Evaluation of Cryptographic Techniques

Jean-Sebastien Coron

Gemplus Card International
34 rue Guynemer, 92447 Issy-les-Moulineaux, France
{jean-sebastien.coron}@gemplus.com

Abstract. This report evaluates the security of RSA signature schemes PKCS#1 v1.5, ANSI X9.31, ISO 9796-1 and ISO 9796-2. We investigate the hash format used in those signature schemes. This is as required by Cryptrec.

1 Introduction

RSA was invented in 1977 by Rivest, Shamir and Adleman [24], and is now the most widely used public-key cryptosytem.

A very common practice for signing with RSA is to first hash the message, add some padding, and then raise the result to the power of the decryption exponent. This paradigm is the basis of numerous standards such as PKCS#1 v1.5, ANSI X9.31, ISO 9796-1 and ISO 9796-2.

A signature scheme is said to be secure if it is infeasible to produce a valid signature of a message without knowing the private key. This task should remain infeasible even if the attacker can obtain the signature of any message of his choice. This security notion was formalized by Goldwasser, Micali and Rivest in [12] and called *existential unforgeability under an adaptive chosen message attack*. It is the strongest security notion for a signature scheme and it is now considered as standard. Formally, this notion captures the property that an attacker cannot produce a valid signature, even after obtaining the signature of (polynomially many) messages of his choice.

In this report, we investigate the security of RSA signature schemes PKCS#1 v1.5, ANSI X9.31, ISO 9796-1 and ISO 9796-2. We investigate the possibility of forging signatures when using those standards. Moreover, in some cases, the user is free to choose some parameters. We determine which choice of parameters gives a secure signature scheme.

A significant line of research in cryptography consists in proving the security of cryptosystems. A proof of security is usually a computational reduction from solving a well established problem to breaking the cryptosystem. In our case, breaking the cryptosystem means forging signatures. Well established problems of cryptographic relevance include factoring large integers, computing discrete logarithms in prime order groups, or extracting roots modulo a composite integer. In our case, the underlying problem consists in factoring integers, or inverting the RSA function.

A security proof provides a strong guarantee for the security of a RSA-based signature scheme: the signature scheme is secure, unless inverting RSA is easy (or factoring is easy), which seems unlikely. Since [12], many signature schemes have been proven secure, such as PSS [2].

In this report, we also investigate the security of RSA signature schemes PKCS#1 v1.5, ANSI X9.31, ISO 9796-1 and ISO 9796-2 from the point of view of security proofs. We investigate the possibility of obtaining a security proof for those standards.

2 Attacks against RSA signature schemes

In this section, we review the most significant attacks against RSA signature schemes. The application of those attacks to the standards PKCS#1 v1.5, ANSI X9.31, ISO 9796-1 and ISO 9796-2 will be studied in section 3. We denote by $\mu(m)$ the encoding function of the message m. The signature of m is then:

$$s = \mu(m)^d \mod N$$

where N is the RSA modulus and d the private exponent.

First, we review the attacks against RSA signature with fixed-pattern padding, without using a hash function. Since the standards PKCS#1 v1.5, ANSI X9.31, ISO 9796-1 and ISO 9796-2 use a hash function, those attacks are not directly applicable. The possibility of applying these attacks to those standards will be investigated in section 3.

In this model, to sign a message m, the signer concatenates a fixed padding P to the message, and the signature is obtained by computing:

$$s = (P|m)^d \mod N$$

where d is the private exponent and N the modulus.

More generally, we consider RSA signatures in which a simple affine redundancy is used. To sign a message m, the signer first computes:

$$R(m) = \omega \cdot m + a \quad \text{where} \begin{cases} w \text{ is the multiplicative redundancy} \\ a \text{ is the additive redundancy} \end{cases} \tag{1}$$

The signature of m is then:

$$s = R(m)^d \mod N$$

A left-padded redundancy scheme $P|m$ is obtained by taking $\omega = 1$ and $a = P \cdot 2^\ell$, whereas a right-padding redundancy scheme $m|P$ is obtained by taking $\omega = 2^\ell$ and $a = P$.

2.1 De Jonge and Chaum attack against RSA signature with linear redundancy

At Crypto '85, De Jonge and Chaum [9] exhibited a multiplicative attack against RSA signatures with affine redundancy, based on the extended Euclidean algorithm. Their attack applies when the multiplicative redundancy ω is equal to one and the size of the message is at least two-thirds of the size of the RSA modulus N.

$$|\text{message}| \succ \frac{2}{3}|N|$$

For example, a signature can be forged if one uses the affine redundancy of figure 1.

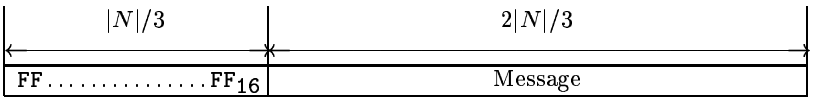

Fig. 1. Example of an RSA padding forgeable by De Jonge and Chaum's method where $\omega = 1$ and $a = \texttt{FF}\ldots\texttt{FF}\ \texttt{00}\ldots\texttt{00}_{16}$

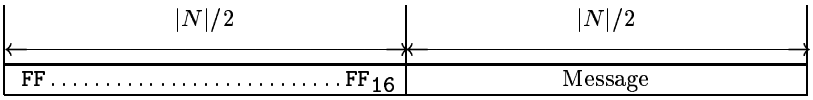

Fig. 2. Example of an RSA padding forgeable by Girault and Misarsky's method where $\omega = 1$ and $a = \texttt{FF}\ldots\texttt{FF}\ \texttt{00}\ldots\texttt{00}_{16}$

2.2 Girault and Misarsky's attack

De Jonge and Chaum's attack was extended by Girault and Misarsky [10] at Eurocrypt '97, using Okamoto-Shiraishi's algorithm [22], which is an extension of the extended Euclidean algorithm. They increased the field of application of multiplicative attacks on RSA signatures with affine redundancy as their attack applies to any value of ω and a, when the size of the message is at least half the size of the modulus (refer to figure 2 for an illustration):

$$|\text{message}| \succ \frac{1}{2}|N|$$

Girault and Misarsky also extended the multiplicative attacks to RSA signatures with modular redundancy:

$$R(m) = \omega_1 \cdot m + \omega_2 \cdot (m \mod b) + a \quad \text{where} \begin{cases} w_1, w_2 & \text{is the multiplicative redundancy} \\ a & \text{is the additive redundancy} \\ b & \text{is the modular redundancy} \end{cases} \tag{2}$$

In this case, the size of the message must be at least half the size of the modulus plus the size of the modular redundancy.

2.3 Misarsky's attack

Girault and Misarsky's attack was extended by Misarsky [20] at Crypto '97 to a redundancy function in which the message m and the modular redundancy $m \mod b$ can be split into different parts, using the LLL algorithm [18]. The attack applies when the size of the message is at least half the size of the modulus plus the size of the modular redundancy.

2.4 Brier, Clavier, Coron, Naccache's attack

This is an extension of Girault and Misarsky's attack against RSA signatures with affine redundancy to messages of size as small as one third of the size of the modulus, as illustrated in figure 3.

$$|\text{message}| \succ \frac{1}{3}|N|$$

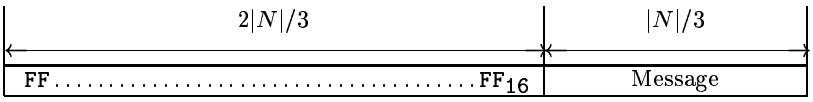

Fig. 3. Example of an RSA padding forgeable where the ω is equal to one and $a = \texttt{FF}\ldots\texttt{FF}\ \texttt{00}\ldots\texttt{00}_{16}$

As Girault and Misarsky's attack, the attack applies for any w and a and runs in polynomial time. However, the attack is existential only, as one cannot choose the message the signature of which is forged, whereas Girault and Misarsky's attack is selective: it is possible to choose the message which signature is forged.

We briefly review the attack. We give a slightly different exposition, which, in our opinion, is simpler than [3]. The attack looks for four distinct messages x, y, z and t, each as small as one third of the size of the modulus, such that:

$$(P + x) \cdot (P + y) = (P + z) \cdot (P + t) \mod N \tag{3}$$

which gives:

$$P \cdot (x + y - z - t) + x \cdot y - z \cdot t = 0 \mod N$$

First, one obtains two integers u and v such that

$$P \cdot u + v = 0 \mod N \text{ with } \begin{cases} 0 < u < N^{\frac{1}{3}} \\ 0 < v < 2 \cdot N^{\frac{2}{3}} \end{cases}$$

As noted in [11], this is equivalent to finding a good approximation of the fraction P/N, and can be done efficiently by developing it in continued fractions, *i.e.* applying the extended Euclidean algorithm to P and N. Now we try to solve the system:

$$\begin{cases} x + y - z - t = u \\ xy - zt = v \end{cases} \tag{4}$$

Solving for x and z, this gives:

$$x = \frac{v - tu}{y - t} + t \tag{5}$$

$$z = \frac{v - yu}{y - t} + y \tag{6}$$

We denote $\Delta = y - t$. For a solution to exist, the integer $v - tu$ must be divisible by $y - t$, which gives $v - tu = 0 \mod \Delta$, which also implies $v - yu = 0 \mod \Delta$. Therefore, we select a random prime Δ less than $N^{1/3}$; then we let $t = v \cdot (u^{-1}) \mod \Delta$ and let $y = \Delta + t$. We let x and z as in (5) and (6). We obtain four integers x, y, z, and t, each of size one third of the size of the modulus, such that:

$$(P + x)(P + y) = (P + z)(P + t) \mod N$$

which enables to forge the signature of message x as:

$$(P + x)^d = (P + z)^d (P + t)^d / (P + y)^d \mod N$$

An example of forgery with a 1024-bit modulus is given in [3]. The attack complexity is polynomial in the size of N.

2.5 Lenstra and Shparlinski's attack

The attack, described in [19], is an extension of the previous attack to selective forgeries. The extension was announced in [3], but the attack was not described in details. The new attack is selective in that the message to be forged can be fixed in advance. However, the attack complexity is no longer polynomial.

The technique is the following. Assume that t is fixed in advance. Using the same notations as previously, we must find y such that $v - t \cdot u = 0 \mod [y - t]$. This is done by computing the factorization of $v - t \cdot u$. If $v - t \cdot u$ factors into the product of two integers α and β of roughly the same size, we can take $y = \alpha + t$ and we obtain $v - t \cdot u = 0 \mod [y - t]$, which gives a forgery as previously.

However, the attack succeeds only if $v - t \cdot u$ is the product of two integers of roughly the same size, which happens with small probability. Moreover, a factorization algorithm must be used, which explains why the attack is no longer polynomial. The technique described in [19] consists in generating various couples (u_k, v_k) such that $P \cdot u_k + v_k = 0 \mod N$ and trying to factor $v_k - t \cdot u_k$ into the product of two integers of roughly the same size. It is shown in [19] that this can be done in heuristic asymptotic runtime:

$$\exp\left((1 + \circ(1))(\log N)^{1/3}(\log\log N)^{2/3}\right)$$

In [19] is given an example of selective forgery for a 1024-bit RSA modulus.

In the previous sections, we have considered RSA signature schemes with linear redundancy. We have seen that the most efficient attacks were Brier, Clavier, Coron and Naccache's attack which enables to make an existential forgery in polynomial time, and Lenstra and Shparlinski's attack which enables to make a selective forgery in sub-exponential time.

However, those attacks do not apply to hash-based signature schemes such as PKCS#1 v1.5, ANSI X9.31, ISO 9796-1 and ISO 9796-2. In the following, we consider attacks against RSA signature schemes using a hash function.

2.6 Desmedt and Odlyzko's attack

This attack is described in [21] and applies to RSA signature schemes in which a hash function is used. Let m be the message to be signed. The goal is to obtain $\mu(m)^d \mod N$ without knowing d.

1. Factor $\mu(m)$ into the product of small primes p_i only.

2. Obtain the values $p_i^d \mod N$ by combining the signatures of messages m_j for which $\mu(m_j)$ is the product of small primes only.

3. Obtain the signature of m by multiplying the values $p_i^d \mod N$ where p_i is a small prime factor of $\mu(m)$.

The attack complexity depends on the size of $\mu(m)$. The attack only applies for small sizes of $\mu(m)$ (otherwise, the probability that $\mu(m)$ is the product of small primes only is too small).

2.7 Coron, Naccache, Stern's attack

The attack, describes in [5], is an extension of Desmedt and Odlyzko's attack. It applies to the ISO 9796-1 and ISO 9796-2 standards. The attack applies to the case in which it is possible to find constants a and b such that

$$t = a \cdot \mu(m) + b \cdot N$$

is small, or when it is possible to find a constant c such that $\mu(m)$ can be written as:

$$\mu(m) = c \cdot t$$

where t is a small integer. By taking $c = a^{-1} \mod N$, one can always consider the case:

$$\mu(m) = c \cdot t \mod N$$

where t is a small integer.

The attack consists in obtaining many messages m_i such that the integer t_i in

$$\mu(m_i) = c \cdot t_i \mod N \tag{7}$$

is y-smooth, where y is a parameter. An integer is said to be y-smooth if all his prime factors are less than y. We denote by $(p_1, \ldots, p_k)$ the list of all prime factors smaller than y. We can write:

$$\mu(m_i) = c \cdot \prod_{j=1}^{k} p_j^{v_{i,j}} \mod N \quad \text{for } 1 \leq i \leq \tau$$

To each $\mu(m_i)$ we associate a $k+1$-dimensional vector $\boldsymbol{V}_i$:

$$\mu(m_i) \longmapsto \boldsymbol{V}_i = \{1, v_{i,1} \mod e, \ldots, v_{i,k} \mod e\}$$

One tries to express one vector $\boldsymbol{V}_\tau$ as a linear combination of the others, by Gaussian elimination:

$$\boldsymbol{V}_\tau = \sum_{i=1}^{\tau-1} \beta_i \boldsymbol{V}_i \mod e \tag{8}$$

From (8) one can write:

$$v_{\tau,j} = \sum_{i=1}^{\tau-1} \beta_i \cdot v_{i,j} - \gamma_j \cdot e \quad \text{for all} \quad 1 \leq j \leq k$$

and denoting:

$$\delta = \prod_{j=1}^{k} p_j^{-\gamma_j}$$

we obtain:

$$\mu(m_\tau) = \delta^e \cdot \prod_{i=1}^{\tau-1} \mu(m_i)^{\beta_i} \mod N$$

Thus, the attacker will ask for the signature of the $\tau - 1$ first messages m_i and forge the signature of m_τ with:

$$\mu(m_\tau)^d = \delta \cdot \prod_{i=1}^{\tau-1} \left(\mu(m_i)^d\right)^{\beta_i} \mod N$$

The attack complexity depends on the probability that the integers t_i are y-smooth. Defining $\psi(x, y) = \#\{v < x, \text{ such that } v \text{ is } y\text{-smooth}\}$, it is known [8] that, for large x, the ratio $\psi(x, \sqrt[t]{x})/x$ is equivalent to Dickman's function defined by :

$$\rho(t) = \begin{cases} 1 & \text{if } 0 \leq t \leq 1 \\ \rho(n) - \displaystyle\int_n^t \frac{\rho(v-1)}{v} dv & \text{if } n \leq t \leq n+1 \end{cases}$$

$\rho(t)$ is thus an approximation of the probability that a u-bit number is $2^{u/t}$-smooth. In particular, denoting:

$$y = L_x[\beta] = \exp\left(\beta \cdot \sqrt{\log x \log\log x}\right)$$

the probability that an integer between one and x is $L_x[\beta]$-smooth is:

$$\frac{\psi(x, y)}{x} = L_x\left[-\frac{1}{2\beta} + \circ(1)\right]$$

If we assume that the integers t_i in (7) are uniformly distributed between one and x, we have to generate on average $L_x[1/(2\beta) + \circ(1)]$ integers t_i.

Using the ECM factorization algorithm [17], a prime factor p of an integer n is extracted in time:

$$L_p[\sqrt{2} + \circ(1)]$$

A y-smooth integer can thus be factorized in time:

$$L_y[\sqrt{2} + \circ(1)] = L_x[\circ(1)]$$

The complexity to find an integer t_i which is y-smooth using ECM is thus:

$$L_x\left[\frac{1}{2\beta} + \circ(1)\right]$$

Moreover, the number τ of integers which are necessary to find a vector which is a linear combination of the others is $\mathcal{O}(y \cdot \log e)$ (see [5] for more details). Therefore, one must solve a system with $r = L_x[\beta + \circ(1)]$ equations in $r = L_x[\beta + \circ(1)]$ unknown. Using Lanzos iterative algorithm [16], the time required to solve this system is $\mathcal{O}(r^2)$ and the space required is roughly $\mathcal{O}(r)$. To summarize, the time required to obtain the $L_x[\beta + \circ(1)]$ necessary equations is

$$L_x\left[\beta + \frac{1}{2\beta} + \circ(1)\right]$$

This system is solved in time

$$L_x[2\beta + \circ(1)]$$

and space

$$L_x[\beta + \circ(1)]$$

The complexity is minimal by taking $\beta = 1/\sqrt{2}$. We obtain a time complexity

$$L_x[\sqrt{2} + \circ(1)]$$

and space complexity:

$$L_x\left[\frac{\sqrt{2}}{2} + \circ(1)\right]$$

The complexity is sub-exponential in the size of the integers t_i. Therefore, the attack will be practical only if we can obtain small t_i.

In the following table, we give the values of the functions $L_x[\sqrt{2}]$ et $L_x[\sqrt{2}/2]$ corresponding to the time complexity and space complexity of the attack, as a function of the size $|x|$ of the integer t_i. This table should be handled with care: this is just an approximation of the attack practical complexity, and the attack may take more time in practice. The table suggests that the attack can be practical when the size of t_i is smaller than 128 bits, but the attack becomes unpractical for larger sizes.

$\|x\|$	$\log_2$ **time**	$\log_2$ **space**
64	26	13
96	34	17
128	41	20
192	52	26
256	62	31
368	77	38

Table 1. Attack complexity

3 Application to existing standards

In this section, we discuss the application of the previous attacks to the existing standards.

3.1 ISO 9796-1

The signature scheme ISO/IEC-9796-1 [14] has been published in 1991. The standard enables message recovery. For a modulus N of size $2\gamma + 1$ bits and a message m of size γ bits, assuming that γ is divisible by 8, the encoding of m is defined as follows:

We denote by ω_i the i-th nibble of m. We let $\ell = \gamma/4$. We denote by $s(x)$ the substitution:

$x =$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$s(x) =$	E	3	5	8	9	4	2	F	0	D	B	6	7	A	C	1

Letting $\bar{s}(x)$ force the most significant bit in $s(x)$ to 1 and $\tilde{s}(x)$ complement the least significant bit of $s(x)$, the standard ISO 9796-1 specifies :

$$\begin{aligned}\mu(m) = \; & \bar{s}(\omega_{\ell-1})\; \tilde{s}(\omega_{\ell-2})\; \omega_{\ell-1}\; \omega_{\ell-2} \\ & s(\omega_{\ell-3})\; s(\omega_{\ell-4})\; \omega_{\ell-3}\; \omega_{\ell-4} \\ & \ldots \\ & s(\omega_3) \quad s(\omega_2) \quad \omega_3 \quad \omega_2 \\ & s(\omega_1) \quad s(\omega_0) \quad \omega_0 \quad 6_{16}\end{aligned}$$

In [5] is described an attack against a variant of ISO 9796-1, in which $\tilde{s}(x)$ is replaced by $s(x)$. This variant differs from the real ISO 9796-1 by one single bit. The attack is an application of Coron, Naccache and Stern's attack of the previous section.

The attack of [5] was extended by Coppersmith, Halevi and Jutla [4] to the real ISO 9796-1 standard. In the following, we give a practical forgery against ISO 9796-1using this technique (no practical forgery was given in [4]). The forgery is given for a 1025-bit modulus with $e = 3$. Let denote the 97-bit constant $\Gamma = 1001001$, where each digit represents a 16-bit word.

Step 1 :. Let x_i be the following integers, with $1 \leq i \leq 273$:

$$\begin{aligned} a_i &= \bar{s}(u_{i,1})\; \tilde{s}(u_{i,2})\; u_{i,1}\; u_{i,2} \\ b_i &= s(u_{i,3})\; s(u_{i,4})\; u_{i,3}\; u_{i,4} \\ c_i &= s(u_{i,5})\; s(u_{i,6})\; u_{i,5}\; u_{i,6} \\ d_i &= s(u_{i,7})\; s(u_{i,8})\; u_{i,8}\; 6 \\ x_i &= a_i\; b_i\; c_i\; d_i \end{aligned}$$

where message$[i] = u_{i,1}\, u_{i,2}\, u_{i,3}\, u_{i,4}\, u_{i,5}\, u_{i,6}\, u_{i,7}\, u_{i,8}$ is given by the following table:

113C2789	2103E5FE	213488FE	215041FE	21A1F6FE	23979965	23A9DF65	26013565	26182D65	261B3865
26235B65	26729D65	26EB1465	30157C81	3038C281	304D5B81	30CF6581	34045BF1	340AC4F1	34596BF1
34B660F1	34E1B0F1	34FF49F1	3814BA6A	38585D6A	3873976A	38A9396A	38E2F86A	38EEE56A	385192BD
3854A9BD	3882F7BD	389E88BD	38BB52BD	3A16E425	3A3C6125	3A797525	3A9B4E25	3AB30125	3ABFBC25
3AD30A25	3D12D3F9	3D6C4AF9	3D8AF3F9	3D91E4F9	3D9E3BF9	3DD521F9	3DE363F9	3DEDAFF9	3F09D025
3F198D25	3F3DFC25	3FCE9B25	410AB2F9	4122BDF9	412F08F9	413EDBF9	41C584F9	41EE50F9	41F296F9
4345DC55	43486155	4372C655	43793F55	4385E655	43EE7B55	4617F255	4627D755	463CF255	4665D455
468AA555	46DB9055	484B4E1A	488ED71A	48E4B91A	48EE6D1A	4A55A165	4A6F6565	4A77DA65	4A905D65
4AC74265	4AEE8465	4D069469	4D147369	4D31AB69	4D420C69	4D499369	4D532169	4D56A869	4D758769
4D84EE69	4DD22969	4F2BF565	4F2C2665	4F758F65	4FA5A565	4FD7BD65	51C43089	51DA7A89	51E7E789
590CC262	59733762	59F54062	5B07E9FA	5B9EFDFA	5BBC4BFA	5BDC93FA	5BFCCEFA	5E062FFA	5E157DFA
5E4550FA	5E7CB6FA	5E963AFA	5ED3F8FA	6015AF51	60326151	60372751	604F6B51	60708951	607F0B51
60931F51	60D7FF51	6297391A	6486D321	6496D721	64F0D121	6758901A	675ED11A	67F7F31A	6C3FB8F7
6C9916F7	6CAA47F7	6CD886F7	806BD551	806F2D51	80A83051	831D3465	833A6E65	837B2565	837F0865
83B16265	83DA9C65	840FAF21	84149621	84704721	84802A21	84A25A21	84F1E221	84FDA321	858D66B8
85EB0BB8	861A4765	8634B865	866AB865	868D6165	86AC2F65	891EF962	89220762	892C2662	893ABD62
8950EA62	89CFD062	89DA4562	8A049B55	8A27EF55	8A32DF55	8A489755	8A523055	8A7F9955	8AB3CA55
8AD3AD55	8AF88555	8DA35BBE	8DC6B0BE	8DDAC3BE	8F1F7855	8F5F5F55	8FC42755	8FEC2655	913BD36E
9158BF6E	9199DF6E	91B4856E	91D1546E	91E5696E	A0B92266	A0BA2B66	A4401E16	A4DFFF16	A4ED5A16
A4F64416	A8668A5D	AD0C6EFE	AD8124FE	ADB3D7FE	ADC5A6FE	ADDAF5FE	D00806F1	D07D68F1	D0D26DF1
D0DDC2F1	D20C395A	D25CE85A	D278785A	D2B6C25A	D2BF0D5A	D2E44D5A	D400B761	D41E1961	D4732D61
D494FC61	D4A85061	D79B1B5A	D79FAA5A	D801D7FD	D815D2FD	D868D1FD	D8F292FD	EA43E961	EA485761
EA4E1261	EB355C8A	EB37F78A	EB73DA8A	EED7308A	EEDBF58A	EEE9118A	EF784561	EF7CB861	EF8FDE61
F10F04FE	F146DAFE	F18C0CFE	F196ACFE	F1B831FE	F1CFA5FE	F1D371FE	F269861A	F26A251A	F28A8D1A
F32E2E21	F3369421	F3EB6821	F52952B8	F55C47B8	F5CC08B8	F6202521	F64ABA21	F6683921	F684CE21
F6DE0521	F6F67621	F7BDBD1A	F7D0F01A	F7D2411A	F7F60F1A	FB6E9AFA	FBA2B8FA	FBF809FA	FC8BA450
FCBC2050	FCD65150	FCEFE550	FD705E6E	FDBACE6E	FDE3756E	FE0395FA	FE0F38FA	FE0FABFA	FE2ECFFA
FE56C3FA	FE9C2EFA	FEEFA7FA							

We obtain $M_i = \Gamma \cdot x_i$, which is a valid encoding for a message m_i, such that $M_i = \mu(m_i)$.

Step 2 : Obtain the 272 signatures $s_i = \mu_{\text{ISO}}(m_i)^d \mod N$ for $1 \leq i \leq 272$.

Step 3 : The signature of m_{273} is given by:

$$\mu(m_{273})^d = \Gamma^{-139} \prod_{i=1}^{587} p_i^{-\mathtt{g}[i]} \prod_{i=1}^{272} s_i^{\mathtt{b}[i]} \mod N$$

where p_i is the i-th prime and $\mathtt{b}[i]$ is given by the following table:

2	2	1	2	1	2	2	2	2	1	2	2	2	1	1	1	2	1	2	1	1	2	2	1	1	1
2	2	2	1	2	1	1	2	2	1	2	1	1	2	1	2	2	2	1	2	2	2	2	2	1	2	2
1	2	1	1	1	2	2	1	1	2	1	2	2	2	2	1	2	1	2	2	2	2	2	1	1	1
1	1	1	1	2	1	1	2	1	2	2	2	1	2	1	1	1	2	1	1	2	1	2	2	2	1
1	1	2	1	1	2	1	1	2	2	1	1	2	1	1	1	2	1	2	2	2	2	2	1	2	2
1	2	2	2	1	1	2	2	1	2	1	1	1	1	2	2	2	2	1	1	2	2	1	2	1	2
2	2	2	1	1	2	1	2	2	2	1	1	1	1	1	2	2	1	1	1	2	1	2	2	2	2
1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	2	1	2	1	2	1	2	2
2	1	2	1	2	2	2	1	2	1	1	2	1	1	2	2	2	1	2	2	2	2	2	1	2	2
2	1	2	2	2	1	1	2	2	2	1	1	1	2	2	1	2	2	1	2	1	2	1	1	1	2
2	1	1	1	1	1	1	1	2	1	2	2														

and $\mathtt{g}[i]$ is given by the following table:

8B	89	4F	3D	20	25	1D	14	14	13	11	0F	10	0B	0D	0B	0A	0B	07	08				
09	07	0B	08	0B	07	05	04	08	08	05	04	08	01	07	04	07	04	02	04				
0A	05	07	07	06	05	05	04	03	05	03	04	05	04	03	04	05	05	03	04				
02	03	03	02	02	02	02	02	03	02	02	02	02	01	01	02	04	05	02	02				
06	04	02	01	01	04	01	02	02	01	04	03	02	02	01	02	01	02	03	02				
00	02	02	02	03	02	01	01	02	03	04	03	02	02	02	02	02	01	01	02				
02	05	00	00	01	01	03	01	02	02	00	01	01	02	01	00	02	03	02	01				
02	02	01	01	02	02	01	02	01	03	01	00	01	01	02	01	01	02	00	02				
02	00	02	00	02	01	02	01	03	01	01	01	01	03	02	00	01	01	02	02				
00	01	02	01	00	01	01	01	01	01	01	01	02	01	01	01	02	01	03	02				
02	01	01	01	03	03	01	00	00	01	01	02	01	01	01	01	02	02	02	01				
02	01	00	01	01	00	01	02	01	02	00	01	01	02	00	04	02	01	01	01				
00	02	00	01	00	00	01	00	01	00	01	01	00	00	01	00	03	00	01	00				
02	03	02	01	01	01	01	01	00	02	01	02	00	00	02	02	00	01	00	01				
02	02	02	01	00	01	01	02	00	02	01	02	00	01	00	00	02	01	01	01				
01	01	00	01	00	01	01	02	00	01	02	00	01	03	02	00	00	02	00	01				
01	00	02	00	00	00	01	00	01	01	00	01	00	01	01	00	02	01	01	00				
02	00	00	00	01	01	01	02	01	01	00	00	00	00	01	01	01	00	01	01				
02	02	01	01	01	01	01	00	00	01	00	00	00	01	01	01	01	00	01	00				
00	01	00	00	00	02	02	00	01	00	00	00	01	01	00	00	00	02	02	00				
00	00	00	01	00	00	01	00	00	00	01	01	01	00	01	02	00	01	00	00				
01	01	01	01	00	01	01	01	00	00	01	01	00	00	01	00	01	00	01	01				
01	00	01	00	01	00	02	00	01	00	01	00	02	01	00	00	01	00	00	00				
00	00	02	01	00	00	00	01	00	00	00	00	00	00	03	00	00	01	00	00				
00	01	00	00	01	02	00	00	01	00	02	00	00	00	00	02	00	01	00	00				
00	00	00	00	01	01	01	00	00	01	02	00	00	00	00	01	00	00	01	00				
00	00	00	00	01	01	00	01	00	00	00	01	00	01	00	00	00	01	00	00				
01	01	00	00	00	00	00	01	00	01	01	00	00	01	00	01	00	00	00	00				
01	01	02	00	00	00	00	01	00	00	00	01	00	01	01	00	00	00	01	00				
01	00	00	01	00	02	00																	

At Eurocrypt 2002, Grieu presented [13] a much more efficient attack against ISO 9796-1. The attack consists in finding all message m, m' such that:

$$\frac{\mu(m)}{\mu(m')} = \frac{a}{b}$$

for small integers a, b. One obtains 2 pairs of messages m_1, m_1' and m_2, m_2' solution of the previous equation, which gives four messages such that:

$$\mu(m_1) \cdot \mu(m_2') = \mu(m_1') \cdot \mu(m_2)$$

and enables to express the signature of m_1 as a function of the signatures of the other 3 messages.

From the two previous attacks we can conclude that ISO 9796-1 is broken and should not be used.

3.2 ISO 9796-2

ISO 9796-2 is a generic padding standard allowing total or partial message recovery. Let denote by L the output size of the hash function. Hash-functions of different sizes are acceptable and parameter L (in the standard k_h) is consequently a variable. Section 5, note 4 of [15] recommends $64 \leq L \leq 80$ for total recovery and $128 \leq L \leq 160$ for partial recovery.

We start with the partial message recovery variant. For simplicity, we assume that N, L and the size of m are all multiples of eight and that the hash function is known to both parties. The message $m = m[1]\|m[2]$ is separated into two parts where $m[1]$ consists of the $N - L - 16$ most significant bits of m and $m[2]$ of all the remaining bits of m. The padding function is :

$$\mu(m) = \mathtt{6A}_{16}\|m[1]\|\mathrm{HASH}(m)\|\mathtt{BC}_{16}$$

The attack against ISO 9796-2 described in [5] is an application of Coron, Naccache and Stern's attack of the previous section.

One divides $(\mathtt{6A}_{16} + 1) \cdot 2^{|N|}$ by N and obtains:

$$(\mathtt{6A}_{16} + 1) \cdot 2^{|N|} = i \cdot N + r \quad \text{with} \quad 0 \leq r < N < 2^{|N|}$$

On defines N' such that:

$$N' = i \cdot N = \mathtt{6A}_{16} \cdot 2^{|N|} + (2^{|N|} - r) = \mathtt{6A}_{16}\|N'[1]\|N'[0]$$

where the size of N' is $|N| + 7$ bits and the size of $N'[1]$ is $|N| - L - 16$ bits. One takes $m[1] = N'[1]$ one obtains:

$$t = i \cdot N - \mu(m) \cdot 2^8 = N'[0] - \mathrm{HASH}(m)\|\mathtt{BC00}_{16}$$

and the size of t is less than $L + 16$ bits.

The attacker modifies $m[2]$ until he finds sufficiently many integers t which are the product of small primes. In other words, one applies Coron, Naccache and Stern's attack to the integers t. The attack complexity is independent of the size of N; it only depends on the hash size L. From table 1, we obtain the following attack complexity, as a function of the hash size. As for table 1, this is only an estimate, and the practical complexity may be much higher. The table suggests that the attack may be practical for $L = 128$, but will be more demanding for $L = 160$. Note that the following complexities are smaller than the complexities obtained in [5]. This is due to the fact that we have obtained a smaller complexity in section 2.7.

L	$\log_2$ **time**	$\log_2$ **space**
128	44	22
160	49	25

Table 2. Attack complexity with partial message recovery

In the full message recovery variant, we assume that the message size is $|N|-L-16$. The encoding of m is then given by:

$$\mu(m) = \mathtt{4A}_{16} \| m \| \mathrm{HASH}(m) \| \mathtt{BC}_{16}$$

Using the same technique as for the partial message recovery, we obtain the following complexities. This suggests that an attack against ISO 9796-2 with full message recovery may be practical for $L = 64$ and $L = 80$, but is likely to be unpractical for $L \geq 128$.

L	$\log_2$ **time**	$\log_2$ **space**
64	35	18
80	39	20
128	52	26

Table 3. Attack complexity with full message recovery

3.3 PKCS#1 v1.5

The signature scheme PKCS#1 v1.5 [23] is defined as follows:

$$\mu(m) = \mathtt{0001}_{16} \| \mathtt{FFFF}_{16} \ldots \mathtt{FFFF}_{16} \| \mathtt{00}_{16} \| c_{\mathrm{SHA}} \| H(m)$$

where c_{SHA} is a constant and $H(m) = \mathrm{SHA}(m)$, or

$$\mu(m) = \mathtt{0001}_{16} \| \mathtt{FFFF}_{16} \ldots \mathtt{FFFF}_{16} \| \mathtt{00}_{16} \| c_{\mathrm{MD5}} \| H(m)$$

where c_{MD5} is a constant and $H(m) = \mathrm{MD5}(m)$.

In [5] is described an attack against PKCS#1 v1.5 in a particular case: the modulus N is of the form $N = 2^k \pm c$, where the size of c is at least half the size of N. However, the attack is not practical, since its complexity is still higher than factoring the modulus.

In the following, we provide an extension of the attack of [5] to PKCS#1 v1.5 for any modulus N, with roughly the same complexity. Therefore, the attack is still not practical and does not endanger the use of PCKS#1 v1.5.

The technique is the following. We write $\mu(m)$ as

$$\mu(m) = c + H(m)$$

where c is a constant and H is the hash function of size ℓ bits. We denote by n the size of the modulus N in bits. We find two integers a and b such that

$$a \cdot c = b \mod N$$

where the size of a is $(n-\ell)/2$ and the size of b is $(n+\ell)/2$. As noted in [11], this is equivalent to finding a good approximation of the fraction c/N, and can be done efficiently by developing it in continued fractions, *i.e.* applying the extended Euclidean algorithm to c and N.

Then we have:

$$a \cdot \mu(m) = b + a \cdot H(m) = t$$

where t is a $(n + \ell)/2$ bit modulus. Therefore, we can apply the attack of section 2.7 directly. Using table 1, we obtain the following attack complexity (table 4) for a 1024-bit modulus, which shows that the attack is not practical.

ℓ	$\log_2$ **time**	$\log_2$ **space**
128	100	50
160	102	51

Table 4. Attack complexity against PCKS#1 v1.5

3.4 ANSI x9.31

The analysis for ANSI x9.31, where:

$$\mu(m) = \mathtt{6B}_{16} \| \mathtt{BBBB}_{16} \ldots \mathtt{BBBB}_{16} \| \mathtt{BA}_{16} \| \mathrm{SHA}(m) \| \mathtt{33CC}_{16}$$

is the same as for PKCS#1 v1.5. The attacks described in [5] and its extension described in the previous section have the same complexity as for PKCS#1 v1.5 and are not practical.

3.5 Attacks against RSA schemes with linear redundancy

In section 2, we have described the existing attacks against RSA signature schemes with linear redundancy. These attacks do not seem to extend to RSA signature schemes using a hash function. For example, given a constant P, a message m_1 and a modulus N, Lenstra and Shparlinski's attack provides three messages m_2, m_3 and m_4 such that

$$(P \| m_1) \cdot (P \| m_2) = (P \| m_3) \cdot (P \| m_4) \mod N$$

For a hash-based signature scheme such as PKCS#1 v1.5, where:

$$\mu(M) = P \| H(M)$$

one could take $m_1 = H(M_1)$ for a given M_1, but then we would have to find three other messages M_2, M_3, M_4 such that $m_i = H(M_i)$, for $i = 1, 2, 3$. This is infeasible unless the hash function is "weak", *i.e.* it is not one-way.

Therefore, it seems reasonable to say that the attacks against RSA schemes with linear redundancy do not extend to RSA signature standards PKCS#1 v1.5, ANSI X9.31 and ISO 9796-2.

3.6 Security proof for partial-domain hash signature scheme

In this section, we show that it is possible to derive a security proof for the RSA signature schemes ISO 9796-2, ANSI x9.31 and PKCS#1 v1.5. The security proof only applies for $e = 2$ and for a hash size larger than 2/3 the size of the modulus. This result will be published [6] at the conference Crypto 2002. We provide in appendix the proceeding version of [6].

More generally, this result applies to any partial-domain hash signature scheme. We say that a hash-and-sign signature scheme is a *partial-domain hash signature scheme* if the encoding function $\mu(m)$ can be written as:

$$\mu(m) = \gamma \cdot H(m) + f(m) \tag{9}$$

where γ is a constant, H a hash function and f some function of m.

We now state the main theorem. It shows that partial-domain hash signature schemes are provably secure in the random oracle model, for $e = 2$, assuming that factoring is hard, if the size of the hash function is larger than 2/3 of the modulus size. The case $e = 2$ corresponds to the Rabin-William signature scheme, which we recall in appendix. In particular, the Rabin-William signature scheme uses a padding function $\mu(m)$ such that for all m, $\mu(m) = 6 \mod 16$. Moreover, we restrict ourselves to small constants γ in (9), *e.g.* $\gamma = 16$ or $\gamma = 256$. This is the case for all the signature standards that we have considered.

Theorem 1. *Let $\mathcal{S}$ be the Rabin-Williams partial-domain hash signature scheme with constant γ and hash size k_0 bits. Assume that there is no algorithm which factors a RSA modulus with probability greater than ε within time t. Then the success probability of a forger against $\mathcal{S}$ making at most q_{hash} hash queries and q_{sig} signature queries within time t' is upper bounded by ε', where:*

$$\varepsilon' = 8 \cdot q_{sig} \cdot \varepsilon + 32 \cdot (q_{hash} + q_{sig} + 1) \cdot k_1 \cdot \gamma \cdot 2^{-\frac{3}{13} \cdot k_1} \tag{10}$$

$$t' = t - k_1 \cdot \gamma \cdot (q_{hash} + q_{sig} + 1) \cdot \mathcal{O}(k^3) \tag{11}$$

and $k_1 = k_0 - \frac{2}{3}k$.

4 Application to Signature Standards

4.1 PKCS#1 v1.5

The standard PKCS#1 v1.5 was not designed to work with Rabin ($e = 2$). However, one can replace the last nibble of $H(m)$ by 6 and obtain a padding scheme which is compatible with the Rabin-Williams signature scheme. The standard is then provably secure if the size of the hash-function is larger than 2/3 of the size of the modulus. This is much larger than the 128 or 160 bits which are recommended in the standard.

4.2 ISO 9796-2 and ANSI x9.31

An application of ISO 9796-2 with the Rabin-Williams signature scheme is described in [15]. Note that since $\mu(m) = 12 \mod 16$ instead of $\mu(m) = 6 \mod 16$, there is a slight change in the verification process. However, the same security bound applies: the scheme is provably secure if the size of the hash-function is larger than 2/3 of the size of the modulus. The same analysis applies for the ANSI x9.31 padding scheme [1].

5 Conclusion

We have investigated the security of RSA signature schemes PKCS#1 v1.5, ANSI x9.31, ISO 9796-1 and ISO 9796-2. We have shown that the standard ISO 9796-1 is insecure and should not be used. We have recalled Coron, Naccache and Stern's attack [5] against ISO 9796-2 which shows that if the hash size L is too small, the standard is insecure. For ISO 9796-2 with partial-message recovery, we recommend to take $L \geq 160$. For ISO 9796-2 with full-message recovery, we recommend to take $L \geq 128$. This makes the attack of [5] unpractical. For PKCS#1 v1.5 and ANSI x9.31, we have seen that the attack of [5] does not apply. To our knowledge, no attack better than factoring the modulus or finding a collision in the hash function, is known for PKCS#1 v1.5 and ANSI x9.31.

Moreover, we have shown that it is possible to obtain a security proof for PKCS#1 v1.5, ANSI x9.31 and ISO 9796-2 in a particular case: $e = 2$ and the hash size is larger than 2/3 the size of the modulus. In this case, the signature standard reaches the highest level of provable security: it is infeasible to forge signature under an adaptive chosen message attack, assuming that factoring is hard.

References

1. ANSI X9.31, *Digital signatures using reversible public-key cryptography for the financial services industry (rDSA)*, 1998.
2. M. Bellare and P. Rogaway, *The exact security of digital signatures - How to sign with RSA and Rabin*. Proceedings of Eurocrypt'96, LNCS vol. 1070, Springer-Verlag, 1996, pp. 399-416.
3. E. Brier, C. Clavier, J.S. Coron and D. Naccache, *Cryptanalysis of RSA Signatures with fixed-pattern padding*, Proceedings of Crypto 2001, LNCS 2139.
4. D. Coppersmith, S. Halevi and C. Jutla, *ISO 9796-1 and the new forgery strategy*, Resarch contribution to P1363, 1999, available at `http://grouper.ieee.org/groups/1363/contrib.html`.
5. J.S. Coron, D. Naccache and J.P. Stern, *On the security of RSA Padding*, Proceedings of Crypto'99, LNCS vol. 1666, Springer-Verlag, 1999, pp. 1-18.
6. J.S. Coron, *Security proof for partial-domain hash signature schemes*, Proceedings of Crypto 2002, Lecture Notes in Computer Science.
7. Y. Desmedt and A. Odlyzko. *A chosen text attack on the RSA cryptosystem and some discrete logarithm schemes*, Proceedings of Crypto'85, LNCS 218, pp. 516–522.
8. K. Dickman, *On the frequency of numbers containing prime factors of a certain relative magnitude*, Arkiv för matematik, astronomi och fysik, vol. 22A, no. 10, pp. 1–14, 1930.
9. W. De Jonge and D. Chaum, *Attacks on some RSA signatures*. Proceedings of Crypto '85, LNCS vol. 218, Springer-Verlag, 1986, pp. 18-27.
10. M. Girault and J.-F. Misarksy, *Selective forgery of RSA signatures using redundancy*, Proceedings of Eurocrypt '97, LNCS vol. 1233, Springer-Verlag, 1997, pp. 495–507.
11. M. Girault, P. Toffin and B. Vallée, *Computation of approximation L-th roots modulo n and application to cryptography*, Proceedings of Crypto '88, LNCS vol. 403, Springer-Verlag, 1988, pp. 100–117.
12. 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, april 1988.
13. F. Grieu, *A chosen message attack on the ISO/IEC 9796-1 signature scheme*, Advances in Cryptology - Eurocrypt 2000, LNCS 1807, pp. 70–80.
14. ISO/IEC 9796, *Information technology - Security techniques - Digital signature scheme giving message recovery, Part 1 : Mechanisms using redundancy*, 1999.

15. ISO/IEC 9796-2, *Information technology - Security techniques - Digital signature scheme giving message recovery, Part 2 : Mechanisms using a hash-function*, 1997.
16. C. Lanczos, *An iterative method for the solution of the eigenvalue problem of linear differential and integral operator*, J. Res. Nat. Bur. Standards, 1950, vol. 45, pp. 255–282.
17. H. W. Lenstra, Jr., *Factoring integers with elliptic curves*, Ann. of Math. (2) 126 (1987) pp. 649-673.
18. A. K. Lenstra, H.W. Lenstra and L. Lovász, *Factoring polynomials with rational coefficients*, Mathematische Annalen, vol. 261, n. 4, 1982, pp. 515–534.
19. A. K. Lenstra and I. E. Shparlinski, *Selective forgery of RSA signatures with fixed-pattern padding*, Proceedings of PKC 2002, LNCS 2274.
20. J.-F. Misarsky, *A multiplicative attack using LLL algorithm on RSA signatures with redundancy*, Proceedings of Crypto '97, LNCS vol. 1294, Springer-Verlag, pp. 221–234.
21. J.-F. Misarsky, *How (not) to design RSA signature schemes*, Public-key cryptography, Springer-Verlag, Lectures notes in computer science 1431, pp. 14–28, 1998.
22. T. Okamoto and A. Shiraishi, *A fast signature scheme based on quadratic inequalities*, Proc. of the 1985 Symposium on Security and Privacy, April 1985, Oakland, CA.
23. RSA Laboratories, PKCS #1 : *RSA cryptography specifications*, version 1.5, November 1993 and version 2.0, September 1998.
24. R. Rivest, A. Shamir and L. Adleman, *A method for obtaining digital signatures and public key cryptosystems*, CACM 21, 1978.
25. RSA Laboratories, PKCS #1 : *RSA cryptography specifications*, version 2.0, September 1998.

Security Proof for Partial-Domain Hash Signature Schemes

Jean-Sébastien Coron
Gemplus Card International
34 rue Guynemer
Issy-les-Moulineaux, F-92447, France
jean-sebastien.coron@gemplus.com

Abstract. We study the security of partial-domain hash signature schemes, in which the output size of the hash function is only a fraction of the modulus size. We show that for $e = 2$ (Rabin), partial-domain hash signature schemes are provably secure in the random oracle model, if the output size of the hash function is larger than 2/3 of the modulus size. This provides a security proof for a variant of the signature standards ISO 9796-2 and PKCS#1 v1.5, in which a larger digest size is used.

Key-words: Signature Schemes, Provable Security, Random Oracle Model.

1 Introduction

A common practice for signing with RSA or Rabin consists in first hashing the message m, then padding the hash value with some predetermined or message-dependent block, and eventually raising the result $\mu(m)$ to the private exponent d. This is commonly referred to as the "hash-and-sign" paradigm:

$$s = \mu(m)^d \mod N$$

For digital signature schemes, the strongest security notion was defined by Goldwasser, Micali and Rivest in [8], as *existential unforgeability under an adaptive chosen message attack*. This notion captures the property that an attacker cannot produce a valid signature, even after obtaining the signature of (polynomially many) messages of his choice.

The random oracle model, introduced by Bellare and Rogaway in [2], is a theoretical framework allowing to prove the security of hash-and-sign signature schemes. In this model, the hash function is seen as an oracle which outputs a random value for each new query. Bellare and Rogaway defined in [3] the Full Domain Hash (FDH) signature scheme, in which the output size of the hash function is the same as the modulus size. FDH is provably secure in the random oracle model assuming that inverting RSA is hard. Actually, a security proof in the random oracle model does not necessarily imply that the scheme is secure in the real world (see [4]). Nevertheless, it seems to be a good engineering principle to design a scheme so that it is provably secure in the random oracle model. Many encryption and signature schemes were proven to be secure in the random oracle model

Other hash-and-sign signature schemes include the widely used signature standards PKCS#1 v1.5 and ISO 9796-2. In these standards, the digest size is only a fraction of

the modulus size. As opposed to FDH, no security proof is known for those standards. Moreover, it was shown in [5] that ISO 9796-2 was insecure if the size of the hash function was too small, and the standard was subsequently revised.

In this paper, we study the security of partial-domain hash signature schemes, in which the hash size is only a fraction of the modulus size. We show that for $e = 2$, partial-domain hash signature schemes are provably secure in the random oracle model, assuming that factoring is hard, if the size of the hash function is larger than 2/3 of the modulus size. The proof is based on a modification of Vallée's generator of small random squares [16]. This provides a security proof for a variant of PKCS#1 v1.5 and ISO 9796-2 signatures, in which the digest size is larger than 2/3 of the size of the modulus.

2 Definitions

In this section we briefly present some notations and definitions used throughout the paper. We start by recalling the definition of a signature scheme.

Definition 1 (Signature Scheme). *A signature scheme* $(\mathtt{Gen}, \mathtt{Sign}, \mathtt{Verify})$ *is defined as follows:*

- The key generation algorithm `Gen` *is a probabilistic algorithm which given* 1^k*, outputs a pair of matching public and private keys,* (pk, sk)*.*

- The signing algorithm `Sign` *takes the message* M *to be signed, the private key* sk*, and returns a signature* $x = \mathtt{Sign}_{sk}(M)$*. The signing algorithm may be probabilistic.*

- The verification algorithm `Verify` *takes a message* M*, a candidate signature* x' *and* pk*. It returns a bit* $\mathtt{Verify}_{pk}(M, x')$*, equal to one if the signature is accepted, and zero otherwise. We require that if* $x \leftarrow \mathtt{Sign}_{sk}(M)$*, then* $\mathtt{Verify}_{pk}(M, x) = 1$*.*

In the previously introduced *existential unforgeability under an adaptive chosen message attack* scenario, the forger can dynamically obtain signatures of messages of his choice and attempt to output a valid forgery. A *valid forgery* is a message/signature pair (M, x) such that $\mathtt{Verify}_{pk}(M, x) = 1$ whereas the signature of M was never requested by the forger. Moreover, in the random oracle model, the attacker cannot evaluate the hash function by himself; instead, he queries an oracle which outputs a random value for each new query.

RSA [14] is undoubtedly the most widely used cryptosystem today:

Definition 2 (RSA). *The RSA cryptosystem is a family of trapdoor permutations, specified by:*
- The RSA generator $\mathcal{RSA}$*, which on input* 1^k*, randomly selects two distinct* $k/2$*-bit primes* p *and* q *and computes the modulus* $N = p \cdot q$*. It picks an encryption exponent* $e \in \mathbb{Z}^*_{\phi(N)}$ *and computes the corresponding decryption exponent* d *such that* $e \cdot d = 1 \mod \phi(N)$*. The generator returns* (N, e, d)*.*
- The encryption function $f : \mathbb{Z}^*_N \to \mathbb{Z}^*_N$ *defined by* $f(x) = x^e \mod N$*.*
- The decryption function $f^{-1} : \mathbb{Z}^*_N \to \mathbb{Z}^*_N$ *defined by* $f^{-1}(y) = y^d \mod N$*.*

An *inverting algorithm* $\mathcal{I}$ for RSA gets as input (N, e, y) and tries to find $y^d \mod N$. Its success probability is the probability to output $y^d \mod N$ when (N, e, d)

are obtained by running $\mathcal{RSA}(1^k)$ and y is set to $x^e \mod N$ for some x chosen at random in $\mathbb{Z}_N^*$.

The Full-Domain-Hash scheme (FDH) [3] was the first practical and provably secure signature scheme based on RSA. It is defined as follows: the key generation algorithm, on input 1^k, runs $\mathcal{RSA}(1^k)$ to obtain (N, e, d). It outputs (pk, sk), where the public key pk is (N, e) and the private key sk is (N, d). The signing and verifying algorithms use a hash function $H : \{0,1\}^* \to \mathbb{Z}_N^*$ which maps bit strings of arbitrary length to the set of invertible integers modulo N.

$\texttt{SignFDH}_{N,d}(M)$
  $y \leftarrow H(M)$
  return $y^d \mod N$

$\texttt{VerifyFDH}_{N,e}(M, x)$
  $y \leftarrow x^e \mod N$
  if $y = H(M)$ then return 1 else return 0.

The following theorem [6] proves the security of FDH in the random oracle model, assuming that inverting RSA is hard. It provides a better security bound than [3].

Theorem 1. *Assume that there is no algorithm which inverts RSA with probability greater than ε within time t. Then the success probability of a FDH forger making at most q_{hash} hash queries and q_{sig} signature queries within running time t' is less than ε', where*

$$\varepsilon' = 4 \cdot q_{sig} \cdot \varepsilon$$
$$t' = t - (q_{hash} + q_{sig} + 1) \cdot \mathcal{O}(k^3)$$

We say that a hash-and-sign signature scheme is a *partial-domain hash signature scheme* if the encoding function $\mu(m)$ can be written as:

$$\mu(m) = \gamma \cdot H(m) + f(m) \tag{1}$$

where γ is a constant, H a hash function and f some function of m. A typical example of a partial-domain hash signature scheme is the ISO 9796-2 standard with full message recovery [11]:

$$\mu(m) = \texttt{4A}_{16} \| m \| H(m) \| \texttt{BC}_{16}$$

The main result of this paper is to show that for $e = 2$, partial-domain hash signature schemes are provably secure, if the hash size is larger than 2/3 of the modulus size. In the following, we recall the Rabin- Williams signature scheme [12]. It uses a padding function $\mu(m)$ such that for all m, $\mu(m) = 6 \mod 16$.

- Key generation: on input 1^k, generate two $k/2$-bit primes p and q such that $p = 3 \mod 8$ and $q = 7 \mod 8$. The public key is $N = p \cdot q$ and the private key is $d = (N - p - q + 5)/8$.

- Signature generation: compute the Jacobi symbol

$$J = \left(\frac{\mu(m)}{N}\right)$$

The signature of m is $s = \min(\sigma, N - \sigma)$, where:

$$\sigma = \begin{cases} \mu(m)^d \mod N & \text{if } J = 1 \\ (\mu(m)/2)^d \mod N & \text{otherwise} \end{cases}$$

- Signature verification: compute $\omega = s^2 \mod N$ and check that:

$$\mu(m) \stackrel{?}{=} \begin{cases} \omega & \text{if } \omega = 6 \mod 8 \\ 2 \cdot \omega & \text{if } \omega = 3 \mod 8 \\ N - \omega & \text{if } \omega = 7 \mod 8 \\ 2 \cdot (N - \omega) & \text{if } \omega = 2 \mod 8 \end{cases}$$

3 Security of Partial-domain Hash Signature Schemes

To prove the security of a signature scheme against chosen message attacks, one must be able to answer the signature queries of the attacker. In FDH's security proof, when answering a hash query, one generates a random $r \in \mathbb{Z}_N$ and answers $H(m) = r^e \mod N$ so that the signature r of m is known. Similarly, for partial-domain hash signature schemes, we should be able to generate a random r such that:

$$\mu(m) = \gamma \cdot H(m) + f(m) = r^e \mod N$$

with $H(m)$ being uniformly distributed in the output space of the hash function. For example, if we take $\mu(m) = H(m)$ where $0 \leq H(m) \leq N^\beta$ and $\beta < 1$, one should be able to generate a random r such that $r^e \mod N$ is uniformly distributed between 0 and N^β.

Up to our knowledge, no such algorithm is known for $e \geq 3$. For $e = 2$, Vallée constructed in [16] a random generator where the size of $r^2 \mod N$ is less than 2/3 of the size of the modulus. [16] used this generator to obtain proven complexity bounds for the quadratic sieve factoring algorithm. Vallée's generator has a quasi-uniform distribution; a distribution is said to be *quasi-uniform* if there is a constant ℓ such that for all x, the probability to generate x lies between $1/\ell$ and ℓ times the probability to generate x under the uniform distribution. However, quasi-uniformity is not sufficient here, as we must simulate a random oracle and therefore our simulation should be indistinguishable from the uniform distribution.

Our contribution is to modify Vallée's generator in order to generate random squares in any interval of size $N^{2/3+\varepsilon}$, with a distribution which is statistically indistinguishable from the uniform distribution. From this generator we will derive a security proof for partial-domain hash signatures, in which the digest size is at least 2/3 of the modulus size.

Remark: for Paillier's trapdoor permutation [13] with parameter $g = 1 + N$, it is easy to show that half-domain hash is provably secure in the random oracle model, assuming that inverting RSA with $e = N$ is hard.

4 Generating Random Squares in a Given Interval

4.1 Notations

We identify $\mathbb{Z}_N$, the ring of integers modulo N with the set of integers between 0 and $N - 1$. We denote by $\mathbb{Z}_N^+$ the set of integers between 0 and $(N - 1)/2$. We denote by Q the squaring operation over $\mathbb{Z}_N$:

$$Q(x) = x^2 \mod N$$

Given positive integers a and h such that $a + h < N$, let B be the set:

$$B = \{x \in \mathbb{Z}_N^+ \mid a \leq Q(x) \leq a + h\}$$

Our goal is to generate integers $x \in B$ with a distribution statistically indistinguishable from the uniform distribution. The *statistical distance* between two distributions X and Y is defined as the function:

$$\delta = \frac{1}{2} \sum_{\alpha} |\Pr[X = \alpha] - \Pr[Y = \alpha]|$$

We say that two ensembles $X = \{X_n\}_{n \in \mathbb{N}}$ and $Y = \{Y_n\}_{n \in \mathbb{N}}$ are *statistically indistinguishable* if their statistical distance δ_n is a negligible function of n.

4.2 Description of B

In this section, we recall Vallée's description of the set B. We denote by b the cardinality of B. The following lemma, which proof can be derived from equation (6) in [16], shows that b is close to $h/2$.

Lemma 1. *Let N be a ℓ-bit RSA modulus. We have for $\ell \geq 64$:*

$$\left| b - \frac{h}{2} \right| \leq 4 \cdot \ell \cdot 2^{\ell/2}$$

In the following, we assume that the bit size of N is greater than 64. As in [16], we introduce Farey sequences [9]:

Definition 3 (Farey sequence). *The Farey sequence $\mathcal{F}_k$ of order k is the ascending sequence of irreducible fractions between 0 and 1 whose denominators do not exceed k. Thus p/q belongs to $\mathcal{F}_k$ if $0 \leq p \leq q \leq k$ and $\gcd(p, q) = 1$.*

The characteristic property of Farey sequences is expressed by the following theorem [9]:

Theorem 2. *If p/q and p'/q' are two successive terms of $\mathcal{F}_k$, then $q \cdot p' - p \cdot q' = 1$*

Given $p/q \in \mathcal{F}_k$, we define the *Farey interval* $I(p, q)$ as the interval of center $pN/(2q)$ and radius $N/(2kq)$. Given the terms p'/q' and p''/q'' of $\mathcal{F}_k$ which precede and follow p/q, we let $J(p, q)$ be the interval:

$$J(p, q) = \left[\frac{N(p + p')}{2(q + q')}, \frac{N(p + p'')}{2(q + q'')} \right]$$

If $p/q = 0/1$, then p/q has no predecessor and we take $p'/q' = 0/1$. Similarly, if $p/q = 1/1$, we take $p''/q'' = 1/1$. The set of intervals $J(p, q)$ forms a partition of $\mathbb{Z}_N^+$. The following lemma [16] shows that intervals $I(p, q)$ and $J(p, q)$ are closely related.

Lemma 2. *$I(p, q)$ contains $J(p, q)$ and its length is at most twice the length of $J(p, q)$.*

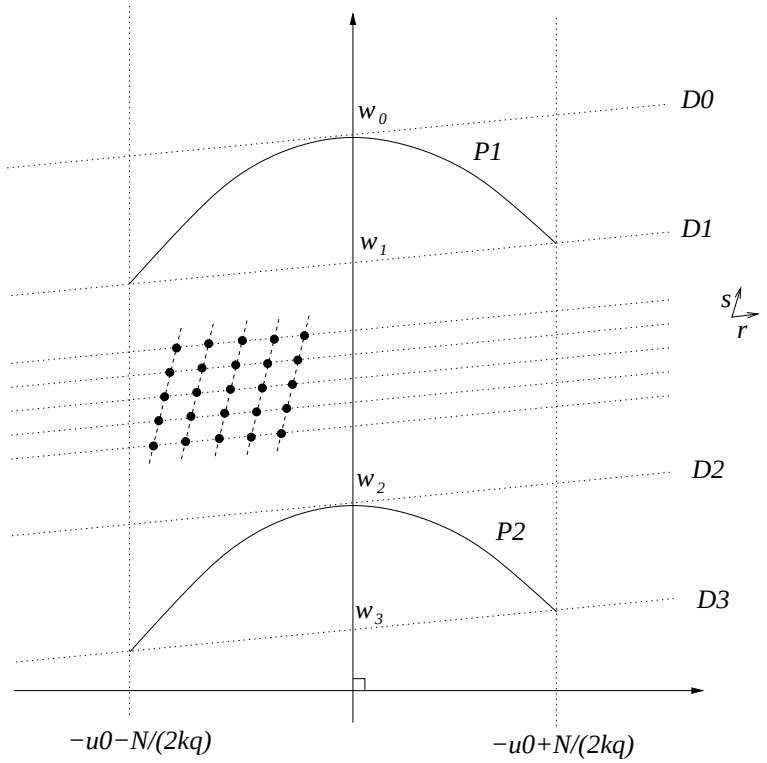

Fig. 1. The intersection between the lattice $L(x_0)$ and the domain between the two parabolas $\mathcal{P}_1$ and $\mathcal{P}_2$

Given $p/q \in \mathcal{F}_k$ with $p/q \neq 0/1$, let x_0 be the integer nearest to the rational $pN/2q$:

$$x_0 - \frac{pN}{2q} = u_0 \quad \text{with } |u_0| \leq \frac{1}{2}$$

Let $L(x_0)$ be the lattice spanned by the two vectors $(1, 2x_0)$ and $(0, N)$. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be the two parabolas of equations:

$$\mathcal{P}_1 : \omega + u^2 + x_0^2 = a + h \quad \text{and} \quad \mathcal{P}_2 : \omega + u^2 + x_0^2 = a$$

Let P be the domain of lattice points comprised between the two parabolas:

$$P = \{(u, \omega) \in L(x_0) \mid a \leq \omega + u^2 + x_0^2 \leq a + h\}$$

The following lemma, which proof is straightforward, shows that the elements of B arise from the intersection of the lattice $L(x_0)$ and the domain comprised between the two parabolas (see figure 1).

Lemma 3. *$x = x_0 + u$ belongs to B iff there exists a unique ω such that the point (u, ω) belongs to P.*

We let $B(p, q)$ be the set of integers in $B \cap J(p, q)$. From Lemma 3 the integers in $B(p, q)$ arise from the domain of lattice points:

$$P(p, q) = \{(u, \omega) \in P \mid x_0 + u \in J(p, q)\}$$

From Lemma 2, the set $P(p,q)$ is included inside the set of lattice points:

$$Q(p,q) = \{(u,\omega) \in P \mid x_0 + u \in I(p,q)\}$$

whose abcissae u are comprised between $-u_0 - N/(2kq)$ and $-u_0 + N/(2kq)$. In the following, we describe the domain $Q(p,q)$, using the following two short vectors of $L(x_0)$ (see figure 1):

$$\boldsymbol{r} = q(1, 2x_0) - p(0, N) = (q, 2qu_0) \tag{2}$$
$$\boldsymbol{s} = q'(1, 2x_0) - p'(0, N) = (q', 2q'u_0 + N/q) \tag{3}$$

where p'/q' is the term of $\mathcal{F}_k$ which precedes p/q.

We consider the lines of the lattice parallel to vector $\boldsymbol{r}$ which intersect the domain $Q(p,q)$. These lines have a slope equal to $2u_0$. The first extremal position of these lines is the tangent D_0 to the first parabola:

$$D_0 : \ \omega - (-u_0^2 - x_0^2 + a + h) = 2u_0(u + u_0)$$

The second extremal position joins the two points of the second parabola with abscissae $-u_0 - N/(2kq)$ and $-u_0 + N/(2kq)$. This line D_3 has also a slope equal to $2u_0$ and satisfies the equation:

$$\omega + (u_0 + \frac{N}{2kq})^2 - a + x_0^2 = 2u_0(u + u_0 + \frac{N}{2kq})$$

The two lines intersect the vertical axis at the respective points:

$$\omega_0 = a - x_0^2 + u_0^2 + h \ \text{ and } \ \omega_3 = a - x_0^2 + u_0^2 - \frac{N^2}{4k^2q^2}$$

All the lines parallel to $\boldsymbol{r}$ that intersect $P(p,q)$ are the ones that intersect the segment $[\omega_3, \omega_0]$ on the vertical axis. We denote by $D(\nu)$ a line parallel to $\boldsymbol{r}$ which intersects the vertical axis at ordinate equal to $\omega_0 - \nu N/q$. The line D_0 is the line $D(\nu_0 = 0)$, whereas the line D_3 is the line $D(\nu_3)$ such that:

$$\nu_3 = \frac{hq}{N} + \frac{N}{4k^2q} \tag{4}$$

Eventually, we denote by $D_1 = D(\nu_1)$ the line which joins the two points of the first parabola with abcissae $-u_0 - N/(2kq)$ and $-u_0 + N/(2kq)$, and by $D_2 = D(\nu_2)$ the tangent to the second parabola, with a slope equal to $2u_0$. We have:

$$\nu_1 = \frac{N}{4k^2q} \ \text{ and } \ \nu_2 = \frac{hq}{N} \tag{5}$$

A real ν is called an index if $D(\nu)$ is a line of $L(x_0)$. The difference between two consecutive indices is equal to one.

4.3 Our New Generator

In this section, we describe our new generator of integers in B. The difference with Vallée's generator is that we use different parameters for k and h, and we do not generate all the integers in B; instead we avoid a negligible subset of B.

First, we describe a generator $\mathcal{G}(p,q)$ of integers in $B(p,q)$, and we show that its distribution is statistically indistinguishable from the uniform distribution. We assume that $N \leq 2 \cdot k \cdot q \cdot \sqrt{h}$, which gives $\nu_1 \leq \nu_2$. Therefore the line D_1 is above the line D_2 (see figure 1). We restrict ourselves to the integers in $B(p,q)$ such that the corresponding points $(u,\omega) \in P(p,q)$ lie on $D(\nu)$ with $\nu_1 \leq \nu \leq \nu_2$. These points are the points on $D(\nu)$ whose abscissae u are such that $x_0 + u \in J(p,q)$.

Generator $\mathcal{G}(p,q)$ of integers in $B(p,q)$:
1. Generate a random index ν uniformly distributed between ν_1 and ν_2.
2. Generate a point $(u,\omega) \in P(p,q)$ on $D(\nu)$ such that $x_0 + u \in J(p,q)$, with the uniform distribution.
3. Output $x_0 + u$.

The following lemma shows that under some conditions on k, h and q, the cardinality $b(p,q)$ of $B(p,q)$ is close to $h \cdot j(p,q)/N$, where $j(p,q)$ is the number of integers in the interval $J(p,q)$. Moreover, under the same conditions, the distribution induced by $\mathcal{G}(p,q)$ is statistically indistinguishable from the uniform distribution in $B(p,q)$. The proof is given in appendix A.

Lemma 4. *Let $\alpha > 0$ and $k = N^{\frac{1}{3}-\alpha}$. Assume that $k \geq 6$, $N^\alpha \geq 3$ and $N^{\frac{2}{3}+13\cdot\alpha} \leq h < N$. Then for all $p/q \in \mathcal{F}_k$ such that $N^{1/3-4\alpha} \leq q \leq k$, we have:*

$$\left| b(p,q) - \frac{h \cdot j(p,q)}{N} \right| \leq \frac{4h \cdot j(p,q)}{N} N^{-3\alpha} \tag{6}$$

Moreover, $\mathcal{G}(p,q)$ generates elements in $B(p,q)$ with a distribution whose distance δ_G from the uniform distribution is at most $7 \cdot N^{-3\alpha}$.

Now we construct a generator $\mathcal{V}$ of $p/q \in \mathcal{F}_k$ such that the probability to generate p/q is close to $b(p,q)/b$. It only generates $p/q \in \mathcal{F}_k$ such that $q \geq N^{1/3-4\alpha}$, so that from the previous lemma, $b(p,q)$ is nearly proportional to the number of integers in $J(p,q)$, and the distribution induced by $\mathcal{G}(p,q)$ is close to the uniform distribution.

Generator $\mathcal{V}$ of $p/q \in \mathcal{F}_k$
1. Generate a random integer $x \in \mathbb{Z}_N^+$ with the uniform distribution.
2. Determine which interval $J(p,q)$ contains x.
3. If $q \geq N^{1/3-4\alpha}$ then output $p/q \in \mathcal{F}_k$, otherwise output $\perp$.

Lemma 5. *Let denote by $\mathcal{D}$ the distribution induced by choosing $p/q \in \mathcal{F}_k$ with probability $b(p,q)/b$. Under the conditions of lemma 4, the statistical distance δ_V between $\mathcal{D}$ and the distribution induced by $\mathcal{V}$ is at most $9 \cdot N^{-3\alpha}$.*

Proof. See appendix B.

Eventually, our generator $\mathcal{G}$ of elements in B combines the two generators $\mathcal{V}$ and $\mathcal{G}(p,q)$:

Generator $\mathcal{G}$ of $x \in B$
1. Generate y using $\mathcal{V}$.
2. If $y = \perp$, then output $\perp$.
3. Otherwise, $y = p/q$ and generate $x \in B(p,q)$ using $\mathcal{G}(p,q)$. Output x.

The following theorem, whose proof is given in appendix C, shows that the distribution induced by $\mathcal{G}$ is statistically indistinguishable from the uniform distribution in B.

Theorem 3. *For any $\varepsilon > 0$, letting $h = N^{\frac{2}{3}+\varepsilon}$ and $\alpha = \varepsilon/13$. If $N^\alpha \geq 3$, then the distance δ between the distribution induced by $\mathcal{G}$ and the uniform distribution in B is at most $16 \cdot N^{-3\cdot\varepsilon/13}$. The running time of $\mathcal{G}$ is $\mathcal{O}(\log^3 N)$.*

5 A Security Proof for Partial-domain Hash Signature Schemes

In this section, using the previous generator $\mathcal{G}$ of random squares, we show that partial-domain hash signature schemes are provably secure in the random oracle model, for $e = 2$, assuming that factoring is hard, if the size of the hash function is larger than 2/3 of the modulus size. Moreover, we restrict ourselves to small constants γ in (1), *e.g.* $\gamma = 16$ or $\gamma = 256$. This is the case for all the signature standards of the next section. We denote by k_0 the hash function's digest size. The proof is similar to the proof of theorem 1 and is given in the full version of this paper [7].

Theorem 4. *Let $\mathcal{S}$ be the Rabin-Williams partial-domain hash signature scheme with constant γ and hash size k_0 bits. Assume that there is no algorithm which factors a RSA modulus with probability greater than ε within time t. Then the success probability of a forger against $\mathcal{S}$ making at most q_{hash} hash queries and q_{sig} signature queries within time t' is upper bounded by ε', where:*

$$\varepsilon' = 8 \cdot q_{sig} \cdot \varepsilon + 32 \cdot (q_{hash} + q_{sig} + 1) \cdot k_1 \cdot \gamma \cdot 2^{-\frac{3}{13}\cdot k_1} \tag{7}$$

$$t' = t - k_1 \cdot \gamma \cdot (q_{hash} + q_{sig} + 1) \cdot \mathcal{O}(k^3) \tag{8}$$

and $k_1 = k_0 - \frac{2}{3}k$.

6 Application to Signature Standards

6.1 PKCS#1 v1.5 and SSL-3.02

The signature scheme PKCS#1 v1.5 [15] is a partial-domain hash signature scheme, with:

$$\mu(m) = \mathtt{0001}_{16} \| \mathtt{FFFF}_{16} \ldots \mathtt{FFFF}_{16} \| \mathtt{00}_{16} \| c_{\mathrm{SHA}} \| H(m)$$

where c_{SHA} is a constant and $H(m) = \mathrm{SHA}(m)$, or

$$\mu(m) = \mathtt{0001}_{16} \| \mathtt{FFFF}_{16} \ldots \mathtt{FFFF}_{16} \| \mathtt{00}_{16} \| c_{\mathrm{MD5}} \| H(m)$$

where c_{MD5} is a constant and $H(m) = \mathrm{MD5}(m)$.

The standard PKCS#1 v1.5 was not designed to work with Rabin ($e = 2$). However, one can replace the last nibble of $H(m)$ by 6 and obtain a padding scheme which is compatible with the Rabin-Williams signature scheme. The standard is then provably secure if the size of the hash-function is larger than 2/3 of the size of the modulus. This is much larger than the 128 or 160 bits which are recommended in the standard. The same analysis applies for the SSL-3.02 padding scheme [10].

6.2 ISO 9796-2 and ANSI x9.31

The ISO 9796-2 encoding scheme [11] is defined as follows:

$$\mu(m) = \mathtt{6A}_{16} \| m[1] \| H(m) \| \mathtt{BC}_{16}$$

where $m[1]$ is the leftmost part of the message, or:

$$\mu(m) = \mathtt{4A}_{16} \| m \| H(m) \| \mathtt{BC}_{16}$$

[11] describes an application of ISO 9796-2 with the Rabin-Williams signature scheme. Note that since $\mu(m) = 12 \mod 16$ instead of $\mu(m) = 6 \mod 16$, there is a slight change in the verification process. However, the same security bound applies: the scheme is provably secure if the size of the hash-function is larger than 2/3 of the size of the modulus. The same analysis applies for the ANSI x9.31 padding scheme [1].

7 Conclusion

We have shown that for Rabin, partial-domain hash signature schemes are provably secure in the random oracle, assuming that factoring is hard, if the size of the hash function is larger than 2/3 of the modulus size. Unfortunately, this is much larger than the size which is recommended in the standards PKCS#1 v1.5 and ISO 9796-2. An open problem is to obtain a smaller bound for the digest size, and to extend this result to RSA signatures.

Acknowledgements

I wish to thanks the anonymous referees for their helpful comments.

References

1. ANSI X9.31, *Digital signatures using reversible public-key cryptography for the financial services industry (rDSA)*, 1998.
2. M. Bellare and P. Rogaway, *Random oracles are practical : a paradigm for designing efficient protocols*. Proceedings of the First Annual Conference on Computer and Commmunications Security, ACM, 1993.
3. M. Bellare and P. Rogaway, *The exact security of digital signatures - How to sign with RSA and Rabin*. Proceedings of Eurocrypt'96, LNCS vol. 1070, Springer-Verlag, 1996, pp. 399-416.
4. R. Canetti, O. Goldreich and S. Halevi, *The random oracle methodology, revisited*, STOC' 98, ACM, 1998.
5. J.S. Coron, D. Naccache and J.P. Stern, *On the security of RSA Padding*, Proceedings of Crypto'99, LNCS vol. 1666, Springer-Verlag, 1999, pp. 1-18.
6. J.S. Coron, *On the exact security of Full Domain Hash*, Proceedings of Crypto 2000, LNCS vol. 1880, Springer-Verlag, 2000, pp. 229-235.
7. J.S. Coron, *Security proof for partial-domain hash signature schemes*. Full version of this paper. Cryptology ePrint Archive, `http://eprint.iacr.org`.
8. 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, april 1988.
9. G.H. Hardy and E.M. Wright, *An introduction to the theory of numbers*, Oxford science publications, fifth edition.
10. K. Hickman, *The SSL Protocol*, December 1995. Available electronically at : `www.netscape.com/newsref/std/ssl.html`

11. ISO/IEC 9796-2, *Information technology - Security techniques - Digital signature scheme giving message recovery, Part 2 : Mechanisms using a hash-function*, 1997.
12. A.J. Menezes, P. C. van Oorschot and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC press, 1996.
13. P. Paillier, *Public-key cryptosystems based on composite degree residuosity classes*, proceedings of Eurocrypt'99, LNCS 1592, pp. 223-238, 1999.
14. R. Rivest, A. Shamir and L. Adleman, *A method for obtaining digital signatures and public key cryptosystems*, CACM 21, 1978.
15. RSA Laboratories, PKCS #1 : *RSA cryptography specifications*, version 1.5, November 1993 and version 2.0, September 1998.
16. B. Vallée, *Generation of elements with small modular squares and provably fast integer factoring algorithms*, Mathematics of Computation, vol. 56, number 194, april 1991, pp. 823-849.

A Proof of lemma 4

From the conditions of lemma 4, we obtain:

$$\frac{hq}{N} \geq N^{9\alpha} \quad \text{and} \quad \frac{N}{k^2 q} \leq N^{6\alpha} \tag{9}$$

which gives $N \leq 2 \cdot k \cdot q \cdot \sqrt{h}$ and then $\nu_1 < \nu_2$.

Recall that $j(p,q)$ denotes the number of integers in interval $J(p,q)$. From lemma 2 the length of $J(p,q)$ is at least $N/(2kq)$ and therefore, $j(p,q) \geq N/(2kq) - 1$, which gives using $k \geq 6$:

$$\frac{j(p,q)}{q} \geq \frac{N^{3\alpha}}{3} \tag{10}$$

Let us denote by $n(\nu)$ the number of points of $P(p,q)$ on a line $D(\nu)$. The distance between the abcissae of two consecutive points of $P(p,q)$ on a line $D(\nu)$ is equal to q. Therefore, for all indices ν, we have $n(\nu) \leq \lfloor j(p,q)/q \rfloor + 1$. Moreover, for $\nu_1 \leq \nu \leq \nu_2$, $n(\nu)$ is either $\lfloor j(p,q)/q \rfloor$ or $\lfloor j(p,q)/q \rfloor + 1$. This gives the following bound for $b(p,q)$:

$$(\nu_2 - \nu_1 - 1) \cdot \left(\frac{j(p,q)}{q} - 1 \right) \leq b(p,q) \leq (\nu_3 + 1) \cdot \left(\frac{j(p,q)}{q} + 1 \right)$$

which gives using (4), (5), (9), (10) and $N^\alpha \geq 3$:

$$\left| b(p,q) - \frac{h \cdot j(p,q)}{N} \right| \leq \frac{4h \cdot j(p,q)}{N} N^{-3\alpha} \tag{11}$$

Let n' be the number of indices ν such that $\nu_1 \leq \nu \leq \nu_2$. We have $n' = \lfloor \nu_2 - \nu_1 \rfloor$ or $n' = \lfloor \nu_2 - \nu_1 \rfloor + 1$. The probability that $\mathcal{G}(p,q)$ generates an element $x \in B(p,q)$ corresponding to a point of index ν is given by:

$$\Pr[x] = P(\nu) = \frac{1}{n' \cdot n(\nu)}$$

for $\nu_1 \leq \nu \leq \nu_2$ and $P(\nu) = 0$ otherwise. The number of integers $x \in B(p,q)$ such that $\Pr[x] = 0$ is then at most:

$$(\nu_1 + \nu_3 - \nu_2 + 2) \cdot \left(\frac{j(p,q)}{q} + 1 \right) \leq N^{6\alpha} \cdot \frac{j(p,q)}{q} \tag{12}$$

For all $\nu_1 \leq \nu \leq \nu_2$, we have using (4), (5), (9), (10), (11) and $N^\alpha \geq 3$:

$$\left| P(\nu) - \frac{1}{b(p,q)} \right| \leq 10 \cdot \frac{N}{h \cdot j(p,q)} \cdot N^{-3\alpha} \tag{13}$$

Eventually, the statistical distance from the uniform distribution is:

$$\delta_G = \frac{1}{2} \sum_{x \in B(p,q)} \left| \Pr[x] - \frac{1}{b(p,q)} \right|$$

and we obtain using (11), (12) and (13):

$$\delta_G \leq 7 \cdot N^{-3\alpha}$$

B Proof of lemma 5

Let us denote $q_m = N^{1/3-4\alpha}$. For $q \geq q_m$, the probability to generate $p/q \in \mathcal{F}_k$ using $\mathcal{V}$ is $j(p,q)/|\mathbb{Z}_N^+|$. Moreover, using lemma 2, the probability that $\mathcal{V}$ generates $\perp$ is at most:

$$\Pr[\perp] = \sum_{\mathcal{F}_k | q < q_m} \frac{2 \cdot j(p,q)}{N+1} \leq 3\frac{q_m}{k} \leq 3 \cdot N^{-3\alpha} \tag{14}$$

Consequently, the statistical distance δ_V between $\mathcal{D}$ and the distribution induced by $\mathcal{V}$ is at most:

$$\delta_V = \frac{1}{2} \sum_{\mathcal{F}_k | q \geq q_m} \left| \frac{2 \cdot j(p,q)}{N+1} - \frac{b(p,q)}{b} \right| + \frac{1}{2} \Pr[\perp] + \frac{1}{2} \sum_{\mathcal{F}_k | q < q_m} \frac{b(p,q)}{b} \tag{15}$$

Let ℓ be the size of N in bits. From lemma 1, we obtain for $\ell \geq 64$:

$$|b - \frac{h}{2}| \leq 4 \cdot \ell \cdot 2^{\ell/2} \leq \frac{1}{2} \cdot N^{2/3} \leq \frac{h}{2} \cdot N^{-3\alpha} \tag{16}$$

For $q \geq q_m$, we obtain from Lemma 4 and (16):

$$\left| \frac{b(p,q)}{b} - \frac{2 \cdot j(p,q)}{N+1} \right| \leq \frac{12 \cdot j(p,q)}{N+1} \cdot N^{-3\alpha} \tag{17}$$

This gives:

$$\sum_{\mathcal{F}_k | q < q_m} \frac{b(p,q)}{b} = 1 - \sum_{\mathcal{F}_k | q \geq q_m} \frac{b(p,q)}{b} \leq 1 - (1 - 6 \cdot N^{-3\alpha}) \cdot \sum_{\mathcal{F}_k | q \geq q_m} \frac{2 \cdot j(p,q)}{N+1}$$

From (14) and using:

$$\sum_{\mathcal{F}_k} \frac{2 \cdot j(p,q)}{N+1} = 1$$

we obtain:

$$\sum_{\mathcal{F}_k | q < q_m} \frac{b(p,q)}{b} \leq 9 \cdot N^{-3\alpha} \tag{18}$$

From equation (15) and inequalities (14), (17) and (18), we obtain:

$$\delta_V \leq 9 \cdot N^{-3\alpha}$$

C Proof of theorem 3

The generator $\mathcal{G}$ combines the generators $\mathcal{V}$ and $\mathcal{G}(p,q)$. Moreover, $\mathcal{V}$ generates $p/q \in \mathcal{F}_k$ such that the statistical distance δ_G of the distribution induced by $\mathcal{G}(p,q)$ from the uniform distribution in $B(p,q)$ is at most $7 \cdot N^{-3\alpha}$. Therefore the statistical distance δ of $\mathcal{G}$ from the uniform distribution in B is at most:

$$\delta \leq \delta_V + \delta_G \leq 16 \cdot N^{-3\varepsilon/13}$$