On Rabin-type Signatures*

Marc Joye[1] and Jean-Jacques Quisquater[2]

[1] Gemplus Card International, Card Security Group
Parc d'Activités de Gémenos, B.P. 100, 13881 Gémenos Cedex, France
marc.joye@gemplus.com
http://www.geocities.com/MarcJoye/
[2] UCL Crypto Group, Université catholique de Louvain
Place du Levant 3, 1348 Louvain-la-Neuve, Belgium
jjq@dice.ucl.ac.be
http://www.uclcrypto.org/

Abstract. This paper specializes the signature forgery by Coron, Naccache and Stern (1999) to Rabin-type systems. We present a variation in which the adversary may derive the private keys and thereby forge the signature on *any* chosen message. Further, we demonstrate that, contrary to the RSA, the use of larger (even) public exponents does not reduce the complexity of the forgery. Finally, we show that our technique is very general and applies to any Rabin-type system designed in a unique factorization domain, including the Williams' M^3 scheme (1986), the cubic schemes of Loxton et al. (1992) and of Scheidler (1998), and the cyclotomic schemes (1995).

Keywords. Rabin-type systems, digital signatures, signature forgeries, factorization.

1 Introduction

In this paper, we specialize the signature forgery of Coron, Naccache and Stern [8] to Rabin-type systems [19, 24]. We present a variation in which the adversary may derive the private keys and thereby forge the signature on *any* chosen message. Further, we demonstrate that, contrary to the RSA, systems using larger (even) public exponents are *equally* susceptible to the presented forgery. We also show that our technique is very general and applies to any Rabin-type systems designed in a unique factorization domain, including the Williams' M^3 scheme [27], the cubic schemes of Loxton et al. [16] and of Scheidler [20], and the cyclotomic schemes [21].

As an application, we analyze the implications of our forgery against the PKCS #1 standard [4]. Finally, and of independent interest, we propose a *generic* technique (i.e., applicable to *any* encoding message method) that reduces the overall complexity of a forgery from n to $\sqrt{n}$.

* A working draft of this work was presented at the ISO/IEC JTC1/SC27/WG2 meeting in August 1999.

The rest of this paper is organized as follows. We first begin by a brief presentation of Rabin-type systems. Next, in Section 3, we review the Coron-Naccache-Stern forgery and turn it into an universal forgery against Rabin-type signature schemes. In Section 4, we generalize the forgery to higher exponents and higher degree schemes. We apply it to PKCS #1 encoding method in Section 5. We also present a generic technique for reducing the complexity of the forgery. Finally, we conclude in Section 6.

2 Rabin-type Systems

In this section, following the IEEE/P1363 specifications for public-key cryptography [2] (see also [17, Chapter 11]), we present a modified version of the Rabin-Williams signature scheme [19, 24]. The scheme consists of three algorithms: the `setup`, the `signature` and the `verification`. For setting up the system, each user generates a pair of public/private keys. The private key is used to sign messages with the signature algorithm. Using the corresponding public key, a signature can then be verified and the signed message recovered with the verification algorithm.

`setup:` Generate two primes p, q such that $p \equiv 3 \pmod 8$ and $q \equiv 7 \pmod 8$ and compute $n = pq$. Define an appropriate "representation" function $R : \mathcal{M} \to \mathcal{M}_R : m \mapsto \widetilde{m} = R(m)$, where $\mathcal{M}$ is the set of valid messages and $\mathcal{M}_R = \{\widetilde{m} = R(m) \in (\mathbb{Z}/n\mathbb{Z})^* : \widetilde{m} \equiv 6 \pmod{16}\}$ is the set of message representatives. The public key is n and the private key is $d = (n-p-q+5)/8$.

`signature:` Compute $\widetilde{m} = R(m)$ and $\widehat{m}$ given by

$$\widehat{m} = \begin{cases} \widetilde{m} \bmod n & \text{if } (\widetilde{m}|n) = 1 \\ \widetilde{m}/2 \bmod n & \text{if } (\widetilde{m}|n) = -1 \end{cases} .$$

The signature on message m is $s = \widehat{m}^d \bmod n$.

`verification:` Compute $m' = s^2 \bmod n$. Then, take

$$\widetilde{m} = \begin{cases} m' & \text{if } m' \equiv 6 \pmod 8 \\ 2m' & \text{if } m' \equiv 3 \pmod 8 \\ n - m' & \text{if } m' \equiv 7 \pmod 8 \\ 2(n - m') & \text{if } m' \equiv 2 \pmod 8 \end{cases} .$$

If $\widetilde{m} \in \mathcal{M}_R$ then the signature is accepted and message m is recovered from $\widetilde{m}$.

Remark 1. A signature scheme with appendix can also be defined along these lines. In that case, the set of messages representatives is given by $\mathcal{M}_R = \{\widetilde{m} \in (\mathbb{Z}/n\mathbb{Z})^* : \widetilde{m} \equiv 10 \pmod{16}\}$.

Remark 2. The recommended function R for message encoding is the international standard ISO/IEC 9796 [3]. Another possible encoding is specified in PKCS #1 v2.0 [4]; this latter, however, only covers signature schemes with appendix.

Remark 3. The presented scheme supposes $p \equiv 3 \pmod 8$ and $q \equiv 7 \pmod 8$. Similar schemes with form-free primes may be found in [26, 13].

Noticing that $2d = (p-1)(q-1)/4+1$, the correctness of the method follows from the next lemma; moreover, it uses the fact that if $n \equiv 5 \pmod 8$ then $(-z|n) = (z|n) = -(2z|n)$.

Lemma 1. *Let $n = pq$, where p, q are distinct primes and $p, q \equiv 3 \pmod 4$. If $(z|n) = 1$, then*

$$z^{(p-1)(q-1)/4} \equiv \pm 1 \pmod n \ .$$

Proof. See [24, Lemma 1]. □

3 Signature Forgeries

This section reviews the Coron-Naccache-Stern forgery [8] when applied to the Rabin-Williams scheme. In the second part, we modify it into an universal forgery so that the signature on any message can be obtained without knowing the private key.

3.1 Coron-Naccache-Stern forgery

As aforementioned, for each message representative $\widetilde{m}_i$, $\widehat{m}_i = \widetilde{m}_i$ if $(\widetilde{m}_i|n) = 1$ and $\widehat{m}_i = \widetilde{m}_i/2$ if $(\widetilde{m}_i|n) = -1$; the corresponding signature is then given by $s_i = {\widehat{m}_i}^d \bmod n$. (Note here that $(\widehat{m}_i|n) = 1$.)

Suppose that an adversary has collected several pairs $(\widehat{m}_i, s_i)$ such that the $\widehat{m}_i$'s are smooth (modulo n). More precisely, suppose she knows

$$\widehat{m}_i \equiv (-1)^{v_{0,i}} \prod_{1 \le j \le B} {p_j}^{v_{j,i}} \pmod n \,, \tag{1}$$

where $p_1 < \cdots < p_B$ are prime and $v_{j,i} \in \mathbb{Z}$, and $s_i = {\widehat{m}_i}^d \bmod n$, for $1 \le i \le \ell$. Then she can forge the signature on a message m_τ, provided that $\widehat{m}_\tau$ is smooth (modulo n), as follows.[1,2]

To each $\widehat{m}_i$, she associates the B-tuple $\vec{V}_i$ given by $\vec{V}_i = (v_{1,i}, \ldots, v_{B,i})$. Let $\vec{V}_\tau$ denote the B-tuple corresponding to $\widehat{m}_\tau$. If there exist integers β_i such that $\vec{V}_\tau \equiv \sum_{1 \le i \le \ell} \beta_i \vec{V}_i \pmod 4$, then

$$v_{j,\tau} = \sum_{1 \le i \le \ell} \beta_i \, v_{j,i} - 4\gamma_j \quad (1 \le j \le B) \tag{2}$$

[1] It is here essential to note that the smoothness requirement must only be satisfied modulo n. For example, 197 is prime in $\mathbb{Z}$ but is 3-smooth as an element of $(\mathbb{Z}/437\mathbb{Z})$, i.e., $197 \equiv 2^9 \cdot 3^{-3} \pmod{437}$.

[2] In [8], the authors only consider positive "messages" $\widehat{m}_i$. We slightly generalize their presentation by introducing the term $(-1)^{v_{0,i}}$ in Eq. (1).

for some $\gamma_j \in \mathbb{Z}$.
Hence, the signature on message m_τ can be expressed as

$$\begin{aligned} s_\tau := \widehat{m}_\tau{}^d \bmod n &\equiv \Big[(-1)^{v_{0,\tau}} \prod_{1\le j\le B} p_j{}^{v_{j,\tau}}\Big]^d \qquad \text{(from Eq. (1))} \\ &\equiv \prod_{1\le j\le B} p_j^{v_{\tau,j}\,d} \qquad \text{(since } d \text{ is even)} \\ &\equiv \prod_{1\le i\le \ell} \prod_{1\le j\le B} p_j{}^{(\beta_i v_{j,i} - 4\gamma_j)d} \qquad \text{(from Eq. (2))} \\ &\equiv \prod_{1\le i\le \ell} s_i{}^{\beta_i} \prod_{1\le j\le B} p_j{}^{-2\gamma_j} \pmod n \ . \end{aligned} \tag{3}$$

The only difference with [8] is that we use the weaker relation $p_j{}^{4d} \equiv p_j{}^2 \pmod n$. (The relation $p_j{}^{2d} \equiv \pm p_j \pmod n$ only holds when $(p_j|n) = 1$; see Lemma 1.)

3.2 Universal forgery

We will now show that we can do much better than forging the signature on a smooth $\widehat{m}_\tau$, namely, forging the signature on *any* chosen $\widehat{m}_\tau$. We need the following proposition.

Proposition 1. *Let $n = pq$, where p, q are distinct primes and $p, q \equiv 3 \pmod 4$ and let $d = (n - p - q + 5)/8$. If $(z|n) = -1$, then*

$$\gcd\big(z^{2d} \mp z \pmod n, n\big) = p \text{ or } q \ .$$

Proof. Since $p, q \equiv 3 \pmod 4$, both $(p-1)/2$ and $(q-1)/2$ are odd. Hence, $z^{2d} \equiv z^{(p-1)(q-1)/4}\, z \equiv (z|p)^{(q-1)/2}\, z \equiv (z|p)\, z \pmod p$, and similarly $z^{2d} \equiv (z|q)\, z \pmod q$. Noting that $(z|p) = -(z|q) = \pm 1$, the proposition is proved. □

The above proposition suggests that if an adversary can derive the signature on an $\widehat{m}_\tau$ such that $(\widehat{m}_\tau|n) = -1$, then she can factor the modulus by computing $\gcd\big(s_\tau{}^2 - \widehat{m}_\tau \pmod n, n\big)$. This, however, is *not* possible. Define

$$\mathcal{J}(n, B) = \Big\{1 \le j \le B : p_j \text{ is prime and } \Big(\frac{p_j}{n}\Big) = -1\Big\} \ . \tag{4}$$

Since $(\widehat{m}_i|n) = 1$ (cf. beginning of § 3.1), we must have

$$\sum_{j\in\mathcal{J}(n,B)} v_{j,i} \equiv 0 \pmod 2 \ . \tag{5}$$

So, any linear combination, as done in Eq. (2), will always yield $\sum_{j\in\mathcal{J}(n,B)} v_{j,\tau} \equiv 0 \pmod 2$, resulting in $(\widehat{m}_\tau|n) = 1$.

But there is another way to consider Proposition 1: If the adversary is able to obtain the signature on $\widehat{m}_\tau = \pm {\widehat{r}_\tau}^2$ for some $\widehat{r}_\tau$ such that $(\widehat{r}_\tau|n) = -1$ (note here that $(\widehat{m}_\tau|n) = (\pm 1|n) \cdot ({\widehat{r}_\tau}^2|n) = 1 \cdot 1 = 1$), then

$$\gcd\bigl(s_\tau - \widehat{r}_\tau \pmod n, n\bigr) \tag{6}$$

will give a non-trivial factor of n. To make this feasible, in addition to verify Eq. (2), the components of $\vec{V}_\tau$ must satisfy

$$v_{j,\tau} \equiv 0 \pmod 2 \quad \text{for all } 1 \le j \le B \tag{7}$$

and

$$\sum_{j \in \mathcal{J}(n,B)} v_{j,\tau} \equiv 2 \pmod 4 \ . \tag{8}$$

The first condition ensures that the "message" corresponding to $\vec{V}_\tau$ is a square (i.e., $\widehat{m}_\tau = \pm {\widehat{r}_\tau}^2$), while the second condition ensures that $(\widehat{r}_\tau|n) = -1$. Replacing $v_{j,\tau}$ by Eq. (2), Eqs (7) and (8) can respectively be rewritten as

$$\sum_{1 \le i \le \ell} \beta_i \, v_{j,i} \equiv 0 \pmod 2 \quad \text{for all } 1 \le j \le B \tag{9}$$

and, defining $2\zeta_i = (\sum_{j \in \mathcal{J}(n,B)} v_{j,i}) \bmod 4$ ($\in \{0, 2\}$ from Eq. (5)),

$$\begin{aligned} &\sum_{j \in \mathcal{J}(n,B)} \sum_{1 \le i \le \ell} \beta_i \, v_{j,i} \equiv \sum_{1 \le i \le \ell} \beta_i \sum_{j \in \mathcal{J}(n,B)} v_{j,i} \equiv \sum_{1 \le i \le \ell} \beta_i \, 2\zeta_i \equiv 2 \pmod 4 \\ &\iff \sum_{1 \le i \le \ell} \beta_i \, \zeta_i \equiv 1 \pmod 2 \ . \end{aligned} \tag{10}$$

To sum up, an adversary can recover the secret factorization of n by carrying out the following steps:

(I) For $1 \le i \le \ell$, write $\widehat{m}_i \equiv (-1)^{v_{0,j}} \prod_{1 \le j \le B} {p_j}^{v_{j,i}} \pmod n$;

(II) Define $\vec{V}_i = (v_{1,i}, \ldots, v_{B,i})$ and $\zeta_i = \dfrac{(\sum_{j \in \mathcal{J}(n,B)} v_{j,i}) \bmod 4}{2}$;

(III) Find $\beta_1, \ldots, \beta_\ell$ such that

  (a) for $1 \le j \le B$, $\sum_{1 \le i \le \ell} \beta_i \, v_{j,i} \equiv 0 \pmod 2$;

  (b) $\sum_{1 \le i \le \ell} \beta_i \, \zeta_i \equiv 1 \pmod 2$;

(IV) Compute $\vec{V}_\tau = (v_{1,\tau}, \ldots, v_{B,\tau}) \in (\mathbb{Z}/4\mathbb{Z})^B$,
where $v_{j,\tau} = \sum_{1 \le i \le \ell} \beta_i \, v_{j,i} - 4\gamma_j$ for some $\gamma_j \in \mathbb{Z}$;

(V) Set $\widehat{r}_\tau = \prod_{1 \le j \le B} {p_j}^{v_{j,\tau}/2} \pmod n$;

(VI) Compute $s_\tau = \prod_{1 \le i \le \ell} {s_i}^{\beta_i} \prod_{1 \le j \le B} {p_j}^{-2\gamma_j} \bmod n$;

(VII) Recover the factors of n by computing $\gcd(s_\tau - \widehat{r}_\tau \pmod n, n)$.

Example 1. Here is a "toy" example to illustrate the forgery. Let $p = 8731$ ($\equiv 3 \pmod 8$) and $q = 3079$ ($\equiv 7 \pmod 8$) yielding a modulus $n = 26882749$. So, the private exponent is given by $d = 3358868$.
We consider p_4-smooth message representatives $\widetilde{m}_i \in \mathcal{M}_R$. We have $\mathcal{J}(n, 4) = \{2, 7\}$. Suppose, we are given:

$\widetilde{m}_i$	$\widehat{m}_i \pmod n$	s_i	$\vec{V}_i$	ζ_i
70	$70\ (= 2 \cdot 5 \cdot 7)$	8417525	$(1, 0, 1, 1)$	1
294	$147\ (= 3 \cdot 7^2)$	11480098	$(0, 1, 0, 2)$	1
486	$243\ (= 3^5)$	16287310	$(0, 5, 0, 0)$	0
630	$630\ (= 2 \cdot 3^2 \cdot 5 \cdot 7)$	1630174	$(1, 2, 1, 1)$	1

Conditions (III-a) and (III-b) yield

$$\begin{cases} \beta_1 + \beta_4 \equiv 0 \pmod 2 \\ \beta_2 + \beta_3 \equiv 0 \pmod 2 \\ \beta_1 + \beta_2 + \beta_4 \equiv 1 \pmod 2 \end{cases} \iff \begin{cases} \beta_1 \equiv \beta_4 \pmod 2 \\ \beta_2 \equiv \beta_3 \equiv 1 \pmod 2 \end{cases} .$$

Taking $\beta_1 = \beta_4 = 0$ and $\beta_2 = \beta_3 = 1$, we have $\vec{V}_\tau = (0, 2, 0, 2)$, which corresponds to $\widehat{m}_\tau = 3^2 \cdot 7^2 = 21^2$ whose signature is given by $s_\tau = {s_2}^1 {s_3}^1 3^{-2} \bmod n = 8076196$. So, the factorization of n is obtained by computing $\gcd(8076196 - 21, n) = 8731\ (= p)$. ♢

We will see below (Algorithm 1) a simple method to compute a solution $(\beta_1, \ldots, \beta_\ell) \in (\mathbb{Z}/2\mathbb{Z})^\ell$. This is a slight modification of Algorithm 2.3.1 in [7, pp. 56–57] (see also [12, Algorithm N, pp. 425–426]).

Using matrix notations, Conditions (III-a) and (III-b) can be rewritten as

$$\underbrace{\begin{pmatrix} v_{1,1} & \cdots & v_{1,\ell} & 0 \\ \vdots & & \vdots & \vdots \\ v_{B,1} & \cdots & v_{B,\ell} & 0 \\ \zeta_1 & \cdots & \zeta_\ell & 1 \end{pmatrix}}_{\substack{\pmod 2 \\ := \mathbf{U}}} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_\ell \\ 1 \end{pmatrix} \equiv \vec{0} \pmod 2 . \tag{11}$$

So, the problem is reduced to find a vector $\vec{K} = (\kappa_1, \ldots, \kappa_\ell, \kappa_{\ell+1}) \in \ker \mathbf{U}$ (i.e., the kernel of matrix $\mathbf{U}$), whose last coordinate, $\kappa_{\ell+1}$, is equal to 1. In that case, we have $\beta_i = \kappa_i$, $1 \le i \le \ell$.

Algorithm 1. This algorithm computes a solution (if any) $\beta_1, \ldots, \beta_\ell$ satisfying Eq. (11). We let $u_{r,c}$ denote the entry (modulo 2) at row r and column c of matrix $\mathbf{U}$.

1. [initialization] Set $c \leftarrow 1$ and for $1 \leq j \leq B+1$, set $t_j \leftarrow 0$.
2. [scanning] If there is some r in the range $1 \leq r \leq B+1$ such that $u_{r,c} = 1$ and $t_r = 0$, then go to Step 3. Otherwise, go to Step 5.
3. [elimination] For all $j \neq r$, if $u_{j,c} = 1$, then add (modulo 2) row r to row j. Set $t_r \leftarrow c$.
4. [loop] If $c \leq \ell$, then set $c \leftarrow c+1$ and go to Step 2.
5. [kernel] Evaluate the vector $\vec{K} = (\kappa_1, \ldots, \kappa_{\ell+1})$ defined by

$$\kappa_i = \begin{cases} u_{j,c} & \text{if } t_j = i > 0 \\ 1 & \text{if } i = c \\ 0 & \text{otherwise} \end{cases} .$$

   If $\kappa_{\ell+1} = 0$ and $c \leq \ell$, then set $c \leftarrow c+1$ and go to Step 2.
6. [output] If $\kappa_{\ell+1} = 1$, then output $\beta_i \leftarrow \kappa_i$ for all $1 \leq i \leq \ell$; otherwise, output `no solution`.

Example 1 (cont'd). If we apply the previous algorithm to Example 1, matrix $\mathbf{U}$ is given by

$$\mathbf{U} = \begin{pmatrix} \boxed{1} & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix} .$$

We then successively obtain for $c = 1, 2, 3$

$$\mathbf{U} = \begin{pmatrix} \boxed{1} & 0 & 0 & 1 & 0 \\ 0 & \boxed{1} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} \boxed{1} & 0 & 0 & 1 & 0 \\ 0 & \boxed{1} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boxed{1} & 0 & 1 \end{pmatrix}, \begin{pmatrix} \boxed{1} & 0 & 0 & 1 & 0 \\ 0 & \boxed{1} & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boxed{1} & 0 & 1 \end{pmatrix}$$

after the elimination step. We also have $t_1 = 1$, $t_2 = 2$, $t_5 = 3$ (which is indicated by the boxes ($t_r = c$)) and $t_3 = t_4 = 0$. At this point, we have $c = 4$ and the kernel step yields the vector $\vec{K} = (u_{1,4}, u_{2,4}, u_{5,4}, 1, 0) = (1, 0, 0, 1, 0)$. Since $\kappa_5 = 0$, we increment c, $c = 5$, and go to the scanning step. Then, since $t_2 = t_5 = 0$ (note that $u_{c,2} = u_{c,5} = 1$), we directly go to the kernel step and obtain the new vector $\vec{K} = (u_{1,5}, u_{2,5}, u_{5,5}, 0, 1) = (0, 1, 1, 0, 1)$. So, we finally find $\beta_1 = 0$, $\beta_2 = 1$, $\beta_3 = 1$ and $\beta_4 = 0$. $\diamondsuit$

3.3 Improvements

The methods we have presented so far are subject to numerous possible improvements. We just mention two of these.

In Eq. (1), we consider only "messages" $\widehat{m}_i$ whose largest prime factor (modulo n) is p_B. As for modern factorization methods, a substantial speed-up can be obtained by also considering the $\widehat{m}_i$'s which are p_B-smooth except for one or two factors [15]. Another speed-up can be obtained by using structured Gaussian elimination to solve Eq. (11); see [18] for an efficient variation directly applicable to our case.

4 Generalizations

4.1 Higher exponents

The signature scheme presented in Section 2 can be generalized to other even public exponents besides $e = 2$. Define $\Lambda = \mathrm{lcm}[(p-1)/2, (q-1)/2]$. It suffices to choose e relatively prime to Λ, the corresponding private exponent d is then given according to $ed \equiv 1 \pmod{\Lambda}$ (see [24]). The scheme remains exactly the same except that $m' = s^2 \bmod n$ must be replaced by $m' = s^e \bmod n$ in the verification stage.

In that setting, Proposition 1 becomes

Proposition 2. *Let $n = pq$, where p, q are distinct primes and $p, q \equiv 3 \pmod 4$ and let e, d such that e is even, $\gcd(e, \Lambda) = 1$ and $ed \equiv 1 \pmod{\Lambda}$. If $(z|n) = -1$, then*

$$\gcd\bigl(z^{ed} \mp z \pmod n, n\bigr) = p \text{ or } q \ .$$

Proof. From $ed \equiv 1 \pmod{\Lambda}$, we deduce that $ed \equiv 1 \pmod{(p-1)/2}$ and so there exists $\gamma \in \mathbb{Z}$ such that $ed = \gamma \frac{p-1}{2} + 1$. Further, since $p \equiv 3 \pmod 4$, $(p-1)/2$ is odd. Hence, γ must be odd since ed is even. Consequently, $z^{ed} \equiv z^{\gamma(p-1)/2}\, z \equiv (z|p)^{\gamma}\, z \equiv (z|p)\, z \equiv \pm z \pmod p$ and similarly $z^{ed} \equiv (z|q)\, z \equiv -(z|p)\, z \equiv \mp z \pmod q$, which completes the proof. □

Since e is even, we can write $e = 2e_1$. It is here worth remarking that anyone can raise an element to the ${e_1}^{\text{th}}$ power (modulo n). So, if an adversary follows Steps (I)–(VI) as described in Section 3, she can recover the factors of n by computing

$$\gcd(S_\tau - \widehat{r}_\tau \pmod n, n)\,, \tag{12}$$

where

$$\begin{aligned} S_\tau := {s_\tau}^{e_1} \bmod n \equiv ({\widehat{m}_\tau}^{d})^{e_1} &\equiv \prod_{1 \le i \le \ell} \prod_{1 \le j \le B} {p_j}^{(\beta_i v_{j,i} - 4\gamma_j) d e_1} \\ &\equiv \prod_{1 \le i \le \ell} {s_i}^{\beta_i e_1} \prod_{1 \le j \le B} {p_j}^{-2\gamma_j} \pmod n \ . \end{aligned}$$

The forgery is thus no more expensive against a scheme with a large public exponent e than against the basic scheme with $e = 2$.

4.2 Higher degree schemes

Rabin-type schemes can be developed in any unique factorization domain. In [27], Williams presents a scheme, called M^3, with public exponent 3 using arithmetic in a quadratic number field. This scheme was later extended to cyclotomic fields by Scheidler and Williams in [21] where they also give a scheme with a public exponent 5. In [20], Scheidler modifies Williams' M^3 scheme so that it works with a larger class of primes. Finally, in [16], Loxton et al. give another cubic scheme; the main difference with [27] being its easy geometrical interpretation. In this paragraph, we will stick on this latter scheme because it most resembles the Rabin-Williams scheme presented in Section 2.

The scheme of Loxton et al. uses the ring of Eisenstein integers, namely $\mathbb{Z}[\omega]$, where $\omega = (-1+\sqrt{-3})/2$ is a primitive cube root of unity. Its correctness relies of the following lemma (compare it with Lemma 1).

Lemma 2. *Let $n = pq$, where p, q are distinct primes in $\mathbb{Z}[\omega]$, $3 \nmid \mathrm{N}n$ and $\mathrm{N}q \equiv 2\,\mathrm{N}p - 1 \pmod 9$. If $(z|n)_3 = 1$, then*

$$z^{(\mathrm{N}p-1)(\mathrm{N}q-1)/9} \equiv \left(\frac{z}{p}\right)_3^{(2/3)(\mathrm{N}p-1)} \pmod n \ .$$

Proof. See [16, Lemma 1]. □

Essentially, that scheme suggests to generate two primes $p, q \in \mathbb{Z}[\omega]$ such that $p \equiv 8 + 6\omega \pmod 9$ and $q \equiv 5 + 6\omega \pmod 9$ and to compute $n = pq$. The public key is n and the private key is $d = [(\mathrm{N}p - 1)(\mathrm{N}q - 1) + 9]/27$, where N denotes the norm. A message m is signed by computing $\widetilde{m} = R(m)$ (for an appropriate function R) and $\widehat{m} = (1-\omega)^{3-t}\,[(1-\omega)\widetilde{m} + 1]$, where $\omega^t = (\widetilde{m}|n)_3$; the signature is $s = \widehat{m}^d \bmod n$. The signature is then verified by cubing s, and if accepted, the message m is recovered.

We need an analogue of Proposition 1.

Proposition 3. *Let $n = pq$, where p, q are primes in $\mathbb{Z}[\omega]$, $\mathrm{N}p \equiv 7 \pmod 9$ and $\mathrm{N}q \equiv 4 \pmod 9$, and let $d = [(\mathrm{N}p-1)(\mathrm{N}q-1)+9]/27$. If $(z|n)_3 = \omega$ or ω^2, then*

$$\gcd\bigl(z^{3d} - \omega^k z \pmod n, n\bigr) = p \text{ or } q \ ,$$

for some $k \in \{0, 1, 2\}$.

Proof. We first note that $(\mathrm{N}p-1)/3 \equiv 2 \pmod 3$ and $(\mathrm{N}q-1)/3 \equiv 1 \pmod 3$. So, $z^{3d} \equiv z^{(\mathrm{N}p-1)(\mathrm{N}q-1)/9}\, z \equiv (z|p)_3^{(\mathrm{N}q-1)/3}\, z \equiv (z|p)_3\, z \pmod p$, and similarly $z^{3d} \equiv (z|q)_3^{(\mathrm{N}p-1)/3}\, z \equiv (z|q)_3^{\,2}\, z \pmod q$. The proposition now follows by observing that $(z|p)_3 \neq (z|q)_3^{\,2}$ because, by hypothesis, $(z|n)_3 = (z|p)_3\,(z|q)_3 = \omega$ or ω^2. □

From this proposition, we can mimic the forgery presented against the Rabin-Williams scheme. The adversary has to find $\beta_1, \ldots, \beta_\ell$ such that (a) for $1 \leq j \leq B$, $\sum_{1 \leq i \leq \ell} \beta_i \, v_{j,i} \equiv 0 \pmod 3$; and (b) $\sum_{1 \leq i \leq \ell} \beta_i \, \zeta_i \equiv 1, 2 \pmod 3$, where $3\zeta_i := (\sum_{i \in \mathcal{J}(n,B)} v_{j,i}) \bmod 9$ and $\mathcal{J}(n,B) := \{1 \leq j \leq B : p_j \text{ is prime in } \mathbb{Z}[\omega] \text{ and } (p_j|n)_3 = \omega \text{ or } \omega^2\}$. She then computes $\vec{V}_\tau = (v_{1,\tau}, \ldots, v_{B,\tau}) \in (\mathbb{Z}/9\mathbb{Z})^B$, where $v_{j,\tau} = \sum_{1 \leq i \leq \ell} \beta_i \, v_{j,i} - 9\gamma_j$ (for some $\gamma_j \in \mathbb{Z}$), $\widehat{r}_\tau = \prod_{1 \leq j \leq B} {p_j}^{v_{j,\tau}/3} \pmod n$ and $s_\tau = \prod_{1 \leq i \leq \ell} {s_i}^{\beta_i} \prod_{1 \leq j \leq B} {p_j}^{-3\gamma_j} \bmod n$. Finally, by computing

$$\gcd(s_\tau - \omega^k \, \widehat{r}_\tau \pmod n, n) \tag{13}$$

for some $k \in \{0, 1, 2\}$, she finds the factors of n.

5 Applications

As an application, we will analyze the consequences of the previously described forgeries (Section 3) when the Rabin-Williams signature scheme is employed with the PKCS #1 v2.0 message encoding method as specified in [4]. We note, however, that the PKCS #1 standard is not expressly intended for use with the Rabin-Williams scheme but rather with the plain RSA scheme.

In the second part, we will present an algorithm which reduces the complexity of the forgery from n to $\sqrt{n}$. This algorithm is not restricted to PKCS #1: it remains applicable *whatever the employed encoding message method*.

5.1 PKCS #1 encoding method

We only briefly review the PKCS #1 message encoding method and refer the reader to [4] for details. PKCS #1 supports signature schemes with appendix. (In such a scheme, the message must accompany the signature in order to verify the validity of the signature.) The set of message representatives is thus given by $\mathcal{M}_R = \{\widetilde{m} \in (\mathbb{Z}/n\mathbb{Z})^* : \widetilde{m} \equiv 10 \pmod{16}\}$ (cf. Remark 1).

Let m be the message being encoded into the message representative $\widetilde{m}$. First, a hash function is applied to m to produce the hash value $H = \mathrm{Hash}(m)$. Next, the hash algorithm identifier and the hash value H are combined into an ASN.1 value and DER-encoded (see [11] for the relevant definitions). Let T denote the resulting DER-encoding. Similarly to what is done in ISO 9796 [3], we concatenate the octet $\mathtt{0A}_{16}$ to obtain the data string $D = T \| \mathtt{0A}_{16}$ (the reason is to ensure that D, viewed as an integer, is congruent to 10 modulo 16). The encoded message EM is then formed by concatenating the block-type octet BT, the padding string PS, the $\mathtt{00}_{16}$ octet and the data string D; or schematically,

$$EM = \mathtt{00}_{16} \| BT \| PS \| \mathtt{00}_{16} \| D \,, \tag{14}$$

where BT is a single octet containing the value $\mathtt{00}_{16}$ or $\mathtt{01}_{16}$. Let $\|n\|$ and $\|D\|$ respectively denote the octet-length of modulus n and D. When $BT = \mathtt{00}_{16}$ then PS consists of $\|n\| - \|D\| - 3$ octets having value $\mathtt{00}_{16}$; when $BT = \mathtt{01}_{16}$ then PS consists of $\|n\| - \|D\| - 3$ octets having value $\mathtt{FF}_{16}$.

The message representative $\widetilde{m}$ is the integer representation of EM.

Remark 4. Three hash functions are recommended: SHA-1 [1], MD2 [5], and MD5 [6]. The default function is SHA-1; MD2 and MD5 are only recommended for compatibility with existing applications based on PKCS #1 v1.5.

In what follows, we will assume that SHA-1 is the used hash function. In that case, the data string D consists of $(15 + 20 + 1) = 36$ octets ($= 288$ bits).

Type 0 encoding With type 0, i.e., when $BT = \mathtt{00}_{16}$, the message representatives, $\widetilde{m}_i$, are 288-bit long, *whatever the size of the modulus*. We have seen that the effectiveness of our forgeries is related to the smoothness (modulo n) of the $\widehat{m}_i$'s that appear in Eq. (1), where $\widehat{m}_i = \widetilde{m}_i$ or $\widetilde{m}_i/2$ according to the value of $(\widetilde{m}_i|n)$. So, when $BT = \mathtt{00}_{16}$, each $\widehat{m}_i$ is at most a 288-bit integer and we can hope that they factor (in $\mathbb{Z}$) into small primes.

Type 1 encoding Type 1 encoding, i.e., $BT = \mathtt{01}_{16}$, is the recommended way to encode a message. In that case, the message representatives are longer. For a 1024-bit modulus, these are 1016-bit integers (the leading octet in EM is $\mathtt{00}_{16}$). Therefore, the $\widehat{m}_i$'s happen unlikely to be smooth. But, what we ultimately need is to find smooth $\widehat{m}_i$'s *modulo* n, that is, we need to find an $\widehat{M}_i \equiv \widehat{m}_i \pmod{n}$ so that $\widehat{M}_i$ is smooth as an element in $\mathbb{Z}/n\mathbb{Z}$ (cf. Footnote ([1])). As already noted in [8], if the 1024-bit modulus has the special form

$$n = 2^{1023} \pm t\,, \tag{15}$$

then it is easy to find an $\widehat{M}_i$ whose magnitude is comparable to that of t. Indeed, when $BT = \mathtt{01}_{16}$, the padding string is formed with $(128 - 36 - 3) = 89$ octets having value $\mathtt{FF}_{16}$. Therefore, considering the data string D_i as a 288-bit integer, we can write from Eq. (14), $\widetilde{m}_i = \{(2^8)^{89} + [(2^8)^{89} - 1]\}(2^8)^{37} + D_i$. Hence, $\widehat{m}_i = (2^{713} - 1)2^{296} + D_i$ or $(2^{713} - 1)2^{295} + D_i/2$ according to $(\widetilde{m}_i|n) = 1$ or -1. So, defining $M_i := 2^{g_i}\,\widehat{m}_i - n$ and setting $\widehat{M}_i \equiv 2^{-g_i}\,M_i \equiv \widehat{m}_i \pmod{n}$, an appropriate choice for g_i will remove the term $2^{713+296}$ (resp. $2^{713+295}$). Namely, we set

$$M_i = \begin{cases} 2^{14}\,\widehat{m}_i - n = 2^{14}\,D_i - 2^{310} \mp t & \text{if } (\widetilde{m}_i|n) = 1 \\ 2^{15}\,\widehat{m}_i - n = 2^{15}\,D_i - 2^{310} \mp t & \text{if } (\widetilde{m}_i|n) = -1 \end{cases}\,. \tag{16}$$

Hence, since a special modulus of the form (15) with a square-free t as small as 400 bits offers the same security as a regular 1024-bit modulus [14], the M_i's as given in Eq. (16) can already be as small as 400-bit integers. Consequently, hoping that the M_i's factor into small integers (in $\mathbb{Z}$),[3] $\widehat{M}_i \equiv 2^{-g_i}\,M_i \equiv \widehat{m}_i \pmod{n}$ will be smooth as elements of $\mathbb{Z}/n\mathbb{Z}$.

[3] The probability that a 400-bit integer is smooth is relatively small; but see § 3.3 for some possible alternatives.

5.2 Arbitrary encoding

For general ("form-free") message encoding methods and moduli, the message representatives have roughly the same length as the modulus. We now present a generic method which on input an arbitrary message representative outputs a "message equivalent" whose length has the size of $\sqrt{n}$.

As in [8], we observe that if we can find integers a_i and b_i so that a_i is smooth and

$$M_i := a_i\,\widehat{m}_i - b_i\,n \tag{17}$$

is smooth as well, then $\widehat{M}_i :\equiv {a_i}^{-1}\,M_i \equiv \widehat{m}_i \pmod{n}$ is also smooth as an element of $\mathbb{Z}/n\mathbb{Z}$.
Explicitly, let $a_i = (-1)^{u_{0,i}} \prod_{1\le j\le B} {p_j}^{u_{j,i}}$ and $M_i = (-1)^{w_{0,i}} \prod_{1\le j\le B} {p_j}^{w_{j,i}}$ be the prime factorizations of a_i and M_i; then, we have

$$\widehat{M}_i \equiv \widehat{m}_i \equiv (-1)^{v_{0,i}} \prod_{1\le j\le B} {p_j}^{v_{j,i}} \pmod{n}\,, \tag{18}$$

where $v_{j,i} = u_{j,i} - w_{j,i}$, as required.

A related problem has been addressed by de Jonge and Chaum in [9, § 3.1] (see also [10]): they describe a method to find *small* integers a_i and M_i satisfying Eq. (17). The "Pigeonhole Principle" (e.g., see [22, Chapter 30]) quantifies how small can be a_i and M_i.

Proposition 4 (Pigeonhole Principle). *Let two integers $n, \widehat{m} \in \mathbb{Z}$. For any positive integer $A < n$, there exist integers a^* and b^* such that*

$$0 < |a^*| \le A \quad \textit{and} \quad |a^*\widehat{m} - b^*n| < \lceil n/A \rceil \ .$$

Proof. Consider the $(A+1)$ "pigeons" given by the numbers $P_a := a\,\widehat{m} \bmod n$ $(0 \le a \le A)$, i.e., each pigeon is an integer between 0 and $(n-1)$. We now form the A pigeonholes given by the integer intervals

$$\mathcal{I}_a = \begin{cases} \big[(a-1)\lceil n/A\rceil, a\lceil n/A\rceil\big[& \text{for } 1 \le a \le (A-1) \\ \big[(A-1)\lceil n/A\rceil, n-1\big] & \text{for } a = A \end{cases} .$$

Since there are more pigeons than pigeonholes, two pigeons are sitting in the same hole. Suppose these are pigeons P_x and P_y and that they are in hole $\mathcal{I}_a$; in other words, $P_x, P_y \in \mathcal{I}_a \iff |P_x - P_y| < \lceil n/A\rceil \iff |(x-y)\widehat{m} \bmod n| < \lceil n/A\rceil$. Noting that $0 \le x, y \le A$ and $x \ne y$, we set $a^* = x - y$ and so $0 < |a^*| \le A$. Hence, $|a^*\widehat{m} \bmod n| < \lceil n/A\rceil$ and the lemma follows by setting $b^* = \lfloor a^*\widehat{m}/n \rfloor$. □

In particular, taking $A = \lceil \sqrt{n}\,\rceil$, there exist $a_i, b_i \in \mathbb{Z}$ such that $|a_i| \le \lceil \sqrt{n}\,\rceil$ and $|M_i| \le \big\lceil n/\lceil \sqrt{n}\,\rceil\big\rceil - 1 = \big\lfloor n/\lceil \sqrt{n}\,\rceil\big\rfloor \le \lfloor \sqrt{n} \rfloor$. This means that for a given 1024-bit modulus n and *any* $\widehat{m}_i$ (corresponding to the message representative $\widetilde{m})_i$), there are integers a_i and b_i such both a_i and $M_i = a_i\,\widehat{m}_i - b_i\,n$ are 512-bit integers, *whatever the message encoding method*. It remains to explain how to find a_i and b_i. A simple means is given by the extended Euclidean algorithm [12, Algorithm X, p. 325].

Algorithm 2. On inputs n and $\widehat{m}_i \in \mathbb{Z}/n\mathbb{Z}$, this algorithm computes a solution a_i, b_i, M_i satisfying Eq. (17) so that $|a_i|$ and $|M_i|$ are $\leq \lceil \sqrt{n} \rceil$. It makes use of a vector (u_1, u_2, u_3) such that $u_1 \widehat{m}_i - u_2 n = u_3$ always holds.

1. [initialization] Set $(u_1, u_2, u_3) \leftarrow (0, -1, n)$ and $(v_1, v_2, v_3) \leftarrow (1, 0, \widehat{m}_i)$.
2. [Euclid] Compute $Q \leftarrow \lfloor u_3/v_3 \rfloor$ and $(t_1, t_2, t_3) \leftarrow (u_1, u_2, u_3) - (v_1, v_2, v_3)Q$. Then set $(u_1, u_2, u_3) \leftarrow (v_1, v_2, v_3)$ and $(v_1, v_2, v_3) \leftarrow (t_1, t_2, t_3)$.
3. [loop] If $u_3 > \lceil \sqrt{n} \rceil$, then return to Step 2.
4. [output] Output $a_i \leftarrow u_1$, $b_i \leftarrow u_2$ and $M_i \leftarrow u_3$.

Example 2. Using the modulus of Example 1 (i.e., $n = 26882749$), suppose we are given a message m_i whose encoding is $\widetilde{m}_i = 26543210$ ($\equiv 10 \pmod{16}$). Since $(\widetilde{m}_i | n) = -1$, we have $\widehat{m}_i = \widetilde{m}_i/2 = 13271605 = 5 \cdot 79 \cdot 33599$. This $\widehat{m}_i$ is not smooth, we thus apply Algorithm 2 and obtain $M_i = a_i \widehat{m}_i + b_i n$ with $a_i = 1821$, $b_i = 899$, $M_i = 1354$. We have $a_i = 3 \cdot 607$ and $M_i = 2 \cdot 677$. Hence, $\widehat{m}_i \equiv 2 \cdot 3^{-1} \cdot 607^{-1} \cdot 677 \pmod{n}$.

u_1 (a_i)	u_2 (b_i)	u_3 (M_i)
1	0	13271605
−2	−1	339539
79	*39*	*29584*
−871	−430	14115
1821	*899*	*1354*
−19081	−9420	575
39983	19739	204
−99047	−48898	167
139030	68637	37
−655167	−323446	19
794197	392083	18
−1449364	−715529	1

Note that Algorithm 2 does not always give the best possible solution. Considering all the steps of the extended Euclidean algorithm (see above), the best solution for our example is given by $a_i = 79$ and $M_i = 29584 = 2^4 \cdot 43^2$ which yields $\widehat{m}_i \equiv 2^4 \cdot 43^2 \cdot 79^{-1} \pmod{n}$. ♢

6 Conclusion

This paper presented a specialized version of the Coron-Naccache-Stern signature forgery. It applies to *any* Rabin-type signature scheme and to *any* (even) public verification exponent. Furthermore, contrary to the case of RSA, the forgery is *universal*: it yields the value of the private key.

References

1. FIPS 180-1. Secure Hash Standard. Federal Information Processing Standards Publication 180-1, U.S. Department of Commerce, April 1995.
2. IEEE Std 1363-2000. IEEE Standard Specifications for Public-Key Cryptography. IEEE Computer Society, August 29, 2000.
3. ISO/IEC 9796. Information technology – Security techniques – Digital signature scheme giving message recovery, 1991.
4. PKCS #1 v2.0. RSA cryptography standard. RSA Laboratories, October 1, 1998. Available at http://www.rsasecurity.com/rsalabs/pkcs/.
5. RFC 1319. The MD2 message digest algorithm. Internet Request for Comments 1321, Burt Kaliski, April 1992. Available at http://www.ietf.org/rfc/rfc1319.txt.
6. RFC 1321. The MD5 message digest algorithm. Internet Request for Comments 1321, Ronald Rivest, April 1992. Available at http://www.ietf.org/rfc/rfc1321.txt.
7. Henri Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, 1993.
8. Jean-Sébastien Coron, David Naccache, and Julien P. Stern. On RSA padding. In M. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 1999.
9. Wiebren de Jonge and David Chaum. Attacks on some RSA signatures. In H. C. Williams, editor, *Advances in Cryptology – CRYPTO '85*, volume 218 of *Lecture Notes in Computer Science*, pages 18–27, 1986.
10. Marc Girault, Philippe Toffin, and Brigitte Vallée. Computation of approximate L-th root modulo n and application to cryptography. In S. Goldwasser, editor, *Advances in Cryptology – CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 110–117, 1990.
11. Burton S. Kaliski Jr. A layman's guide to a subset of ASN.1, BER, and DER. RSA Laboratories Technical Note, RSA Laboratories, November 1993. Available at http://www.rsasecurity.com/rsalabs/pkcs/.
12. Donald E. Knuth. *The Art of Computer Programming, v. 2. Seminumerical Algorithms*. Addison-Wesley, 2nd edition, 1981.
13. Kaoru Kurosawa, Toshiya Itoh, and Masashi Takeuchi. Public key cryptosystem using a reciprocal number with the same intractability as factoring a large number. *Cryptologia*, 12(4):225–233, 1988.
14. Arjen K. Lenstra. Generating RSA moduli with a predetermined portion. In K. Ohta and D. Pei, editors, *Advances in Cryptology – ASIACRYPT '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 1–10. Springer-Verlag, 1998.
15. Arjen K. Lenstra and Mark S. Manasse. Factoring with two large primes. *Mathematics of Computation*, 63:785–798, 1994.
16. J. H. Loxton, David S. Khoo, Gregory J. Bird, and Jennifer Seberry. A cubic RSA code equivalent to factorization. *Journal of Cryptology*, 5(2):139–150, 1992.
17. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
18. Peter L. Montgomery. A block Lanczos algorithm for finding dependencies over GF(2). In L. C. Guillou and J.-J. Quisquater, editors, *Advances in Cryptology – EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pages 106–120, 1995.
19. Michael O. Rabin. Digitized signatures and public-key functions as intractable as factorization. Technical Report LCS/TR-212, M.I.T. Lab. for Computer Science, January 1979.

20. Renate Scheidler. A public-key cryptosystem using purely cubic fields. *Journal of Cryptology*, 11(2):109–124, 1998.
21. Renate Scheidler and Hugh C. Williams. A public-key cryptosystem utilizing cyclotomic fields. *Designs, Codes and Cryptography*, 6:117–131, 1995.
22. Joseph H. Silverman. *A Friendly Introduction to Number Theory*. Prentice-Hall, 1997.
23. Robert D. Silverman and David Naccache. Recent results on signature forgery, April 1999. Available at http://www.rsasecurity.com/rsalabs/bulletins/sigforge.html.
24. Hugh C. Williams. A modification of the RSA public-key encryption procedure. *IEEE Transactions on Information Theory*, IT-26(6):726–729, 1980.
25. ———. Some public-key crypto-functions as intractable as factorization. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology – Proceedings of CRYPTO 84*, volume 196 of *Lecture Notes in Computer Science*, pages 66–70. Springer-Verlag, 1986.
26. ———. Some public-key crypto-functions as intractable as factorization. *Cryptologia*, 9(3):223–237, 1985. An extended abstract appears in [25].
27. ———. An M^3 public key encryption scheme. In H. C. Williams, editor, *Advances in Cryptology – CRYPTO '85*, volume 218 of *Lecture Notes in Computer Science*, pages 358–368. Springer-Verlag, 1986.