This paper appears in the proceedings of *PKC2000, LNCS , Springer Verlag, 2000*

Short Proofs of Knowledge for Factoring

Guillaume Poupard and Jacques Stern

École Normale Supérieure, Département d'informatique
45 rue d'Ulm, F-75230 Paris Cedex 05, France
email: {Guillaume.Poupard,Jacques.Stern}@ens.fr

Abstract. The aim of this paper is to design a proof of knowledge for the factorization of an integer n. We propose a statistical zero-knowledge protocol similar to proofs of knowledge of discrete logarithm *a la* Schnorr. The efficiency improvement in comparison with the previously known schemes can be compared with the difference between the Fiat-Shamir scheme and the Schnorr one. Furthermore, the proof can be made non-interactive. From a practical point of view, the improvement is dramatic: the size of such a non-interactive proof is comparable to the size of the integer n and the computational resources needed can be kept low; three modular exponentiations both for the prover and the verifier are enough to reach a high level of security.

1 Introduction

Zero-knowledge (ZK) proofs have first been proposed in 1985 by Goldwasser, Micali and Rackoff [14]. Those proofs are interactive protocols between a prover who wants to convince a verifier that an object belongs to a language (proof of membership) or that he knows a secret information (proof of knowledge), without revealing anything about his secret knowledge. Such proofs have practical applications since they allow to solve many cryptographic problems such as ZK identification [9], digital signature [25] or robust distributed cryptography [26]. Many ZK proof systems have been published so far that are related to the presumably intractable problems on which public key cryptography is based, such as the computation of discrete logarithms [25], of square roots [9] and of e^{th} roots [15] modulo a composite integer.

In this paper, we consider the most popular such problem: the factorization of integers, i.e. how to prove to a verifier that one's knows some prime numbers whose product is a public number without giving any information about this decomposition. Proofs of knowledge for the factorization of an integer have been known for a long time. But, even if they are claimed efficient according to complexity theoretical arguments, none of them can be considered practical for many applications because of their significant communication complexity: the proof is much longer than the object it deals with.

Previously known proofs of knowledge for the factorization of an integer n are closely related to the property that n has less than a prescribed number of prime factors. Van de Graaf and Peralta [29] and Galil et al. [11] first provided efficient proofs that a given integer is of the form $p^r q^s$. Then Boyar et al. [4] proposed a protocol to prove that an integer is square-free. The combination of those two protocols typically allows to prove that an integer is an RSA modulus. Those results have recently been enhanced with protocols which prove that the factors are *quasi-safe* primes [12], have about the same size [16] or are exactly safe primes [5].

All the above protocols are based on the basic observation that, modulo a given integer n, a random number has a square root with probability $2^{-\eta}$, where η is the number of different prime factors of n, and that such a square root can be efficiently computed only when the factorization of n is known. A verifier can ask for square roots of randomly chosen numbers and count the proportion of correct answers of the prover. This allows to prove that a number has less than η different prime factors but the basic round has to be repeated many times in order to reach a sufficient level of security and confidence.

A derived protocol to prove the knowledge of the factorization of n has been proposed by Tompa and Woll [28]. It is based on ZK proofs for quadratic residuosity that come from [14] and which can be interpreted as proofs of membership of an integer x to the set of the quadratic residues modulo an integer n or as proofs of knowledge of a square root of x modulo n. According to Tompa and Woll, the verifier first randomly chooses an element r in $\mathbb{Z}_n$ and sends $x = r^2 \bmod n$ to the prover. Then he proves that x is a quadratic residue modulo n. Once the prover is convinced that the verifier knows a square root of x, himself computes a square root s of x and proves that he also knows a square root of x.

The completeness of such a protocol is based on the existence of a polynomial time algorithm for computing square roots modulo n when the factorization of n is known. Furthermore, this scheme is zero-knowledge since the view of any prover can be easily simulated using the simulator of the proof for quadratic residuosity proposed in [14]. This explains why the verifier needs to prove that he knows a square root of x; otherwise, he would learn whether x is a quadratic residue or not and the protocol would not be zero-knowledge. Finally, the scheme has error probability smaller than $7/8$ but greater than $3/4$. Consequently, it has to be repeated many times in order to reach a high level of security. For realistic parameters, this implies Giga-bytes of communication. From a theoretical point of view, let k be a security parameter such that the cheating probability is smaller than $1/2^k$. We can prove that the complexity for both computation and communication is $\theta(k \times |n|^2)$ where $|n|$ denotes the number of digits in the binary expansion of n. Even if some optimizations can be added, it does not seem possible to go under $\theta(k \times |n|)$.

Notice that Boudot and Traoré [3] have recently proposed a scheme that does not need to prove that n has less than a prescribed number of prime factors. The idea, somehow comparable with what we do in this paper, is to prove the knowledge of a common discrete logarithm d of g_1 and g_2, two randomly chosen integers, in basis g_1^e and g_2^e, where d is such that $e \times d = 1 \bmod \lambda(n)$.

1.1 Our results

We propose an interactive proof of knowledge for the factorization of any public integer n whose prime factors cannot be found by simple trial division. It is statistical zero-knowledge. The protocol is a proof of knowledge of a small discrete logarithm of $z^n \bmod n$ for a few randomly chosen elements z modulo n. Consequently, it is related to variants of the Schnorr proof of knowledge for discrete logarithms.

Our scheme is very efficient. When suitably optimized, its communication complexity is only $O(k+|n|)$ bits. In this setting, the size of our proof is similar to the size of the integer n. The improvement in comparison with the previously known schemes can therefore be compared with the difference of efficiency between the Fiat-Shamir scheme and the Schnorr one. Furthermore, the computational complexity is proved to be $O((|n|+k) \times k)$ multiplications modulo n both for the prover and the verifier but we provide strong heuristic evidence to show that $O((|n|+k) \times k/\log k)$ is enough. This might appear a small improvement but it has drastic consequences in practical terms: only three modular exponentiations both for the prover and the verifier are needed to obtain a very high level of security.

In section 2 we state results of independent interest on the probability to generate the multiplicative group $\mathbb{Z}_{p^e}{}^*$ with few elements (complete proofs appear in the appendix A). Those results are used in the security analysis of the interactive proof of knowledge we describe in section 3. We also propose a communication efficient variant (section 3.3) and a non-interactive version (section 3.4). In section 4 we give heuristic arguments to improve the computational efficiency of the scheme and, in section 5, we show the practical efficiency of those proofs in actual applications. Finally, in section 6, we propose a variant suggested by Adi Shamir in order to make the proof work even when n has a small prime factors.

1.2 Notations and Definitions

For any integer n,
- we use $\mathbb{Z}_n$ to denote the set of the integers modulo n,
- we use $\mathbb{Z}_n{}^*$ to denote the multiplicative group of invertible elements of $\mathbb{Z}_n$,
- we use $\varphi(n)$ to denote the Euler totient function, i.e. the cardinality of $\mathbb{Z}_n{}^*$,
- we use $\lambda(n)$ to denote Carmichael's lambda function defined as the largest order of the elements of $\mathbb{Z}_n{}^*$.

It is well known that if the prime factorization of an odd integer n is $\prod_{i=1}^{\eta} {q_i}^{f_i}$ then $\varphi(n) = \prod_{i=1}^{\eta} {q_i}^{f_i-1}(q_i-1)$ and $\lambda(n) = \mathrm{lcm}_{i=1..\eta}\left({q_i}^{f_i-1}(q_i-1)\right)$.

If $(g_i)_{i\in[1,K]}$ is a K-tuple of $\left(\mathbb{Z}_n{}^*\right)^K$, we use $\left\langle (g_i)_{i\in[1,K]} \right\rangle$ to denote the subgroup of $\mathbb{Z}_n{}^*$ that is generated by the g_is, i.e.

$$\left\langle (g_i)_{i\in[1,K]} \right\rangle = \left\{ x \in \mathbb{Z}_n{}^* \middle| \exists(\lambda_1, ...\lambda_K) \quad x = \prod_{i=1}^{K} g_i^{\lambda_i} \bmod n \right\}$$

In the following, p_i is the i^{th} prime number ($p_1 = 2, p_2 = 3,\ldots$). For any finite set S, $\mathcal{C}\text{ard}(S)$ denotes the number of elements of S. We finally denote by $\zeta(K)$ the Riemann Zeta function defined by $\zeta(K) = \sum_{d=1}^{+\infty} \frac{1}{d^K}$ for any integer $K \geq 2$.

2 On the generation of $\mathbb{Z}_{p^e}{}^*$

We state known facts about the generation of the cyclic multiplicative group $\mathbb{Z}_{p^e}{}^*$ where p is an odd prime number and $e \geq 1$. Using generators it is possible to precisely estimate the probability to generate the full group $\mathbb{Z}_{p^e}{}^*$ by a single randomly chosen element.

Theorem 1 *For any prime number $p \geq 7$, for any $e \geq 1$,*

$$\Pr_{g \in \mathbb{Z}_{p^e}{}^*} \{\langle g \rangle = \mathbb{Z}_{p^e}{}^*\} = \frac{\varphi(\varphi(p^e))}{\varphi(p^e)} > \frac{1}{7 \ln \ln p}$$

Next, we generalize this result when K elements are randomly chosen instead of one. We obtain the following lower bound, independent of p and e:

Theorem 2 *For any odd prime number p, for any $e \geq 1$, for any $K \geq 2$,*

$$\Pr_{\{g_i\}_{i\in[1,K]} \in \left(\mathbb{Z}_{p^e}{}^*\right)^K} \left\{ \left\langle (g_i)_{i\in[1,K]} \right\rangle = \mathbb{Z}_{p^e}{}^* \right\} > \frac{1}{\zeta(K)} > 1 - \frac{K+1}{K-1} \times \frac{1}{2^K}$$

Finally, we obtain a lower bound for the probability that K elements generate a *large* subgroup of $\mathbb{Z}_{p^e}{}^*$, i.e. a subgroup of size greater than $\mathcal{C}\text{ard}\left(\mathbb{Z}_{p^e}{}^*\right)/C$ for a fixed parameter C:

Theorem 3 *For any odd prime number p, for any $e \geq 1$, $C \geq 1$ and $K \geq 2$,*

$$\textit{with } \mathcal{P} = \Pr_{\{g_i\}_{i\in[1,K]} \in \left(\mathbb{Z}_{p^e}{}^*\right)^K} \left\{ \mathcal{C}ard \left\langle (g_i)_{i\in[1,K]} \right\rangle \geq \frac{\mathcal{C}ard(\mathbb{Z}_{p^e}{}^*)}{C} \right\}$$

$$\mathcal{P} > \frac{1}{\zeta(K)} \times \sum_{d=1}^{C} \frac{1}{d^K} > 1 - \frac{1}{(K-1)C^{K-1}\zeta(K)}$$

All the proofs appear in appendix A.

3 Proofs of knowledge for factoring

3.1 Description

Let k be a security parameter. Let n be an integer whose number of digits in its binary expansion is denoted $|n|$. Let A, B, ℓ and K be integers which depend *a priori* on k and $|n|$. Let $z_1, \ldots z_K$ be K elements randomly chosen in ${\mathbb{Z}_n}^*$. We describe an interactive proof of knowledge for the factorization of n.

A round of proof (see figure 1) consists for the prover in randomly choosing an integer r in $[0, A[$ and computing, for $i = 1..K$, the *commitments* $x_i = z_i^r \bmod n$. Then he sends the x_is to the verifier who answers a *challenge* e randomly chosen in $[0, B[$. The prover computes $y = r + (n - \varphi(n)) \times e$ (in $\mathbb{Z}$) and sends it to the verifier who checks $0 \leq y < A$ and, for $i = 1..K$, $z_i^{y - n \times e} = x_i \bmod n$. A complete proof consists in repeating ℓ times the elementary round.

Prover		Verifier
$r \in_R [0, A[$		
$x_i = z_i^r \bmod n$, for $i = 1..K$	$\xrightarrow{(x_i)_{i \in [1,K]}}$	
	$\xleftarrow{e}$	$e \in_R [0, B[$
$y = r + (n - \varphi(n)) \times e$	$\xrightarrow{y}$	check $0 \leq y < A$ and
		$z_i^{y-ne} \overset{?}{=} x_i \bmod n$, for $i = 1..K$

Fig. 1. Interactive proof of knowledge for factoring (elementary round)

This scheme is a variant of the Schnorr proof of knowledge of discrete logarithms. It consists in proving the knowledge of small discrete log, namely $n - \varphi(n)$, of $z_i^n \bmod n$, for K randomly chosen integers z_i in ${\mathbb{Z}_n}^*$. In the following section, we prove that it is a statistical ZK proof of knowledge of the factorization of n.

3.2 Security proofs

In order to prove the security of the protocol, we follow the approach of Feige, Fiat and Shamir [8], first proving completeness, then soundness and, finally, the zero-knowledge property. In order to simplify the notations, we do not write the dependencies on k but when we say that an expression f is negligible, this means that f depends on k and that, for any constant c and for large enough k, $f(k) < 1/k^c$. Our computing model is the probabilistic polynomial time Turing machine (PPTM), whose running time is a polynomial in k and $|n|$.

Theorem 4 (Completeness) *The execution of the protocol between a prover who knows the factorization of n and a verifier is successful with overwhelming probability if $(n - \varphi(n))\ell B/A$ is negligible.*

Proof: At the end of each round, the verifier obtains $x_i = z_i^r \bmod n$ and $y = r + (n - \varphi(n)) \times e$ which can be easily computed by the prover if he knows the factorization of n. From Euler's theorem, we know that $z_i^{\varphi(n)} = 1 \bmod n$ so $z_i^y = z_i^{r+(n-\varphi(n))e} = x_i \times z_i^{ne} \bmod n$. Consequently, $z_i^{y-n\times e} = x_i \bmod n$.

If the prover follows the protocol, the proof fails only if $y \geq A$ at some round of the proof. The probability of failure of such an event taken over all possible choices of r is smaller than $(n - \varphi(n))B/A$. Consequently the execution of the protocol is successful with probability $\geq \left(1 - \frac{(n-\varphi(n))B}{A}\right)^\ell \geq 1 - \frac{(n-\varphi(n))\ell B}{A}$. Thus, if $(n-\varphi(n))\ell B/A$ is negligible, the probability of success is overwhelming. $\square$

The proof of soundness consists in proving that, if the verifier accepts the proof, then, with overwhelming probability, the prover must know the factorization of n. Intuitively, after the commitment of the x_is, if the prover is accepted with probability $> 1/B$, he must be able to answer two different questions e and e' with y and y' smaller than A such that $z_i^{y-ne} = x_i = z_i^{y'-ne'} \bmod n$ for $i = 1..K$. Let $\lambda_0 = |(y - y') - n(e - e')|$; this integer is such that, for $i = 1..K$, $z_i^{\lambda_0} = 1 \bmod n$. The following lemma formally states those ideas, where ε is implicitly assumed to depend on k and $|n|$:

Lemma 1 *Assume that some* PPTM *adversary* $\widetilde{P}$ *is accepted with probability* $\varepsilon' = 1/B^\ell + \varepsilon$, $\varepsilon > 0$ *and that* $A < n$. *Then there exists an algorithm which, with probability* $> \varepsilon^2/(6\varepsilon'^2)$, *outputs* $\lambda_0 \in]0, A + nB]$ *such that, for* $i = 1..K$, $z_i^{\lambda_0} = 1 \bmod n$. *The expected running time is* $< 2/\varepsilon \times \tau$, *where* τ *is the average running time of an execution of the proof.*

Proof: Assume that some PPTM adversary $\widetilde{P}(\omega)$, running on random tape ω, is accepted with probability $\varepsilon' = 1/B^\ell + \varepsilon$. We write $Succ(\omega, (e_1, ...e_\ell)) \in \{true, false\}$ the result (successful of not) of the identification of $\widetilde{P}(\omega)$ when the challenges $e_1, ...e_\ell$ are used.

We consider the following algorithm (largely inspired from [25]):
Step 1. Pick a random tape ω and a tuple e of ℓ integers $e_1, ...e_\ell$ in $\{0, ..B - 1\}$ until $Succ(\omega, e)$. Let u be the number of probes.
Step 2. Probe up to u random ℓ-tuples e' different from e until $Succ(\omega, e')$. If after the u probes a successful e' is not found, the algorithm fails.
Step 3. Let j be one of the indices such that $e_j \neq e_j{}'$; we note y_j and $y_j{}'$ the related correct answers of $\widetilde{P}$. The algorithm outputs $\lambda_0 = |(y_i - y_i{}') - n(e_i - e_i{}')|$.

If this algorithm does not fails, the prover is able to correctly answer two challenges e_j and $e_j{}'$ for the same commitment x_j with the answers y_j and $y_j{}'$. This means that $z_i^{y_j - n\times e_j} = x_j = z_i^{y_j{}' - n\times e_j{}'} \bmod n$ for all $i = 1..K$ so the integer λ_0 is such that $z_i^{\lambda_0} = 1 \bmod n$. Furthermore, λ_0 is smaller than $A + nB$ because $\lambda_0 = |(y_i - y_i{}') - n(e_i - e_i{}')|$ for integers y_i and $y_i{}'$ smaller than A and integers e_i and $e_i{}'$ smaller than B. Finally, since $A < n$, $\lambda_0 = 0$ would imply $e_i = e_i{}'$ so $\lambda_0 > 0$.

We now analyze the complexity of the algorithm. By assumption, the probability of success of $\widetilde{P}$ is ε' so the first step finds ω and e with the same probability.

The expected number E of repetitions is $1/\varepsilon'$ and the number u of probes is equal to N with probability $\varepsilon' \times (1-\varepsilon')^{N-1}$.

Let Ω be the set of random tapes ω such that $\Pr_e \{Succ(\omega, e)\} \geq \varepsilon' - \varepsilon/2 = 1/B^\ell + \varepsilon/2$. The probability for the random tape ω found in step 1 to be in Ω conditioned by the knowledge that $Succ(\omega, e) = true$ can be lower bounded:

$$\begin{aligned}\Pr_{\omega,e} \{\omega \in \Omega | Succ(\omega, e)\} &= 1 - \Pr_{\omega,e} \{\omega \notin \Omega | Succ(\omega, e)\} \\ &= 1 - \Pr_{\omega,e} \{Succ(\omega, e) | \omega \notin \Omega\} \times \frac{\Pr_{\omega,e} \{\omega \notin \Omega\}}{\Pr_{\omega,e} \{Succ(\omega, e)\}} \\ &\geq 1 - \left(\frac{1}{B^\ell} + \frac{\varepsilon}{2}\right) \times 1/\varepsilon' = \frac{\varepsilon}{2 \times \varepsilon'}\end{aligned}$$

With probability $> \varepsilon/(2\varepsilon')$, the random tape ω is in Ω and in this case, by definition of the set Ω, the probability for a tuple of challenges $e' \neq e$ to lead to success is $\geq \varepsilon/2$. The probability to obtain such a tuple e' after less than N probes is $\geq 1 - (1 - \varepsilon/2)^N$.

Consequently, the probability to obtain a random tape ω in Ω and to find e' is greater than

$$\frac{\varepsilon}{2\varepsilon'} \times \sum_{N=1}^{+\infty} (1-\varepsilon')^{N-1} \times \varepsilon' \times \left[1 - (1 - \frac{\varepsilon}{2})^N\right] = \frac{\varepsilon^2}{4\varepsilon'(\varepsilon' + \varepsilon/2 - \varepsilon \times \varepsilon'/2)} > \frac{\varepsilon^2}{6\varepsilon'^2}$$

In conclusion, the algorithm finds λ_0 with probability $> \varepsilon^2/(6\varepsilon'^2)$ and the total expected number of executions of the proof between $\widetilde{P}$ and a verifier is smaller than $2/\varepsilon'$. $\square$

Theorem 5 (Soundness) *Assume that some* PPTM *adversary* $\widetilde{P}$ *is accepted with non-negligible probability. If* $\ell \times \log B = \theta(k)$, $K = \theta(k + \log(|n|))$, $\log(A)$ *is a polynomial in* k *and* $|n|$ *and* $A < n$, *there exists a* PPTM *which factors* n *with overwhelming probability.*

Proof: Let $\pi(k)$ is the probability of success of $\widetilde{P}$. If $\pi(k)$ is non-negligible, there exists an integer d such that $\pi(k) \geq 1/k^d$ for infinitely many values k.

Let $n = \prod_{j=1}^{\eta} q_j^{e_j}$ be the prime factorization of n. Notice that η is the number of different prime factors of n. Let us consider the K randomly chosen elements z_i; from theorem 2, we know that, modulo $q_j^{e_j}$, they generate $\mathbb{Z}_{q_j^{e_j}}{}^*$ with probability greater than $1 - (K+1)/(K-1) \times 1/2^K$. Consequently, the z_is generate multiplicative groups modulo $q_j^{e_j}$ for $j = 1..\eta$ with probability greater than $1 - \eta \times (K+1)/(K-1) \times 1/2^K$.

The probability for $\widetilde{P}$ to be accepted while the z_is generate all groups $\mathbb{Z}_{q_j^{e_j}}{}^*$ is larger than $\pi(k) - \eta \times (K+1)/(K-1) \times 1/2^K$. The number η of different prime factors of n is less than $\log_2(n) = |n|$ so, if $K = \theta(k + \log(|n|))$, for infinitely many values k, $\eta \times (K+1)/(K-1) \times 1/2^K \leq 1/3k^d$.

Furthermore, for k large enough, $1/B^\ell < 1/3k^d$ if $\ell \times \log B = \theta(k)$. So, taking $\varepsilon = \pi(k)/3$ in lemma 1 we conclude that it is possible to obtain $\lambda_0 \in]0, A+nB]$ in polynomial time $O(1/\varepsilon) = O(k^d)$.

Then, for $i = 1..K$, $z_i^{\lambda_0} = 1 \bmod n$ so, for any $j = 1..\eta$, $z_i^{\lambda_0} = 1 \bmod q_j^{e_j}$. Let z be any element of $\mathbb{Z}_n{}^*$; since the z_is generate $\mathbb{Z}_{q_j^{e_j}}{}^*$ for a fixed j, z can be written as a product $\prod_{i=1}^{K} z_i^{\alpha_{i,j}}$ modulo $q_j^{e_j}$ and consequently $z^{\lambda_0} = 1 \bmod q_j^{e_j}$. Using the Chinese remainder theorem, we obtain that $z^{\lambda_0} = 1 \bmod n$ for any z in $\mathbb{Z}_n{}^*$. This means that λ_0 is a non-zero multiple of the Carmichael lambda function of n. It is well known that knowledge of a multiple of $\lambda(n)$ allows to factor n in time $O(\eta \times \log \lambda_0)$ modular multiplications using the Miller's factoring algorithm [18] which we recall in appendix B.

Finally, we obtain the factorization of n in time $O(|n| \times \log(A+nB))$ modular multiplications modulo n. □

Theorem 6 (Zero-Knowledge) *The protocol is statistically zero-knowledge if* $(n - \varphi(n))\ell B/A$ *is negligible and* $\ell \times B$ *is a polynomial in* k.

Proof: We first remind that the zero-knowledge property is verified if the *view* of any verifier considered as a random variable is perfectly approximable by the output of a PPTM which does not know the factorization of n. A protocol is only *statistically zero-knowledge* if the view and the output of the PPTM are only statistically indistinguishable. We refer the reader to [14] 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 the factorization of n, $\widetilde{V}$ does not randomly choose the challenges. If we focus on the j^{th} round of identification, $\widetilde{V}$ has already obtained data, noted $Data_j$, from previous interactions with P. Then the prover sends the commitments $X_j = (x_1, ..x_K)$ and $\widetilde{V}$ chooses, possibly using $Data_j$ and X_j, the challenge $e_j(Data_j, X_j)$.

Here is a simulation of the j^{th} round of identification: choose random values $e_j{}' \in [0, B[$ and $y_j{}' \in [0, A[$, compute, for $i = 1..K$, $x_i{}' = z_i^{y_j{}' - n e_j{}'} \bmod n$. If $e_j(Data_j, (x_1{}', ..x_K{}')) \neq e_j{}'$ then try again with another pair $(e_j{}', y_j{}')$, else return $((x_1{}', ..x_K{}'), e_j{}', y_j{}')$.

We observe that a good triplet $((x_1{}', ..x_K{}'), e_i{}', y_i{}')$ is obtained with probability $1/B$. Consequently, the expected time complexity of the all simulation is $O(\ell B)$.

Furthermore, it can be formally proved that such a simulation is statistically indistinguishable from the transcript of a real proof if $(n - \varphi(n))\ell B/A$ is negligible. Therefore, a verifier with infinite computation power cannot learn significant information after a polynomial number of authentications. □

Note. This theorem shows that if we choose $\ell = 1$ and B exponential in the security parameter k, we cannot prove the zero-knowledge property. Notice that there is exactly the same problem with the Schnorr scheme.

Choice of the parameters and Complexity of the scheme. The choice of the parameters ℓ and B must be such that $\ell \times \log B = \theta(k)$ in order to make the protocol sound. The choice of A is a bit more difficult; A must be much larger than $(n-\varphi(n))\ell B$ to guarantee the completeness and the zero-knowledge property but A must also be smaller than n to guarantee the soundness. Consequently, n must verify $(n-\varphi(n)) \times 2^k \ll n$. The proof we propose cannot be used with any integer n but we can notice that the previous equation is verified by all the integers which does not have small prime factors. More precisely, if $n = \prod_{i=1}^{\eta} q_i^{e_i}$, we can prove that $\frac{1}{q_1} < \frac{n-\varphi(n)}{n} = 1 - \prod_{i=1}^{\eta}\left(1 - \frac{1}{q_i}\right) < \sum_{i=1}^{\eta} \frac{1}{q_i}$. Consequently, if $(n-\varphi(n)) \times 2^k \ll n$, all the prime factors of n must be $\gg 2^k$.

If all the prime factors of n are greater than a bound $F(k)$, we know that $(n-\varphi(n))/n < \eta/F(k)$ so we require $F(k) \gg \eta \times 2^k$. Anyway, in practical applications, such a proof is used to prove the knowledge of integers like RSA modulus with large prime factors; if n has small prime factors, the proof is not zero-knowledge but n can not be considered as a good modulus! Informally, this means that the proof is correct and zero-knowledge if the factorization of n is intractable. Notice that a prover cannot try to cheat choosing an integer n with small factors since the soundness is guaranteed by $A < n$.

The execution of the protocol requires the transmission of exactly $\ell \times (K \times |n| + |B| + |A|)$ bits. If we assume that $\ell \times \log B = \theta(k)$, $K = \theta(k + \log |n|)$ and $|A| = \theta(|n|)$ (with $A < n$), we obtain a communication complexity equal to $\theta(\ell \times (k + \log |n|) \times |n|)$. From the computational point of view, both the prover and the verifier need to compute K exponentiations modulo n with $|A|$-bits exponents.

3.3 Optimized version

In the interactive proof of knowledge we have described is section 3.1, we can observe that the largest part of the communication concerns the commitments x_i. Using an idea of Fiat and Shamir whose security has been formalized by Girault and Stern in [13], we can replace those commitments by the hash value $H(x_1, .., x_K)$ where H is an appropriate collision-free hash function. We obtain a new scheme (see figure 2), much more efficient in term of communication than the initial one. An important consequence is that the communication complexity of the modified protocol is independent of the parameter K and is the same than for the Schnorr scheme, i.e. $O(k + |n|)$.

We can finally observe that the commitments can be precomputed in order to reduce the on-line computation to a very simple non-modular arithmetic operation (like in [23]).

In practical applications, an important point is the choice of the z_is. They need to be randomly chosen in order to make the proof sound. A first solution consists in using a mutually trusted source of random bits by the prover and the verifier. Such a strategy is used in non-interactive zero-knowledge proofs [2]. In practice, z_i can be pseudo-randomly generated from a seed of the form $h(n, i)$ where h is a hash function such as SHA-1 [19].

Prover		Verifier
$r \in_R [0, A[$		
$X = H\left((z_i^r \bmod n)_{i=1..K}\right)$	$\xrightarrow{X}$	
	$\xleftarrow{e}$	$e \in_R [0, B[$
$y = r + (n - \varphi(n)) \times e$	$\xrightarrow{y}$	check $0 \leq y < A$ and
		$H\left((z_i^{y-ne} \bmod n)_{i=1..K}\right) \stackrel{?}{=} X$

Fig. 2. Optimized interactive proof of knowledge for factoring

3.4 Non-interactive proof of knowledge for factoring

The interactive proof can be made non-interactive using the Fiat-Shamir heuristic [8, 9]. The verifier's challenge e is replaced with the hash value of the commitment $X = H(x_1, .., x_k)$ and of the public data using a collision-resistant hash function H'. The size of such a proof, $< k + |n|$ bits, is very small:

> A non-interactive proof of knowledge of the factorization of n is
> a pair (e, y) with $0 \leq e < B$, $0 \leq y < A$ and
> $e = H'(n, z_1, .., z_K, H(z_1^{y-ne} \bmod n, .., z_K^{y-ne} \bmod n)$

It is widely believed that such a transformation 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 [1, 20, 21] even if such analysis cannot be considered as an absolute proof of security [6].

The important difference between the interactive and the non-interactive setting is that in the second case we do not have to care about dishonest verifiers since the hash function H' is assumed to produce random challenges. It is easy to modify the proof of theorem 6 in order to demonstrate that if $\ell = 1$ and $B = 2^k$, the protocol is *honest-verifier statistically zero-knowledge* :

Theorem 7 *The protocol is honest-verifier statistically zero-knowledge if* $(n - \varphi(n))\ell B/A$ *is negligible.*

Then the so-called forking lemma technique described by Pointcheval and Stern [20, 21] can be applied in order to prove the security of the non-interactive version of the scheme, in the random oracle model, even when $\ell = 1$.

4 Heuristic efficiency improvements

We have seen in the previous section that the proof of knowledge we propose has a communication complexity independent of the number K of integers z_i. Furthermore, for security reasons (theorem 5), K must be approximately equal to the security parameter k. In order to reduce the computational complexity of the protocol, we now show how to reduce K to a very small value such as 3 in practice. The underlying idea is to increase the work load of the extractor of theorem 5 by a factor $\sqrt{C}$ in order to reduce K by a factor $\log C$. Unfortunately, the algorithm we propose is based on well known techniques such as the Pollard's rho method whose complexity can only be analyzed using heuristic arguments.

Theorem 8 *Assume that some adversary is accepted with non-negligible probability ε. If $\ell \times \log B = \theta(k)$ and $K = \theta\left(\frac{k+\log(|n|)}{\log C}\right)$, there exists an algorithm which, heuristically, factors n in time $O(1/\varepsilon + \sqrt{C} \times |n| \times \log(A+nB))$ with non-negligible probability.*

If C is a polynomial in the security parameter k, this proves that we can choose $K = \theta((k + \log|n|)/\log k)$.

Let us first remind two algorithms.

Floyd's cycle-finding algorithm

Let f be a random function from a set S to S. Let x_0 be a random element of S and consider the elements x_i recursively defined by $x_{i+1} = f(x_i)$. Such a sequence consists of a *tail* of expected length $\sqrt{\pi \mathcal{C}\mathrm{ard}(S)/8}$ followed by a *cycle* of the same expected length (see for example [10]). This immediately leads to an algorithm able to find a collision $x_i = x_j$ in time $O(\sqrt{\mathcal{C}\mathrm{ard}(S)})$ and with memory $O(\sqrt{\mathcal{C}\mathrm{ard}(S)})$ (see [17] for more details).

The previous algorithm can be improved in order to use just a constant amount of memory. Floyd's cycle-finding algorithm consists in starting with the pair (x_1, x_2) and iteratively computing (x_i, x_{2i}) from (x_{i-1}, x_{2i-2}) until $x_m = x_{2m}$. It can be proved that the expected running time of this algorithm is also $O(\sqrt{\mathcal{C}\mathrm{ard}(S)})$ while the memory needed is constant since no values have to be stored in memory.

This algorithm can still be improved in order to find indexes i and j such that $x_i = x_j$ but $x_{i-1} \neq x_{j-1}$. The idea consists in finding in a first step an index m such that $x_m = x_{2m}$ with Floyd's algorithm. Then, we iteratively test if $x_i = x_{i+m}$ for increasing values of i. The time complexity is always $O(\sqrt{\mathcal{C}\mathrm{ard}(S)})$ and the memory needed is still constant.

Pollard's rho factoring algorithm

Floyd's algorithm can be used to factor integers. The Pollard's rho algorithm [22] consists in choosing $S = \mathbb{Z}_n$ and $f(x) = x^2 + 1 \bmod n$. We do not search collisions $x_i = x_j \bmod n$ but only indexes i and j such that $\gcd(x_i - x_j, n) > 1$, i.e. a collision modulo a prime factor of n. Since this gcd is equal to n with negligible probability, we obtain a non trivial factor of n. A recursive use of the algorithm allows to completely factor n.

If we assume that $f(x) = x^2 + 1 \bmod p$ behaves like a random function, the computational complexity required to find a factor p of n is $O(\sqrt{p})$ modular multiplication. As a consequence, this algorithm allows to find small factors much more efficiently than with trial division.

Proof of theorem 8

We now describe the algorithm announced in theorem 8. The procedure we explain allows to break the integer n in two factors. A complete factorization of n is obtained by using it recursively.

We have proved in section 2 that the z_is generate large subgroups modulo each prime power factor of n with overwhelming probability, even for small values of K, but we do not have any similar result modulo n. We note G the multiplicative group of the integers $z^{\lambda_0} \bmod n$ for $z \in {\mathbb{Z}_n}^*$. We know that $\mathcal{C}\text{ard}\{z \text{ s.t. } z^{\lambda_0} = 1 \bmod n\} \times \mathcal{C}\text{ard}\, G = \varphi(n)$. Consequently, $\Pr_{z \in {\mathbb{Z}_n}^*} \{z^{\lambda_0} = 1 \bmod n\} = 1/\mathcal{C}\text{ard}\, G$.

Two cases may occur:

- if $\mathcal{C}\text{ard}\, G$ is a *small* set, a multiple of $\lambda(n)$ can be computed from λ_0. This can be done using a Floyd's algorithm to compute the order of $x^{\lambda_0} \bmod n$ for a randomly chosen elements x of ${\mathbb{Z}_n}^*$. Such an algorithm succeeds in expected time $O(\sqrt{\lambda(n)/\gcd(\lambda_0, \lambda(n))})$. Then we can use the Miller's factoring algorithm to factor n with a multiple of $\lambda(n)$.
- otherwise, λ_0 does not have enough common factors with $\lambda(n)$ to make the Carmichael's lambda function of n easy to compute so we use the following algorithm to overcome the problem.

First, it is easy to test if the modulus n has more than η prime factors using the elliptic curve factoring algorithm or just the Pollard's rho factoring algorithm as soon as the size of the factors are *small enough*. For example, for 1024 bits modulus, $\eta = 16$ is reasonable.

Let $n = \prod_{j=1}^{\eta} q_j^{e_j}$ be the prime factorization of the modulus n. From theorem 3, we know that, modulo $q_j^{e_j}$, the K elements $z_1, \ldots z_K$ generate a subgroup of ${\mathbb{Z}_{q_j^{e_j}}}^*$ which size is greater than $\varphi(q_j^{e_j})/C$ with probability greater than $1 - ((K-1)C^{K-1}\zeta(K))^{-1}$. Consequently, with probability greater than $1 - \eta \times ((K-1)C^{K-1}\zeta(K))^{-1}$, the z_is generate large subgroups modulo each $q_j^{e_j}$. For example, with $\eta = 16$, $K = 3$ and $C = 2^{42}$, this probability is larger than $1 - 1/2^{80}$.

We now use a variant of Pollard's rho algorithm to factor n. Let z be a randomly chosen element of ${\mathbb{Z}_n}^*$. We define a sequence of elements modulo n by $w_0 = z^{\lambda_0} \bmod n$ and $w_{i+1} = \left(w_i^2 + 1\right)^{\lambda_0} \bmod n$. Then we look for indexes i and j such that $\gcd(w_i \pm w_j, n) \neq 1$ and $\gcd(w_{i-1} \pm w_{j-1}, n) = 1$ using the Floyd's cycle-finding algorithm.

Heuristically, since the cycles modulo each factor $q_j^{e_j}$ are not too small, $w_i \neq w_j \bmod n$. In this case, $\gcd(w_i - w_j, n)$ is a non-trivial factor of n.

If we consider this algorithm modulo $q_j^{e_j}$, we see that it is a Floyd's cycle-finding algorithm in a space of size smaller than C. Consequently, cycles are searched in parallel for all the factors $q_j^{e_j}$ of n and the solution is found in average time $O(\sqrt{C})$ exponentiations to the power λ_0 modulo n using a fixed (small) amount of memory. It is important to notice that we just need that the z_i generate large subgroups modulo each factor of n and not necessarily a large subgroup of ${\mathbb{Z}_n}^*$.

Other method. A. Joux suggested a different analysis based on Pollard's $p-1$ factoring method and that leads to the same conclusions about the choice of K.

5 Performances

Let us consider the following typical application: a prover wishes to generate a non-interactive proof of knowledge of the factorization of a 1024-bit integer n ($|n| = 1024$). In order to reach a high level of security, we choose $k = 80$, $\ell = 1$ and $B = 2^k = 2^{80}$ in order to obtain a probability of success for a dishonest prover smaller than $1/2^{80}$ (lemma 1). The choice of A is directed by the results of theorems 4, 5 and 7 on the completeness, soundness and zero-knowledge property. We have to take A much larger than $(n - \varphi(n))\ell B$ and smaller than n, e.g. $A = 2^{1024}$. Finally, the choice of $C = 2^{42}$ allows to take $K = 3$ according to theorem 8.

We can only consider integers n with less than 16 prime factors. The protocol is secure for the prover if all the factors are much more than 84-bits long, e.g. 128-bits long. For numbers with smaller factors, a dishonest prover could not cheat but a (possibly dishonest) verifier could learn non negligible information about the factorization of n. Notice that for any cryptographic application which requires composite integers to perform secure computations, we cannot reasonably assume the intractability of the factorization of an integer with prime factors shorter than 2^{128}.

With this choice of parameters, a proof requires 3 exponentiations modulo n for the prover and for the verifier. The proof is very short ($80 + 1024 = 1104$ bits long) and of about the same size than the integer n.

6 A variant to prove the knowledge of the factorization of any integer

As we previously said, our protocol can only be used with integers n such that $(n - \varphi(n)) \times 2^k \ll n$, i.e. integers without small prime factors. Adi Shamir suggested an interesting variant to deal with such numbers:

Let a, b, ℓ and K be integers. Let z_1,...z_K be K elements randomly chosen in $\mathbb{Z}_n{}^*$. A round of proof consists for the prover in randomly choosing an integer r in $[0, 2^a[$ and computing, for $i = 1..K$, the *commitments* $x_i = z_i^r \bmod n$. Then he sends the x_is to the verifier who answers a *challenge* e randomly chosen in $[0, 2^b[$. The prover computes an answer $y \in [0, 2^a[$ such that $y = r + c \times \varphi(n) - e \times 2^a$ for a suitable value of c. He sends it to the verifier who checks $0 \leq y < 2^a$ and, for $i = 1..K$, $z_i^{y+e\times 2^a} = x_i \bmod n$. A complete proof consists in repeating ℓ times the elementary round.

This scheme is based on the ability to compute an integer y equal to r modulo $\varphi(n)$, with its b leading bits fixed to e, when $\varphi(n)$ is known. The correctness is satisfied as soon as $2^a \gg \varphi(n)$ so we impose $2^a \gg n$. The proof of soundness is similar to the proof of theorem 5. Finally, the protocol is also statistically zero-knowledge when $2^a \gg n$.

Acknowledgments

We would like to thank Adi Shamir and Antoine Joux for their helpful comments and suggestions.

References

1. M. Bellare and P. Rogaway. Random Oracles are Practical: a paradigm for designing efficient protocols. In *Proc. of the 1st CCCS*, pages 62–73. ACM press, 1993.
2. M. Blum, A. De Santis, S. Micali, and G. Persiano. Non-Interactive Zero-Knowledge. *SIAM journal of computing*, 20(4):1084–1118, 1991.
3. F. Boudot and J. Traoré. Efficient Publicly Verifiable Secret Sharing Schemes with Fast or Delayed Recovery. In *Proc of ICICS'99*. Springer-Verlag, 1999.
4. J. Boyar, K. Friedl, and C. Lund. Practical Zero-Knowledge Proofs: Giving Hints and Using Deficiencies. *Journal of Cryptology*, 4(3):185–206, 1991.
5. J. Camenisch and M. Michels. Proving in Zero-Knowledge That a Number Is the Product of Two Safe Primes. In *Eurocrypt '99*, LNCS 1592, pages 107–122. Springer-Verlag, 1999.
6. R. Canetti, O. Goldreich, and S. Halevi. The Random Oracle Methodology Revisited. In *Proc. of the 30th STOC*, pages 209–218. ACM Press, 1998.
7. H. Cohen. *A Course in Computational Algebraic Number Theory*. Graduate Texts in Mathematics 138. Springer-Verlag, 1993.
8. U. Feige, A. Fiat, and A. Shamir. Zero-Knowledge Proofs of Identity. *Journal of Cryptology*, 1:77–95, 1988.
9. A. Fiat and A. Shamir. How to Prove Yourself: practical solutions of identification and signature problems. In *Crypto '86*, LNCS 263, pages 186–194. Springer-Verlag, 1987.
10. P. Flajolet and A. Odlyzko. Random Mapping Statistics. In *Eurocrypt '89*, LNCS 434, pages 329–354. Springer-Verlag, 1990.
11. Z. Galil, S. Haber, and M. Yung. A Private Interactive Test of a Boolean Predicate and Minimum-Knowledge Public-Key Cryptosystems. In *Proc. of the 26th FOCS*, pages 360–371. IEEE, 1985.
12. R. Gennaro, D. Micciancio, and T. Rabin. An Efficient Non-Interactive Statistical Zero-Knowledge Proof System for Quasi-Safe Prime Products. In *Proc. of the 5th CCCS*, pages 67–72. ACM press, 1998.
13. 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.
14. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. In *Proc. of the 17th STOC*, pages 291–304. ACM Press, 1985.
15. L. C. Guillou and J.-J. Quisquater. A "Paradoxal" Identity-Based Signature Scheme Resulting from Zero-Knowledge. In *Crypto '88*, LNCS 403, pages 216–231. Springer-Verlag, 1989.
16. M. Liskov and D. Silverman. A Statistical Limited-Knowledge Proof for Secure RSA Keys. Technical report, RSA Laboratories, 1998.
17. A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

18. G. Miller. Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences*, 13:300–317, 1976.
19. NIST. Secure Hash Standard (SHS). Federal Information Processing Standards PUBlication 180–1, april 1995.
20. D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. In *Eurocrypt '96*, LNCS 1070, pages 387–398. Springer-Verlag, 1996.
21. D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 1999. to appear, available at `http://www.dmi.ens.fr/ pointche`.
22. J. M. Pollard. A Monte Carlo Methods for Factorization. *BIT*, 15:331–334, 1975.
23. 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.
24. J.N. Rosser and L. Schoenfeld. Approximate Formulas for some Functions of Prime Numbers. *Illinois Journal of Mathematics*, 6(1):64–94, march 1962.
25. C. P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.
26. M. Stadler. Publicly verifiable secret sharing. In *Eurocrypt '96*, LNCS 1070, pages 190–199. Springer-Verlag, 1996.
27. D. R. Stinson. *Cryptography, Theory and Practice*. CRC Press, 1995.
28. M. Tompa and H. Woll. Random Self-Reducibility and Zero Knowledge Interactive Proofs of Possession of Information. In *Proc. of the 28rd FOCS*, pages 472–483. IEEE, 1987.
29. J. van de Graaf and R. Peralta. A Simple and Secure Way to Show the Validity of Your Public Key. In *Crypto '87*, LNCS 293, pages 128–134. Springer-Verlag, 1988.

A On the generation of $\mathbb{Z}_{p^e}{}^*$ (proofs)

A.1 Generation of $\mathbb{Z}_{p^e}{}^*$ with one randomly chosen element

We first recall known properties of the Euler function:

Fact 1 – *If p is prime and $e \in \mathbb{N}^*$, then* $\varphi(p^e) = p^e - p^{e-1} = p^{e-1}(p-1)$
– *If* $\gcd(m,n) = 1$ *then* $\varphi(m \times n) = \varphi(m) \times \varphi(n)$ *(φ is said to be* multiplicative*)*
– *If $n = q_1^{e_1} \times q_2^{e_2} \times \ldots \times q_k^{e_k}$ is the prime factorization of n, then*

$$\varphi(n) = n \times \left(1 - \frac{1}{q_1}\right)\left(1 - \frac{1}{q_2}\right)\ldots\left(1 - \frac{1}{q_k}\right)$$

A first question about the generation of $\mathbb{Z}_{p^e}{}^*$ is the following: for a randomly chosen element g of $\mathbb{Z}_{p^e}{}^*$, what is the probability for g to be a generator of $\mathbb{Z}_{p^e}{}^*$, i.e. $< g >= \mathbb{Z}_{p^e}{}^*$? The classical answer below allows to count the number of generators of $\mathbb{Z}_{p^e}{}^*$.

Fact 2 *Suppose that g is a generator of $\mathbb{Z}_{p^e}{}^*$. Then $h = g^i \bmod p^e$ is also a generator of $\mathbb{Z}_{p^e}{}^*$ if and only if* $\gcd(i, \varphi(p^e)) = 1$. *It follows that the number of generators of $\mathbb{Z}_{p^e}{}^*$ is $Card\left(\mathbb{Z}_{\varphi(p^e)}{}^*\right) = \varphi(\varphi(p^e))$.*

$$\Pr_{g \in \mathbb{Z}_{p^e}{}^*}\{< g >= \mathbb{Z}_{p^e}{}^*\} = \frac{\varphi(p^{e-1}(p-1))}{p^{e-1}(p-1)} = \begin{cases} \varphi(p-1)/(p-1) & \textit{if } e = 1 \\ \varphi(p-1)/p & \textit{if } e > 1 \end{cases}$$

It can be noted that this probability can be small. For example, if p is such that $p-1$ is the product of the first α prime numbers ($p-1 = \prod_{i=1}^{\alpha} p_i$), the probability to generate $\mathbb{Z}_p{}^*$ with a single element is $\varphi(p-1)/(p-1) = \prod_{i=1}^{\alpha}(1-1/p_i) \cong \frac{e^{-\gamma}}{\ln(\alpha \ln \alpha)}$ where γ is Euler's constant (see for example [24]). For $\alpha = 55$, p is 340 bits long and this probability is about $1/10$. This means that we need to try more than 500 elements in order to find a generator with probability very close to 1 such as $1 - 1/2^{80}$.

More precisely, the following fact proved in [24] allows to lower bound the probability of fact 2:

Fact 3 *For all integers* $x \geq 5$, $\dfrac{\varphi(x)}{x} > \dfrac{1}{6 \ln \ln x}$ *but there does not exist any constant* C *such that* $\varphi(x) > C \times x$ *for any integer* x.

Theorem 1 *For any prime number* $p \geq 7$, *for any* $e \geq 1$,

$$\underset{g \in \mathbb{Z}_{p^e}{}^*}{Pr} \{\langle g \rangle = \mathbb{Z}_{p^e}{}^*\} = \frac{\varphi(\varphi(p^e))}{\varphi(p^e)} > \frac{1}{7 \ln \ln p}$$

Proof: The probability for a randomly chosen element g of $\mathbb{Z}_{p^e}{}^*$ to generate the all group is greater than $\varphi(p-1)/p$ so, using the previous fact, if $p \geq 7$ the probability is larger than $(1-1/p) \times 1/(6 \ln \ln(p-1)) > 1/(7 \ln \ln p)$. $\square$

A.2 Generation of $\mathbb{Z}_{p^e}{}^*$ with K randomly chosen elements

A natural question is how the probability of fact 2 is modified if we choose K elements $g_1, ... g_K$ of $\mathbb{Z}_{p^e}{}^*$ instead of one. To answer this problem, we first generalize the Euler totient function and define φ_K for all integers $K \geq 1$ by:

- If p is prime and $e \geq 1$, then $\varphi_K(p^e) = p^{Ke} - p^{K(e-1)}$
- If $\gcd(m, n) = 1$ then $\varphi_K(m \times n) = \varphi_K(m) \times \varphi_K(n)$ (implies $\varphi_K(1) = 1$)

We note that for $K = 1$, $\varphi_1 = \varphi$.

Lemma 2 *If* $n = q_1^{e_1} \times q_2^{e_2} \times ... \times q_k^{e_k}$ *is the prime factorization of* n, *then*

$$\varphi_K(n) = n^K \times \left(1 - \frac{1}{q_1^K}\right)\left(1 - \frac{1}{q_2^K}\right) ... \left(1 - \frac{1}{q_k^K}\right)$$

The functions φ_K allow to generalize fact 2 to the case of K generators:

Lemma 3 *The number of* K*-tuples of* $(\mathbb{Z}_{p^e}{}^*)^K$ *that generate* $\mathbb{Z}_{p^e}{}^*$ *is* $\varphi_K(\varphi(p^e))$.

Proof: Let $(g_1, ... g_K)$ be a K-tuple of $(\mathbb{Z}_{p^e}{}^*)^K$. Let g be a generator of $\mathbb{Z}_{p^e}{}^*$; for $i = 1, ... K$, we define $\alpha_i \in \mathbb{Z}_{\varphi(p^e)}$ by the relation $g^{\alpha_i} = g_i \bmod p^e$.

We first notice that $(g_1, ... g_K)$ generates $\mathbb{Z}_{p^e}{}^*$ if and only if the ideal generated by $\alpha_1, \alpha_2, ...\ \alpha_K$ in the ring $\mathbb{Z}_{\varphi(p^e)}$ is the entire ring. Bezout equality shows that this occurs iff $\gcd(\alpha_1, ... \alpha_K, \varphi(p^e)) = 1$.

Let us count the number of K-tuples $(\alpha_1, \ldots \alpha_K) \in \left(\mathbb{Z}_{\varphi(p^e)}\right)^K$ such that $\gcd(\alpha_1, \ldots \alpha_K,$
$\varphi(p^e)) = 1$. Let $\prod_{i=1}^{t} q_i^{f_i}$ be the prime factorization of $\varphi(p^e)$. We know that

$$\gcd(x, \prod_{i=1}^{t} q_i^{f_i}) = 1 \Leftrightarrow \forall i \leq t \ \gcd(x, q_i^{f_i}) = 1 \Leftrightarrow \forall i \leq t \ \gcd(x \bmod q_i^{f_i}, q_i^{f_i}) = 1$$

Using the Chinese remainder theorem, the problem reduces to counting the number of K-tuples $(\beta_1, \ldots \beta_K)$ of $\left(\mathbb{Z}_{q_i^{f_i}}\right)^K$ such that $\gcd(\beta_1 \bmod q_1^{f_1}, \ldots \beta_K \bmod q_K^{f_K}, q_i^{f_i}) = 1$ for $i = 1, \ldots t$. The K-tuples that do not verify this relation for a fixed index i are of the form $(q_i\gamma_1, \ldots q_i\gamma_K)$ where $(\gamma_1, \ldots \gamma_K) \in \left(\mathbb{Z}_{q_i^{f_i - 1}}\right)^K$ and there are exactly $q_i^{K(f_i-1)}$ such K-tuples.

Finally there are $\prod_{i=1}^{t} q_i^{Kf_i} - q_i^{K(f_i-1)}$ K-tuples of $\left(\mathbb{Z}_{\varphi(p^e)}\right)^K$ such that $\gcd(\alpha_1, \ldots \alpha_K,$
$\prod_{i=1}^{t} q_i^{f_i}) = 1$ and this is equal to $\prod_{i=1}^{t} \varphi_K(q_i^{f_i}) = \varphi_K(\varphi(p^e))$ since φ_K is multiplicative. □

Theorem 2 *For any odd prime number p, for any $e \geq 1$, for any $K \geq 2$,*

$$\textit{with } \mathcal{P} = \underset{\{g_i\}_{i \in [1,K]} \in \left(\mathbb{Z}_{p^e}{}^*\right)^K}{Pr} \left\{ \left\langle (g_i)_{i \in [1,K]} \right\rangle = \mathbb{Z}_{p^e}{}^* \right\}$$

$$\mathcal{P} = \frac{\varphi_K(\varphi(p^e))}{\varphi(p^e)^K} > \frac{1}{\zeta(K)} > 1 - \frac{K+1}{K-1} \times \frac{1}{2^K}$$

Proof: Let us first introduce a notation: for any integer x, let S_x be the set of the indices i such that p_i is a factor of x.

From the previous lemma, we know that the probability for a K-tuple of $(\mathbb{Z}_{p^e}{}^*)^K$ to generate $\mathbb{Z}_{p^e}{}^*$ is $P = \frac{\varphi_K(\varphi(p^e))}{\varphi(p^e)^K}$. Lemma 2 shows that P is equal to the product $\prod_{i \in S_{\varphi(p^e)}} 1 - \frac{1}{p_i^K}$. The inverse of each term $1 - 1/p_i^K$ can be expanded in power series: $(1 - 1/p_i^K)^{-1} = \sum_{j=0}^{+\infty} (1/p_i^K)^j$. The probability P is a product of series with positive terms, $P = \left(\prod_{i \in S_{\varphi(p^e)}} \sum_{\alpha_i = 0}^{+\infty} \frac{1}{p_i^{\alpha_i \times K}} \right)^{-1}$ so we can distribute terms and obtain that P^{-1} is the sum of $1/d^K$ where d ranges over integers whose prime factors are among p_is, $i \in S_{\varphi(p^e)}$. This sum is smaller than the unrestricted sum $\sum_{d=1}^{+\infty} 1/d^K = \zeta(K)$. Finally, we obtain $P > 1/\zeta(K)$.

The Riemann Zeta function is bounded by the following integral: $\zeta(K) = \sum_{d=1}^{+\infty} 1/d^K < 1 + 1/2^K + \int_2^{+\infty} dx/x^K = 1 + \frac{K+1}{K-1} \times \frac{1}{2^K}$. Since for all $x > -1$, $1/(1+x) \geq 1 - x$, $1/\zeta(K) > 1 - \frac{K+1}{K-1} \times \frac{1}{2^K}$. □

This result provides a lower bound independent of p and e. This is quite surprising since for $K = 1$, theorem 1 proves that such a non-zero bound does not exist.

A.3 Generation of a large subgroup of $\mathbb{Z}_{p^e}{}^*$ with K randomly chosen elements

Another statement of theorem 2 is that we need to randomly choose K elements in $\mathbb{Z}_{p^e}{}^*$ in order to to generate the group with probability greater than $1-1/2^K$. For a probability very close to one such that $1-1/2^{80}$, K becomes quite large even if experiments show that a few randomly chosen elements always generate very large subgroups of $\mathbb{Z}_{p^e}{}^*$. We now make this observation precise by establishing a lower bound of the probability $\mathcal{P}_K^C(p^e)$ that K elements generate a subgroup of size greater than $\mathcal{C}\mathrm{ard}\left(\mathbb{Z}_{p^e}{}^*\right)/C$. We first generalize lemma 3 for subgroups of $\mathbb{Z}_{p^e}{}^*$.

Lemma 4 *For any divisor d of $\varphi(p^e)$, the number of K-tuples of $\left(\mathbb{Z}_{p^e}{}^*\right)^K$ that generate a subgroup of $\mathbb{Z}_{p^e}{}^*$ of order d is $\varphi_K(d)$.*

Proof: Let $(g_1, \ldots g_K)$ be a K-tuple of $\left(\mathbb{Z}_{p^e}{}^*\right)^K$ and g be a generator of $\mathbb{Z}_{p^e}{}^*$; for $i = 1, \ldots K$ we define $\alpha_i \in \mathbb{Z}_{\varphi(p^e)}$ by the relation $g^{\alpha_i} = g_i \bmod p^e$. The K-tuple $(g_1, \ldots g_K)$ generates a subgroup of $\mathbb{Z}_{p^e}{}^*$ of order d if and only if the size of the ideal generated by the α_is in the ring $\mathbb{Z}_{\varphi(p^e)}$ is d. Bezout equality shows that this is equivalent to $\gcd(\alpha_1, \ldots \alpha_K, \varphi(p^e)) = \varphi(p^e)/d$.

Let us factor $\varphi(p^e)$ and d: $\varphi(p^e) = \prod_{i=1}^t q_i^{f_i}$ and $d = \prod_{i=1}^t q_i^{d_i}$ with $f_i \geq d_i \geq 0$. The number of K-tuples $(\beta_1, \ldots \beta_K)$ of $\left(\mathbb{Z}_{q_i^{f_i}}\right)^K$ such that $\gcd(\beta_1, \ldots \beta_K, q_i^{f_i}) = q_i^{f_i - d_i}$ is equal to the number of K-tuples of $\left(\mathbb{Z}_{q_i^{d_i}}\right)^K$ whose gcd with $q_i^{d_i}$ is 1, i.e. $\varphi_K(q_i^{d_i})$. The total number of K-tuples of $\left(\mathbb{Z}_{p^e}{}^*\right)^K$ that generate a subgroup of $\mathbb{Z}_{p^e}{}^*$ of order $\varphi(p^e)/d$ is consequently $\prod_{i=1}^t \varphi_K(q_i^{d_i}) = \varphi_K(d)$. □

Let K and C be two integers ≥ 1; we note

$$\mathcal{P}_K^C(n) = \Pr_{\{g_i\}_{i\in[1,K]} \in (\mathbb{Z}_n{}^*)^K} \left\{ \mathcal{C}\mathrm{ard} \left\langle (g_i)_{i\in[1,K]} \right\rangle \geq \frac{\mathcal{C}\mathrm{ard}\left(\mathbb{Z}_n{}^*\right)}{C} \right\}$$

Theorem 3 *For any $C \geq 1$ and $K \geq 2$,*

$$\mathcal{P}_K^C(p^e) > \frac{1}{\zeta(K)} \times \sum_{d=1}^{C} \frac{1}{d^K} > 1 - \frac{1}{(K-1)C^{K-1}\zeta(K)}$$

Proof: The probability $\mathcal{P}_K^C(p^e)$ is

$$\sum_{\delta \geq \varphi(p^e)/C} \left(\Pr_{\{g_i\}_{i\in[1,K]} \in (\mathbb{Z}_n{}^*)^K} \left\{ \mathcal{C}\mathrm{ard} \left\langle (g_i)_{i\in[1,K]} \right\rangle = \delta \right\} \right)$$

Lemma 4 shows that this sum is equal to $\displaystyle\sum_{\delta | \varphi(p^e), \delta \geq \varphi(p^e)/C} \frac{\varphi_K(\delta)}{\varphi(p^e)^K}$ and this expression can also be written $\displaystyle\sum_{d_0 | \varphi(p^e), d_0 \leq C} \frac{1}{d_0^K} \times \frac{\varphi_K(\varphi(p^e)/d_0)}{\varphi(p^e)/d_0{}^K}$ if we replace δ by $\varphi(p^e)/d_0$.

Let us calculate

$$\mathcal{P}_K^C(p^e) \times \zeta(K) = \sum_{d_0 | \varphi(p^e), d_0 \leq C} \frac{1}{d_0^K} \times \prod_{i \in S_{\varphi(p^e)/d_0}} \left(1 - \frac{1}{p_i^K}\right) \times \zeta(K)$$

In the Riemann Zeta function $\zeta(K) = \sum_{\beta=1}^{+\infty} 1/\beta^K$ the index β can be written as the product of two integers, d_1 which is relatively prime with $\varphi(p^e)/d_0$ and d_2 whose factors are among the p_is for $i \in S_{\varphi(p^e)/d_0}$. As in the proof of theorem 2, we note that the sum $\sum 1/d_2^K$ is equal to the inverse of $\prod_{i \in S_{\varphi(p^e)/d_0}} \left(1 - \frac{1}{p_i^K}\right)$ so we obtain that $\mathcal{P}_K^C(p^e) \times \zeta(K) = \sum_{d_0 | \varphi(p^e), d_0 \leq C} \left(\frac{1}{d_0^K} \times \sum_{\gcd(d_1, \varphi(p^e))=1} \frac{1}{d_1^K}\right)$. Finally let us observe that all the integers smaller than C can be uniquely decomposed in the product of a divisor d_0 of $\varphi(p^e)$ smaller than C and of an integer d_1 relatively prime with $\varphi(p^e)$. As a consequence, $\mathcal{P}_K^C(p^e) \times \zeta(K)$ is greater than $\sum_{d=1}^{C} 1/d^K$.

The end of the proof is a consequence of calculus techniques for comparing integrals and series:

$$\mathcal{P}_K^C(p^e) > \frac{\sum_{d=1}^{C} \frac{1}{d^K}}{\zeta(K)} = \frac{\zeta(K) - \sum_{d=C+1}^{+\infty} \frac{1}{d^K}}{\zeta(K)}$$

$$> 1 - \frac{1}{\zeta(K)} \times \int_C^{+\infty} \frac{dx}{x^K} = 1 - \frac{1}{(K-1)C^{K-1}\zeta(K)}$$

□

B Miller's factoring algorithm

Let n be an integer whose factorization has to be found and $L = 2^s \times r$, with r an odd integer, a multiple of $\lambda(n)$. We can first assume that n is odd because if L is a multiple of $\lambda(2^\alpha n) = \text{lcm}(\lambda(2^\alpha), \lambda(n))$ it is still a multiple of $\lambda(n)$. The following algorithm is due to Miller [18]:

Algorithm $Fact(n, L)$
1. choose w at random in $[1, n-1]$
2. if $1 < \gcd(w, n) < n$ then return $\gcd(w, n)$
3. compute $v = w^r \bmod n$
4. if $v = 1 \bmod n$ then return `Fail`
5. while $v \neq 1 \bmod n$ do $v_0 = v$ and $v = v^2 \bmod n$
6. if $v_0 = -1 \bmod n$ then return `Fail` else return $\gcd(v_0 + 1, n)$

This algorithm allows to find a non-trivial factor of n, i.e. a factor different from 1 and n. It can be recursively used to completely factor n.

Lemma 5 *Let n be a k-bit integer and $L < X$ be a multiple of $\lambda(n)$. Then $Fact(n, L)$ outputs the factorization of n in expected time $O(\eta \times \log X)$ modular multiplications, where η is the number of distinct prime factors of n.*

Proof: Let $\prod_{i=1}^{\eta} p_i^{e_i}$ be the prime factorization of n. We first prove that the algorithm $Fact(n, L)$ returns a non-obvious factor of n with probability $> 1 - 1/2^{\eta}$, after at most $O(\log X)$ arithmetical operations. The underlying idea is the same as in the Rabin-Miller primality test; if w is an element of $\mathbb{Z}_n{}^*$, $w^L = 1 \bmod n$ so if we find α such that $w^{2^{\alpha}r} \neq \pm 1$ and $w^{2^{\alpha+1}r} = 1$, we obtain $(w^{2^{\alpha}r} + 1)(w^{2^{\alpha}r} - 1) = 0 \bmod n$ and thus $\gcd(n, w^{2^{\alpha}r} + 1)$ is a non-trivial factor of n.

The proof generalizes the one presented in [27, chapter 4] when n is an RSA modulus. We first need the following notations in order to analyze the algorithm: $\lambda(p_i^{e_i}) = \varphi(p_i^{e_i}) = p_i^{e_i-1}(p_i - 1) = 2^{\alpha_i}p'_i$ with p'_i odd, $\mathcal{P} = \prod_{i=1}^{\eta} p'_i$, $\beta = \min\{\alpha_i, 1 \leq i \leq \eta\}$, g_i is a generator of the cyclic group $\mathbb{Z}_{p_i^{e_i}}{}^*$, u_i is such that $w = g^{u_i} \bmod p_i^{e_i}$ and $0 \leq u_i < \lambda(p_i^{e_i})$.

Let us count the number of w for which the algorithm fails. This happens if $w^r = 1 \bmod n$ or $w^{2^t r} = -1 \bmod n$ for an integer t in $[0, s[$. Because of the Chinese remainder theorem, $w^r = 1 \bmod n$ is equivalent to $\forall i, w^r = 1 \bmod p_i^{e_i}$. It can be seen that $w^r = 1 \bmod p_i^{e_i}$ if and only if there exists μ_i in $[0, p'_i[$ such that $u_i = \mu_i 2^{\alpha_i}$. So $w^r = 1 \bmod n$ has $\prod_{i=1}^{\eta} p'_i = \mathcal{P}$ solutions.

Using same ideas, $w^{2^t r} = -1 \bmod n$ is equivalent to $w^{2^t r} = -1 \bmod p_i^{e_i}$ for all $i = 1..\eta$. The latter equation has no solution if $t \geq \alpha_i$ and it has $\lambda(p_i)/(2 \times 2^{\alpha_i - t - 1}) = 2^t p'_i$ solutions if $t < \alpha_i$. So $w^{2^t r} = -1 \bmod p_i^{e_i}$ as no solution if $t \geq \beta$ and $2^{\eta t}\mathcal{P}$ solutions if $t < \beta$. Finally, the number of values w for which the algorithm fails is less than $\mathcal{P} + \sum_{t=0}^{\beta-1} 2^{\eta t}\mathcal{P}$. We can easily prove that this is less than $n/2^{\eta-1}$. The algorithm succeeds with probability $> 1 - 1/2^{\eta-1}$ and performs less than $s + 1 \leq \log_2 X \leq \log_2 L$ modular multiplications. If we use the algorithm until we find a non trivial factor of n, the expected number of executions if smaller than $1/(1 - 1/2^{\eta-1}) \leq 2$.

Fact can be used to recursively factor n. With this aim, we also need a prime power detection algorithm such as the one proposed in [7, page 41] and which is also based on Rabin-Miller ideas. If we want to factor an integer n, we first test if n is a prime power and if not we call $Fact(n, L)$ as long as it fails. After about two tries, we obtain $n = n' \times n''$ and we recursively call the factorization procedure. Notice that a multiple L of $\lambda(n)$ is also a multiple of $\lambda(n')$ if n' is a factor of n. The proposed algorithm needs on average less than 2η calls to *Fact* so finally we can factor n with $O(\eta \times \log X)$ modular multiplications. □