From fixed-length to arbitrary-length RSA padding schemes

Jean-Sébastien Coron
Ecole Normale Supérieure
45 rue d'Ulm
Paris, F-75005, France
coron@clipper.ens.fr

Francois Koeune
UCL Crypto Group
Bâtiment Maxwell, place du Levant 3
Louvain-la-Neuve, B-1348, Belgium
fkoeune@dice.ucl.ac.be

David Naccache
Gemplus Card International
34 rue Guynemer
Issy-les-Moulineaux, F-92447, France
david.naccache@gemplus.com

Abstract. A common practice for signing with RSA is to first apply a hash function or a redundancy function to the message, add some padding and exponentiate the resulting padded message using the decryption exponent. This is the basis of several existing standards.
In this paper we show how to build a secure padding scheme for signing arbitrarily long messages with a secure padding scheme for fixed-size messages. This focuses more sharply the question of finding a secure encoding for RSA signatures, by showing that the difficulty is not in handling messages of arbitrary length, but rather in finding a secure redundancy function for short messages, which remains an open problem.

Key words : Signature scheme, provable security, padding scheme.

1 Introduction

Since the discovery of public-key cryptography by Diffie and Hellman [4], one of the most important research topics has been the design of practical and provably secure cryptosystems. A proof of security is usually a computational reduction between breaking the cryptosystem and solving a well established problem such as factoring large integers, computing the discrete logarithm modulo a prime p or extracting a root modulo a composite integer. RSA [10] is based on this last problem.

A common practice for signing with RSA is to first apply a hash (or a redundancy) function to the message m, add some padding and raise the

padded message to the decryption exponent. This is the basis of numerous standards such as ISO/IEC-9796-1 [6], ISO 9796-2 [7] and PKCS#1 V2.0 [8].

Many padding schemes have been designed and many have been broken (see [9] for a survey). The Full Domain Hash (FDH) scheme and the Probabilistic Signature Scheme (PSS) [2] were among the first practical and provably secure signature schemes. Those schemes are provably secure in the random oracle model [1], in which the hash function is assumed to behave as a truly random function.

However, security proofs in the random oracle model are not "real" proofs, and can be only considered as heuristic, since in the real world the random oracle is replaced by a function which can be computed by all parties. A recent result by Canneti, Goldreich and Halevi [3] shows that a security proof in the random oracle does not necessarily imply security in the "real world".

In this paper we do not model hash functions as random oracles nor assume the existence of collision-resistant hash-functions. Instead, we assume the existence of a secure deterministic padding function μ for signing fixed-length message and show how to build a secure padding scheme for signing arbitrarily long messages. This focuses more sharply the question of finding a secure encoding for RSA signatures, by showing that the difficulty is not in handling messages of arbitrary length, but rather in finding a secure redundancy function for short messages, which remains an open problem.

2 Definitions

2.1 Signature schemes

The digital signature of a message m is a string which depends on m and on some secret known only to the signer, in such a way that anyone can check the validity of the signature. The following definitions are based on [5].

Definition 1 (signature scheme). *A signature scheme is defined by the following :*

- The key generation algorithm Generate *is a probabilistic algorithm which given* 1^k*, outputs a pair of matching public and secret keys,* $\{\mathsf{pk}, \mathsf{sk}\}$.

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

- *The verification algorithm* Verify *takes a message M, a candidate signature x' and the public key* pk*. It returns a bit $\mathsf{Verify}_{\mathsf{pk}}(M, x')$, equal to 1 if the signature is accepted, and 0 otherwise. We require that if $x \leftarrow \mathsf{Sign}_{\mathsf{sk}}(M)$, then $\mathsf{Verify}_{\mathsf{pk}}(M, x) = 1$.*

2.2 Security of signature schemes

The security of signature schemes was formalized in an asymptotic setting by Goldwasser, Micali and Rivest [5]. Here we use the definitions of [2] which provide a framework for the concrete security analysis of digital signatures. Resistance against adaptative chosen-message attacks is considered : a forger $\mathcal{F}$ can dynamically obtain signatures of messages of its choice and attempts to output a valid forgery. A *valid forgery* is a message/signature pair $\{M, x\}$ such that $\mathsf{Verify}_{\mathsf{pk}}(M, x) = 1$ whilst the signature of M was never requested by $\mathcal{F}$.

Definition 2. *A forger $\mathcal{F}$ is said to (t, q_{sig}, ϵ)-break the signature scheme* {Generate, Sign, Verify} *if after at most q_{sig} signature queries and t processing time, it outputs a valid forgery with probability at least ϵ.*

Definition 3. *A signature scheme* {Generate, Sign, Verify} *is (t, q_{sig}, ϵ)-secure if there is no forger who (t, q_{sig}, ϵ)-breaks the scheme.*

2.3 The RSA cryptosystem

RSA [10] is the most widely used public-key cryptosytem. It may be used to provide both secrecy and digital signatures.

Definition 4 (The RSA cryptosystem). *RSA is a family of trapdoor permutations. It is specified by :*

- *The RSA generator $\mathcal{RSA}$, which on input 1^k, randomly selects 2 distinct $k/2$-bit primes p and q and computes the modulus $N = p \cdot q$. It randomly picks an encryption exponent $e \in \mathbb{Z}^*_{\phi(N)}$ and computes the corresponding decryption exponent d such that $e \cdot d = 1 \bmod \phi(N)$. The generator returns $\{N, e, d\}$.*

- *The encryption function $f : \mathbb{Z}^*_N \to \mathbb{Z}^*_N$ defined by $f(x) = x^e \bmod N$.*

- *The decryption function $f^{-1} : \mathbb{Z}^*_N \to \mathbb{Z}^*_N$ defined by $f^{-1}(y) = y^d \bmod N$.*

2.4 The standard RSA signature scheme

Let μ be a padding function taking as input a message of size $k+1$ bits and returning an integer of size k bits. We consider in figure 1 the classical RSA signature scheme {Generate, Sign, Verify} which signs fixed-length $k+1$-bits messages.

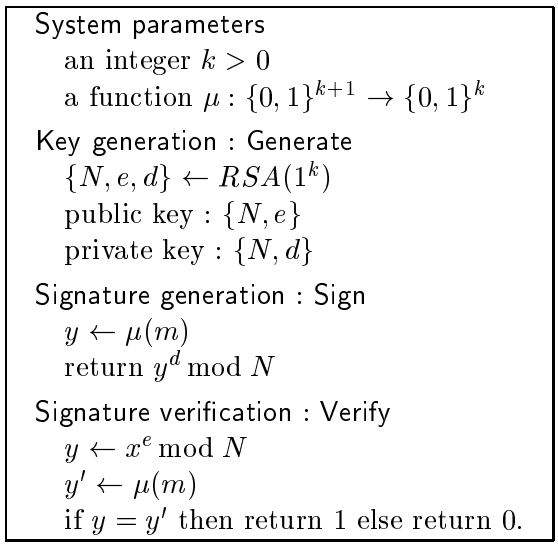

Fig. 1. The classical RSA scheme using function μ for signing fixed-length messages.

3 The new construction

We construct in figure 2 a new signature scheme {Generate', Sign', Verify'} using function μ. The new construction enables to sign messages of size $2^a \cdot (k-a)$ bits where a is comprised between 0 and $k-1$ and k is the size of the modulus in bits. The maximum length that can be handled is then 2^{k-1} bits for $a = k-1$ or $a = k-2$. The construction can be recursively iterated to sign messages of arbitrary length. For bit strings m_1 and m_2, we let $m_1||m_2$ denote the concatenation of m_1 and m_2.

This construction preserves the resistance against adaptive chosen message attack of the signature scheme :

System parameters
  an integer $k > 0$
  an integer $a \in [0, k-1]$
  a function $\mu : \{0,1\}^{k+1} \rightarrow \{0,1\}^k$

Key generation : Generate$'$
  $\{N, e, d\} \leftarrow RSA(1^k)$
  public key : $\{N, e\}$
  private key : $\{N, d\}$

Signature generation : Sign$'$
  Split the message m into blocks of size $k-a$ bits
  such that $m = m[1]||\ldots||m[r]$.
  let $\alpha = \prod_{i=1}^{r} \mu(0||i||m[i]) \bmod N$
  where i in $0||i||m[i]$ is the a-bit string representing i.
  let $y \leftarrow \mu(1||\alpha)$
  return $y^d \bmod N$

Verification : Verify$'$
  $y \leftarrow x^e \bmod N$
  let $\alpha = \prod_{i=1}^{r} \mu(0||i||m[i]) \bmod N$
  let $y' \leftarrow \mu(1||\alpha)$
  if $y = y'$ then return 1 else return 0.

Fig. 2. The new construction using function μ for signing long messages.

Theorem 1. *If the signature scheme* {Generate, Sign, Verify} *is* (t, q_{sig}, ϵ) *secure, then the signature scheme* {Generate', Sign', Verify'} *which signs messages of length* $2^a \cdot (k-a)$ *bits is* $(t', q'_{sig}, \epsilon')$ *secure, where :*

$$t' = t - 2^a \cdot q_{sig} \cdot \mathcal{O}(k^2) \tag{1}$$

$$q'_{sig} = q_{sig} - 2^{a+1} \tag{2}$$

$$\epsilon' = \epsilon \tag{3}$$

Proof. Let $\mathcal{F}'$ be a forger that $(t', q'_{sig}, \epsilon')$-breaks the signature scheme {Generate', Sign', Verify'} . We construct a forger $\mathcal{F}$ that (t, q_{sig}, ϵ)-breaks the signature scheme {Generate, Sign, Verify} using $\mathcal{F}'$. The forger $\mathcal{F}$ has oracle access to a signer $\mathcal{S}$ for the signature scheme {Generate, Sign, Verify}

and its goal is to produce a forgery for {Generate, Sign, Verify} . The forger $\mathcal{F}$ will answer the signature queries of $\mathcal{F}'$ itself.

The forger $\mathcal{F}$ is given as input $\{N, e\}$ where N, e were obtained by running Generate. It starts running $\mathcal{F}'$ with the public key $\{N, e\}$.

When $\mathcal{F}'$ asks the signature of the j-th message m_j with $m_j = m_j[1]||\ldots||m_j[r_j]$, $\mathcal{F}$ computes :

$$\alpha_j = \prod_{i=1}^{r_j} \mu(0||i||m_j[i]) \bmod N$$

and requests from $\mathcal{S}$ the signature $s_j = \mu(1||\alpha_j)^d \bmod N$ of the message $1||\alpha_j$, and returns s_j to $\mathcal{F}'$. Let q be the total number of signatures requested by $\mathcal{F}'$.

Eventually $\mathcal{F}'$ outputs a forgery $\{m', s'\}$ for the signature scheme {Generate', Sign', Verify'} with $m' = m'[1]||\ldots||m'[r']$, from which $\mathcal{F}$ computes :

$$\alpha' = \prod_{i=1}^{r'} \mu(0||i||m'[i]) \bmod N$$

We distinguish two cases :

First case : $\alpha' \notin \{\alpha_1, \ldots, \alpha_q\}$. In this case $\mathcal{F}$ outputs the forgery $\{1||\alpha', s'\}$ and halts. This is a valid forgery for the signature scheme {Generate, Sign, Verify} since $s' = \mu(1||\alpha')^d$ and the signature of $1||\alpha'$ was never asked to the signer $\mathcal{S}$.

Second case : $\alpha' \in \{\alpha_1, \ldots, \alpha_q\}$, so there exist c such that $\alpha' = \alpha_c$. Let denote $m = m_c$, $\alpha = \alpha_c$ and $r = r_c$. We have :

$$\prod_{i=1}^{r'} \mu(0||i||m'[i]) \bmod N = \prod_{i=1}^{r} \mu(0||i||m[i]) \bmod N \tag{4}$$

The message m' is distinct from the message m because the signature of m has been requested by $\mathcal{F}'$ whereas the signature of m' was never requested by $\mathcal{F}$, since m' is the message of the forgery. Consequently there exist an integer j such that :

$$0||j||m'[j] \notin \{0||1||m[1], \ldots, 0||r||m[r]\} \tag{5}$$

or

$$0||j||m[j] \notin \{0||1||m'[1], \ldots, 0||r'||m'[r']\} \tag{6}$$

We assume that condition (5) is satisfied (condition (6) leads to the same result). In this case $\mathcal{F}$ asks $\mathcal{S}$ for the signatures x'_i of the messages $0||i||m'[i]$ for $i \in [1, r']$ and $i \neq j$, and the signatures x_i of the messages $0||i||m[i]$ for $i \in [1, r]$. Since from (4) :

$$\mu(0||j||m'[j]) = \Big(\prod_i \mu(0||i||m[i]\Big)\Big(\prod_{i \neq j} \mu(0||j||m'[j])\Big)^{-1} \bmod N$$

the forger $\mathcal{F}$ can compute the signature of $0||j||m'[j]$ from the other signatures :

$$x'_j = \mu(0||j||m'[j])^d = \Big(\prod_i x_i\Big)\Big(\prod_{i \neq j} x'_j\Big)^{-1} \bmod N$$

and $\mathcal{F}$ finally outputs the forgery $\{0||j||m'[j], x'_j\}$. This is a valid forgery for the signature scheme {Generate, Sign, Verify} since the signature of $0||j||m'[j]$ was never asked from the signer $\mathcal{S}$.

We assume that μ can be computed in time linear in k, as is the case for most padding functions. The running time of $\mathcal{F}$ is then the running time of $\mathcal{F}'$ plus the time necessary for the multiplications modulo N, which is quadratic.

□

Note that q_{sig} must me greater than 2^{a+1} so that equation (2) holds. The security reduction is tight : the probability of success of $\mathcal{F}$ is exactly the probability of success of $\mathcal{F}'$.

4 Conclusion and further research

We have reduced the problem of designing a secure deterministic general-purpose RSA padding scheme to the problem of designing a one block secure padding scheme, by providing an efficient and secure tool to extend the latter into the former. As stated previously, this focuses more sharply the question of finding a secure encoding for RSA signatures, by showing that the difficulty is not in handling messages of arbitrary length, but rather in finding a secure redundancy function for short messages, which remains an open problem.

Our construction assumes that the padding function μ takes as input messages larger than the modulus; padding schemes such as ISO/IEC 9697-1 are consequently uncovered. A possible line of research could be a construction similar to ours, using a small (1024-bit) inner modulus and a larger (2048-bit) outer modulus.

5 Acknoledgements

We thank Jean-Marc Robert and Geneviève Arboit for useful discussions and the anonymous referees for their comments.

References

1. 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.
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. R. Canetti, O. Goldreich and S. Halevi, *The Random Oracle Methodology, Revisited*, STOC '98, ACM, 1998.
4. W. Diffie and M. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory, IT-22, 6, pp. 644-654, 1976.
5. 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.
6. ISO/IEC 9796, *Information technology - Security techniques - Digital signature scheme giving message recovery, Part 1 : Mechanisms using redundancy*, 1999.
7. ISO/IEC 9796-2, *Information technology - Security techniques - Digital signature scheme giving message recovery, Part 2 : Mechanisms using a hash-function*, 1997
8. RSA Laboratories, PKCS #1 : *RSA cryptography specifications*, version 2.0, September 1998.
9. J.F. Misarsky, *How (not) to design signature schemes*, proceedings of PKC'98, Lecture Notes in Computer Science vol. 1431, Springer Verlag, 1998.
10. R. Rivest, A. Shamir and L. Adleman, *A method for obtaining digital signatures and public key cryptosystems*, CACM 21, 1978.