# On-line/off-line RSA-like

Marc Girault and Jean-Claude Paillès

*France Telecom R&D, 42 rue des Coutures*
*BP 6243, F-14066 Caen Cedex 4, France*
marc.girault@francetelecom.com
jeanclaude.pailles@francetelecom.com

**Abstract**

We present a variant of RSA which can be used in an on-line/off-line mode. More precisely, we present an asymmetric authentication/digital signature scheme which combines the following properties : a) the key pair is a RSA key pair, b) it is secure if factoring is hard, but c) almost all the computations can be made in advance. Such properties make this scheme a very attractive alternative to RSA when the execution time of the prover/signer is a critical parameter (e.g. in contactless transactions).

***Keywords.*** Authentication, digital signatures, off-line/on-line, on the fly, RSA, server-aided verification.

## 1 Introduction

It is well-known that one drawback of the RSA digital signature scheme [15] is the large number of operations that the signer must carry out, leading to a frequently excessive execution time. One solution is to use a specific cryptoprocessor (in addition to the general-purpose (micro-)processor) in order to speed up the signature generation, but this is costly : smart cards that use such a cryptoprocessor may be between 10 and 50% as expensive as standard smart cards.

A second solution is to use special moduli $n$, such as those in the form $n = pqr$ or $n = p^2q$, where $p$, $q$ and $r$ are prime numbers, and to apply the Chinese Remainder Technique, in order to make the signature generation two or three times as efficient as with a standard modulus (in the form $n = pq$).

Another solution is to make things such that most of the signer computations can be performed in advance, that is before the actual time of signing or interacting with a verifier: this is known as the on-line/off-line concept [2]. At Crypto 2001, Shamir and Tauman have defined a generic method (say ST) for using any digital signature scheme in an off-line/on-line way, by the means of a trapdoor hash-function [16].

Here, we propose an authentication/digital signature scheme which has the following properties:

a) the key pair is a RSA key pair (there is no extra key or parameter)

b) it is secure if factoring is hard (consequently it is as least as secure as RSA)

c) almost all computations can be made in advance ("off-line") and the "on-line" computation takes essentially no time

d) the results of the off-line computations can be made very short (down to 50 bits under some assumptions)

e) the verification can be server-aided (so as to be almost as efficient as a Guillou-Quisquater [8] verification)

Note that property a) does not mean that the scheme generates same signatures as RSA scheme would have itself generated, only that the key pairs are RSA key pairs. We also point out that RSA key pairs issued from standard key generation algorithms do work with our scheme, but that "unusual" key pairs may not work (e.g. when the public exponent is not prime).

Such properties make this scheme very attractive to replace RSA, in case RSA cannot be used for performance and/or cost reasons. The reasons why are the following : first, there is no need to change the Public Key Infractructure (PKI), since the public keys are RSA keys. This is a crucial point, as carrying out a PKI is a very challenging and long process. Second, there is no risk to replace RSA with this scheme, since its security is equivalent to factorisation intractability (or equivalent to RSA in some variants). Finally, there is even no need to change the key generator nor the key formatting used in the data memories, since the private key is also a RSA private key.

This scheme is different from the ST construction applied to RSA in that, on one hand, the signatures from our scheme are not RSA signatures[1], but on the other hand, no additional specific hash-function, keys and parameters (compared to regular RSA) are required. Furthermore, many optimisations are possible (see properties d and e), while ST makes verification much slower than with RSA. In summary, ST construction is much more general while the present scheme is dedicated to RSA but is much more compact and efficient.

## 2 The new scheme

### 2.1 Preliminaries

Let $(n, e, d)$ be a RSA key, i.e. let $n$ be a composite modulus, $e$ and $d$ integers such that $ed = 1 \pmod{\varphi(n)}$, where $\varphi(n)$ is the Euler function of $n$. We address the following issue: how to demonstrate the knowledge of the private key $d$, after revealing the public key $(n, e)$ ?

[1] Actually, the ST method appends to the regular signature the result of a specific hash-function (in addition to the hash-function used for hashing the message), so that a ST-signature applied to RSA is not reduced to a single RSA signature, and the verification process is not reduced to a single RSA verification process.

The first possibility is well-known: the prover computes $y = c^d \pmod n$, where $c$ is for example a random integer $c$ chosen by the verifier in the interval $[0, n-1]$. This is the famous RSA process. By replacing $c$ with the image of a message $m$ by an appropriate function (such as those specified in the standard PKCS#1), $y$ becomes the RSA signature of $m$.

Another possibility is to show the knowledge of $d$ in the zero-knowledge paradigm. The most practical approach consists in choosing an (appropriate) integer $f$ smaller than $n$, and prove in one way or another that:

$$f^{ed} = f \pmod n$$

Again, there are two possibilities : either rewrite this equation as follows : $(f^d)^e = f \pmod n$, and prove that you know $f^d \pmod n$ without revealing it (i.e. roughly that you know a RSA signature of $f$ without revealing it) : this is achieved by the Guillou-Quisquater (GQ) protocol. Or rewrite this equation as follows : $(f^e)^d = f \pmod n$ and prove that you know a discrete logarithm of $f$ in base $g = f^e \pmod n$ without revealing it : this is achieved by the scheme presented in this paper, which can from the previous discussion be viewed as "dual" to the GQ scheme.

Note that close issues are addressed in [14] and [1], but in a different context and with different solutions. Note also that [13] provides a scheme which is related to the present one, but the private key is different from a RSA private key (explicitly it is equal to $n - \varphi(n)$).

## 2.2 Description

The authentication scheme is obtained by iterating $t$ times the following three-pass protocol[2]:

| | | |
|---|---|---|
| Parameters: | $n$ a composite modulus<br>$e$ a prime integer and $d$ an integer such that $ed = 1 \pmod{\varphi(n)}$<br>$f, g \in \mathbb{Z}_n^*$ with $g = f^e \pmod n$ | |
| Secret key: | $d$ | |
| Public key: | $(n, e)$ | |
| **Prover** | | **Verifier** |
| Choose $r \in [0, \varphi(n)[$ | | |
| Compute $x = g^r \pmod n$ | | |
| | $\xrightarrow{x}$ | |
| | | Choose $c \in [0, e[$ |
| | $\xleftarrow{c}$ | |
| Check $c \in [0, e[$ | | |
| Compute $y = (r - cd) \pmod{\varphi(n)}$ | | |
| | $\xrightarrow{y}$ | |
| | | Check $y < n$ and $g^y f^c = x \pmod n$ |

**Figure 1 - On-line/off-line RSA-like scheme (authentication protocol)**

[2] as usual, $\mathbb{Z}_n^*$ denotes the multiplicative group of the ring of integers modulo $n$.

Typically:

- $n$ is at least 1024-bit long, so that factorization algorithms are inefficient
- $e$ is 16, 32 or 64-bit long (e.g. $e = 65537$)
- $f = 2$ (or $g = 2$) in order to speed-up exponentiations (these choices are not compatible with any value of $n$, see discussion below).

More generally, $f$ will preferably be fixed to a given value for a group of users, so that neither $f$ nor $g$ needs to be part of the public key. The way $n$ is generated will depend on this value of $f$. More precisely, $n$ and $f$ must be chosen so that the order of $f$ modulo $n$ be "sufficiently" large (and the discrete logarithm modulo $n$ in base $f$ a "sufficiently" hard problem). Ideally, $f$ will be of order close or equal to the maximum possible value $\lambda(n)$, where $\lambda(n)$ denotes the "Carmichaël function" of $n$. For example, we will choose $n$ as the product of two distinct large safe primes $p$ and $q$ [3], and $g$ of order $\lambda(n) = \frac{(p-1)(q-1)}{2}$ or $\frac{1}{2}\lambda(n)$. It happens that, for such a choice of $n$, "almost any" integer between 2 and $n$ is such a "good" $g$ (and that a very simple test allows to check it). Moreover, $f = 2$ or $g = 2$ is *always* good.

Of course, $\varphi(n)$ can be replaced everywhere in the protocol by $\lambda(n)$, or more generally by any multiple of the order of $g$ modulo $n$.

As claimed, the most consuming part of the protocol (from the prover's point of view) is the computation of $x$ and can be made before interacting with the verifier. Sometimes, the couple $(r, x)$ is called a coupon [11]. The possibility of pre-computing coupons is the main advantage of this variant of RSA. With regular RSA, no values can be pre-computed and the on-line computation is time-expensive. With this scheme, there is only one modular multiplication and one modular addition to do at the time of the transaction. Even in a standard smart card, the time of this operation is masked by all the rest (other computations, R/W operations, communication time etc.).

Note that $x$ can be hashed and thus replaced by $x = H(g^r \pmod n)$ where $H$ is a hash-function. In that case, the verification equation becomes $H(g^y f^c \pmod n) = x$. It allows to store many more coupons in the same memory size, especially if $r$ is generated by a pseudo-random number generator and can be regenerated at the time of the transaction (so that only the values of $x$ have to be stored). Finally, by using some tricks (see e.g. [9] and [5]), $x$ can be made as short as 80-90 bits and even 40-50 bits under the assumption that the verifier controls that the protocol be executed within a limited period of time.

Note also that a message $M$ can be authenticated (not only the identity) by using a (possibly different) hash-function $H'$ in the following way : $x = H'(g^r \pmod n, M)$. In that case the equation verification becomes $H'(g^y f^c \pmod n, M) = x$.

Finally, the computation of $x$ can be accelerated by using the well-known Chinese Remainder Technique.

[3] A prime $p$ is safe if $\frac{p-1}{2}$ is prime.

### 2.3 Security properties

Let us briefly consider the three conditions that a zero-knowledge protocol must satisfy.

a) Completeness is straight-forward :

$$g^y f^c = g^{r-dc \bmod(\varphi(n))} f^c = g^{r-dc} f^c = f^{er-edc+c} = f^{er+c(1-ed)} = f^{er} = g^r = x \pmod n$$

b) Soundness can be proved by exhibiting the following (informal) extractor : assume that a machine can answer correctly to two different challenge values $c$ and $c'$ for the same value of $x$. This means that this machine knows $y$ and $y'$ such that :

$$g^y f^c = g^{y'} f^{c'} \pmod n$$

which is equivalent to $f^{ey+c} = f^{ey'+c'} \pmod n$, i.e. $f^{e(y-y')+(c-c')} = 1 \pmod n$.
Assume (without loss of generality) that $c > c'$. Since $c - c'$ is greater than 0 and less than $e$, the exponent of $f$ is not equal to zero modulo $e$ and therefore is not equal to zero. Hence it is a multiple of the order of $f$ modulo $n$. In case $n$ and $f$ are chosen as recommended in section 2.2, this exponent is a multiple of $\lambda(n)$(or $\frac{1}{2}\lambda(n)$). Note that this multiple is "small" since $y$ and $y'$ are less than $n$. By a very classical result from Miller [10], such a multiple reveals the factorization of $n$. As a consequence, assuming factorization is hard, no other entity than the prover can answer to two different values of the challenge and the success probability of an impersonator is bounded by $(\frac{1}{e})^t$. Typical values are $e = 65537$ and $1 \leq t \leq 4$.

c) Now, the zero-knowledge property. In a few words, the simulator chooses at random $c$ in $[0, e[$ and $y$ in $[0, n[$. Then he computes $x$ from the verification equation and sends it to the verifier. If and only if the verifier sends the challenge $c$, the triplet $(x, c, y)$ is an output of the simulator. The expected time is $t \times e$, and its distribution probability is statistically indistinguishable from the true distribution. Roughly speaking, this is because $n$ is "very close" to $\varphi(n)$ (the difference is in the order of the square root of $n$ if the two factors of $n$ have the same size). As usual, only honest-verifier zero-knowledgeness is achieved when the number $t$ of rounds is constant (e.g. equal to 1). General (statistical) zero-knowledge would require typically that $t$ and $e$ grow polynomially with the size of $n$.

### 2.4 On the fly version

Here, we show an "on the fly" variant, which is inspired from the Girault-Poupard-Stern (GPS) scheme ([4],[12],[6]). It allows not to make any modular reduction during the on-line computation. The counterpart is that $r$ must be chosen a little bit larger and consequently the response $y$ also.

| | | |
|---|---|---|
| Parameters: | $n$ a composite modulus | |
| | $e$ a prime integer and $d$ an integer such that $ed = 1 \pmod{\varphi(n)}$ | |
| | $f, g \in \mathbb{Z}_n^*$ with $g = f^e \pmod{n}$ | |
| | $A$ an integer $>> e.n$ | |
| Secret key: | $d$ | |
| Public key: | $(n, e)$ | |
| **Prover** | | **Verifier** |
| Choose $r \in [0, A[$ | | |
| Compute $x = g^r \pmod{n}$ | | |
| | $\xrightarrow{x}$ | |
| | | Choose $c \in [0, e[$ |
| | $\xleftarrow{c}$ | |
| Check $c \in [0, e[$ | | |
| Compute $y = r - cd$ | | |
| | $\xrightarrow{y}$ | |
| | | Check $y \in [-(e-1)(n-2), A[$ |
| | | Check $g^y f^c = x \pmod{n}$ |

**Figure 2 - On the fly RSA-like**

Note that the on the fly version can be used in conjunction with any of the variants or extensions described in this paper.

### 2.5 Case $f = 2$

The choice $f = 2$ allows to speed up the computation of $x$ as well as the verification.

| | | |
|---|---|---|
| Parameters: | $n$ a composite modulus | |
| | $e$ a prime integer and $d$ an integer such that $ed = 1 \pmod{\varphi(n)}$ | |
| Secret key: | $d$ | |
| Public key: | $(n, e)$ | |
| **Prover** | | **Verifier** |
| Choose $r \in [0, \varphi(n)[$ | | |
| Compute $r' = er \pmod{\varphi(n)}$ | | |
| Compute $x = 2^{r'} \pmod{n}$ | | |
| | $\xrightarrow{x}$ | |
| | | Choose $c \in [0, e[$ |
| | $\xleftarrow{c}$ | |
| Check $c \in [0, e[$ | | |
| Compute $y = (r - cd) \pmod{\varphi(n)}$ | | |
| | $\xrightarrow{y}$ | |
| | | Check $y < n$ and $2^{ey+c} = x \pmod{n}$ |

**Figure 3 - On-line/off-line RSA-like with $f = 2$**

# 3 Extensions

## 3.1 Digital signature scheme

By using a standard method from Fiat and Shamir [3], the authentication scheme can be easily turned into a digital signature scheme as follows:

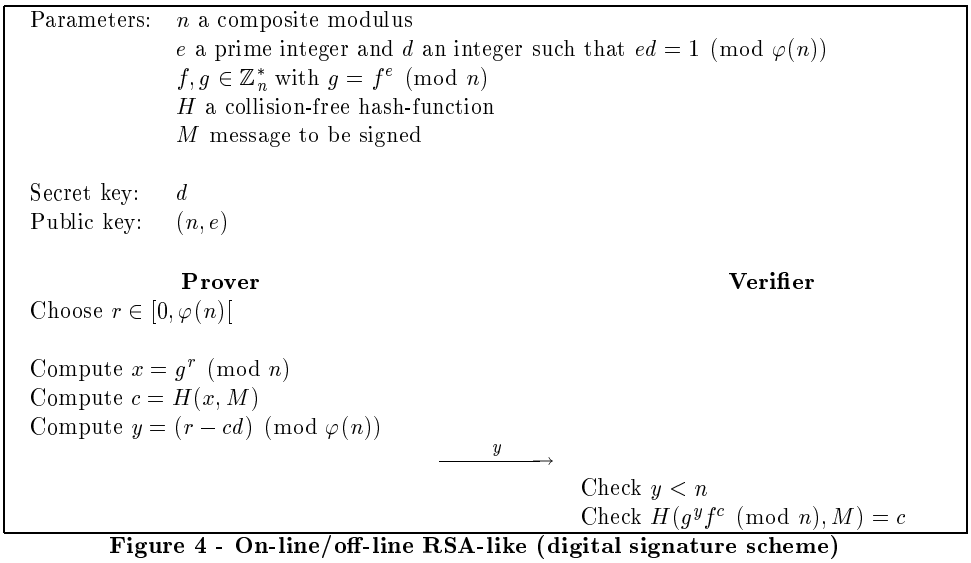


**Figure 4 - On-line/off-line RSA-like (digital signature scheme)**

The same sizes/values of parameters can be used, except $c$, which should be at least 160-bit long, in order to prevent from finding collisions on $H$ by using a birthday attack.

## 3.2 Server-aided verification

Verification step is not so fast in this scheme, since it involves an exponentiation with a somewhat large exponent. In contrast, verification is fast in the GQ scheme. A key observation made in [7] is that GPS verification step can be transformed into a GQ verification step, provided the exponentiation can be delegated to a (powerful) server. This implies to add at least a new (public) parameter to those already existing in the GPS scheme, namely the "GQ exponent". Another observation is that this server can be any third party, including an untrusted one. This is useful in environments where a) the whole transaction must be very rapid, b) the (secure) verification chip is not powerful enough but is embedded in a device including another (insecure but powerful) chip. Such a situation may occur e.g. in a card-reader device or in a mobile telephone.

In the RSA-like scheme, that a public exponent is already part of the public key allows to integrate the server-aided verification option without modifying the set-up: the decision to use (option 2) or not (option 1) this possibility can be made at the very last moment by the verifier, and therefore needs not be anticipated. With option 1, the security of the scheme is

equivalent to factorisation, while in option 2 it is equivalent to RSA[4]. The resulting scheme can be described as follows:

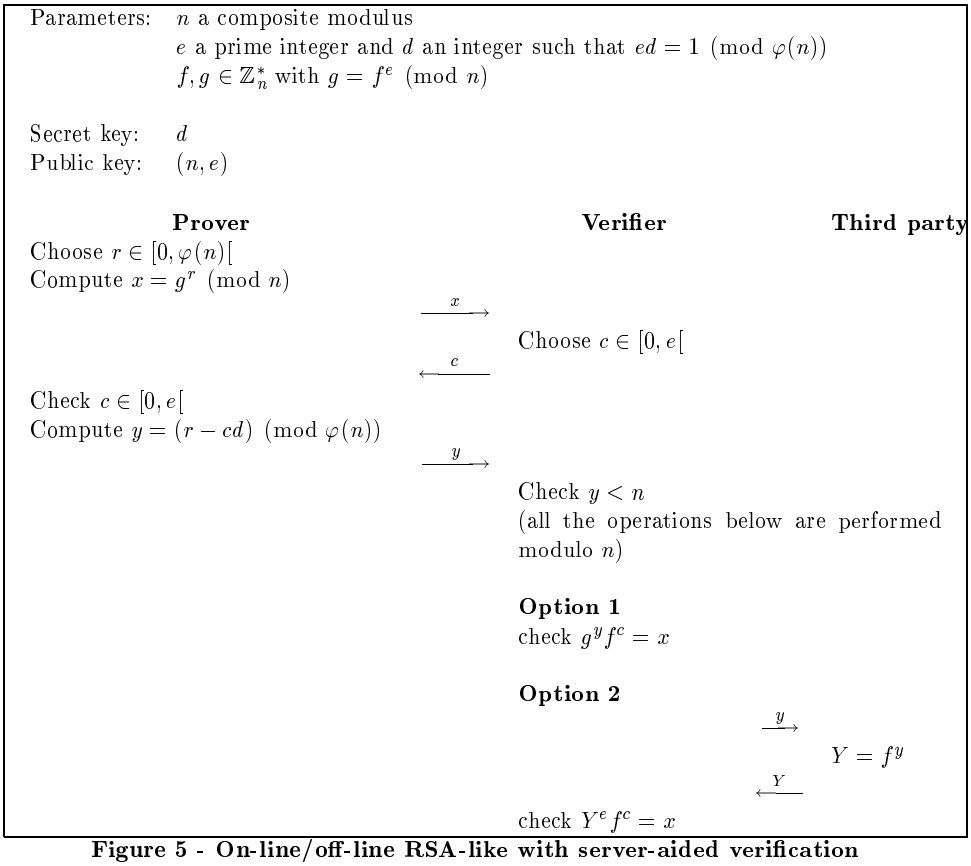


**Figure 5 - On-line/off-line RSA-like with server-aided verification**

## 4 Conclusion

We have presented an authentication/digital signature scheme which combines compatibility with RSA (the keys are the same), factorisation-based security and on-line/off-line computation. Moreover it supports the so-called "server-aided verification" option, which is of interest when transaction time is a critical parameter. For all these reasons, this scheme is a good candidate to replace RSA in all environments where RSA is not efficient enough.

[4]The basic idea of the security proof in option 2 is that a collusion between a dishonest prover and a third party able to be accepted by an honest verifier can be used as an extractor of $e^{th}$ roots modulo $n$.

## 5 Acknowledgements

Guillaume Poupard and Jacques Stern are greatly acknowledged for the observations they made on this scheme (and reproduced in the paper here and there). We also thank Sébastien Canard for helping editing the LaTeX version.

## References

[1] F. Boudot and J. Traoré. Efficient Publicly Verifiable Secret Sharing Schemes with Fast or Delayed Recovery. In ICICS'99, LNCS 1726. Springer-Verlag, 1999.

[2] S. Even, O. Goldreich, and S. Micali. On-line/off-line Digital Signatures. In Crypto '89, LNCS 435, pages 263-277. Springer-Verlag, 1990.

[3] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In Crypto '86, LNCS 263, pages 186-194. Springer-Verlag, 1987.

[4] M. Girault. Self-Certified Public Keys. In Eurocrypt '91, LNCS 547, pages 490-497. Springer-Verlag, 1992.

[5] M. Girault. Low-Size Coupons for Low-Cost Smart Cards. In Cardis 2000, pages 39-49. Kluwer Academic, 2000.

[6] M. Girault, G. Poupard, and J. Stern. Global Payment System (GPS): un Protocole de Signature à la Volée. In Trusting Electronic Trade '99, 1999.

[7] M. Girault and J.-J. Quisquater. GQ+GPS, Eurocrypt 2002, rump session.

[8] L. C. Guillou and J.-J. Quisquater. A "Paradoxical" Identity-Based Signature Scheme Resulting from Zero-Knowledge. In Crypto '88, LNCS 403, pages 216-231. Springer-Verlag, 1989.

[9] M. Girault and J. Stern. On the length of cryptographic hash-values used in identification schemes. In Crypto'94, LNCS 839, pages 202-215. Springer-Verlag, 1994.

[10] G. Miller. Riemann's Hypothesis and Tests for Primality. Journal of Computer and System Sciences, 13:300-317, 1976.

[11] D. M'Raihi and D. Naccache. Couponing scheme reduces computational power requirements. In Proc. of Cardtech'94, pages 99-104, 1994.

[12] G. Poupard and J. Stern. Security Analysis of a Practical "on the fly" Authentication and Signature Generation. In Eurocrypt '98, LNCS 1403, pages 422-436. Springer-Verlag, 1998.

[13] G. Poupard and J. Stern. On The Fly Signatures based on Factoring. In Proceedings of 6th ACM-CCS, pages 37-45. ACM press, 1999.

[14] G. Poupard and J. Stern. Short Proofs of Knowledge for Factoring. In PKC 2000, LNCS 1751, pages 147-166. Springer-Verlag, 2000.

[15] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Communications of the ACM, 21(2):120-126, February 1978.

[16] A. Shamir and Y. Tauman. Improved On-line/Off-line Signature Schemes. In Crypto 2001, LNCS 2139, pages 355-367. Springer-Verlag, 2001.