# A Practical and Tightly Secure Signature Scheme Without Hash Function*

[Published in M. Abe Ed., *Topics in Cryptology – CT-RSA 2007*, vol. 4377 of *Lecture Notes in Computer Science*, pp. 339–356, Springer-Verlag, 2007.]

Benoît Chevallier-Mames[1,**] and Marc Joye[2]

[1] Gemalto, Security Labs
Avenue du Jujubier, 13705 La Ciotat Cedex, France
benoit.chevallier-mames@gemalto.com
[2] Thomson R&D France
Technology Group, Corporate Research, Security Laboratory
1 avenue de Belle Fontaine, 35576 Cesson-Sévigné, France
marc.joye@thomson.net

**Abstract.** In 1999, two signature schemes based on the flexible RSA problem (a.k.a. strong RSA problem) were independently introduced: the Gennaro-Halevi-Rabin (GHR) signature scheme and the Cramer-Shoup (CS) signature scheme. Remarkably, these schemes meet the highest security notion in the *standard model*. They however differ in their implementation. The CS scheme and its subsequent variants and extensions proposed so far feature a loose security reduction, which, in turn, implies larger security parameters. The security of the GHR scheme and of its twinning-based variant are shown to be tightly based on the flexible RSA problem but additionally (i) either assumes the existence of *division-intractable* hash functions, or (ii) requires an *injective* mapping into the prime numbers in both the signing *and* verification algorithms. In this paper, we revisit the GHR signature scheme and completely remove the extra assumption made on the hash functions without relying on injective prime mappings. As a result, we obtain a *practical* signature scheme (and an on-line/off-line variant thereof) whose security is *solely* and *tightly* related to the strong RSA assumption.

*Keywords:* Digital signatures, standard model, strong RSA assumption, tight reduction, Gennaro-Halevi-Rabin signature scheme, Cramer-Shoup signature scheme, on-line/off-line signatures.

## 1 Introduction

Digital signatures are one of the most useful and fundamental primitives resulting from the invention of public-key cryptography by Diffie and Hellman [DH76]

* The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT.

** Also with École Normale Supérieure, Département d'Informatique, 45 rue d'Ulm, 75230 Paris 05, France.

in 1976. Rivest, Shamir and Adleman [RSA78] gave the first practical implementation of such a primitive. However, at that time, the security analysis of signature schemes was studied more heuristically: a scheme was declared "secure" if no attacks were found.

Provably signature schemes. Formal security notions for signature schemes were later introduced by Goldwasser, Micali and Rivest in their seminal paper [GMR88]. They also proposed a signature scheme provably meeting their security notion (see also [Gol86,NY89]). This tree-based signature scheme was subsequently improved by Dwork and Naor [DN94], Cramer and Damgård [CD96], and Catalano and Gennaro [CG05].

Random oracle model. More efficient schemes were proven secure in the so-called random oracle model [BR93,FS87]. The random oracle model assumes that the output of a hash function behaves like a random generator. Provably secure signature schemes relying on this extra assumption are presented and discussed in [BR96,PS96,GJ03,KW03,BLS04], with different underlying problems: the discrete logarithm problem [PS96], the RSA problem [BR96,KW03], the CDH problem [GJ03,KW03], the DDH problem [KW03], or the CDH problem on certain elliptic curves [BLS04]. The idealized random oracle model has however certain limitations [CGH98,PV05].

Standard model. Efficient signature schemes without random oracles are due to Gennaro, Halevi and Rabin [GHR99] and to Cramer and Shoup [CS00]. They are both based on the strong RSA assumption, which assumes that it is impossible to find an $e$-th modular root of a given element, even if $e$ can be chosen arbitrarily by the attacker (provided of course that $e \geq 2$). Subsequent improvements and modifications include the works of [NPS01,Zhu01,CL02,Fis03], with some better performances and signature size, or additional features for particular use cases.

More recently, the introduction of cryptographic bilinear mappings has allowed the emergence of new techniques to achieve security without random oracles. More precisely, the study of pairings gave rise to signature scheme based on the strong Diffie-Hellman assumption [BB04] and even more recently on the computational Diffie-Hellman assumption [Wat05,BSW06].

**Our contribution.** This paper presents a new signature scheme based on the strong RSA assumption, in the standard model. In contrast to the Cramer-Shoup scheme and its variants, our security proof yields a tight reduction. Moreover, our scheme does not rely on special-type hash functions nor injective prime functions. In this sense, it is easier to implement than the Gennaro-Halevi-Rabin scheme and its known variants, as one needs not to design such functions. Finally, our scheme features an efficient on-line/off-line variant.

**Organization.** The rest of this paper is organized as follows. In Section 2, we introduce some background on signature schemes, provable security and RSA-related problems. In Section 3, we briefly review the Gennaro-Halevi-Rabin and the Cramer-Shoup signature schemes. Section 4 is the main part of our paper. We introduce our new signature scheme (that we call TSS) and prove its security in the standard model. We also compare our scheme with prior RSA-based schemes in the standard model, and present an on-line/off-line variant. Finally, we conclude in Section 5.

## 2 Preliminaries

In this section, we introduce notations and definitions that are used throughout the paper. For convenience, we often identify an integer with its binary representation: $a \in \{0,1\}^\ell$ is also viewed as an integer in the range $[0, 2^\ell - 1]$. We say that $a$ is an $\ell$-bit integer if $a$ is an integer in the range $[2^{\ell-1}, 2^\ell - 1]$. An (odd) prime $p$ is a strong prime if $(p-1)/2$ is prime. An RSA modulus $n = pq$ is safe if it is the product of two equal-size strong primes.

### 2.1 Signature schemes

A signature scheme $\mathsf{Sig} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ is defined by the three following algorithms:

1. *Key generation algorithm* Gen. On input security parameter $k$, algorithm Gen produces a pair $(\mathsf{pk}, \mathsf{sk})$ of matching public and private keys.
2. *Signing algorithm* Sign. Given a message $m$ in a set $\mathcal{M}$ of messages and a pair of matching public and private keys $(\mathsf{pk}, \mathsf{sk})$, Sign produces a signature $\sigma$. The signing algorithm can be probabilistic.
3. *Verification algorithm* Verify. Given a signature $\sigma$, a message $m \in \mathcal{M}$ and a public key pk, Verify checks whether $\sigma$ is a valid signature on $m$ with respect to pk.

Several security notions have been defined for signature schemes, mostly based on the work by Goldwasser, Micali and Rivest [GMR88]. It is now customary to ask for the infeasibility of existential forgeries, even against adaptive chosen-message adversaries:

– An *existential forgery* is a signature on a new message, valid and generated by the adversary. The corresponding security goal is called *existential unforgeability* (EUF).
– A *weak existential forgery* is a new message/signature pair, valid and generated by the adversary. The corresponding security goal is called *strong existential unforgeability* (sEUF).
– 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 the *adaptive chosen-message attacks* (CMA), where the adversary can ask the signer to sign any message of her choice, in an adaptive way.

As a consequence, we say that a signature scheme is *secure* if it prevents (weak) existential forgeries against chosen-message attacks (EUF-CMA or sEUF-CMA) with overwhelming probability.

– A signature scheme Sig is $(\tau, q_s, \varepsilon)$-*secure* if the success probability

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

  for any adversary $\mathcal{A}$ with running time bounded by $\tau$, 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.
– A signature scheme Sig is *strongly* $(\tau, q_s, \varepsilon)$-*secure* if the success probability

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

  for any adversary $\mathcal{A}$ with running time bounded by $\tau$, making (at most) $q_s$ queries to a signing oracle $\mathsf{Sign}(\mathsf{sk}; \cdot)$, and returning a valid message/signature pair $(m_*, \sigma_*)$ where $m_*$ was not submitted to the signing oracle and/or $\sigma_*$ was not returned by the signing oracle.

### 2.2 Provable security

Reduction. Basically, the idea behind provable security (or reductionist security [KM04]) is to prove that a scheme is secure by exhibiting a so-called reduction that uses a chosen-message attacker against the signature scheme, in order to solve a hard cryptographic problem. In the *standard model*, the attacker has access to a signing oracle that is simulated by the reduction, answering the $q_s$ signature queries on chosen messages, and receiving eventually a signature forgery.

Tightness of reduction. Two classes of provably secure signature schemes can be distinguished. The first class proposes reductions that are said *loose*, as they can turn an attacker into an algorithm solving a cryptographic problem, but whose ratio

$$\rho = \frac{\text{running time}}{\text{success probability}}$$

is far greater than those required for the attacker to produce a forgery. Hence, this kind of reduction only proves the security asymptotically. The second class of provable signature schemes features so-called *tight* reductions, using the attacker to solve the cryptographic problem with roughly the same ratio $\rho$.

Signature schemes in the standard model. Of course, secure schemes with a reduction in the standard model are the preferred ones, even if proofs in the random oracle model are arguments for a good design. Indeed, important differences between the idealized random oracle model and the real life have been pointed out in the literature [CGH98,PV05].

There are just a handful of practical signature schemes with a security reduction in the standard model, and most of them rely on *flexible* problems, *one-more* problems or *q-type* problems. A notable exception is the recent signature scheme of Waters [Wat05] (see also [BSW06]), the security of which relies on the computational Diffie-Hellman [DH76] assumption, in bilinear groups.

Flexible problems are cryptographic problems admitting several solutions, the solver has just to find one of them. Its main representative is certainly the flexible RSA problem [BP97,FO97].

**Definition 1 (Flexible RSA problem – SRSA).** *Being given a safe RSA modulus $n$ and an element $y \in \mathbb{Z}_n^*$, the* flexible RSA problem *is defined as finding an element $x \in \mathbb{Z}_n^*$ and an integer $e > 1$ such that $x^e \equiv y \pmod{n}$.*

*The* strong RSA assumption *conjectures that there is no attacker that can $(\tau, \varepsilon)$-solve the flexible* RSA *problem (i.e., with success probability smaller than $\varepsilon$ and running time bounded by $\tau$) with $\tau$ polynomial and $\varepsilon$ non-negligible.*

One-more problems are problems where the solver has $n$ accesses to an oracle that solves a hard problem, in order to solve $n+1$ instances of this hard problem. Typical examples include the one-more RSA, the one-more DL and the one-more CDH [BNPS03].

Finally, $q$-type problems are problems where the solver receives a large instance of $q$ data, and must find a value satisfying a certain relation with this instance. Notably, the returned value could be a (new) data of the same kind. A typical example of this last type of problem is the strong Diffie-Hellman problem [BB04].

Devising an efficient signature scheme based on an harder type of problem such as the RSA problem [RSA78], the discrete logarithm problem or the computational Diffie-Hellman problem — *in a group without pairing* — in the standard model still remains an open question.

## 3 Signature Schemes Based on the Strong-RSA Assumption

We briefly review several *efficient* signature schemes the security of which relies on the strong RSA assumption, in the standard model. We refer the reader to the original papers for a complete description.

We begin with the two main schemes introduced in 1999: the Gennaro-Halevi-Rabin (GHR) signature scheme [GHR99] and the Cramer-Shoup (CS) signature scheme [CS00]. The CS scheme was subsequently modified in several directions, including the variants by Zhu [Zhu01,Zhu03], Camenisch and Lysyanskaya [CL02] and Fischlin [Fis03]. As the Twin-GHR scheme [NPS01], our new scheme (see Section 4) as for it can be viewed as a variant of the GHR scheme.

In order to allow an easier comparison between the different schemes, we deviate from the original papers and use similar notation when describing the signature schemes. For an RSA modulus $n$, we let $\mathrm{QR}_n \subset \mathbb{Z}_n^*$ denote the set of quadratic residues modulo $n$. The message space is denoted by $\mathcal{M}$; $\mathcal{H}$ denotes a hash function.

### 3.1 Gennaro-Halevi-Rabin signature scheme

The message space is $\mathcal{M} = \{0,1\}^*$ and $\mathcal{H} : \mathcal{M} \to \{0,1\}^{\ell_h}$ is division-intractable.

**Key generation:** The public key is $\mathsf{pk} = \{n, u\}$ where $n = (2p'+1)(2q'+1)$ is a safe RSA modulus and $u$ is a random element in $\mathbb{Z}_n^*$. The private key is $\mathsf{sk} = \{p', q'\}$.

**Signing:** The signature on a message $m \in \mathcal{M}$ is given by $\sigma = u^{c^{-1} \bmod 2p'q'} \bmod n$ where $c = \mathcal{H}(m)$.

**Verification:** Signature $\sigma$ on message $m \in \mathcal{M}$ is accepted iff $\sigma^{\mathcal{H}(m)} \equiv u \pmod{n}$.

In the original description, hash function $\mathcal{H}$ has to be a *division-intractable* hash function (see [GHR99] for a precise definition). Following [CN00], the easiest way to achieve this additional property is to define $\mathcal{H}$ as a hash function that maps bitstrings to prime numbers.[3]

Tight variant. The $\mathsf{GHR}$ signature scheme as above has a *loose* security reduction to the flexible-$\mathsf{RSA}$ problem [GHR99,Cor00]. However, in [GHR99], the authors also propose techniques to achieve tightness in their signature scheme.

Basically, their idea is to make use of a chameleon hash function [KR00]. Let $P$ be an $\ell_p$-bit prime, let $Q$ be an $\ell_q$-bit prime divisor of $P-1$, and let $\langle g \rangle$ denote the cyclic subgroup generated by an element $g \in \mathbb{Z}_P^*$ of order $Q$. They so obtain a scheme such that an attacker against it can be used to solve either the discrete logarithm problem in subgroup $\langle g \rangle$ or the flexible $\mathsf{RSA}$ problem modulo $n$, with roughly the same success probability and running time.

The message space is $\mathcal{M} = \mathbb{Z}_Q$ and $\mathcal{H} : \langle g \rangle \to \{0,1\}^{\ell_h}$. The scheme then becomes:

**Key generation:** The public key is $\mathsf{pk} = \{n, u, g, y, P\}$ where $n = (2p'+1)(2q'+1)$ is a safe RSA modulus, $u$ is a random element in $\mathbb{Z}_n^*$ and $y$ is a random element in $\langle g \rangle \subseteq \mathbb{Z}_P^*$. The private key is $\mathsf{sk} = \{p', q'\}$.

**Signing:** The signature on a message $m \in \mathcal{M}$ is given by $\sigma = (r, s)$ where $r$ is a random element in $\mathbb{Z}_Q$ and $s = u^{c^{-1} \bmod 2p'q'} \bmod n$ with $c = \mathcal{H}(g^m y^r \bmod P)$.

**Verification:** Signature $\sigma = (r, s)$ on message $m \in \mathcal{M}$ is accepted iff $s^{c'} \equiv u \pmod{n}$ with $c' = \mathcal{H}(g^m\, y^r \bmod P)$.

Clearly, the chameleon hash function could be of different nature: for a joint use with $\mathsf{GHR}$ scheme, most interesting cases are certainly chameleon hash functions based on $\mathsf{RSA}$ or factorization problems (*e.g.*, [KS06]).

[3] It would also be possible to remove the hash function, and to sign instead only prime messages. However, this solution suffers from practicality.

TWINNING-BASED VARIANT. The twinning paradigm was introduced by Naccache, Pointcheval and Stern in [NPS01]. It particulary fits to GHR signatures.

Let $\mathcal{P}$ be an injective function that maps the set $\{0,1\}^{2\ell_m}$ into the prime numbers. Consider for example the function

$$\mathcal{P} : \{0,1\}^{2\ell_m} \to \{\text{primes}\},\ \mu \mapsto \texttt{NextPrime}(\mu\, 2^{\tau})$$

with $\tau \approx 5 \log_2(2\ell_m)$, which can be evaluated with less than $40\,\ell_m$ primality tests [NPS01, Appendix B]. The Twin-GHR scheme is then defined as follows:

**Key generation:** The public key is $\mathsf{pk} = \{n, N, u_1, u_2\}$ where $n = (2p'+1)(2q'+1)$ and $N = (2P'+1)(2Q'+1)$ are two safe RSA moduli, $u_1$ is a random element in $\mathbb{Z}_n^*$ and $u_2$ is a random element in $\mathbb{Z}_N^*$. The private key is $\mathsf{sk} = \{p', q', P', Q'\}$.

**Signing:** The signature on a message $m \in \mathcal{M} = \{0,1\}^{\ell_m}$ is given by $\sigma = (\mu_1, s_1, s_2)$, computed, for a random $\mu_1 \in \{0,1\}^{2\ell_m}$ and $\mu_2 = (m\|m) \oplus \mu_1$, as $s_1 = {u_1}^{\mathcal{P}(\mu_1)^{-1} \bmod 2p'q'} \bmod n$ and $s_2 = {u_2}^{\mathcal{P}(\mu_2)^{-1} \bmod 2P'Q'} \bmod N$.

**Verification:** Signature $\sigma = (\mu_1, s_1, s_2)$ on message $m \in \mathcal{M}$ is accepted iff $s_1^{\mathcal{P}(\mu_1)} = u_1 \bmod n$ and $s_2^{\mathcal{P}((m\|m)\oplus\mu_1)} = u_2 \bmod N$.

The Twin-GHR signature scheme has a *tight* security reduction to the flexible-RSA problem [NPS01].

### 3.2 Cramer-Shoup signature scheme

The message space is $\mathcal{M} = \{0,1\}^*$ and $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{\ell_h}$.

**Key generation:** The public key is $\mathsf{pk} = \{n, e, x, h\}$ where $n = (2p'+1)(2q'+1)$ is a safe RSA modulus, $e$ is an $(\ell_h+1)$-bit prime and $x, h$ are random elements in $\mathrm{QR}_n$. The private key is $\mathsf{sk} = \{p', q'\}$.

**Signing:** The signature on a message $m \in \mathcal{M}$ is given by $\sigma = (c, u, v)$ where $c$ is a random $(\ell_h + 1)$-bit prime, $u$ is a random element in $\mathrm{QR}_n$ and $v = (x\, h^{\mathcal{H}(w)})^{c^{-1} \bmod p'q'} \bmod n$ with $w = u^e h^{-\mathcal{H}(m)} \bmod n$.

**Verification:** Signature $\sigma = (c, u, v)$ on message $m \in \mathcal{M}$ is accepted iff (1) $c$ is an odd $(\ell_h + 1)$-bit integer and (2) $v^c\, h^{-\mathcal{H}(w')} \equiv x \pmod{n}$ with $w' = u^e\, h^{-\mathcal{H}(m)} \bmod n$.

This signature scheme, as shown in [CS00], is secure under the strong RSA assumption. We refer the reader to the original paper for details, and just mention that unfortunately, the reduction is loose, with a loss factor equal to $\frac{1}{q_s}$, where $q_s$ is the number of signature queries the attacker is allowed to make.[4] On

[4] To be complete, the reduction of CS scheme is made of two types of reductions, one of which tightly reduces the security to the flexible RSA problem, while the other one reduces the security to the (plain) RSA problem — but with a $\frac{1}{q_s}$ factor. Even if flexible RSA should be easier than RSA, there is no estimation of the difference of difficulty between these problems. Note also that a loose reduction implies the use of larger RSA moduli.

the other hand, one of the advantages of the $\mathsf{CS}$ scheme compared to the $\mathsf{GHR}$ scheme is that the hash function $\mathcal{H}$ needs not to map to prime numbers nor to be division-intractable, but merely to be collision-resistant.

Camenisch-Lysyanskaya signature scheme. Camenisch and Lysyanskaya introduce in [CL02] a variant of $\mathsf{CS}$ signature scheme (that we abbreviate in $\mathsf{CL}$). Independently, in a Chinese journal [Zhu01], Zhu propose a similar scheme (see also [Zhu03]).

**Key generation:** The public key is $\mathsf{pk} = \{n, x, g, h\}$ where $n = (2p'+1)(2q'+1)$ is a safe RSA modulus, prime and $x, g, h$ are random elements in $\mathrm{QR}_n$. The private key is $\mathsf{sk} = \{p', q'\}$.

**Signing:** The signature on a message $m \in \mathcal{M} = \{0,1\}^{\ell_m}$ is given by $\sigma = (c, t, v)$ where $c$ is a random $\ell_c$-bit prime with $\ell_c \geq (\ell_m + 2)$, $t$ is a random $(\ell_n + \ell_m + \ell)$-bit integer and $v = (x\, g^t\, h^m)^{c^{-1} \bmod p'q'} \bmod n$.

**Verification:** Signature $\sigma = (c, t, v)$ on message $m \in \mathcal{M}$ is accepted iff (1) $c$ is an odd $\ell_c$-bit integer and (2) $v^c\, g^{-t}\, h^{-m} \equiv x \pmod{n}$.

Fischlin signature scheme. A last variant is due to Fischlin in [Fis03]. The message space is $\mathcal{M} = \{0,1\}^*$ and $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{\ell_h}$.

**Key generation:** The public key is $\mathsf{pk} = \{n, x, g, h\}$ where $n = (2p'+1)(2q'+1)$ is a safe RSA modulus, prime and $x, g, h$ are random elements in $\mathrm{QR}_n$. The private key is $\mathsf{sk} = \{p', q'\}$.

**Signing:** The signature on a message $m \in \mathcal{M}$ is given by $\sigma = (c, t, v)$ where $c$ is a random $(\ell_h + 1)$-bit prime, $t$ is a random $\ell_h$-bit integer and $v = (x\, g^t\, h^{t \oplus \mathcal{H}(m)})^{c^{-1} \bmod p'q'} \bmod n$.

**Verification:** Signature $\sigma = (c, t, v)$ on message $m \in \mathcal{M}$ is accepted iff (1) $c$ is an odd $(\ell_h + 1)$-bit integer and (2) $v^c\, g^{-t} h^{-(t \oplus \mathcal{H}(m))} \equiv x \pmod{n}$.

A comparison of all the previous schemes is presented in Table 1, Section 4.3.

## 4 The TSS Signature Scheme

This section is the core of our paper. We introduce the $\mathsf{TSS}$ (for *Tightly Secure Signature*) scheme and prove its security in the standard model. Our reduction is tightly related to the flexible $\mathsf{RSA}$ problem.

The $\mathsf{GHR}$ scheme requires the use of a hash function that maps to prime numbers (or at least, that is division-intractable). The $\mathsf{CS}$ scheme and its variants, on the other hand, feature a loose security reduction: there is a loss factor of $1/q_s$. Our goal, when designing $\mathsf{TSS}$, is to combine the practicality of the $\mathsf{CS}$ scheme with the tightness of the $\mathsf{GHR}$ scheme.

### 4.1 Description

Intuitively, there are two ideas behind our scheme. First, as in the $\mathsf{CS}$-like schemes (cf. Section 3), we want to use a fresh, prime exponent $c$ in each signature generation and then to give a $c^{\text{th}}$ root modulo a safe RSA modulus $n$ as part of the signature. Further, we want prime $c$ to be free of any particular relation (except its size), in order to allow the use of fast prime generation algorithms. The second idea is, as in the $\mathsf{GHR}$ scheme, to use a chameleon function in order to tighten the security reduction. In order to base the security on the strong $\mathsf{RSA}$ assumption, we define a second safe RSA modulus $N$ and make use of an $\mathsf{RSA}$-type chameleon function.

The message space is $\mathcal{M} = \{0,1\}^{\ell_m}$. Let also $\ell_n$ be a security parameter. A detailed description of the $\mathsf{TSS}$ scheme is given below.

<u>Gen:</u> On input $\ell_n$ and $\ell_m$:
- choose an (odd) $(\ell_m + 1)$-bit prime $E$;
- generate two random $\ell_n$-bit safe RSA moduli $n = (2p' + 1)(2q' + 1)$ and $N = (2P' + 1)(2Q' + 1)$ such that $\gcd(P'Q', E) = 1$;
- compute $D = E^{-1} \bmod 2P'Q'$;
- choose at random two elements $u \in \mathbb{Z}_n^*$ and $g \in \mathbb{Z}_N^*$.

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

<u>Sign:</u> Let $m \in \{0,1\}^{\ell_m}$ be the message to be signed:
- choose a random prime $c$ in $[(N + 1)/2, N[$;
- compute $s = u^{c^{-1} \bmod 2p'q'} \bmod n$ and $r = (c\, g^{-(m+1)})^D \bmod N$.

The signature on $m$ is $\sigma = (r, s) \in \mathbb{Z}_N^* \times \mathbb{Z}_n^*$.

<u>Verify:</u> Let $\sigma = (r, s)$ be a putative signature on message $m \in \{0,1\}^{\ell_m}$. Then:
(i) check that $(r, s) \in [0, N[ \times [0, n[$;
(ii) check that $s^c \equiv u \pmod{n}$ where $c = g^{m+1}\, r^E \bmod N$.

If the two conditions hold then signature $\sigma$ is accepted.

*Remark 1.* If Condition (i) is removed in the verification phase, *i.e.*, $(r, s) \overset{?}{\in} [0, N[ \times [0, n[$, the corresponding security level becomes $\mathsf{EUF\text{-}CMA}$. It is easy to see that the scheme is no longer $\mathsf{sEUF\text{-}CMA}$ because if $(r, s)$ is a valid signature on a message $m$ then so is $(r + r_0\, N, s + s_0\, n)$ on the same message $m$, for arbitrary integers $r_0$ and $s_0$.

*Remark 2.* As in the $\mathsf{CS}$ scheme, there is no need to check the primality of $c$ in the verification algorithm. In the $\mathsf{TSS}$ scheme, the verification step is even simpler, as one needs to verify neither the parity nor the bitsize of $c$.

Typically, we set $\ell_m = 160$ and $\ell_n = 1024$, for using $\mathsf{TSS}$ with short messages, but there is actually no limitation on $\ell_m$: if signing long messages is required, one could set $\ell_m$ to a large value,[5] and achieve security under the sole strong-$\mathsf{RSA}$

[5] Actually this is basically as in the Camenisch-Lysyanskaya signature scheme [CL02].

assumption. Alternatively, it is also possible to use $\mathsf{TSS}$ preceded by a hashing step of the message, at the price of assuming in addition the collision resistance of the underlying hash function.

The previous scheme is subject to numerous variants. One can for example (slightly) speed up the signature algorithm by pre-computing $g^{-1} \bmod N$ or $g^{-D} \bmod N$ and then by evaluating $r$ as $r = (c\,(g^{-1})^{m+1})^D \bmod N$ or as $r = c^D\,(g^{-D})^{m+1} \bmod N$.

### 4.2 Security analysis

We show that the security of our scheme is *tightly* related the strong $\mathsf{RSA}$ assumption. That is, given an $\ell_n$-bit safe RSA modulus $\hat{n}$ and a random element $\hat{y} \in \mathbb{Z}_{\hat{n}}^*$, we want to find a pair $(\hat{x}, \hat{e}) \in \mathbb{Z}_{\hat{n}}^* \times \mathbb{Z}_{>1}$ satisfying $\hat{y} \equiv \hat{x}^{\hat{e}} \pmod{\hat{n}}$. More formally, we prove the following theorem.

**Theorem 1.** *Suppose that the flexible* $\mathsf{RSA}$ *problem is* $(\tau, \varepsilon)$*-hard. Then, for any* $q_s$*, the* $\mathsf{TSS}$ *signature scheme is strongly* $(\tau_{\mathcal{A}}, q_s, \varepsilon_{\mathcal{A}})$*-secure, where*

$$\varepsilon \geq \frac{\varepsilon_{\mathcal{A}}}{2} \quad \textit{and} \quad \tau \lesssim \tau_{\mathcal{A}} + \mathrm{O}\big({\ell_n}^5 + q_s\,{\ell_n}^3 \max(\log q_s, \ell_n)\big) \enspace .$$

*Proof.* As usual, the proof is by contradiction. We assume that there exists a polynomial-time adversary $\mathcal{A}$ that is able to produce a weak existential forgery with non-negligible success probability $\varepsilon_{\mathcal{A}}$ within time $\tau_{\mathcal{A}}$ after $q_s$ queries to a signing oracle. We then use $\mathcal{A}$ to $(\tau, \varepsilon)$-solve the flexible $\mathsf{RSA}$ problem, *i.e.*, to find a pair $(\hat{x}, \hat{e})$ on input challenge $(\hat{n}, \hat{y})$.

- We toss a coin $b \in \{0, 1\}$ and run Simulation $b$ defined as follows.

  **Simulation 0**

  - We let $n = \hat{n}$. We choose an (odd) $(\ell_m + 1)$-bit prime $E$. Next, we generate a random $\ell_n$-bit safe RSA modulus $N = (2P' + 1)(2Q' + 1)$ such that $\gcd(P'Q', E) = 1$. We compute $D = E^{-1} \bmod 2P'Q'$. We choose a random element $g \in \mathbb{Z}_N^*$. Finally, for all $i \in \{1, \ldots, q_s\}$, we let $c_i$ be a random prime in $[(N + 1)/2, N[$ and define

    $$u = \hat{y}^{\prod_i c_i} \bmod n \enspace .$$

    We create the public key $\mathsf{pk} = \{N, n, u, g, E\}$. It is easy to see that the key generation is perfectly simulated.

  - When $\mathcal{A}$ requests the signature on a message $m_j \in \{0, 1\}^{\ell_m}$, for $j \in \{1, \ldots, q_s\}$, we simulate the signing oracle by computing

    $$r_j = (c_j\, g^{-(m_j+1)})^D \bmod N \quad \text{and} \quad s_j = \hat{y}^{\prod_{i \neq j} c_i} \bmod n \enspace .$$

    We return $\sigma_j = (r_j, s_j)$ as the signature on $m_j$. Here too, the simulation is perfect.

**Simulation 1**

- We let $N = \hat{n}$ and $g = \hat{y}$. We choose a random $\ell_n$-bit safe RSA modulus $n = (2p'+1)(2q'+1)$ and an (odd) $(\ell_m + 1)$-bit prime $E$. (W.l.o.g., we may assume that (odd) prime $E \in \mathbb{Z}^*_{2P'Q'}$ as otherwise we would have $E = P'$ or $E = Q'$, which yields the factorization of $N$.) Finally, we choose a random element $u \in \mathbb{Z}^*_n$.
  We create the public key $\mathsf{pk} = \{N, n, u, g, E\}$. The key generation is perfectly simulated.
- When $\mathcal{A}$ requests the signature on a message $m_j \in \{0,1\}^{\ell_m}$, for $j \in \{1, \ldots, q_s\}$, we simulate the signing oracle as follows.
  1. We choose a random element $r_j \in \mathbb{Z}^*_N$ and define $c_j = g^{m_j+1}\, {r_j}^E \bmod N$;
  2. If $c_j$ is not a prime lying in $[(N+1)/2, N[$, then we go back to Step 1.

  Next, we compute $s_j = u^{{c_j}^{-1} \bmod 2p'q'} \bmod n$ and return $\sigma_j = (r_j, s_j)$ as the signature on $m_j$. The simulation is perfect.

– Eventually, adversary $\mathcal{A}$ outputs with probability $\varepsilon_{\mathcal{A}}$ a valid signature forgery $\sigma_* = (r_*, s_*) \in [0, N[ \times [0, n[$ on a message $m_* \in \{0,1\}^{\ell_m}$, with $(m_*, \sigma_*) \neq (m_i, \sigma_i)$ for all $i \in \{1, \ldots, q_s\}$. We compute $c_* := g^{m_*+1}\, {r_*}^E \bmod N$.

- If $c_* \neq c_j$ for all $j \in \{1, \ldots, q_s\}$, if $c_* > 1$, and if $b = 0$ (*i.e.*, Simulation 0 was run) then it follows that $\gcd(c_*, \prod_i c_i) = 1$, since $c_* \in [2, N[$ and all $c_i$'s are primes in set $[(N+1)/2, N[$. Hence, from extended Euclidean algorithm, we get integers $\alpha$ and $\beta$ s.t. $\alpha\, c_* + \beta \prod_i c_i = 1$. Therefore, noting that $\hat{y}^{\prod_i c_i} \equiv u \equiv {s_*}^{c_*} \pmod{n}$ and $n = \hat{n}$, we have
$$\hat{y} \equiv \hat{y}^{\alpha\, c_* + \beta \prod_i c_i} \equiv \left(\hat{y}^{\alpha}\, {s_*}^{\beta}\right)^{c_*} \pmod{\hat{n}} \ .$$
  The pair $(\hat{x}, \hat{e})$ with $\hat{x} := \hat{y}^{\alpha}\, {s_*}^{\beta} \bmod \hat{n}$ and $\hat{e} := c_*$ is thus a solution to the flexible $\mathsf{RSA}$ problem.
- If $c_* = c_j$ for some $j \in \{1, \ldots, q_s\}$ (and thus $s_* = s_j$) and if $b = 1$ (*i.e.*, Simulation 1 was run) then, remembering that $N = \hat{n}$ and $g = \hat{y}$, we get
$$\begin{cases} g^{m_j+1}\, {r_j}^E \equiv g^{m_*+1}\, {r_*}^E \pmod{N} \implies \hat{y}^{(m_j - m_*)} \equiv \left(\dfrac{r_*}{r_j}\right)^E \pmod{\hat{n}}\,, \\ s_j = s_* \ . \end{cases}$$
  Note that we cannot have $m_* = m_j$ as otherwise we would have $r_* = r_j$ and so $(m_*, \sigma_*) = (m_j, \sigma_j)$, a contradiction. Therefore, since $E$ is an $(\ell_m + 1)$-bit integer, we can find integers $\alpha$ and $\beta$ by the extended Euclidean algorithm so that $\alpha\, E + \beta\, (m_j - m_*) = \gcd(E, m_j - m_*) = 1$. As a result, we have
$$\hat{y} \equiv \hat{y}^{\alpha\, E + \beta\, (m_j - m_*)} \equiv \left(\hat{y}^{\alpha}\, (r_*/r_j)^{\beta}\right)^E \pmod{\hat{n}}$$
  and the pair $(\hat{x}, \hat{e})$ with $\hat{x} := \hat{y}^{\alpha}\, (r_*/r_j)^{\beta} \bmod \hat{n}$ and $\hat{e} = E$ is a solution to the flexible $\mathsf{RSA}$ problem.

- If $c_* = 0$ and if $b = 0$ (*i.e.*, Simulation 0 was run) then, letting $\Lambda = \prod_i c_i$, we compute $\hat{d} := \hat{e}^{-1} \bmod \Lambda$ for an arbitrary $\hat{e} > 1$ such that $\gcd(\hat{e}, \Lambda) = 1$. So, the pair $(\hat{x}, \hat{e})$ with $\hat{x} := \hat{y}^{\hat{d}} \bmod \hat{n}$ is a solution to the flexible RSA problem:
$$\hat{x}^{\hat{e}} \equiv \hat{y}^{\hat{e}\hat{d}} \equiv \hat{y}^{\hat{e}\hat{d} \bmod \Lambda} \equiv \hat{y} \pmod{n}$$
because $c_* = 0$ implies $u = 1$ and thus $\hat{y}^{\Lambda} \bmod \hat{n} = 1$ (remember that $n = \hat{n}$ when $b = 0$).
- If $c_* = 1$ and if $b = 1$ (*i.e.*, Simulation 1 was run) then, using extended Euclidean algorithm, we can find integers $\alpha$ and $\beta$ s.t $\alpha\, E + \beta\, (m_* + 1) = \gcd(E, m_* + 1) = 1$.[6] Hence, since $c_* = 1 = \hat{y}^{m_*+1}\, {r_*}^{E} \bmod N$, we get
$$\hat{y} \equiv \hat{y}^{\alpha\, E + \beta\, (m_*+1)} \equiv \left(\hat{y}^{\alpha}\, {r_*}^{-\beta}\right)^{E} \pmod{\hat{n}} \ .$$
Consequently, the pair $(\hat{x}, \hat{e})$ with $\hat{x} := \hat{y}^{\alpha}\, {r_*}^{-\beta} \bmod \hat{n}$ and $\hat{e} = E$ is a solution to the flexible RSA problem.

Since $\mathcal{A}$'s view is perfectly simulated, the success probability of the reduction is clearly $\varepsilon_{\mathcal{A}}/2$.

For Simulation 0, we need to generate $\ell_n$-bit safe RSA modulus $N$, $(\ell_m{+}1)$-bit prime $E$, $\ell_n$-bit modular inverse $D$ and $\ell_n$-bit parameter $u$ in the key generation; we also need, for each signature query, compute $r_j$ and $s_j$. We assume that we have algorithms so that the generation of safe prime is quintic, the generation of a prime is quartic and the evaluation of a modular exponentiation or of a modular inverse is cubic, in the bitlength. The evaluation of $u$ and the $q_s$ $s_j$'s amounts to $\mathrm{O}(q_s \log q_s)$ $\ell_n$-bit exponentiations using the trick of [CLP05, § 3.3]. Hence, the running time required by the reduction is (approximatively) $\tau_{\mathcal{A}} + \mathrm{O}({\ell_n}^5 + q_s \log q_s\, {\ell_n}^3)$.

For Simulation 1, further assuming that primality testing is cubic in the bitlength, we similarly obtain that the running time required by the reduction is (approximatively) $\tau_{\mathcal{A}} + \mathrm{O}({\ell_n}^5 + q_s\, {\ell_n}^4)$. □

### 4.3 Comparison with other schemes

In Table 1, we compare the advantages and drawbacks of the schemes presented in Section 3 with our TSS scheme, including the differences in tightness of security reduction in the standard model, the size of signatures and the size of public/private keys. When applicable, we also give necessary conditions the hash function should fulfill (in addition to collision resistance).

From this table, it appears that the TSS scheme is proven secure *solely* under the strong-RSA assumption, with a *tight* security reduction. Furthermore, this is not done at the price of extra properties on a hash function, as the division-intractability for the GHR scheme. Twin-GHR is also tightly and solely related

[6] This last case explains why $(m + 1)$ (and not merely $m$) appears in the description of TSS.

**Table 1.** Performance comparison.

| | Security | | Typical values | Bitsizes[a] | | |
|---|---|---|---|---|---|---|
| | Tightness | Assumption[b] | | $\sigma$ | $\mathsf{pk}$ | $\mathsf{sk}$[c] |
| GHR (basic) | $\mathrm{O}(\frac{1}{q_s})$ | $\mathsf{Div} + \mathsf{SRSA}$ | $\ell_n \gg 1024$ | $\ell_n$ | $2\ell_n$ | $\frac{1}{2}\,\ell_n$ |
| GHR (tight) | $\mathrm{O}(1)$ | $\mathsf{Div} + \mathsf{DL} + \mathsf{SRSA}$ | $\ell_n = \ell_p = 1024$<br>$\ell_q = 160$ | $\ell_n + \ell_q$ | $2\ell_n + 3\ell_p$ | $\frac{1}{2}\,\ell_n$ |
| Twin-GHR | $\mathrm{O}(1)$ | SRSA | $\ell_n = 1024$<br>$\ell_m = 160$ | $2\ell_n + 2\ell_m$ | $4\ell_n$ | $\ell_n$ |
| CS | $\mathrm{O}(\frac{1}{q_s})$ | SRSA | $\ell_n \gg 1024$<br>$\ell_h = 160$ | $2\ell_n + \ell_h$ | $3\ell_n + \ell_h$ | $\frac{1}{2}\,\ell_n$ |
| CL | $\mathrm{O}(\frac{1}{q_s})$ | SRSA | $\ell_n \gg 1024$<br>$\ell_m = 160$, $\ell = 80$ | $2\ell_n + 2\ell_m + \ell$ | $4\ell_n$ | $\frac{1}{2}\,\ell_n$ |
| Fischlin | $\mathrm{O}(\frac{1}{q_s})$ | SRSA | $\ell_n \gg 1024$<br>$\ell_h = 160$ | $\ell_n + 2\ell_h$ | $4\ell_n$ | $\frac{1}{2}\,\ell_n$ |
| TSS | $\mathrm{O}(1)$ | SRSA | $\ell_n = 1024$<br>$\ell_m = 160$ | $2\ell_n$ | $4\ell_n + \ell_m$ | $\ell_n$ |

[a] To ease the reading, the bitsizes are rounded up to a few bits.
[b] Div stands for the division intractability assumption and DL for the discrete logarithm assumption.
[c] In the description of GHR and the CS-like schemes (Section 3), we have $\mathsf{sk} = \{p', q'\}$; however, it is possible to only store the value of $p'$ and to recover $q'$ from $p'$ (and pk). Similarly, for Twin-GHR, $\mathsf{sk} = \{p', P'\}$ is sufficient, and for TSS, it is possible to recover $\mathsf{sk} = \{p', q', D\}$ from $p'$, $P'$ (and pk).

to the strong RSA assumption. Twin-GHR and TSS however differ in their implementation. Compared to the former, TSS does not rely on an injective prime generation and needs no prime generation at all in the verification algorithm. Further, TSS offers shorter signatures.

On the minus side, our scheme produces longer signatures than Fischlin or GHR (but shorter than CS or CL). Another drawback is computational. TSS requires the generation of a large random prime. Even using efficient methods (*e.g.*, [JPV00]), this may be time-consuming for low-cost cryptographic devices. We present in the next section an on-line/off-line version of our scheme to address this issue.[7]

### 4.4 On-line/Off-line version

We present hereafter a variant of our scheme that allows the signer to carry out costly computations *before* knowing the message to be signed. This type of

[7] We observe that in Twin-GHR, only part of the signature can be precomputed (namely, $s_1$); parameter $s_2$ is dependent on the message to be signed.

signature scheme is usually referred to as *on-line/off-line scheme* [EGM96,ST01]. Using this paradigm, once the message is known, only a very fast on-line phase is needed. This property is paramount for time-constrained applications or for low-cost smartcards.

The message space is $\mathcal{M} = \{0,1\}^{\ell_m}$. Let $\ell_n$ and $\ell$ be two security parameters. Typical values are $\ell = 80$ and $\ell_n = 1024$. Our $\mathsf{TSS}$ scheme, in its on-line/off-line version, then goes as follows.

<u>Gen:</u> On input $\ell_n$ and $\ell_m$:
- choose an (odd) $(\ell_m + 1)$-bit prime $E$
- generate two random $\ell_n$-bit safe RSA moduli $n = (2p' + 1)(2q' + 1)$ and $N = (2P' + 1)(2Q' + 1)$ such that $\gcd(P'Q', E) = 1$;
- compute $D = E^{-1} \bmod 2P'Q'$;
- choose at random two elements $u \in \mathbb{Z}_n^*$ and $g \in \mathbb{Z}_N^*$.

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

<u>Sign (off-line part):</u> To prepare a coupon:
- choose a random prime $c$ in $[(N+1)/2, N[$;
- pick a random $(\ell_n + \ell_m + \ell)$-bit integer $k'$;
- compute $s = u^{c^{-1} \bmod 2p'q'} \bmod n$ and $r = g^{(k'-D)}\, c^D \bmod N$.

The coupon is $(k', r, s)$.

<u>Sign (on-line part):</u> Let $m \in \{0,1\}^{\ell_m}$ be the message to be signed:
- take a fresh coupon $(k', r, s)$;
- compute $k = k' + D \cdot m$.

The signature on $m$ is $\sigma = (k, r, s) \in [0, 2^{\ell_n+\ell_m+\ell+1}[ \times \mathbb{Z}_N^* \times \mathbb{Z}_n^*$.

<u>Verify:</u> Let $\sigma = (k, r, s)$ be a putative signature on message $m \in \{0,1\}^{\ell_m}$. Then:
- compute $c = g^{m+1}\,(r\,g^{-k})^E \bmod N$;
- check that $s^c \equiv u \pmod{n}$.

If this condition holds then signature $\sigma$ is accepted.

It is worth remarking that the key generation in the on-line/off-line version is *exactly* the one of the regular version: the public/private keys are the same in both versions.

Security reduction. We now show that this on-line/off-line version *tightly* meets the $\mathsf{EUF\text{-}CMA}$ security notion under the strong $\mathsf{RSA}$ assumption. Actually, we prove that an $\mathsf{EUF\text{-}CMA}$ adversary $\mathcal{A}^*$ against the on-line/off-line version is an $\mathsf{sEUF\text{-}CMA}$ adversary against the regular version of our signature scheme. In more detail, given a public key $\widehat{\mathsf{pk}} = \{\hat{N}, \hat{n}, \hat{u}, \hat{g}, \hat{E}\}$ and (at most) $q_s$ chosen-message calls to a $\mathsf{TSS}$ signing oracle, we want to produce a $\mathsf{TSS}$ signature forgery $\hat{\sigma}_* = (\hat{r}_*, \hat{s}_*)$ on a message $\hat{m}_*$, using $\mathcal{A}^*$.

- We let $\mathsf{pk} = \widehat{\mathsf{pk}}$ (*i.e.*, $\{N, n, u, g, E\} = \{\hat{N}, \hat{n}, \hat{u}, \hat{g}, \hat{E}\}$) as the public key for the on-line/off-line version.

– When $\mathcal{A}^*$ requests an [on-line/off-line] signature on a message $m_j \in \{0,1\}^{\ell_m}$, for $j \in \{1,\ldots,q_s\}$, we call the $\mathsf{TSS}$ signing oracle on input message $\hat{m}_j := m_j$ and get back a $\mathsf{TSS}$ signature $\hat{\sigma}_j = (\hat{r}_j, \hat{s}_j) \in [0, \hat{N}[ \times [0, \hat{n}[$ such that

$$\begin{cases} \hat{c}_j := \hat{g}^{m_j+1}\, \hat{r}_j{}^{\hat{E}} \bmod \hat{N} \text{ is a prime in } [(\hat{N}+1)/2, \hat{N}[\text{, and} \\ \hat{s}_j{}^{\hat{c}_j} \equiv \hat{u} \pmod{\hat{n}} \ . \end{cases}$$

Next, we pick a random $(\ell_n + \ell_m + \ell)$-bit integer $k_j$. We compute $r_j = \hat{r}_j\, \hat{g}^{k_j} \bmod \hat{N}$ and let $s_j = \hat{s}_j$. We return $\sigma_j = (k_j, r_j, s_j)$ as the on-line/off-line signature on message $m_j$.
It is easy to see that $\sigma_j$ is a valid signature since $c_j := g^{m_j+1}\,(r_j\, g^{-k_j})^E \bmod N = \hat{g}^{m_j+1}\, \hat{r}_j{}^{\hat{E}} \bmod \hat{N} = \hat{c}_j$ is a prime in $[(N+1)/2, N[$ and $s_j{}^{c_j} \equiv \hat{s}_j{}^{\hat{c}_j} \equiv \hat{u} \equiv u \pmod{n}$.

– Eventually, with probability $\varepsilon_{\mathcal{A}^*}$ and within time $\tau_{\mathcal{A}^*}$, $\mathcal{A}^*$ returns an on-line/off-line signature forgery $\sigma_* = (k_*, r_*, s_*)$ on a message $m_* \in \{0,1\}^{\ell_m}$, with $m_* \neq m_j$ for all $j \in \{1,\ldots,q_s\}$.

– From $\sigma_* = (k_*, r_*, s_*)$, we form the signature forgery $\hat{\sigma}_* = (\hat{r}_*, \hat{s}_*)$, where

$$\hat{r}_* = r_*\, \hat{g}^{-k_*} \bmod \hat{N} \quad \text{and} \quad \hat{s}_* = s_* \ ,$$

on message $\hat{m}_* := m_*$. Again, it is easy to see that this is a valid signature. Furthermore, as $m_* \neq m_j$, it obviously follows that $(\hat{m}_*, \hat{\sigma}_*) \neq (\hat{m}_j, \hat{\sigma}_j)$, for all $j \in \{1,\ldots,q_s\}$.

Tightness of the reduction. The statistical distance between the $k_j$'s returned by the signature simulation and the $k_j$'s that would be returned by an actual signer is bounded by $2^{-\ell}$, for each signature. Hence, there exists a reduction that succeeds with probability $\varepsilon \geq \varepsilon_{\mathcal{A}^*} - 2^{-\ell} q_s$ and within time $\tau \lesssim \tau_{\mathcal{A}^*} + (q_s+1)\,\mathrm{O}({\ell_n}^3)$, neglecting the time required to generate random numbers. As the regular version is tightly related to the flexible $\mathsf{RSA}$ problem, the on-line/off-line version is tightly $\mathsf{EUF\text{-}CMA}$ secure under the strong $\mathsf{RSA}$ assumption.

$\mathsf{EUF\text{-}CMA}$ vs. $\mathsf{sEUF\text{-}CMA}$. The security proof assumes an $\mathsf{EUF\text{-}CMA}$ adversary (as opposed to an $\mathsf{sEUF\text{-}CMA}$ adversary) against our on-line/off-line signature scheme. Even testing the ranges of $(k, r, s)$ in the verification step would not achieve $\mathsf{sEUF\text{-}CMA}$ security. Indeed, imagine an $\mathsf{sEUF\text{-}CMA}$ adversary returning a signature forgery $\sigma_* = (k_*, r_*, s_*) \neq \sigma_j$ on message $m_* = m_j$, for some $j \in \{1,\ldots,q_s\}$. Then, the $\mathsf{TSS}$ signature forgery $\hat{\sigma}_* = (\hat{r}_*, \hat{s}_*)$ on message $\hat{m}_* = m_*$ returned by the above reduction is not mandatorily a *valid* forgery, *i.e.*, such that $(\hat{m}_*, \hat{\sigma}_*) \neq (\hat{m}_j, \hat{\sigma}_j)$, since $\hat{m}_* = \hat{m}_j$ and $\sigma_* \neq \sigma_j \iff (k_*, \hat{r}_*\, \hat{g}^{k_*} \bmod \hat{N}, \hat{s}_*) \neq (k_j, \hat{r}_j\, \hat{g}^{k_j} \bmod \hat{N}, \hat{s}_j)$ but

$$(k_*, \hat{r}_*\, \hat{g}^{k_*} \bmod \hat{N}, \hat{s}_*) \neq (k_j, \hat{r}_j\, \hat{g}^{k_j} \bmod \hat{N}, \hat{s}_j) \ \not\Longrightarrow\ (\hat{r}_*, \hat{s}_*) \neq (\hat{r}_j, \hat{s}_j) \ .$$

It is even more apparent with a counter-example: if $\sigma = (k, r, s)$ is a valid on-line/off-line signature on message $m$ so is $\sigma' = (k+1, g\, r \bmod N, s)$ on the *same* message $m$. Hence, the on-line/off-line version of TSS we describe is not sEUF-CMA secure, but only EUF-CMA secure.

For most cryptographic applications, existential unforgeability is sufficient. Our TSS signature scheme can however be converted into an on-line/off-line scheme to accommodate *strong* unforgeability (sEUF) by using standard techniques [ST01], at the price of longer — and thus *different* — keys.

## 5 Conclusion

This paper presented a practical sEUF-CMA signature scheme whose security is solely and tightly related to the SRSA assumption, in the standard model. In contrast to the CS scheme and its variants, the security of our TSS scheme is optimal and, contrary to the GHR scheme, this optimal bound does not result from the use of so-called division-intractable hash functions. Indeed, the TSS scheme does not require the use of hash functions by its very specification. Actually, TSS scheme is much closer, in its properties, to the twinning-based version of GHR, even if constructed in a completely different manner. The main differences between the two schemes lie in the implementation and in the signature size. Moreover, the TSS scheme also comes with an on-line/off-line version for time-constrained applications or low-cost cryptographic devices. Remarkably, this on-line/off-line version uses exactly the same set of keys as the regular version.

## References

[BB04] D. Boneh and X. Boyen. Short signatures without random oracles. In *Advances in Cryptology – EUROCRYPT 2004*, LNCS 3027, pp. 56–73. Springer-Verlag, 2004.

[BC92] J. Bos and D. Chaum. Provably unforgeable signatures. In *Advances in Cryptology – CRYPTO '92*, LNCS 740, pp. 1–14. Springer-Verlag, 1993.

[BLS04] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. *Journal of Cryptology* **17**(4):297–319, 2004.

[BM92] M. Bellare and S. Micali. How to sign given any trapdoor permutation. *Journal of the ACM* **39**(1):214–233, 1992.

[BNPS03] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology* **16**(3):185–215, 2003.

[BP97] N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In W. Fumy, editor, *Advances in Cryptology – EUROCRYPT '97*, LNCS 1233, pp. 480–494. Springer-Verlag, 1997.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *1st ACM Conference on Computer and Communications Security*, pp. 62–73. ACM Press, 1993.

[BR96] ———. The exact security of digital signatures: How to sign with RSA and Rabin. In *Advances in Cryptology – EUROCRYPT '96*, LNCS 1070, pp. 399–416. Springer-Verlag, 1996.

[BSW06] D. Boneh, E. Shen, and B. Waters. Strongly unforgeable signature schemes based on comptutational Diffie-Hellman. In *Public Key Cryptography − PKC 2006*, LNCS 3958, pp. 229–240. Springer, 2006.

[CD96] R. Cramer and I. Damgård. New generation of secure and practical RSA-based signatures. In *Advances in Cryptology − CRYPTO '96*, LNCS 1109, pp. 173–185. Springer-Verlag, 1996.

[CG05] D. Catalano and R. Gennaro. Cramer-Damgård signatures revisited: Efficient flat-tree signatures based on factoring. In *Public Key Cryptography − PKC 2005*, LNCS 3386, pp. 313–327. Springer-Verlag, 2005.

[CGH98] R. Canetti, O. Golreich, and S. Halevi. The random oracle methodology, revisited. In *30th Annual ACM Symposium on Theory of Computing*, pp. 209–217. ACM Press, 1998.

[CL02] J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *Security in Communication Networks (SCN 2002)*, LNCS 2676, pp. 268–289. Springer-Verlag, 2002.

[CLP05] J.-S. Coron, D. Lefranc, and G. Poupard. A new baby-step giant-step algorithm and some applications to cryptanalysis. In *Cryptographic Hardware and Embedded Systems − CHES 2005*, LNCS 3659, pp. 47–60. Springer, 2005.

[CN00] J.-S Coron and D. Naccache. Security analysis of the Gennaro-Halevi-Rabin signature scheme. In *Advances in Cryptology − EUROCRYPT 2000*, LNCS 1807, pp. 91–101. Springer-Verlag, 2000.

[Cor00] J.-S Coron. On the exact security of full domain hash. In *Advances in Cryptology − CRYPTO 2000*, LNCS 1880, pp. 229–235. Springer-Verlag, 2000.

[CS00] 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.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory* **IT-22**(6):644–654, 1976.

[DN94] C. Dwork and M. Naor. An efficient existentially unforgeable signature scheme and its applications. In *Advances in Cryptology − CRYPTO '94*, LNCS 839, pp. 234–246. Springer-Verlag, 1994.

[EGM96] S. Even , O. Goldreich, and S. Micali. On-line/off-line digital signatures. *Journal of Cryptology* **9**(1):35–67, 1996.

[Fis03] M. Fischlin. The Cramer-Shoup strong-RSA signature scheme revisited. In *Public Key Cryptography − PKC 2003*, LNCS 2567, pp. 116–129. Springer-Verlag, 2003.

[FO97] E. Fujisaki and T. Okamoto. Statistical zero-knowledge protocols to prove modular polynomial equations. In *Advances in Cryptology − CRYPTO '97*, LNCS 1294, pp. 16–30. Springer-Verlag, 1997.

[FS87] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology − CRYPTO '86*, LNCS 263, pp. 186–194. Springer-Verlag, 1987.

[GHR99] R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. In *Advances in Cryptology − EUROCRYPT '99*, LNCS 1592, pp. 123–139. Springer-Verlag, 1999.

[GJ03] E.-J. Goh and S. Jarecki. A signature scheme as secure as the Diffie-Hellman problem. In *Advances in Cryptology − EUROCRYPT 2003*, LNCS 2656, pp. 401–415. Springer-Verlag, 2003.

[GMR88] 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.

[Gol86] O. Goldreich. Two remarks concerning the Goldwasser-Micali-Rivest signature scheme. In *Advances in Cryptology – CRYPTO '86*, LNCS 263, pp. 104–110. Springer-Verlag, 1986.

[JPV00] M. Joye, P. Paillier, and S. Vaudenay. Efficient generation of prime numbers. In *Cryptographic Hardware and Embedded Systems – CHES 2000*, LNCS 1965, pp. 340–354. Springer-Verlag, 2000.

[KM04] N. Koblitz and A. Menezes. Another look at "provable security". Cryptology ePrint Archive 2004/152, 2004. To appear in Journal of Cryptology.

[KR00] H. Krawczyk and T. Rabin. Chameleon signatures. In *Symposium on Network and Distributed System Security – NDSS 2000*, pp. 143–154. Internet Society, 2000.

[KS06] K. Kurosawa and K. Schmidt-Samoa. New online/offline signature schemes without random oracles. In *Public Key Cryptography – PKC 2006*, LNCS 3958, pp. 330–346. Springer, 2006.

[KW03] J. Katz and N. Wang. Efficiency improvements for signature schemes with tight security reductions. In *10th ACM Conference on Computer and Communications Security*, pp. 155–164. ACM Press, 2003.

[Mer87] R. Merkle. A digital signature based on a conventional encryption function. In *Advances in Cryptology – CRYPTO '87*, LNCS 293, pp. 369–378. Springer-Verlag, 1987.

[NPS01] 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*, pp. 20–27. ACM Press, 2001.

[NY89] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *21st Annual ACM Symposium on Theory of Computing*, pp. 33–43. ACM Press, 1989.

[PS96] D. Pointcheval and J. Stern. Security proofs for signature schemes. In *Advances in Cryptology – EUROCRYPT '96*, LNCS 1070, pp. 387–398. Springer-Verlag, 1996.

[PV05] P. Paillier and D. Vergnaud. Discrete-log-based signatures may not be equivalent to discrete log. In *Advances in Cryptology – ASIACRYPT 2005*, LNCS 3788, pp. 1–20. Springer-Verlag, 2005.

[Rom90] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd Annual ACM Symposium on Theory of Computing*, pp. 387–394. ACM Press, 1990.

[RSA78] 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.

[ST01] A. Shamir and Y. Tauman. Improved online/offline signature schemes. In *Advances in Cryptology – CRYPTO 2001*, LNCS 2139, pp. 355–367. Springer-Verlag, 2001.

[Wat05] B. Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology – EUROCRYPT 2005*, LNCS 3494, pp. 114–127. Springer, 2005.

[Zhu01] H. Zhu. New digital signature scheme attaining immunity against adaptive chosen message attack. *Chinese Journal of Electronics* **10**(4):484–486, 2001.

[Zhu03] ———. A formal proof of Zhu's signature scheme. Cryptology ePrint Archive, Report 2003/155, 2003.