On The Fly Signatures based on Factoring

Guillaume Poupard Jacques Stern

École Normale Supérieure, Département d'informatique
45 rue d'Ulm, F-75230 Paris Cedex 05, France

Guillaume.Poupard@ens.fr Jacques.Stern@ens.fr

ABSTRACT

In response to the current need for fast, secure and cheap public-key cryptography largely induced by the fast development of electronic commerce, we propose a new *on the fly* signature scheme, i.e. a scheme that requires very small on-line work for the signer. It combines provable security based on the factorization problem, short public and secret keys, short transmission and minimal on-line computation. It is the first RSA-like signature scheme that can be used for both efficient and secure applications based on low cost or contactless smart cards.

1 INTRODUCTION

The rapid world-wide development of electronic transactions, largely associated with the growth of the Internet, stimulates a strong demand for fast, secure and cheap public-key cryptography. Besides confidentiality, cryptographers also face two important problems: authentication and signature or, in plain words, how to prove one's identity and how to digitally sign a document such as an electronic message or a purchase order. Several proposals have already addressed the problem of designing efficient and secure signature scheme, putting forward elegant solutions [32, 30, 11, 33, 34, 20, 6, 16, 17, 10, 25, 13, 1].

In order to assess the performances of those schemes, several properties have to be considered. The most important concern is security. Basically, a system is supported by the claim that nobody has been able to jeopardize it so far. This is of course important but, in many applications, it is not a satisfactory guarantee. A much better paradigm tries to prove security in a mathematical sense, i.e. to establish theorems claiming that illegal actions such as impersonation are as difficult as solving a specific problem whose difficulty is well-established. Among these problems are integer factorization or the computation of discrete logarithms in a finite group. Half way between heuristic validation and formal proofs are proofs in models where concrete objects are replaced by ideal substitutes: applying this paradigm to hash functions yields the random oracle model described by Bellare and Rogaway in [2]. Although this approach may not be considered as offering absolute proofs of security for specific schemes, it provides a strong indication that their overall design is not flawed.

Next, the size of the data involved in the scheme is of crucial practical significance. We usually need short public and private keys, mainly when they have to be stored in portable devices like smart cards which have small storage capabilities. We also want to reduce the amount of transmissions and the length of the signatures. The latter is an important parameter in applications for which many signatures have to be stored (e.g. electronic commerce) or transmitted (e.g. pay TV).

Another key property is the time complexity since it directly controls the cost of the devices on which a scheme may be implemented. Here, we have to distinguish between precomputations that can be performed off-line and stored in memory and calculations that have to be done on-line during authentication or signature. The latter is often the bottleneck of many applications, especially when smart cards are used. Naccache et al. [24] proposed to precompute *use & throw coupons* in order to make the DSA [25] signature process much more efficient. This first attempt for designing *on the fly* signature schemes still requires a modular multiplication so that it does not allow very fast signature without the help of a crypto-processor.

The first on the fly signature scheme with minimal on-line computation, which we call GPS, was proposed by Girault [17] and proven secure by Poupard and Stern [29]. The security analysis shows that, if an attacker is able to forge valid signatures for a *non-negligible fraction* of the possible public keys, then he is able to compute discrete logs mod N and therefore to factor N. On the other hand if an attacker is only able to forge signatures for a fixed key then he must be able to compute the discrete log of this key or to solve the so-called strong RSA problem as it was noticed by Camenisch and Michels in [7]. Thus, the underlying problem depends on the model of attack that is considered.

The coupon-based signature algorithm GPS allows to implement public-key signature scheme on low cost smart cards without crypto-processor. Another promising application is the implementation of such schemes on contactless

Proceedings of the 6th ACM Conference on
Computer and Communications Security.
(November 1 –4, 1999, Kent Ridge Digital Labs, Singapore)

smart cards. Such cards just look like credit cards but they have an electronic microchip and an embedded antenna. These components allow the card to communicate with an antenna/coupler unit without any physical contact. Contactless cards are the ideal solution when transactions must be processed very quickly, as in mass-transit or toll collection but, since the power supply comes from electromagnetic induction, heavy-consumption crypto-processors cannot be used.

Our Results

In this paper, we propose the first signature scheme that combines provable security based on the intractability of factorization, short keys, short signature size and minimal on-line computation. This provides a solution for applications which require very efficient and secure signature generation while using only low cost individual devices. In comparison with GPS, our proposal appears more secure in the *one-key attack scenario*.

Intuitively, the signature scheme we propose is based on a non-interactive proof of knowledge of a *small* discrete 2

logarithm of $z^N \bmod N$, where N is the product of two safe primes. The proof is similar to Schnorr's proof of knowledge but it is carried out in $\mathbb{Z}_N$ and the answer to the challenge is computed in $\mathbb{Z}$, like in GPS.

We now briefly describe the organization of the paper. In section 2 we propose a new identification scheme and we prove that it is secure against active adversaries provided the factorization of the product of two large primes is hard. In section 3, we introduce a derived signature scheme and we show that, if an adversary is able to forge a signature under an adaptively chosen message attack, then he is able to factor large numbers.

The aim of this work is not just to propose one more signature scheme. Accordingly, in section 4, we compare the security and the performances of our scheme with its most common competitors in order to convince the reader that our protocol compares advantageously with others in many applications related to electronic commerce.

Section 5 is more practical in character: we discuss how to choose secure parameters in order to withstand the most efficient known attacks against factorization (5.1), we explain how use coupons (5.2), and, finally, we give the performances observed in experiments with smart cards.

2 THE IDENTIFICATION SCHEME

We first introduce some notation and definitions. For any integer x, $|x|$ is the number of bits $(\lfloor \log_2(x) \rfloor + 1)$ of x. For any integer N, we use $\varphi(N)$ to denote the Euler totient function, i.e. the cardinality of the set $\mathbb{Z}_N^*$ of invertible integers modulo N. A prime number p is said to be a *safe* prime if $(p-1)/2$ is also a prime number. Our computing model is the probabilistic polynomial time Turing machine ($\textsc{Pptm}(k)$), whose running time is bounded by a polynomial in the security parameter k.

Description

Let A, B, k and ℓ be four integers. We will latter explain how those parameters are related. Those values are public parameters; there are no system-wide keys.

Each user chooses two k-bit safe primes $P = 2p+1$ and $Q = 2q+1$. Then he computes his $2k$-bit public key $N = P \times Q$ and his $(k+1)$-bit secret key $S = N - \varphi(N) = P + Q - 1$. Finally, each user chooses a public element z of $\mathbb{Z}_N^*$ whose order is divisible by $p \times q$. In section 5.1, we will explain how such data can be efficiently generated.

A round of identification (see figure 1) consists of several steps. First, the prover randomly chooses an integer r in $[0, A[$ and computes the *commitment* $x = z^r \bmod N$. Then, he transmits x to the prover who answers a *challenge* e randomly chosen in $[0, B[$. The prover computes $y = r + S \times e$ (in $\mathbb{Z}$) and sends it to the verifier who checks $z^{y - N \times e} = x \bmod N$ and $y < A$. A complete identification is obtained by repeating ℓ times the elementary round.

Prover		Verifier
$r \in_R [0, A[$		
$x = z^r \bmod N$	$\xrightarrow{\quad x \quad}$	
	$\xleftarrow{\quad e \quad}$	$e \in_R [0, B[$
$y = r + Se$	$\xrightarrow{\quad y \quad}$	check $y < A$ and
		$z^{y-Ne} \stackrel{?}{=} x \bmod N$

Figure 1: Identification scheme

The performances of this scheme are given in section 5. Just notice that the on-line computation of the prover reduces to the non-modular multiplication $\alpha = S \times e$ followed by the non-modular addition $y = r + \alpha$. Those arithmetical operation are very efficient and can be implemented on very basic microprocessors.

Security Analysis

In order to prove the security of the protocol, complexity theory security proofs are given in appendix A.1. We essentially observe three properties:

- An honest user is accepted with overwhelming probability.
- Given a public key N, if an attacker is accepted with non-negligible probability, then he can be used to efficiently factor N. In other words, if we assume that factoring large integers is intractable, such attacks cannot exist.
- Even if a prover is identified many times, no information about his secret can be learned by eavesdroppers or verifiers.

3 THE SIGNATURE SCHEME

We now turn the identification scheme into a signature scheme using a technique originally proposed by Fiat and Shamir [11, 12] and used by Schnorr [33] and others. In order to go from identification to signature, the challenges e are no longer randomly chosen by a verifier but computed through a hash function H such as SHA-1 [26] or MD5 [31]. Such a function maps a binary strings of arbitrary length to binary string of some fixed length. For security reasons it must be computationally impossible to find two strings S_1 and S_2 with the same image $H(S_1) = H(S_2)$. If α and β are two integers, we note $H(\alpha, \beta)$ the image by H of the concatenation of their two bit string representations.

Input:
— the signer's public key (N, z)
— the signer's private key S
— the message representative, which is an integer $m \geq 0$

Assumptions: private key S and public key (N, z) are valid and associated with each other; $m \geq 0$

Output: the signature, which is a triplet of integers (x, e, y), where $1 \leq x < N$, $0 \leq e < B$ and $0 \leq y < A$

Operation. The signature (x, e, y) shall be computed by the following or an equivalent sequence of steps:

1. Randomly generate an integer r from the range $[0, A[$.
2. Compute $x = z^r \bmod N$.
3. Compute $e = H(m, x)$.
4. Compute $y = r + Se$.
5. If $y \geq A$ return to step 1.
6. Output (x, e, y).

Figure 2: Signature generation

Description

The signature of a message m is computed by taking a random r in $[0, A[$ and computing $x = z^r \bmod N$, $e = H(m, x)$ and $y = r + Se$. This produces the signature (x, e, y) that can be checked by anybody with the equations $e = H(m, x)$, $y < A$ and $z^{y-Ne} = x \bmod N$. An IEEE P1363 like description [1] appears in figures 2 and 3.

We now note $[0, B[$ the output range of H. In the identification scheme, B was a fixed constant but, in the signature setting, we need to let B depend on the security parameter k since there is only one round.

Input:
— the signer's public key (N, z)
— the message representative, which is an integer $m \geq 0$
— the signature to be verified, which is a triplet of integers (x, e, y)

Assumptions: public key (N, z) is valid; $m \geq 0$

Output: "valid" if m and (x, e, y) are consistent given the public key; "invalid" otherwise

Operation. The output shall be computed by the following or an equivalent sequence of steps:

1. If x is not in $[1, N-1]$ or e is not in $[0, B[$ or y is not in $[0, A[$, output "invalid" and stop.
2. Compute $e' = H(m, x)$.
3. Compute $x' = z^{y-Ne} \bmod N$.
4. If $e' = e$ and $x' = x$, then output "valid"; else, output "invalid".

Figure 3: Signature verification

Security Analysis

Complexity theory security proofs are given in appendix A.2. The basic idea is to prove that if the hash function H is replaced by a random function and if an attacker is able to forge a valid signature then he must know the factorization of the associated public key. Once again, if we assume that factoring large integers is intractable, forging valid signatures is impossible.

4 Comparison with other signature schemes

RSA and Rabin schemes

The famous RSA signature scheme [32] is based on the intractability of the RSA problem which consists in computing e^{th} roots modulo a product of two unknown large primes. An impressive literature has appeared on this scheme so we just recall a few basic facts.

Firstly, the RSA problem has never been proven to be equivalent to the factorization problem. Boneh and Venkatesan have even provided evidence that breaking low-exponent RSA might not be equivalent to factoring integers [5].

Secondly, when the RSA scheme is used with modulus N, each user has to choose a pair (d, e) such that $d \times e = 1 \bmod \varphi(N)$. The integer d is the secret key and should not be chosen much smaller than N (see the attacks against low exponent RSA of Wiener [36] and Boneh-Durfee [4]). On the contrary, the public exponent e can be chosen very small.

Finally, signature generation consists in raising some number to the power d modulo N. This computation is heavy. It can be improved using the Chinese reminder theorem (CRT) but this strategy implies the storage of the two factors of N.

Rabin [30] has proposed an interesting variant of RSA with public exponent $e = 2$. Its security relies on the difficulty of computing square-roots modulo N which is equivalent to factoring N. We refer to [3] for a precise analysis of the security of those two schemes in the random oracle model.

Feige-Fiat-Shamir and Guillou-Quisquater schemes

The Feige-Fiat-Shamir signature scheme [11] is derived from the Fiat-Shamir zero-knowledge authentication scheme [12]. Like Rabin's scheme, it is based on the difficulty of computing square-roots modulo N. The secret key consists of k elements of $\mathbb{Z}_N^*$ and the related public key is the list of their squares. As a consequence, for a reasonable value of k such that $k = 80$, both the public and the private keys are about 10 kilobytes long. In comparison with RSA, the advantage comes from the low computational complexity of the signature generation.

Guillou and Quisquater [20] have proposed a variant with smaller keys but whose security is only based on the RSA problem.

El Gamal, DSA and Schnorr schemes

Those three signature schemes are based on the intractability of computing discrete logarithms in various finite subgroups of $\mathbb{Z}_p^*$ where p is a prime number. El Gamal's protocol [10] directly uses discrete logs in $\mathbb{Z}_p^*$ but the authors of DSA [25] have noticed that this type of scheme is much more efficient using a generator of multiplicative order q where q is a 160-bit factor of $p - 1$.

Schnorr's signature scheme [33, 34] is very similar to DSA but produces short signatures (about 30 bytes with reasonable size of parameters).

Other schemes

Many other signature schemes have been proposed. Most of them are just variants of those already mentioned. Let us

	Underlying problem	Complexity of pre-comput.	Complexity of on-line comput.	Size (in bits) of Public Key	Size (in bits) of Secret Key	Size (in bits) of Signature	Verification complexity
Proposed scheme ($\lvert N\rvert = 1024$, $\lvert A\rvert = 672$)	**FACT**	$\frac{1008\times}{(\text{mod } 1024)}$	**80 bits × 512 bits** non modular	**1024**	513	752	$\frac{1656\times}{(\text{mod } 1024)}$
RSA ($\lvert N\rvert = 1024$, $e = 3$)	RSA	0	$\frac{1536\times}{(\text{mod } 1024)}$	**1024**	1024	1024	$\boldsymbol{\frac{2\times}{(\text{mod } 1024)}}$
Rabin ($\lvert N\rvert = 1024$)	**FACT**	0	$\frac{1536\times}{(\text{mod } 1024)}$	**1024**	1024	1024	$\boldsymbol{\frac{1\times}{(\text{mod } 1024)}}$
ESIGN ($\lvert N\rvert = \lvert P^2Q\rvert = 1024$)	Approx. RSA	$\frac{3\times}{(\text{mod } 1024)}$	$\frac{1\times}{(\text{mod } 342)}$	**1024**	1024	1024	$\boldsymbol{\frac{3\times}{(\text{mod } 1024)}}$
Feige-Fiat-Shamir ($\lvert N\rvert = 1024$, $k = 80$)	**FACT**	$\frac{1\times}{(\text{mod } 1024)}$	$\frac{41\times}{(\text{mod } 1024)}$	82944	81920	1104	$\frac{42\times}{(\text{mod } 1024)}$
Guillou-Quisquater ($\lvert N\rvert = 1024$, $k = 80$)	RSA	$\frac{192\times}{(\text{mod } 1024)}$	$\frac{121\times}{(\text{mod } 1024)}$	2176	1024	1104	$\frac{313\times}{(\text{mod } 1024)}$
El Gamal ($\lvert p\rvert = 768$)	DLOG mod p	$\frac{1152\times}{(\text{mod } 768)}$	$\frac{2\times}{(\text{mod } 768)}$	2304	**160**	1536	$\frac{3457\times}{(\text{mod } 768)}$
DSA ($\lvert p\rvert = 768$, $\lvert q\rvert = 160$)	DLOG mod p & mod q	$\frac{240\times}{(\text{mod } 768)}$	$\frac{2\times}{(\text{mod } 160)}$	2464	**160**	**320**	$\frac{480\times}{(\text{mod } 768)}$
Schnorr ($\lvert p\rvert = 768$, $\lvert q\rvert = 160$)	DLOG mod q	$\frac{240\times}{(\text{mod } 768)}$	$\frac{1\times}{(\text{mod } 160)}$	2464	**160**	**240**	$\frac{361\times}{(\text{mod } 768)}$
GPS ($\lvert N\rvert = 1024$)	DLOG mod N & Strong RSA	$\frac{504\times}{(\text{mod } 1024)}$	**80 bits × 160 bits** non modular	3072	**160**	**419**	$\frac{625\times}{(\text{mod } 1024)}$

Figure 4: Performances of signature schemes

close this review by mentioning the efficient scheme ESIGN [13] based on an original number theoretical problem which can be described as "approximately" solving RSA.

Another important family of schemes are variants of the previous ones, with computations performed over different finite groups, mainly over elliptic curves, like EC-DSA [1]. For comparisons between classical schemes and their elliptic curves counterparts see for example [37].

The Girault-Poupard-Stern scheme

The Girault-Poupard-Stern (GPS) scheme [17, 29] can be considered as a variant of Schnorr's protocol but it has been designed in order to reduce on-line computation to an absolute minimum and therefore allow on the fly signature using coupons. The basic idea is to design a proof of knowledge of a discrete log modulo a composite integer in such a way that the order of the used generator does not have to be known. A similar idea has also been used for secret sharing [14, 15] and group signature [7].

The security analysis of GPS shows that, if an attacker is able to forge valid signatures for a *non-negligible fraction* of the possible public keys, then he is able to compute discrete logs mod N and consequently to factor N. On the other hand, if an attacker is only able to forge signatures for a fixed key then he must be able to compute the discrete log of this key or to solve the so-called strong RSA problem (as noticed by Camenisch and Michels in [7]). As a consequence, the underlying problem depends on the model of attack that is considered.

In GPS, the modulus can be either a part of the user's public key or else system-wide parameters. In the first case, the public key is three times larger than it is in the present scheme. In the second case, the factorization of N is an ideal target for attacks.

Comparison of the schemes

Figure 4 provides a comparison of the signature schemes that were mentioned. Our estimates for the computational complexity of precomputations, on-line computations and signature verifications, only count modular multiplications. If an operation requires α multiplications modulo a β-bit integer, we write $\frac{\alpha\times}{(\text{mod } \beta)}$. Only for our new scheme and for GPS, we include the non-modular multiplication that has to be performed. The cost of such an operation is negligible in comparison with the modular computations required by the other protocols.

It should be clear that our figures have been obtained from naive algorithms and that they can be optimized. Anyway, we think that they represent an accurate estimate of what is efficient or not in all those schemes.

It appears that our proposal is based on the factorization, probably the most studied number-theoretical problem. The public keys are very short and the secret keys are only three times larger than what they are in discrete log based schemes. The main advantage is that even if the computational complexity of signature generation is similar to RSA or Rabin, almost all the computational work can be done off-line. As a Consequence, our scheme can be seen as the first coupon based RSA signature.

5 Applications

5.1 Choice of the Parameters

Size of N. The public key $N = P \times Q$ must be large enough to make factorization beyond computational reach. The number field sieve algorithm [21] allowed to factor an RSA modulus with $k = 232$ in February 1999. Consequently, the choice $k = 384$ is a lower bound and $k = 512$ would probably be a much reliable value for long-life signature applications.

Choice of P and Q. Integers P and Q must be safe primes. A simple way to find such integers consists in testing randomly chosen k-bit integers using a randomized algorithm such that the Miller-Rabin's one. Such an integer is a safe prime with probability $2/\ln(2^{k-1}) \times 1/\ln(2^k) = 4.163/(k \times (k-1))$. For more efficient algorithms, we refer to [22].

Public key certification. The validity of the proofs of security we propose is based on the hypothesis that N is the product of two safe primes. Consequently, the public key has to be certified by an authority. In order to convince it that N is of correct form, the user can of course reveal the factorization. A much more secure solution is to use the algorithm of Camenisch and Michels [8]. But this proof, even if it is efficient in the sense of the complexity theory, is too large and complex to be generated by a smart card without crypto-processor. Consequently, in practical applications, the public key would be generated by a computer, certified by an authority and then stored in a smart card.

Choice of z. During key setup, each user also has to select an element $z \in \mathbb{Z}_N^*$ of multiplicative order divisible by pq. It can be seen that this is true if and only if $\gcd(z-1, N) = \gcd(z+1, N) = 1$. Consequently, the probability for a randomly chosen element order to be large is $4(p-1)(q-1)/(4pq) \approx 1 - 4/\sqrt{N}$. This overwhelming probability can be used to reduce that public key to N only since z can be generated from N using a publicly known pseudo-random generator. Furthermore, anyone can verify that the order of z is large, without knowing the factorization of N.

Choice of A and B. From the security analysis it appears that the parameters must be such that $A < N$ and $2^k B/A$ is negligible. We advise to choose $|B| = 80$ as a minimal value and $|B| = 128$ for more secure applications. A good choice for the size of A is $|A| = 80 + k + |B|$.

5.2 Use & Throw Coupons

In order to decrease the number of communication bits, Fiat and Shamir [12] have suggested not to send the entire commitment in the first step of the identification but only a hash value. This trick can of course be used with our scheme. Let H' be a hash function and $|H'|$ be the size of its output. The modifications are very simple: the commitment x is replaced by $x' = H'(x)$ and the verifying equation becomes $x' = H'(z^{y-Ne} \bmod N)$.

Using the notion of r-collision-freeness, which applies to functions for which it cannot be possible to find r pairwise distinct values with the same image, Girault and Stern [18] have analyzed precisely the consequences of such a modification on the security of identification schemes.

As was already observed, all commitments can be computed off-line, by the individual device or by an authority. In fact, we just have to compute and keep in memory coupons of the form $(r, H'(z^r \bmod N))$. This can further be improved if the random values r are generated by a pseudo-random generator. This leads just to memorize the seed of the generator and the commitments, i.e. about only 10 bytes per signature! Furthermore, in some applications where public key directories are available to verifiers, N and z no longer need to be stored in memory by the signer's device.

Finally, the signature (x', e, y) of a message m can be reduced to (e, y) by using the verifying equations $y < A$ and $e = H(m, H'(z^{y-Ne} \bmod N))$. An IEEE P1363 like description [1] appears in figures 5 and 6.

Input:
— the signer's private key S
— a signer's use & throw coupon (r, x')
— the message representative, which is an integer $m \geq 0$

Assumptions: private key S is valid; the coupon (r, x') is valid; $m \geq 0$

Output: the signature, which is a pair of integers (e, y), where $0 \leq e < B$ and $0 \leq y < A$

Operation. The signature (e, y) shall be computed by the following or an equivalent sequence of steps:

1. Compute $e = H(m, x')$.
2. Compute $y = r + Se$.
3. If $y \geq A$ return to 1.
4. Output (e, y).

Figure 5: Optimized signature generation with Coupons

Input:
— the signer's public key (N, z)
— the message representative, which is an integer $m \geq 0$
— the signature to be verified, which is a pair of integers (e, y)

Assumptions: public key (N, z) is valid; $m \geq 0$

Output: "valid" if m and (e, y) are consistent given the public key; "invalid" otherwise

Operation. The output shall be computed by the following or an equivalent sequence of steps:

1. If e is not in $[0, B[$ or y is not in $[0, A[$, output "invalid" and stop.
2. Compute $e' = H(m, H'(z^{y-Ne} \bmod N))$.
3. If $e' = e$, then output "valid"; else, output "invalid".

Figure 6: Optimized signature verification

5.3 Smart Card Application

In order to convince the reader that the present scheme requires very low computation and very limited communication we have implemented it on low cost smart cards based on a Motorola 6805 chip. Figure 7 shows that the running time is very short. Note that we have not included the computation time for the hash function; this would probably be the bottleneck of many extremely fast applications.

Parameters	$k = 384$ $\|B\|$=80 $\|A\|$=544	$k = 512$ $\|B\|$=128 $\|A\|$=720
Size of the public key	768 bits	1024 bits
Size of the secret key	385 bits	513 bits
Size of a coupon Number of coupons stored in 4 KB	72 bits **455**	88 bits **372**
Number of CPU cycles Running time (3.57 MHz)	32276 **9 ms**	65726 **18 ms**
Size of a signature Transmission (106 kbaud)	627 bits 6 ms	851 bits 8 ms
Total running time	**15 ms**	**26 ms**

Figure 7: Performances on a 8-bit microprocessor based smart card without crypto-processor

CONCLUSION

We have proposes a new signature scheme that combines provable security based on the factorization problem, short public and secret keys, short transmission and minimal on-line computation. It is the first RSA-like signature scheme that can be used for both efficient and secure applications based on low cost or contactless smart cards.

ACKNOWLEDGMENTS

We would like to thank the anonymous referees for their helpful comments and suggestions.

REFERENCES

[1] IEEE P1363 Draft (Standard Specifications For Public Key Cryptography), August 1998. Available from `http://grouper.ieee.org/groups/1363/index.html`.

[2] Bellare, M., and Rogaway, P. Random Oracles are Practical: a paradigm for designing efficient protocols. In *Proc. of the 1st CCCS* (1993), ACM press, pp. 62–73.

[3] Bellare, M., and Rogaway, P. The Exact Security of Digital Signatures – How to Sign with RSA and Rabin. In *Eurocrypt '96* (1996), LNCS 1070, Springer-Verlag, pp. 399–416.

[4] Boneh, D., and Durfee, G. Cryptanalysis of RSA with Private Key d Less than $n^{0.292}$. In *Eurocrypt '99* (1999), LNCS 1592, Springer-Verlag, pp. 1–11.

[5] Boneh, D., and Venkatesan, R. Breaking RSA May Not Be Equivalent to Factoring. In *Eurocrypt '98* (1998), LNCS 1403, Springer-Verlag, pp. 59–71.

[6] Brickell, E. F., and McCurley, K. S. An Interactive Identification Scheme Based on Discrete Logarithms and Factoring. *Journal of Cryptology 5* (1992), 29–39.

[7] Camenisch, J., and Michels, M. A Group Signature Scheme with Improved Efficiency. In *Asiacrypt '98* (1998), LNCS 1514, Springer-Verlag.

[8] Camenisch, J., and Michels, M. Proving in Zero-Knowledge That a Number Is the Product of Two Safe Primes. In *Eurocrypt '99* (1999), LNCS 1592, Springer-Verlag, pp. 107–122.

[9] Canetti, R., Goldreich, O., and Halevi, S. The Random Oracle Methodology Revisited. In *Proc. of the 30th STOC* (1998), ACM Press, pp. 209–218.

[10] El Gamal, T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *IEEE Transactions on Information Theory* (july 1985), vol. IT–31, no. 4, pp. 469–472.

[11] Feige, U., Fiat, A., and Shamir, A. Zero-Knowledge Proofs of Identity. *Journal of Cryptology 1* (1988), 77–95.

[12] Fiat, A., and Shamir, A. How to Prove Yourself: practical solutions of identification and signature problems. In *Crypto '86* (1987), LNCS 263, Springer-Verlag, pp. 186–194.

[13] Fujioka, A., Miyaguchi, S., and Okamoto, T. ESIGN: An Efficient Digital Signature Implementation for Smart Cards. In *Eurocrypt '91* (1992), LNCS 547, Springer-Verlag, pp. 446–457.

[14] Fujisaki, E., and Okamoto, T. Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. In *Crypto '97* (1997), LNCS 1294, Springer-Verlag, pp. 16–30.

[15] Fujisaki, E., and Okamoto, T. A Practical and Provably Secure Scheme for Publicly Verifiable Secret Sharing and Its Applications. In *Eurocrypt '98* (1998), LNCS 1403, Springer-Verlag, pp. 32–46.

[16] Girault, M. An Identity-Based Identification Scheme Based on Discrete Logarithms Modulo a Composite Number. In *Eurocrypt '90* (1991), LNCS 473, Springer-Verlag, pp. 481–486.

[17] Girault, M. Self-certified public keys. In *Eurocrypt '91* (1992), LNCS 547, Springer-Verlag, pp. 490–497.

[18] Girault, M., and Stern, J. On the Length of Cryptographic Hash-Values used in Identification Schemes. In *Crypto '94* (1994), LNCS 839, Springer-Verlag, pp. 202–215.

[19] Goldwasser, S., Micali, S., and Rackoff, C. The Knowledge Complexity of Interactive Proof Systems. In *Proc. of the 17th STOC* (1985), ACM Press, pp. 291–304.

[20] Guillou, L. C., and Quisquater, J.-J. A "Paradoxal" Identity-Based Signature Scheme Resulting from Zero-Knowledge. In *Crypto '88* (1989), LNCS 403, Springer-Verlag, pp. 216–231.

[21] Lenstra, A., and Lenstra, H. *The Development of the Number Field Sieve*, vol. 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, 1993.

[22] Menezes, A., van Oorschot, P., and Vanstone, S. *Handbook of Applied Cryptography.* CRC Press, 1997.

[23] Miller, G. Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences 13* (1976), 300–317.

[24] Naccache, D., M'Raïhi, D., Vaudenay, S., and Raphaeli, D. Can DSA be improved ? In *Eurocrypt '94* (1995), LNCS 950, Springer-Verlag, pp. 77–85.

[25] NIST. Digital Signature Standard (DSS). Federal Information Processing Standards PUBlication XX, Draft, august 1991.

[26] NIST. Secure Hash Standard (SHS). Federal Information Processing Standards PUBlication 180–1, april 1995.

[27] Pointcheval, D., and Stern, J. Security Proofs for Signature Schemes. In *Eurocrypt '96* (1996), LNCS 1070, Springer-Verlag, pp. 387–398.

[28] Pointcheval, D., and Stern, J. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology* (1999). to appear, available at `http://www.dmi.ens.fr/~pointche`.

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

[30] Rabin, M. O. Digitalized Signatures and Public Key Functions as Intractible as Factorization. Tech. rep., Massachusetts Institute of Technology – Laboratory for Computer Science, january 1979. MIT/LCS/TR-212.

[31] Rivest, R. The MD5 Message-Digest Algorithm. RFC 1321, april 1992.

[32] Rivest, R., Shamir, A., and Adleman, L. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM 21*, 2 (february 1978), 120–126.

[33] Schnorr, C. P. Efficient Identification and Signatures for Smart Cards. In *Crypto '89* (1990), LNCS 435, Springer-Verlag, pp. 235–251.

[34] Schnorr, C. P. Efficient Signature Generation by Smart Cards. *Journal of Cryptology 4*, 3 (1991), 161–174.

[35] Stinson, D. R. *Cryptography, Theory and Practice.* CRC Press, 1995.

[36] Wiener, M. Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information Theory 36*, 3 (1990), 553–558.

[37] Wiener, M. Performance Comparison of Public-Key Cryptosystems. *CryptoBytes 4*, 1 (summer 1998), 1–5.

A Security proofs

A.1 Security Analysis of the Identification scheme

In order to prove the security of the identification protocol of section 2 against active adversaries, we follow the approach of Feige, Fiat and Shamir [11], successively proving completeness, soundness and the zero-knowledge property. We consider that the security parameter is k and that A and ℓ are functions of k. For technical reasons related the zero-knowledge property, B is considered a constant, as in the analysis of the Schnorr's scheme. Our assumptions are classical but may look surprising at first glance as in actual applications ℓ is usually set to one. Note that in the analysis of the signature scheme, we no longer need such an assumption.

In order to simplify notations, we do not write the dependences on k; in statements such as "f is negligible" we implicitly mean that f depends on k and that, for any constant c and for large enough k, $f(k) < 1/k^c$.

Theorem 1 (Completeness) *Assume $2^k\ell B/A$ is negligible. The execution of the protocol between a prover who knows the factorization of the public key N and a verifier is successful with overwhelming probability.*

Proof. At the end of each round, the verifier obtains $x = z^r \bmod N$ and $y = r + S \times e$. From Euler's theorem, we know that $z^{\varphi(N)} = 1 \bmod N$. It follows that $z^y = z^{r+Se} = z^r \times z^{e(N-\varphi(N))} = x \times z^{Ne} \bmod N$. Therefore $z^{y-N\times e} = x \bmod N$.

If the prover knows the secret $S < 2^{k+1}$ and follows the protocol, he can only fail if $y \geq A$ at some round of the proof. For any value of the secret $S < 2^{k+1}$, this probability of failure taken over all possible choices of r is smaller than $2^{k+1}B/A$. Consequently the execution of the protocol is successful with probability $\geq \left(1 - \frac{2^{k+1}B}{A}\right)^\ell \geq 1 - \frac{2^{k+1}\ell B}{A}$. Finally, if $2^k\ell B/A$ is negligible, this probability of success is overwhelming. □

The proof of soundness consists in proving that, if someone is correctly identified then, with overwhelming probability, he must know the secret key associated with the public key, i.e. the factorization of N, as stated in the following lemma:

Lemma 1 *Let N be an RSA modulus and L be any multiple of $\varphi(N)$. Then there exists a Turing machine which, on input (N, L), outputs the factorization of N in time $O(|L|)$.*

Proof. This lemma is due to Miller [23] and is also proved in [35]. □

Lemma 2 *Assume that some* Pptm(k) *adversary $\widetilde{P}$ is accepted with probability $\geq 1/B^\ell + \varepsilon$, $\varepsilon > 0$. Then there exists a* Pptm(k) *which factors the public key in time $O(kB\ell\tau/\varepsilon + k + |B|)$, where τ is the average running time of a round of identification.*

Proof. Assume that some Pptm(k) adversary $\widetilde{P}(\omega)$, running on random tape ω, is accepted with probability $\geq 1/B^\ell + \varepsilon$. We explain how to use $\widetilde{P}$ in order to obtain a machine which on input (N, z) answers (Y, E) such that $z^{Y-NE} = 1 \bmod N$, with $-A < Y < A$ and $-B < E < B$. This computation takes time $O(kB\ell\tau/\varepsilon)$ where τ is the average running time of a round of identification.

We first choose a random tape ω. We then let i vary from 1 to ℓ. For each i, we let $\widetilde{P}(\omega)$ produce the i^{th} commitment x_i and note S_i the state reached by $\widetilde{P}(\omega)$. We ask $\widetilde{P}(\omega)$ the B possible challenges and, each time, we check the answer and we reset $\widetilde{P}(\omega)$ at state S_i. After those B steps, three cases may appear:

- if $\widetilde{P}(\omega)$ has correctly answered two challenges e and e', with y and y', return $(Y, E) = (y - y', e - e')$.

- if $\widetilde{P}(\omega)$ cannot answer any challenge, return `Fail`.

- if $\widetilde{P}(\omega)$ answers exactly one challenge, keep on with the loop. If the end of the loop is reached, return `Fail`.

It can be formally proved that with probability $\geq \varepsilon/2$ this machine returns (Y, E) such that $z^{Y-NE} = 1 \bmod N$ after at most $O(B\ell\tau)$ time units. If we repeat k/ε times with other random tapes, (Y, E) is obtained with overwhelming probability in time $O(kB\ell\tau/\varepsilon)$. Furthermore, Y is the difference of two correct answers y and y' smaller than A so $-A < Y < A$. Finally, $-B < E < B$ and $E \neq 0$.

Let L be $NE - Y$. This integer is such that $z^L = 1 \bmod N$. If N is greater than A, L is a non-zero multiple of the multiplicative order of z in $\mathbb{Z}_N^*$. Since this order is divisible by pq and $\varphi(N) = 4pq$, $4 \times L$ is a multiple of $\varphi(N)$. Miller's algorithm of lemma 1 allows to factor N in time $O(|L|) = O(k + |B|)$. □

Note. This lemma is used to prove the soundness of the identification protocol when B is considered as a constant and ℓ is a function of the security parameter k. A more complex proof proposed by Schnorr in [34] states that if $\ell = 1$ and B is a function of k, if an adversary is accepted with probability $\varepsilon > 2/B$, the public key N can be factored in time $O(1/\varepsilon + k + |B|) \sim O(1/\varepsilon)$. In actual applications ℓ is usually set to one so this last result is much more convincing. For example, if we use a 1000 bits moduli, an attacker accepted with probability as small as $1/2^{40}$ would be able to factor a thousand bits RSA moduli in reasonable time $O(2^{40})$.

Theorem 2 (Soundness) *Assume that some* $\text{PPTM}(k)$ *adversary* $\widetilde{P}$ *is accepted with non-negligible probability, that* $\log(k) = o(\ell)$ *and that* ℓ *is smaller than a polynomial in* k. *Then there exists a* $\text{PPTM}(k)$ *which factors the public key with overwhelming probability.*

Proof. If $\pi(k)$ is the non-negligible probability of success of $\widetilde{P}$, there exists an integer d such that $\pi(k) \geq 1/k^d$ for infinitely many values k. Furthermore, for k large enough, $1/B^\ell < 1/2k^d$ because $\log(k) = o(\ell)$. So, taking $\varepsilon = \pi(k)/2$ in lemma 2, we conclude that one can factor N in time $O(kB\ell\tau/\varepsilon + k + |B|)$. If we assume that ℓ is polynomial in k, then we have found a $\text{PPTM}(k)$ which factors the public key with overwhelming probability. □

Theorem 3 (Zero-knowledge) *The protocol is statistically zero-knowledge if* $2^k\ell BT/A$ *is negligible, where* $T(k)$ *is the maximal number of repetitions of the protocol with the same keys.*

Proof. We first recall that the zero-knowledge property states that the *view* of any verifier considered as a random variable is perfectly similar to the output of a $\text{PPTM}(k)$ which does not know the secret key. A protocol is only *statistically zero-knowledge* if the view and the output of the $\text{PPTM}(k)$ are only statistically indistinguishable. We refer the reader to [19] for more details.

We describe the polynomial time simulation of the communication between a prover P and a dishonest verifier $\widetilde{V}$. We assume that, in order to try to obtain information about S, $\widetilde{V}$ does not randomly choose the challenges. If we focus on the i^{th} round of identification, $\widetilde{V}$ has already obtained data, noted $Data_i$, from previous interactions with P. Then the prover sends the commitment x_i and $\widetilde{V}$ chooses, possibly using $Data_i$ and t_i, the challenge $e_i(Data_i, t_i)$.

Here is a simulation of the i^{th} round of identification: choose random values $e_i' \in [0, B[$ and $y_i' \in [0, A[$, compute $x_i' = z^{y_i' - Ne_i'} \bmod N$. If $e_i(D_i, x_i') \neq e_i'$ then try again with another pair (e_i', y_i'), else return (x_i', e_i', y_i'). It can be formally proved that such a polynomial time simulation is statistically indistinguishable from the transcript of a real proof.

As a consequence, a verifier with infinite computation power cannot learn significant information after a polynomial number of authentications. □

Note. In the previous proof we observe that a good triplet (x_i', e_i', y_i') is obtained with probability $1/B$. Consequently, the expected time complexity of the all simulation is $O(\ell B)$. This explains why we need to repeat ℓ times the elementary round of identification, in order to obtain a negligible probability of cheating $1/B^\ell$ and a polynomial time simulation. Note that we have chosen to make B constant but it can also be considered a polynomial in k.

A.2 Security Analysis of the Signature scheme

It is widely believed that the heuristic transformation we used to design our signature scheme from the interactive identification scheme guarantees an accurate level of security as soon as H is random enough. Furthermore, the security of this approach can be formalized using the random oracle model [2, 27] even if such analysis cannot be considered as an actual proof of security as it is pointed out in [9].

In the identification scheme, B was a fixed constant but, in the signature setting, we need to let B depend on the security parameter k, as A, and to set $\ell = 1$. The following lemma proves that our scheme satisfies all the properties required to apply the the technique developed by Pointcheval and Stern [27] and known as the *forking lemma*:

Lemma 3 *Assume* $\ell = 1$. *The protocol is a 3-pass honest-verifier statistically zero-knowledge identification scheme.*

Proof. Let us consider an honest verifier, i.e. a verifier who ask randomly chosen challenges e. Using a proof similar to Schnorr's in [34] we can show that if an adversary is accepted with probability $\varepsilon > 2/B$, the public key N can be factored in time $O(1/\varepsilon + k + |B|)$.

Furthermore, the proof of theorem 3 can be simplified in the setting of honest verifier in order to provide a constant time simulation. □

In order to prove the security of the scheme in section 3, we show that, if someone is able to forge valid signatures after having obtained signatures of messages of his choice, then we can use the attacker to efficiently factor the public key. The random oracle model [2] is used to model the behavior of the hash function H so that the proof validates the overall design.

An attacker who existentially forges the signature scheme can be modeled as a $\text{PPTM}(k)$ $\mathcal{A}(\omega)$, running on random tape ω. For infinitly many values of the security parameter

k the machine is able to find with probability $\varepsilon(k)$ a message m and a valid signature (x, e, y).

Two distinct scenario of attacks are considered, the *no-message attack* during which $\mathcal{A}(\omega)$ can ask Q queries to a random oracle and the *adaptively chosen message attack* where $\mathcal{A}(\omega)$ can also ask the signatures of R messages he chooses to a signature oracle. We note T the maximal number of messages signed with a fixed key and $\tau_{\mathcal{A}}$ the average running time of an attacker $\mathcal{A}$.

Theorem 4 *If an existential forgery of the signature scheme under a* no-message attack *has a probability of success $\varepsilon \geq 7Q/B$ then the public key N can be factored within expected time $O(Q\tau_{\mathcal{A}}/\varepsilon + k + |B|)$.*

Proof. (sketch) The proof consists in making the attacker run with different random oracles in order to obtain valid signatures (x, e_i, y_i) of a message m with the same commitment x, which in term produces equations of the form $z^{y_i - Ne_i} = z^{y_j - Ne_j} \bmod N$. Then, as in the proof of soundness of the identification scheme, this leads to the factorization of N. We refer to [28] for details. □

Theorem 5 *Assume that $2^k BT/A$ is negligible. If an existential forgery of the signature scheme under an* adaptatively chosen message attack *has a probability of success $\varepsilon \geq 10(R+1)(R+Q)/B$ then the public key N can be factored within expected time $O(Q\tau_{\mathcal{A}}/\varepsilon + k + |B|)$.*

Proof. (sketch) The signature oracle can be replaced by a PPTM(k) which simulates valid signatures if $2^k BT/A$ is negligible. This modifies only negligibly the probability of succes of the attacker: overwise he would be able to distinguish between actual and simulated signatures. As for the previous proof, we refer to [28] for details. □

Note. Because of the factor Q in the reduction, the expected time of the resulting factorization algorithm is a bit disappointing. Consequently, this result must be viewed as a complexity theory security proof and not as an *exact* security analysis.