particular type of forgery.

*Proof (of Lemma 1).* On input a strong RSA modulus $N$ and a random element $z \in \mathbb{Z}_N^*$, one has to find a pair $(w, e) \in \mathbb{Z}_N^* \times \mathbb{Z}_{>1}$ such that $z \equiv w^e \pmod N$.

We define $E = \prod_{1 \leq i \leq q_s} e_i$ for $q_s$ randomly chosen $(\ell+1)$-bit primes, $e_1, \ldots, e_{q_s}$. We also define $g = z^{2E} \bmod N$ and $X = g^r \bmod N$ for some random integer $r \in \{1, \ldots, N^2\}$. We let $\mathsf{pk} = \{N, g, X\}$. The signature on a given message $m_i$ is easily simulated as $\sigma_i = (y_i, u_i, e_i)$ where $u_i$ is chosen at random in $\mathbb{Z}_{e_i}^*$ and $y_i = (z^{2(ru_i + H(m_i))})^{d_i} \bmod N$ with $d_i = E/e_i \in \mathbb{Z}$. [Observe that ${y_i}^{e_i} \equiv g^{ru_i + H(m_i)} \equiv X^{u_i}\, g^{H(m_i)} \pmod N$.]

Now, if $\sigma_* = (y_*, u_*, e_*)$ is a valid signature on message $m_*$ then ${y_*}^{e_*} \equiv X^{u_*}\, g^{H(m_*)} \equiv z^{2E(ru_* + H(m_*))} \pmod N$. Moreover, noting that $e_*$ is odd, if $e_* \neq e_i$ for all $i \in \{1, \ldots, q_s\}$, it follows that $\delta := \gcd\bigl(2E(ru_* + H(m_*)), e_*\bigr) = \gcd(ru_* + H(m_*), e_*) < e_*$ with overwhelming probability. Indeed, letting $r = r_1\, p'q' + r_0$ with $r_0 = r \bmod p'q'$ and $r_1 = \lfloor r/(p'q') \rfloor$, $e_*$ may depend on $r_0$ but as $r_0$ and $r_1$ are essentially independent, $\gcd(ru_* + H(m_*), e_*) \neq e_*$ with overwhelming probability. Hence, extended Euclid's algorithm yields $\alpha, \beta \in \mathbb{Z}$ satisfying $\alpha\bigl(2E(ru_* + H(m_*))\bigr) + \beta\, e_* = \delta$, which, in turn, yields

$$z \equiv z^{\alpha \frac{2E(ru_* + H(m_*))}{\delta} + \beta \frac{e_*}{\delta}} \equiv g^{\alpha \frac{ru_* + H(m_*)}{\delta}}\, z^{\beta \frac{e_*}{\delta}} \equiv ({y_*}^{\alpha}\, z^{\beta})^{\frac{e_*}{\delta}} \pmod N\ .$$

Therefore, $w := {y_*}^{\alpha}\, z^{\beta} \bmod N$ and $e := e_*/\delta > 1$ solves the strong RSA problem: $w^e \equiv z \pmod N$. □

The second case, namely when the attacker returns a signature forgery with a value for $e_*$ already returned by the signing oracle, is more difficult to handle. The problem is to find a way to simulate the signing oracle so that the returned forgery helps to solve a given strong-RSA challenge.

**Random Oracle Model** As a first step, to make the analysis easier, we make use of the random oracle methodology.

Again, the goal is, given a strong RSA modulus $N$ and a random element $z \in \mathbb{Z}_N^*$, to find a pair $(w, e) \in \mathbb{Z}_N^* \times \mathbb{Z}_{>1}$ satisfying $w^e \equiv z \pmod N$ —or a collision in hash function $H$. As in the proof of Lemma 1, we choose $q_s$ random $(\ell+1)$-bit primes $e_1, \ldots, e_{q_s}$. We also select at random an index $j \in \{1, \ldots, q_s\}$, hoping that the attacker will output a forgery with $e_* = e_j$. We define

$$E' = \prod_{\substack{1 \le i \le q_s \\ i \ne j}} e_i \,, \quad E = E'\, e_j \quad \text{and} \quad Z = z^{2E'} \bmod N \ . \tag{1}$$

We also define $g = Z^{u_j} \bmod N$ and $X = h^{2E}\, Z^{-c_j} \bmod N$ for some random integers $u_j \in \mathbb{Z}_{e_j}^*$, $h \in \mathbb{Z}_N^*$, and $c_j \in \{0,1\}^\ell$. We let $\mathsf{pk} = \{N, g, X\}$.

*Modified scheme I.* The random oracle serves to return $c_j$ on input $m_j$, namely, the $j$-th message queried to the signing oracle. However, as this value cannot be forced if this message $m_j$ was already queried to the random oracle, we slightly modify the basic signature scheme by including the value of $e_j$ as an input to hash function $H$. The signature $\sigma = (y, u, e)$ on a message $m$ is defined as in § 3.1 except that $y$ is replaced with

$$y = \left(X^u\, g^{H(m\|e)}\right)^{e^{-1} \bmod p'q'} \bmod N \ . \tag{2}$$

Note that Lemma 1 remains valid with this modification.

If $(\ell + 1)$ is sufficiently large (and thus $e_j$ will be sufficiently large), the event that the attacker queries directly the random oracle with $m_j \| e_j$ (where $m_j$ denotes the $j$-th message submitted to the signing oracle) is very unlikely.

For $i \in \{1, \ldots, q_s\}$, with $i \ne j$, the signature on a message $m_i$ is simulated similarly to what was done in the proof of Lemma 1: $\sigma_i = (y_i, u_i, e_i)$ where $y_i = \left(h^{2e_j u_i}\, z^{2(-c_j u_i + u_j H(m_i\|e_i))}\right)^{E'/e_i} \bmod N$ for some random integer $u_i \in \mathbb{Z}_{e_i}^*$. Observe that ${y_i}^{e_i} \equiv h^{2E' e_j u_i}\, z^{-2E' c_j u_i} z^{2E' u_j H(m_i\|e_i)} \equiv h^{2Eu_i}\, Z^{-c_j u_i}\, Z^{u_j H(m_i\|e_i)} \equiv X^{u_i}\, g^{H(m_i\|e_i)} \pmod N$. When $i = j$, the corresponding signature is $\sigma_j = (y_j, u_j, e_j)$ where $y_j = h^{2E' u_j} \bmod N$ (and $H(m_j\|e_j) := c_j$). It is easily verified that ${y_j}^{e_j} \equiv h^{2Eu_j} \equiv (h^{2Eu_j}\, Z^{-c_j u_j})\, Z^{c_j u_j} \equiv X^{u_j}\, g^{H(m_j\|e_j)} \pmod N$.

Let $\sigma_* = (y_*, u_*, e_*)$ with $e_* = e_j$ denote a signature forgery on a message $m_*$. From ${y_*}^{e_*} \equiv X^{u_*}\, g^{H(m_*\|e_*)} \pmod N$ and ${y_j}^{e_j} \equiv X^{u_j}\, g^{H(m_j\|e_j)} \pmod N$, letting $c_* := H(m_*\|e_*)$, we obtain

$$\left(\frac{y_*}{y_j}\right)^{e_*} \equiv X^{u_* - u_j}\, g^{c_* - c_j} \equiv h^{2E(u_* - u_j)}\, Z^{-c_j(u_* - u_j)}\, Z^{u_j(c_* - c_j)} \pmod N$$

$$\Longrightarrow \left(\frac{y_*}{y_j}\, h^{2E'(u_j - u_*)}\right)^{e_*} \equiv Z^{-c_j(u_* - u_j) + u_j(c_* - c_j)} \equiv z^{2E'(u_j c_* - u_* c_j)} \pmod N \ .$$

Now, using extended Euclid's algorithm, we can find two integers $\alpha$ and $\beta$ such that

$$\delta := \gcd\big(2E'(u_j c_* - u_* c_j), e_*\big) = \alpha\big(2E'(u_j c_* - u_* c_j)\big) + \beta\, e_* \ . \tag{3}$$

Since $e_*$ is prime, the only two possible values for $\delta$ are 1 and $e_*$. If $\delta = 1$ then we are done; a solution $(w, e)$ to the strong RSA problem, satisfying $w^e \equiv z \pmod N$, is given by $w = \left((y_*/y_j)\, h^{2E'(u_j - u_*)}\right)^{\alpha} z^{\beta} \bmod N$ and $e = e_*$. If $\delta = e_*$ the previous methodology would lead to a trivial solution with $e := e_*/\delta = 1$. We do not know how to conclude. Actually, we will see in § 4.1 that such a value for $\delta$ can be turned into an actual attack to obtain the signature on any chosen message.

**Standard Model** The construction of the public key in the previous simulation can be slightly modified so as to allow one to generate a valid signature with $e_j$ without resorting on random oracles. Defining $E'$, $E$ and $Z$ as per Eq. (1), we define $g = Z^v \bmod N$ and $X = h^{2E}\, Z^{-1} \bmod N$ for some random integers $v \in \mathbb{Z}^*_{e_j}$ and $h \in \mathbb{Z}^*_N$. We let $\mathsf{pk} = \{N, g, X\}$.

The signature on message $m_j$ in the basic scheme (resp. modified scheme I) is simulated as $\sigma_j = (y_j, u_j, e_j)$ where

$$u_j = vc_j \bmod e_j \quad \text{with } c_j = H(m_j) \tag{4}$$

(resp. $c_j = H(m_j \| e_j)$) and $y_j = h^{2E'u_j}\, Z^t \bmod N$ with $t = (vc_j - u_j)/e_j \in \mathbb{Z}$. Again, it is easily verified that ${y_j}^{e_j} \equiv h^{2Eu_j}\, Z^{vc_j - u_j} \equiv X^{u_j}\, g^{c_j} \pmod N$. The simulation of signatures on message $m_i$ with $i \neq j$ is done as before. This is essentially the simulation given in [32].[1]

If $\sigma_* = (y_*, u_*, e_*)$ with $e_* = e_j$ denotes a signature forgery on a message $m_*$ then we get

$$\left(\frac{y_*}{y_j}\, h^{2E'(u_j - u_*)}\right)^{e_*} \equiv Z^{-(u_* - u_j)}\, Z^{v(c_* - c_j)} \equiv z^{2E'(vc_* - u_* - vc_j + u_j)} \pmod N\ .$$

From this, assuming that all $e_i$'s are different, it follows that

$$\delta := \gcd\big(2E'(vc_* - u_* - vc_j + u_j), e_*\big) = \gcd(vc_* - u_* - vc_j + u_j, e_*) \tag{5}$$

$$= \gcd(vc_* - u_*, e_*) \tag{6}$$

since from Eq. (4), noting that $e_j = e_*$, we have $-vc_j + u_j \equiv 0 \pmod{e_*}$.

The author of [32] incorrectly deduces from Eq. (5) that because of the choice of $v$ (cf. Footnote ([1])) the probability that $e_* \mid (vc_* - u_* - vc_j + u_j)$ is almost $1/2^{\ell+1}$ and so we must have $\delta = 1$ with high probability, implying that his scheme is secure because this yields a non-trivial $e_*$-root of $z$. This argument is incorrect. As shown in Eq. (6), $\delta$ is equal to $e_*$ (and thus $\neq 1$) if the attacker outputs a signature forgery $(y_*, u_*, e_*)$ with $e_* = e_j$ such that $vc_* - u_* \equiv 0 \pmod{e_*}$. Therefore, only the value of $v \pmod{e_*}$ matters and this value is known to the attacker from $\sigma_j$: from Eq. (4), we have $u_j/c_j \equiv v \pmod{e_*}$.

[1] We however note that, in [32], in the generation of $\mathsf{pk}$, $v$ is chosen as a $(\ell + 2)$-bit integer coprime with $e_j$ rather than a random integer in $\mathbb{Z}^*_{e_j}$.

## 4 [In]Secure Signature Schemes

### 4.1 Breaking Tan's Signature Scheme

Tan's signature scheme [32] (see also Appendix A) is almost the same as modified scheme I. The only difference is the presence of $\mathsf{pk}$ as an input of the hash function (compare Eq. (2) with Eq. (15) in appendix). The aim is to prevent impersonation attacks in a multi-user setting [25].

In the previous section, the analysis in the standard model tells us that the security proof offered in [32] is incorrect. But this does not mean that Tan's signature scheme is insecure. The analysis in the random oracle model already indicated that a signature forgery $\sigma_* = (y_*, u_*, e_*)$ with

$$e_* = e_j \quad \text{and} \quad u_j c_* - u_* c_j \equiv 0 \pmod{e_*} \tag{7}$$

implies that $\delta$, as defined by Eq. (3), is equal to $e_*$ and so the proof was not conclusive.

We now study the implications of such a choice for the forgery. Given a valid signature $\sigma = (y, u, e)$ on a message $m$, we let $c = H(\mathsf{pk}\|m\|e)$ and define $s = -\frac{u}{c} \bmod e$ and $t = (s\,c + u)/e$. We have:

$$y^e \equiv X^u\, g^c \equiv X^{t\,e - s\,c}\, g^c \pmod{N} \iff (y\,X^{-t})^e \equiv (g\,X^{-s})^c \pmod{N}\ .$$

Moreover, using extended Euclid's algorithm, we find integers $\alpha$ and $\beta$ such that $\alpha\,c + \beta\,e = \gcd(c, e) = 1$. Hence, using the previous relation, we get $g\,X^{-s} \equiv (g\,X^{-s})^{\alpha\,c + \beta\,e} \equiv (y\,X^{-t})^{\alpha\,e}\,(g\,X^{-s})^{\beta\,e} \pmod{N}$ and consequently

$$(g\,X^{-s})^{1/e} \equiv (y\,X^{-t})^{\alpha}\,(g\,X^{-s})^{\beta} \pmod{N}\ . \tag{8}$$

It is therefore easy for anyone (i.e., without the knowledge of private key $\mathsf{sk}$) to compute the signature $(y_*, u_*, e_*)$ on *any* chosen message $m_*$ as

$$\begin{cases} e_* = e & \\ u_* = -s\,c_* \bmod e & \text{with } c_* = H(\mathsf{pk}\|m_*\|e) \\ y_* = X^{t_*}\left[(y\,X^{-t})^{\alpha}\,(g\,X^{-s})^{\beta}\right]^{c_*} \bmod N & \text{with } t_* = (s\,c_* + u_*)/e \end{cases}\ . \tag{9}$$

*Proof.* As $e_* = e$ and as $u_*$ is an integer modulo $e$, the three first conditions of the signature verification (cf. Appendix A) are satisfied. Moreover, from Eqs (8) and (9), we get

$${y_*}^{e_*} \equiv \left[X^{t_*}\,(g\,X^{-s})^{c_*/e}\right]^e \equiv X^{e\,t_* - s\,c_*}\, g^{c_*} \equiv X^{u_*}\, g^{H(\mathsf{pk}\|m_*\|e)} \pmod{N}$$

as it is required. □

### 4.2 Secure Strong-RSA Signature Schemes

One way to prevent the previous attack consists in fixing parameter $u$ to a constant value in $]0, 2^\ell]$. Since parameter $e$ is supposed to be an $(\ell + 1)$-bit prime, Eq. (7) would then imply $c - c_* \equiv 0 \pmod{e_*}$ and thus $c = c_*$. This in turn implies a collision in hash function $H$, a contradiction. To avoid to check that $\gcd(e_*, u) = 1$ in the signature verification, we can set $u = 1$ so that this condition is automatically satisfied. We so obtain the following scheme.

*Modified scheme II.* The signature $\sigma = (y, e)$ on a message $m$ is defined as in § 3.1 except that $y$ is replaced with

$$y = (X\, g^{H(m\|e)})^{e^{-1} \bmod p'q'} \bmod N \ . \tag{10}$$

Compared to modified scheme I, this presents the further advantage of reducing the signature size.

Also, it is easy to check that the proof of Lemma 1 carries over with this modification. Moreover, as aforementioned, we can conclude the proof in the random oracle model when the attacker returns a signature forgery with $e_* = e_j$ for some $j \in \{1, \ldots, q_s\}$. To do so, we require that $\ell$ is sufficiently large so that $(i)$ the probability to get twice the same random value for random prime $e$ is sufficiently small, and $(ii)$ the probability that the attacker guesses the value of $e_j$ and submits $(m_j\|e_j)$ to the random oracle (before submitting $m_j$ to the signing oracle) is sufficiently small.[2] From Eq. (3), it then follows that $\delta = \gcd(2E'(c_* - c_j), e_*) = \gcd(c_* - c_j, e_*) = 1$ since $\gcd(2E', e_*) = 1$ (with overwhelming probability) and $c_* - c_j \not\equiv 0 \pmod{e_*}$ as otherwise this would mean $c_* = c_j \iff H(m_*\|e_*) = H(m_j\|e_*)$ with $m_* \neq m_j$, which contradicts the assumption that $H$ is collision-resistant. Since $\delta = 1$, a solution to the strong-RSA problem can be found by the standard technique: $w := (y_*/y_j)^{\alpha}\, z^{\beta} \bmod N$ and $e := e_*$, where $\alpha$ and $\beta$ are as per Eq. (3) with $u_j = u_* = 1$.

We note that modified scheme II is essentially the scheme of Cao and Liu [6] (see also Appendix A). It is even a bit more efficient as the input to hash function $H$ is shorter.

*Remark 1.* One may expect to get a better scheme by fixing parameter $u$ to 0. Although this invalidates the previous proof, this does not necessarily imply that the resulting scheme is insecure. It is however easily seen that this is indeed the case. A signature on a message $m$ then becomes $\sigma = (y, e)$ with $y = (g^{H(m\|e)})^{e^{-1} \bmod p'q'} \bmod N$. From signature $\sigma$, we can compute integers $\alpha$ and $\beta$ such that $\alpha\, e + \beta\, H(m\|e) = 1$ and hence evaluate $G := g^{e^{-1}} \bmod N$ as $G \equiv g^{e^{-1}} \equiv g^{e^{-1}(\alpha\, e + \beta\, H(m\|e))} \equiv g^{\alpha}\, y^{\beta} \pmod N$. Once $G$ is known, it is easy to forge the signature on any message $m_*$ as $\sigma_* = (y_*, e_*)$ with $y_* = G^{H(m_*\|e)} \bmod N$ and $e_* = e$.

As there are better known signature schemes in the *random oracle model* — for example, the FDH scheme the security of which is proven with respect to the (ordinary) RSA assumption [2, 22], we do not discuss any further possible improvements of modified scheme II.

The previous signature simulation for the basic scheme in the standard model does not work (see Eq. (4)) when $u$ is fixed to 1. The same statement holds for modified scheme II. The problem is that the value of $c_j = H(m_j)$ is unknown ahead of time. There are several ways to overcome this problem. For example, CS scheme can be seen as a variation of the basic scheme with $u = 1$ where

[2] These two conditions are readily fulfilled with the typical value $\ell = 160$.

hash function $H$ is replaced with a chameleon hash function [23] (see also [24]). Zhu and Camenisch/Lysyanskaya take another approach. Their schemes can be seen as variation of the basic scheme where $X^u \pmod N$ is replaced with $x\,X^u \pmod N$ for a fixed $x$ so as to prevent the construction of signature from previous signatures (as we did in § 4.1). So we consider the following signature scheme.

*Modified scheme III.* The public key requires an additional random quadratic residue $x \in \mathbb{Z}_N^*$, i.e., $\mathsf{pk} = \{N, x, X, g\}$. The signature $\sigma = (y, u, e)$ on a message $m$ is defined as in § 3.1 except that $y$ is replaced with

$$y = (x\,X^u\,g^{H(m)})^{e^{-1} \bmod p'q'} \bmod N \ . \tag{11}$$

A detailed description and a complete security are presented in Appendix B.

The annoying case in the security proof is the case of an attacker returning a signature forgery $\sigma_* = (y_*, u_*, e_*)$ with $u_* = u_j$ and $e_* = e_j$ from a previous signature $\sigma_j = (y_j, u_j, e_j)$. As highlighted in Appendix B, our slight modification brought to Zhu's signature scheme (compare Eq. (11) where $u \in \mathbb{Z}_e^*$ with Eq.(13) where $u \in \{0,1\}^\ell$) leads to be a better security proof than the one offered in [36]. In [15, § 2.4], Fischlin addresses the annoying case by replacing $(u, H(m))$ in Eq. (11) with $(u, H(m) \oplus u)$. The collision resistance of $H$ implies that if $m \neq m'$ then either $u \neq u'$ or $H(m) \oplus u \neq H(m') \oplus u'$. From Eqs (11) and (14) (in appendix), we conclude that modified scheme III and Fischlin's scheme represent to date the most efficient CS-like signature schemes.

## 5 Conclusion

This paper considered a CS-like signature scheme. From it, we rediscovered — and sometimes improved — several known strong-RSA signature schemes, including a slightly more efficient variant of Cao-Liu signature scheme and a revised version of Zhu's signature scheme with a better security proof. We also showed that a previous strong-RSA scheme is completely insecure.

## References


1. N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In W. Fumy, editor, *Advances in Cryptology – EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. Springer-Verlag, 1997.
2. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *1st ACM Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.
3. M. Bellare and P. Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In U. Maurer, editor, *Advances in Cryptology – EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer-Verlag, 1996.

4. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *Security in Communication Networks (SCN 2002)*, volume 2676 of *Lecture Notes in Computer Science*, pages 268–289. Springer-Verlag, 2002.
5. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *30th Annual ACM Symposium on Theory of Computing (STOC '98)*, pages 209–217, 1998.
6. Z. Cao and L. Liu. A strong RSA signature scheme and its applications. In *8th ACIS International Conference on Software Enginnering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pages 111–115. IEEE Computer Society, 2007.
7. D. Catalano and R. Gennaro. Cramer-Damgård signatures revisited: Efficient flat-tree signatures based on factoring. In S. Vaudenay, editor, *Public Key Cryptography – PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pages 313–327. Springer-Verlag, 2005.
8. B. Chevallier-Mames and M. Joye. A practical and tightly secure signature scheme without hash function. In M. Abe, editor, *Topics in Cryptology – CT-RSA 2007*, volume 4377 of *Lecture Notes in Computer Science*, pages 339–356. Springer-Verlag, 2007.
9. J.-S Coron and D. Naccache. Security analysis of the Gennaro-Halevi-Rabin signature scheme. In B. Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 91–101. Springer-Verlag, 2000.
10. R. Cramer and I. Damgård. New generation of secure and practical RSA-based signatures. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 173–185. Springer-Verlag, 1996.
11. R. Cramer and V. Shoup. Signature scheme based on the strong RSA assumption. *ACM Transactions on Information and System Security*, 3(3):161–185, 2000. An earlier version appears in *6th ACM Conference on Computer and Communications Security*, pp. 46–51, ACM Press, 1999.
12. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.
13. Y. Dodis, R. Oliveira, and K. Pietrzak. On the generic insecurity of the full domain hash. In V. Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 449–466. Springer-Verlag, 2005.
14. C. Dwork and M. Naor. An efficient existentially unforgeable signature scheme and its applications. In Y. Desmedt, editor, *Advances in Cryptology – CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 234–246. Springer-Verlag, 1994.
15. M. Fischlin. The Cramer-Shoup strong-RSA signature scheme revisited. In Y. Desmedt, editor, *Public Key Cryptography – PKC 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 116–129. Springer-Verlag, 2003.
16. E. Fujisaki and T. Okamoto. Statistical zero-knowledge protocols to prove modular polynomial equations. In Jr B. Kaliski, editor, *Advances in Cryptology – CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30. Springer-Verlag, 1997.
17. R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. In M. Bellare, editor, *Advances in Cryptology – EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 123–139. Springer-Verlag, 1999.

18. O. Goldreich. Two remarks concerning the Goldwasser-Micali-Rivest signature scheme. In A. Odlyzko, editor, *Advances in Cryptology – CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 104–110. Springer-Verlag, 1986.
19. S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen message attacks. *SIAM Journal of Computing*, 17(2):281–308, 1988.
20. D. Hofheinz and E. Kiltz. Programmable hash functions and their applications. In D. Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 21–38. Springer-Verlag, 2008.
21. M. Joye and H.-M. Lin. On the TYS signature scheme. In M. Gavrilova et al., editors, *Computational Science and Its Applications – ICCSA 2006*, volume 3982 of *Lecture Notes in Computer Science*, pages 338–344. Springer-Verlag, 2006.
22. J. Katz and N. Wang. Efficiency improvements for signature schemes with tight security reductions. In *10th ACM Conference on Computer and Communications Security*, pages 155–164. ACM Press, 2003.
23. H. Krawczyk and T. Rabin. Chameleon signatures. In *Symposium on Network and Distributed System Security – NDSS 2000*, pages 143–154. Internet Society, 2000.
24. K. Kurosawa and K. Schmidt-Samoa. New online/offline signature schemes without random oracles. In M. Yung et al., editors, *Public Key Cryptography – PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 330–346. Springer-Verlag, 2006.
25. A. Menezes and N. Smart. Security of signature schemes in a multi-user setting. *Designs, Codes and Cryptography*, 33(3):261–274, 2004.
26. D. Naccache, D. Pointcheval, and J. Stern. Twin signatures: An alternative to the hash-and-sign paradigm. In *8th ACM Conference on Computer and Communications Security*, pages 20–27. ACM Press, 2001.
27. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *21st Annual ACM Symposium on Theory of Computing (STOC '89)*, pages 33–43. ACM Press, 1989.
28. P. Paillier. Impossibility proofs for RSA signatures in the standard model. In M. Abe, editor, *Topics in Cryptology – CT-RSA 2007*, volume 4377 of *Lecture Notes in Computer Science*, pages 31–48. Springer-Verlag, 2007.
29. C. Popescu. A modification of the Cramer-Shoup digital signature scheme. *Studia Univ. Babeş-Bolyai Informatica*, XLVII(2):27–35, 2002.
30. R.L. Rivest, A. Shamir, and L.M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
31. C.H. Tan. A secure signature scheme. In S. Onoe et al., editors, *2006 International Conference on Wireless Communications and Mobile Computing (IWCMC 2006)*, pages 195–200. ACM Press, 2006.
32. C.H. Tan. A new signature scheme without random oracles. *International Journal of Security and Networks*, 1(3/4):237–242, 2006.
33. C.H. Tan, X. Yi, and C.K. Siew. A new provably secure signature scheme. *IEICE Trans. Fundamentals*, E86-A(10):2633–2635, 2003.
34. P. Yu and S.R. Tate. Online/offline signature schemes for devices with limited capabilities. In T. Malkin, editor, *Topics in Cryptology – CT-RSA 2008*, volume 4964 of *Lecture Notes in Computer Science*, pages 301–317. Springer-Verlag, 2008.
35. H. Zhu. New digital signature scheme attaining immunity against adaptive chosen message attack. *Chinese Journal of Electronics*, 10(4):484–486, 2001.
36. H. Zhu. A formal proof of Zhu's signature scheme. Cryptology ePrint Archive, Report 2003/155, 2003.

## A Cramer-Shoup Signature Scheme and Variants

In this appendix, we review the basic Cramer-Shoup signature scheme [11], as well as known variants thereof. We refer the reader to the original papers for further details.

The schemes require a strong RSA modulus $N = pq$ where $p = 2p' + 1$ and $q = 2q' + 1$ are two $\ell'$-bit primes with $p'$ and $q'$ prime. Let also $H$ denote a collision-resistant hash function, $H : \{0,1\}^* \rightarrow \{0,1\}^\ell$, mapping arbitrary-length bit-strings to $\ell$-bit strings. The message space is $\mathcal{M} = \{0,1\}^*$.

*Cramer-Shoup signature scheme.* The public key is $\mathsf{pk} = \{N, E, x, g\}$ where $E$ is an $(\ell+1)$-bit prime and $x$, $g$ are random quadratic residues in $\mathbb{Z}_N^*$. The private key is $\mathsf{sk} = \{p', q'\}$.

The signature on a message $m \in \mathcal{M}$ is given by $\sigma = (y, r, e)$ where $e \neq E$ is a random $(\ell+1)$-bit prime, $r$ is a random quadratic residue in $\mathbb{Z}_N^*$ and

$$y = (x\, g^{H(m')})^{e^{-1} \bmod p'q'} \bmod N \quad \text{with } m' = r^E\, g^{-H(m)} \bmod N \ . \tag{12}$$

Signature $\sigma = (y, r, e)$ on message $m \in \mathcal{M}$ is accepted if and only if $(i)$ $e$ is an odd $(\ell+1)$-bit integer different from $E$ and $(ii)$ $y^e\, g^{-H(\dot{m}')} \equiv x \pmod N$ with $\dot{m}' = r^E\, g^{-H(m)} \bmod N$.

*Zhu's and Camenisch-Lysyanskaya signature schemes.* Camenisch and Lysyanskaya introduced in [4] a variant of the above signature scheme. Independently, in a Chinese journal [35], Zhu proposed a similar scheme (see also [36]).

Zhu's signature scheme goes as follows. The public key is $\mathsf{pk} = \{N, x, X, g\}$ where $x$, $g$, $X$ are random quadratic residues in $\mathbb{Z}_N^*$. The private key is $\mathsf{sk} = \{p', q'\}$.

The signature on a message $m \in \mathcal{M}$ is given by $\sigma = (y, u, e)$ where $e$ is a random $(\ell+1)$-bit prime, $u$ is a random integer in $\{0,1\}^\ell$, and

$$y = (x\, X^u\, g^{H(m)})^{e^{-1} \bmod p'q'} \bmod N \ . \tag{13}$$

Signature $\sigma = (y, u, e)$ on message $m \in \mathcal{M}$ is accepted if and only if $(i)$ $e$ is an odd $(\ell+1)$-bit integer, $(ii)$ $u \in \{0,1\}^\ell$,[3] and $(iii)$ $y^e \equiv x\, X^u\, g^{H(m)} \pmod N$.

Camenisch-Lysyanskaya signature scheme is broadly the same; the differences are that $(i)$ $e$ is chosen as a random $(\ell+2)$-bit prime, and $(ii)$ $u$ is a random $(2\ell' + \ell + k)$-bit integer for some security parameter $k$. A Camenisch-Lysyanskaya signature is thus longer but presents the advantage that it can be evaluated into an on-line/off-line fashion [21]. Another on-line/off-line version of a slightly different scheme is described in [33]. However, the modification revealed itself to be fatal, as shown in [21]. Yet another on-line/off-line version is given in [34].

[3] Although not explicitly mentioned in [36], a close inspection of the security proof shows that this additional check is required.

*Fischlin's signature scheme.* Another variant is due to Fischlin [15]. The public key is $\mathsf{pk} = \{N, x, X, g\}$ as in the previous scheme. The private key is $\mathsf{sk} = \{p', q'\}$.

The signature on a message $m \in \mathcal{M}$ is given by $\sigma = (y, u, e)$ where $e$ is a random $(\ell+1)$-bit prime, $u$ is a random $\ell$-bit integer, and

$$y = (x\, X^u\, g^{H(m)\oplus u})^{e^{-1} \bmod p'q'} \bmod N \ . \tag{14}$$

Signature $\sigma = (y, u, e)$ on message $m \in \mathcal{M}$ is accepted if and only if $(i)$ $e$ is an odd $(\ell+1)$-bit integer, $(ii)$ $u$ is an $\ell$-bit integer, and $(iii)$ $y^e \equiv x\, X^u\, g^{H(m)\oplus u} \pmod N$.

*Tan's signature scheme.* A last variant of Cramer-Shoup signature scheme is due to Tan [32] (an earlier version appears in [31]). The public key is $\mathsf{pk} = \{N, X, g\}$ where $g$ and $X$ two random quadratic residues in $\mathbb{Z}_N^*$ (of order $p'q'$). The private key is $\mathsf{sk} = \{p', q'\}$.

The signature on a message $m$ is given by $(y, u, e)$ where $e$ is a random $(\ell+1)$-bit prime, $u$ is a random integer in $]3, \ldots, e[$ and

$$y = (X^u\, g^{H(\mathsf{pk}\|m\|e)})^{e^{-1} \bmod p'\,q'} \bmod N \ . \tag{15}$$

Signature $\sigma = (y, u, e)$ on message $m \in \mathcal{M}$ is accepted if and only if $(i)$ $e$ is an $(\ell+1)$-bit odd integer, $(ii)$ $u < e$, $(iii)$ $\gcd(u, e) = 1$, and $(iv)$ $y^e \equiv X^u\, g^{H(\mathsf{pk}\|m\|e)} \pmod N$.

Tan's signature scheme shares many similarities with an earlier scheme due to Popescu [29]. The main differences are: $(i)$ Popescu's signature has parameter $u = 1$, and $(ii)$ Popescu's signature includes a random integer $r$ as an additional input to hash function $H$. Roughly speaking, using the previous notations, a Popescu's signature on a message $m$ is given by $\sigma = (y, e, r)$ with

$$y = (x\, g^{H(m\|e\|r)})^{e^{-1} \bmod p'q'} \bmod N \ .$$

Another related scheme is the recent scheme of Cao and Liu [6]. With the previous notations, a Cao-Liu signature on a message $m$ is given by $\sigma = (y, e)$ with

$$y = (x\, g^{H(m\|e\|x)})^{e^{-1} \bmod p'q'} \bmod N \ .$$

## B Revised Zhu's Signature Scheme

We give below a detailed description of a revised version of Zhu's signature scheme [35, 36]. This is the scheme referred to as *modified scheme III* in § 4.2.

**Key Generation** Let $\ell$ and $\ell'$ with $\ell' > \ell + 1$ denote two security parameters. Let $H : \{0,1\} \to \{0,1\}^\ell$ be a collision-resistant hash function. Define $N = pq$ for two random strong $\ell'$-bit primes $p = 2p' + 1$ and $q = 2q' + 1$ where $p'$ and $q'$ are prime. Let $g$, $X$ and $x$ be three random quadratic residues in $\mathbb{Z}_N^*$. The public key is $\mathsf{pk} = \{N, x, X, g\}$ and the private key is $\mathsf{sk} = \{p', q'\}$.

**Signature Generation** To sign a message $m \in \{0,1\}^*$ (viewed as an arbitrary bit string), a random $(\ell+1)$-bit prime $e$ is chosen as well as a random integer $u \in \mathbb{Z}_e^*$. Next, using her private key $\mathsf{sk}$, the signer calculates

$$y = (x\, X^u\, g^{H(m)})^{e^{-1} \bmod p'q'} \bmod N \ .$$

The signature on message $m$ is $\sigma = (y, u, e)$.

**Signature Verification** To verify a putative signature $\sigma = (y, u, e)$ on a message $m$, it is checked that $(i)$ $e$ is an odd $(\ell+1)$-bit integer, $(ii)$ $0 < u < e$, and $(iii)$ $y^e \equiv x\, X^u\, g^{H(m)} \pmod N$. If all verifications succeed, the signature is accepted.

**Proposition 1.** *The revised Zhu's signature scheme is* EUF-CMA *secure under the strong-RSA assumption.*

*Proof.* Let $(z, N)$ where $N$ is a safe RSA modulus denote the strong-RSA challenge. The goal is to find a pair $(w, e) \in \mathbb{Z}_N^* \times \mathbb{Z}_{>1}$ such that $z \equiv w^e \pmod N$.

We assume that there exists a polynomial-time EUF-CMA attacker against the scheme. The attacker can submit $q_s$ signature queries on chosen messages $m_i$ to a signing oracle, returning $\sigma_i = (y_i, u_i, e_i)$, for $1 \leq i \leq q_s$. Next, the attacker outputs with non-negligible probability a signature forgery $\sigma_* = (y_*, u_*, e_*)$. We will then use the attacker to solve the strong-RSA problem, thereby proving the security of the scheme.

We distinguish three types of forgers.

We choose at random $q_s$ $(\ell+1)$-bit primes $e_1, \ldots, e_{q_s}$ and an index $j$ in $\{1, \ldots, q_s\}$. For simplicity, we assume that all $e_i$'s are different. We define $E' = \prod_{\substack{1 \leq i \leq q_s \\ i \neq j}} e_i$ and $E = \prod_{1 \leq i \leq q_s} e_i$. We pick two random elements $f$ and $h$ in $\mathbb{Z}_N^*$.

**Type I:** $e_* = e_j$ and $u_* = u_j$ for some $j \in \{1, \ldots, q_s\}$

**Setup** We choose at random $v \in \mathbb{Z}_{e_j}^*$ and define

$$g = z^{2E'} \bmod N \ , \quad X = f^{2E}\, g \bmod N \ , \quad x = h^{2E}\, g^{-v} \bmod N \ .$$

We let $\mathsf{pk} = \{N, x, X, g\}$.

**Signature queries** On input message $m_i$,

- if $i \neq j$, we return $\sigma_i = (y_i, u_i, e_i)$ with

$$y_i = \left((h\, f^{u_i})^{e_j}\, z^{-v+H(m_i)+u_i}\right)^{2E'/e_i} \bmod N$$

for some random $u_i \in \mathbb{Z}_{e_i}^*$;

- if $i = j$, we compute $u_j = v - H(m_j) \bmod e_j$ and $t := (u_j - v + H(m_j))/e_j \in \mathbb{Z}$, and return $\sigma_j = (y_j, u_j, e_j)$ with

$$y_j = (h\, f^{u_j})^{2E'}\, g^t \bmod N \ .$$

**Outcome** We get

$$\left(\frac{y_*}{y_j}\right)^{e_*} \equiv g^{H(m_*)-H(m_j)} \equiv z^{2E'(H(m_*)-H(m_j))} \pmod{N} \ .$$

Using extended Euclid's algorithm, we find integers $\alpha$ and $\beta$ satisfying $\gcd\big(2E'(H(m_*) - H(m_j)), e_*\big) = \alpha\, 2E'(H(m_*) - H(m_j)) + \beta\, e_* = \gcd(H(m_*) - H(m_j), e_*) = 1$ since $H$ is collision-resistant. Hence, $w := (y_*/y_j)^{\alpha}\, z^{\beta} \bmod N$ and $e := e_*$ is a solution to the strong-RSA challenge.

**Type II:** $e_* = e_j$ and $u_* \neq u_j$ for some $j \in \{1, \ldots, q_s\}$

**Setup** We choose at random $u_j \in \mathbb{Z}^*_{e_j}$ and define

$$X = z^{2E'} \bmod N \ , \quad g = f^{2E} \bmod N \ , \quad x = h^{2E}\, X^{-u_j} \bmod N \ .$$

We let $\mathsf{pk} = \{N, x, X, g\}$.

**Signature queries** On input message $m_i$, we return $\sigma_i = (y_i, u_i, e_i)$ with

$$y_i = \begin{cases} \big((h\, f^{H(m_i)})^{e_j}\, z^{u_i - u_j}\big)^{2E'/e_i} \text{ for some } u_i \in_R \mathbb{Z}^*_{e_i} & \text{if } i \neq j \ , \\ (h\, f^{H(m_j)})^{2E'} \bmod N & \text{if } i = j \ . \end{cases}$$

**Outcome** We get

$$\left(\frac{y_*}{y_j}\, f^{2E'(H(m_j)-H(m_*))}\right)^{e_*} \equiv X^{u_*-u_j} \equiv z^{2E'(u_*-u_j)} \pmod{N} \ .$$

From extended Euclid's algorithm, we obtain $\gcd(2E'(u_* - u_j), e_*) = \alpha\, 2E'(u_* - u_j) + \beta\, e_* = \gcd(u_* - u_j, e_*) = 1$ since $u_* \not\equiv u_j \pmod{e_*}$. Hence, $w := \big[(y_*/y_j)\, f^{2E'(H(m_j)-H(m_*))}\big]^{\alpha}\, z^{\beta} \bmod N$ and $e := e_*$ is a solution to the strong-RSA challenge.

**Type III:** $e_* \neq e_j$ for all $j \in \{1, \ldots, q_s\}$

**Setup** We choose a random $r_1, r_2 \in \{1, \ldots, N^2\}$ and define

$$g = z^{2E} \bmod N \ , \quad X = g^{r_1} \bmod N \quad x = g^{r_2} \bmod N \ .$$

We let $\mathsf{pk} = \{N, x, X, g\}$.

**Signature queries** On input message $m_i$, we return $\sigma_i = (y_i, u_i, e_i)$ with

$$y_i = g^{r_1\, u_i + r_2 + H(m_i)} \bmod N$$

for some random $u_i \in \mathbb{Z}^*_{e_i}$.

**Outcome** We get

$$y_*{}^{e_*} \equiv x\, X^{u_*}\, g^{H(m_*)} \equiv z^{2E(r_1\, u_* + r_2 + H(m_*))} \pmod{N} \ .$$

As in the proof of Lemma 1, with overwhelming probability, we have $\delta := \gcd(2E(r_1\, u_* + r_2 + H(m_*)), e_*) = \alpha\, 2E(r_1\, u_* + r_2 + H(m_*)) + \beta\, e_* = \gcd(r_1\, u_* + r_2 + H(m_*), e_*) < e_*$. Hence, $w := y_*{}^{\alpha}\, z^{\beta} \bmod N$ and $e := e_*/\delta$ is a solution to the strong-RSA challenge.

In all cases, the forger succeeds in breaking the strong-RSA assumption with non-negligible probability, which proves the security of the scheme. □