Fast Monte-Carlo Primality Evidence Shown in the Dark

Wenbo Mao
Trusted E-Services
HP Laboratories Bristol
HPL-1999-30(R.1)
27th October, 1999*

Email:wm@hplb.hpl.hp.com

interactive protocols, proof of knowledge, Monte-Carlo primality test

We construct an efficient proof of knowledge protocol for demonstrating Monte-Carlo evidence that a number n is the product of two odd primes of roughly equal size. The evidence is shown "in the dark", which means that the structure is verified without the prime factors of n disclosed. The cost of a proof amounts to $12k \log_2 n$ multiplications of integers of size of n where k is the number of the iterations in the proof and relates to an error probability bounded by $\max(1/2^k, 24/n^{1/4})$. To achieve cost and error probability similar to these, previous techniques require two additional conditions: (1) n is a Blum integer, and (2) a mutually trusted $k \log_2 n$-bit long random source is accessible by the proving/verification participants. In failure of (1), k must be increased substantially in order to keep error probability comparably small (e.g., k should be increased to 3000 for an error probability to remain at the level of $1/2^{60}$). In absence of (2), an additional $k \log_2 n$ iterations are needed for the participants to agree on the needed random input. We note that the drop of (1) due to this work will have a significance on applications.

* Internal Accession Date Only

Fast Monte-Carlo Primality Evidence Shown in the Dark

Wenbo Mao
Hewlett-Packard Laboratories,
Filton Road, Stoke Gifford,
Bristol BS34 8QZ, United Kingdom.
wm@hplb.hpl.hp.com

October 12, 1999

Abstract

We construct an efficient proof of knowledge protocol for demonstrating Monte-Carlo evidence that a number n is the product of two odd primes of roughly equal size. The evidence is shown "in the dark", which means that the structure is verified without the prime factors of n disclosed. The cost of a proof amounts to $12k \log_2 n$ multiplications of integers of size of n where k is the number of the iterations in the proof and relates to an error probability bounded by $\max(1/2^k, 24/n^{1/4})$. To achieve cost and error probability similar to these, previous techniques require two additional conditions: (1) n is a Blum integer, and (2) a mutually trusted $k \log_2 n$-bit long random source is accessible by the proving/verification participants. In failure of (1), k must be increased substantially in order to keep error probability comparably small (e.g., k should be increased to 3000 for an error probability to remain at the level of $1/2^{60}$). In absence of (2), an additional $k \log_2 n$ iterations are needed for the participants to agree on the needed random input. We note that the drop of (1) due to this work will have a significance in applications.

Key Words Interactive protocols, Proof of knowledge, Monte-Carlo primality test.

1 Introduction

In public-key cryptography, the private component of an individual user's cryptographic key should be known only to the user. On the other hand, the user's public key should be certified by a known authority for authentication. The authority may naturally demand that a public/private key pair have a valid private component that conforms to a set of agreed criteria. Proof of knowledge is a powerful tool that allows a user to run a protocol with a verification party, convincing the latter of the validity of the private key without the former disclosing it (thus the evidence of validity is shown in the dark). For instance, the ISO standardisation document 9798 part 3

([10]) recommends that public-key certification include knowledge proof for possession of the private component that matches the public key to be certified.

A key certification authority may also want to require that a user's key be generated at uniformly random which effectively prevents deliberate generation of week keys. Blackburn and Galbraith proposed a protocol for two parties to jointly generate a composite number which guarantees that the prime factors of the number are uniformly chosen [2]. At the end, only one party will know the two prime factors of the number that has been jointly generated and be able to prove to the other the two-prime-product structure.

We propose a protocol for efficient proof of the validity of private keys for cryptosystems based on the difficulty of integer factorisation (such as the RSA cryptosystem [15]), where the criterion to be validated is that an integer is the product of two primes of roughly equal size. The cost of a proof amounts to $12k \log_2 n$ multiplications of integers of size of n where k is the number of the iterations in the proof and relates to an error probability bounded by $\max(1/2^k, 24/n^{1/4})$. This is the first protocol that proves the two-prime-product structure of a number with the cost at the level of $O(k \log_2 n)$ multiplications and the error probability at the level of $1/2^k$ (considering $k = 60$ and $n > 2^{512}$, $1/2^k \gg 24/n^{1/4}$) regardless of whether the number in question is a Blum integer [3]. Previous techniques for proving such a structure have a much higher cost for non-Blum integers (details to be discussed in Section 2). The improved efficiency for reasoning about non-Blum integers due to this work manifests a particular suitability for using the proposed protocol in the proof of valid RSA keys which are generated at uniformly random (e.g., for the protocol of Blackburn and Galbraith [2]).

The remainder of the paper is organised as follows. In Section 2 we provide an analysis on the costs of the previous protocols for proving that an integer is the product of exactly two primes. In Section 3 we describe a new protocol of better performance. We analyse the security of the new protocol in Section 4 and consider its performance in Section 5. Finally, we conclude in Section 6.

2 Cost of Proving Two-Prime-Product Structure

Given the computational intractability of factoring large integers, to date there exists no known algorithm that inputs a given number n and terminates in a polynomial time in the size of n with an output answering whether n is the product of exactly two odd primes. Nevertheless, there exists practically efficient interactive protocols that run in polynomial time and allow a participant who knows the factorisation of n to prove such a structure to another without disclosing the factorisation information to the latter.

An early idea of proving n in such a structure is based on an observation due to Adleman (see [1]). He suggested to use the fact that if n has exactly two different

prime factors (may include their powers) then exactly a quarter of elements in the multiplicative group mod n are quadratic residues (square numbers mod n). On the other hand if n has more than two prime factors then at most one-eighth of them are quadratic residues. Thus, a prover, knowing the factorisation of n, can show a verifier the structure via binomial trials that for a set of k elements randomly chosen from the multiplicative group mod n, roughly $k/4$ of them are quadratic residues (shown by disclosing to the verifier their square roots). Using a normal distribution as an approximation to the probability of binomial trials (a standard method), Berger et al [1] established that if $\frac{\sqrt{21}-1}{20}k$ or more such elements are shown to be quadratic residues then the proof should be accepted with the probability of error between $e^{-k/74}$ and $e^{-k/75}$. Thus, k should be in thousands ($k = 3000$ was suggested in [1]) in order for the error probability to be negligibly small. (We note $e^{-3000/74} < 1/2^{58} < e^{-3000/75}$ and regard an amount at this level to be negligibly small.) Since the cost for computing a square root mod n is measured by $O(\log_2 n)$ multiplications of integers mod n, the total cost for proving the two-prime-product structure of a number n via showing quadratic residue information will be $O(k \log_2 n)$ (multiplications mod n) with an error probability between $e^{-k/74}$ and $e^{-k/75}$.

Van de Graaf and Peralta [9] observed that if n is a Blum integer, that is, n is the product of two distinct prime factors (again may include their powers), both congruent to 3 mod 4, then any element in the multiplicative group mod n with the positive Jacobi symbol has the property that either itself or its negation is a quadratic residue modulo n. Their protocol for proof of Blum integer is based on this fact. A number of other previous protocols for proving two-prime-product structure also use this idea (e.g., [5, 8, 12]). Note that provided n is not a square number (which is easy to test against), exactly half of the elements in the multiplicative group mod n can have positive Jacobi symbol which is also easy to evaluate. Thus, given such n, the above demonstration actually shows that a quarter of elements in the group are quadratic residues (since a quadratic residue must have positive Legendre symbol mod all prime factors, and only half of elements mod a prime have positive Jacobi symbol). If n is not in a two-prime-product structure then it is certainly not a Blum integer. Omitting details, for any group element of positive Jacobi symbol mod such n (which is non-Blum and non-square), a prover will have at most a 50% chance of correctly demonstrating the above. Clearly, such a proof using k random challenges will result in an error probability bounded by $1/2^k$, which approaches zero much faster than $e^{-k/74}$ (see the comparison between them in the previous paragraph).

The simplest way to show quadratic residue evidence to display a square root of a quadratic residue. (In the protocol of Van de Graaf and Peralta [9] for proving Blum integer, the verifier should check that the Jacobi symbol of a square root of a random challenge comply with a pre-agreed random sign. This follows Blum's observation that if n is a Blum integer, then any quadratic residue has square roots of positive and negative Jacobi symbols [3]. In the protocol of Gennaro et al [8], a verifier should

require that for each challenge g sent as challenge, a square root of either $\pm g$ or $\pm 2g$ mod n will be replied. It is possible for a prover to correctly respond such challenges if one of the prime factors of n is congruent to 5 mod 8, and the other, to 7 mod 8. These form an additional constraint to n being a Blum integer.) Note that two different square roots of a quadratic residue mod n can lead to factoring n with a non-trivial probability. So it will be dangerous for a prover to disclose a square root of a challenge which is solely selected by the verifier. The two protocols in [8, 9] assume the existence of a mutually trusted random source which is accessible by the prover and verifier. We believe that it will be costly to implement a mutually trusted random source between two mutually untrusted parties. The cost can be estimated by a protocol that allows the two parties to generate mutually trusted random elements without using a trusted third party. Blum's idea of coin flipping [3] is such a protocol and is used by [1, 7]. Each instantiation of that protocol generates a truely random bit. Each random challenge of size of n generated this way takes $\log_2 n$ iterations and the same number of multi-precision operations of integers mod n (evaluation of $\log_2 n$ Jacobi symbols). Together $k \log_2 n$ iterations are needed for merely agreeing on k mutually trusted random challenges.

Above we have analysed the cost for the previous protocols to prove an integer in the two-prime-power structure, i.e., $n = p^r q^s$ where p, q are distinct primes and r, s, integers. To further prove $r = s = 1$ one can use the protocol of Boyar et al [4] for proving square-free integers. Furthermore, to show that p and q are of roughly equal size one can use Damgård's method of "checking commitment" protocol [6]. We shall ignore the costs of applying these two additional protocols because they are less expensive than that for proving the two-prime-power structure, in particular for the case of non-Blum integers.

3 New Protocol

We describe in this section a new protocol for proving that a number is the product of two odd primes of roughly equal size.

3.1 Notations

Let P be a positive integer. Z_P^* denotes the multiplicative group of elements mod P. For $a \in Z_P^*$, $Ord_P(a)$ denote the order of a mod P.

Let a and b be integers. $a \mid b$ denotes a dividing b and $a \not| b$ otherwise; (a, b) denotes the greatest common divisor of a and b; $(\frac{a}{b})$ denotes the Jacobi symbol of a mod b; $\ell(a)$ denotes the size of a, which is the number of the bits in the binary representation of a.

Let x be a real number. $\lfloor x \rfloor$ denotes the integer part of x (thus $\ell(a) = \lfloor \log_2(a) \rfloor + 1$); $|x|$ denotes the absolute value of x.

Let S be a set. $\#S$ denotes the cardinality of S.

Finally, $Pr[E]$ denotes the probability for event E to occur.

3.2 Parameter Setup

Let Alice be a prover and Bob be a verifier. Alice has constructed $n = pq$ such that p and q are distinct odd primes with $|\ell(p) - \ell(q)| \leq 2$ (i.e., the sizes of the two primes differ by at most 2 bits).

First, Alice shall help Bob to set up a multiplicative group of order n. For her part, Alice only needs to generate a prime P with $n \,|\, (P-1)$. This prime can be constructed by testing the primality of $P = 2\alpha n + 1$ for $\alpha = 1, 2, \cdots$, until P is found to be prime. By the prime number theorem (general form due to Dirichlet, see e.g., p.28 of [11]), for fixed n with $P = 2\alpha n + 1 \leq N$, there are roughly

$$\pi_n(N) \approx \frac{1}{\phi(2n)} \cdot \frac{N}{\ln N}$$

such P's which are under N and are primes. Note that $N \geq 2\alpha n + 1$ and $n > \phi(2n)$. So

$$\pi_n(N) > \frac{2\alpha}{\ln(2\alpha n + 1)}.$$

Since Alice's primality test procedure uses $\alpha = 1, 2, \cdots$, the above inequality indicates a non-trivial probability for two primes to show up upon α reaching $\ln(2n \ln n)$. So we can be sure that α is small (likely to be bounded by $\ln(2n \ln n)$). It will be computationally easy for Alice to find the prime P. Once P is found to be prime, Alice shall send the numbers n and P to Bob.

Upon receipt of n and P, Bob shall test the primality of P. Upon passing of the test, he chooses a random element $f < P$, and sets

$$g = f^{(P-1)/n} \bmod P.$$

Bob then sends g to Alice.

Upon receipt of g, Alice shall check $Ord_P(g) = n$. If this does not hold, Alice may not be able to pass a proof later. Above we have reasoned that $2\alpha = (P-1)/n$ is small ($\ll n$). Thus, for $n = pq$, there can only be a few factors of $P - 1$ which are less than n and are fully known to Alice. So it will be computationally easy for Alice to check $Ord_P(g) = n$. Upon passing this simple checking, Alice shall set

$$A = g^p \bmod P, \qquad B = g^q \bmod P.$$

Alice then sends the pair (A, B) to Bob.

Upon receipt of (A, B), Bob shall check the following:

$$A \neq B, \quad A \neq 1, \quad B \neq 1.$$

If these checks are passed, the system parameters have been properly set up. Specified below is a protocol that, on input of n, A, B, P, demonstrates that n is the product of two odd primes of roughly equal size.

3.3 Protocol Specification

Now we specify the new protocol.

A proof will be abandoned if any check by Alice fails, or rejected if any check by Bob fails, or otherwise accepted. For clarity, we shall omit the trailing $\bmod P$ operation in the protocol specification.

Two_Prime_Product(n, g, A, B, P)
Repeat the following steps k times

1. Bob picks $h \in Z_n^*$ at random with $\left(\frac{h}{n}\right) = -1$ and sends it to Alice;

2. Alice checks $\left(\frac{h}{n}\right) = -1$, picks u, v at random such that

   $$\ell(u) = \ell((p-1)/2), \quad \ell(v) = \ell((q-1)/2),$$

   and sets

   $$U = g^{2u}, \;\; V = g^{2v}, \;\; H_U = B^{(h^u \bmod n)},$$
   $$H_V = A^{(h^v \bmod n)}, \quad H_{UV} = h^u h^v \bmod n;$$

   Alice sends to Bob: U, V, H_U, H_V, H_{UV};

3. Bob picks a challenge $c \in \{0, 1\}$ at random and sends it to Alice;

4. Alice sends Bob the responses

   $$r = u + c(p-1)/2, \qquad s = v + c(q-1)/2;$$

5. Bob checks:

   5.1 $\ell(r) \le \lfloor \ell(n)/2 \rfloor + 2, \qquad \ell(s) \le \lfloor \ell(n)/2 \rfloor + 2;$

   5.2 $$g^{2r+1} \equiv \begin{cases} Ug & c = 0 \\ UA & c = 1 \end{cases}, \quad g^{2s+1} \equiv \begin{cases} Vg & c = 0 \\ VB & c = 1 \end{cases};$$

   5.3 $$B^{(h^r \bmod n)} \equiv \begin{cases} H_U & c = 0 \\ H_U^{\pm 1} & c = 1 \end{cases}, \quad A^{(h^s \bmod n)} \equiv \begin{cases} H_V & c = 0 \\ H_V^{\mp 1} & c = 1 \end{cases};$$

   ($H_U^{\pm 1}$ and $H_V^{\mp 1}$ mean the two exponents taking opposite signs)

   5.4 $$h^r h^s \equiv \begin{cases} H_{UV} \pmod n & c = 0 \\ H_{UV} h^{(n-1)/2} \pmod n & c = 1 \end{cases}.$$

End.

We shall see in the next section that the two congruences checked in step 5.3 actually evaluate the Jacobi (Legendre) symbols $\left(\frac{h}{p}\right)$ and $\left(\frac{h}{q}\right)$. Using challenges of the negative Jacobi symbol has the virtue of not disclosing the quadratic residue information of the challenges. In contrast, many square-root displaying protocols (e.g., [8, 9]) disclose such information.

The protocol allows for the two factors to have size differences satisfying $|\ell(p) - \ell(q)| \leq 2$. Larger size differences, if desirable, can be accommodated by adjusting the inequalities in step 5.1.

4 Analyses

We analyze the security the protocol, which consists of the properties of completeness, soundness, and privacy.

4.1 Completeness

Theorem 1 *If Alice inputs the correct values into the protocol as specified. Then the proof will be accepted.*

Proof We show that Bob will be satisfied by the checks performed in protocol step 5.1 through step 5.4.

First, we show the inequalities in 5.1. Alice has set p and q such that $pq = n$

$$-2 \leq \ell(p) - \ell(q) \leq 2. \tag{1}$$

Obviously

$$\ell(n) \leq \ell(p) + \ell(q) \leq \ell(n) + 1. \tag{2}$$

Adding (1) to (2) yields

$$2\ell(p) \leq \ell(n) + 3,$$

or

$$\ell(p) \leq \lfloor \ell(n) \rfloor / 2 + 2. \tag{3}$$

Alice has chosen $\ell(u) = \ell((p-1)/2)$. With p odd, $(p-1)/2$ is a whole number and $\ell((p-1)/2) = \ell(p) - 1$. So when $c = 0$

$$\ell(r) = \ell(u) = \ell((p-1)/2) = \ell(p) - 1,$$

and When $c = 1$

$$\ell(r) = \ell(u + (p-1)/2) \leq \ell((p-1)/2) + 1 = \ell(p).$$

So for both cases, (3) will imply

$$\ell(r) \leq \lfloor \ell(n) \rfloor / 2 + 2.$$

Analogously we can show

$$\ell(s) \leq \lfloor \ell(n) \rfloor / 2 + 2.$$

In the following, we shall only examine the cases under $c = 1$, since $c = 0$ will render the congruences in 5.2 through 5.4 to hold trivially.

In 5.2, noting that $g^p \equiv A \pmod{P}$ and the structures of U and r, it is easy to see that the first congruence will hold. The second congruence holds similarly.

To see that the congruences in 5.3 will hold, observe

$$B^{(h^{(p-1)/2} \bmod n)} \equiv B^{(h^{(p-1)/2} \bmod p)} \equiv B^{\left(\frac{h}{p}\right)} \pmod{P}. \tag{4}$$

The first congruence in (4) is due to $Ord_P(B) = p \mid n$. Then, since p is prime, the second congruence in (4) follows from Euler's criterion. Therefore, the first congruence in 5.3 (for $c = 1$) is:

$$\begin{aligned} B^{(h^r \bmod n)} &\equiv B^{(h^{u+(p-1)/2} \bmod n)} \\ &\equiv (B^{(h^{(p-1)/2} \bmod n)})^{(h^u \bmod n)} \\ &\equiv (B^{(h^{(p-1)/2} \bmod p)})^{(h^u \bmod n)} \\ &\equiv (B^{\left(\frac{h}{p}\right)})^{(h^u \bmod n)} \\ &\equiv (B^{(h^u \bmod n)})^{\left(\frac{h}{p}\right)} \\ &\equiv H_U^{\left(\frac{h}{p}\right)} \pmod{P}, \end{aligned}$$

while the second congruence in 5.3 (for $c = 1$) is, analogously,

$$A^{(h^s \bmod n)} \equiv H_V^{\left(\frac{h}{q}\right)} \pmod{P}.$$

The exponents of the both right-hand sides must take opposite signs since Jacobi symbols only take values ± 1 and h has been chosen to satisfy

$$-1 = \left(\frac{h}{n}\right) = \left(\frac{h}{p}\right)\left(\frac{h}{q}\right).$$

Therefore the congruences in 5.3 will hold.

Finally, any $h \in Z_n^*$ will satisfy

$$h^{p+q} \equiv h^{n+1} \pmod{n}.$$

With $(p-1)/2$, $(q-1)/2$ and $(n-1)/2$ being whole numbers, it is easy to rewrite the above into

$$h^{[(p-1)/2+(q-1)/2]} \equiv h^{(n-1)/2} \pmod{n}.$$

Therefore the congruence in 5.4 will hold. □

4.2 Soundness

We now show that protocol Two_Prime_Product provides a Monte-Carlo method for testing the primality of the orders of A and B. We firstly note that all the numbers and variables to appear in this section are non-negative integers. In particular, $\log_g(A)$ and $\log_g(B)$ denote some positive integers p and q less than $Ord_P(g)$ satisfying $A \equiv g^p \pmod P$ and $B \equiv g^q \pmod P$.

Lemma 1 *Without the knowledge of the factorisation of n, the element g fixed by Bob satisfies*

$$Pr[\ Ord_P(g)\ \textit{divides}\ x\ ] = x/n,$$

for any x divides n.

Proof Without the knowledge of the factorisation of n, Bob's procedure for fixing g is via $g = f^{(P-1)/n} \bmod P$ using f which is chosen at random from Z_P^* (review Section 2.2). Then $g^n \equiv 1 \pmod P$ by Fermat's Theorem. In the cyclic group Z_P^* there are exactly $n = \sum_{d\,|\,n} \phi(d)$ elements of orders dividing n. Only these elements can be the candidates for g. For the same reason, for any $x \,|\, n \,|\, P-1$, there are exactly $x = \sum_{d\,|\,x} \phi(d)$ elements in Z_P^* of orders dividing x. The claimed probability is thus calculated as that of picking x objects from n. □

Lemma 2 *Denote $Ord_P(B) = x$ and $Ord_P(A) = y$. Upon acceptance of a proof on running* Two_Prime_Product(n, g, A, B, P), *Bob accepts that his random choice of h with $(h, n) = 1$ and $\left(\frac{h}{n}\right) = 1$ satisfies*

$$\begin{cases} h^{[(\log_g(A)-1)/2]} \equiv \pm 1 \pmod x \\ h^{[(\log_g(B)-1)/2]} \equiv \mp 1 \pmod y \end{cases}.$$

The probability for failing this does not exceed $1/2^k$ where k is the number of iterations used in the protocol.

Proof The first congruence in 5.2 shows that Alice knows both $\log_g(U)$ (shown when $c = 0$) and $\log_g(UA) = \log_g(U) + \log_g(A)$ (shown when $c = 1$), and has added $\log_g(A)$ to the response whenever $c = 1$ is the case. Suppose Alice does not know $\log_g(A)$. Then in each iteration she can only answer Bob's random challenge with at most $1/2$ chance of correctness. Thus, after having verified k times of correct responses to his random challenges, Bob should agree that the probability for Alice not having used $\log_g(A)$ in her response (when $c = 1$) is at most $1/2^k$.

The first congruence in 5.3 further shows that H_U is generated from B with the use of an exponent which is in turn generated from Bob's randomly chosen challenge h. Since $(h, n) = 1$, $(h^r \bmod n,\ n) = 1$. Therefore

$$Ord_P(H_U) = Ord_P(B) = x.$$

Clearly, the quantity $\log_g(A)$ in $2r + 1$ (when $c = 1$) amounts to $(\log_g(A) - 1)/2$ in r. Therefore the first congruence in 5.3 shows that for h satisfying $(h, n) = 1$:

$$h^{[(\log_g(A)-1)/2]} \equiv \pm 1 \pmod x.$$

Analogously we can use the second congruence in 5.3 to establish that for the same h

$$h^{[(\log_g(B)-1)/2]} \equiv \mp 1 \,(\mathrm{mod}\, y). \qquad \Box$$

In the rest of this section we will continue denoting

$$Ord_P(B) = x, \quad Ord_P(A) = y.$$

Following the Solovay-Strassen primality test technique [16] we define the following set

$$H_x = \{\, h \in Z_x^* \mid (h, x) = 1, \;\; h^\alpha \equiv \pm 1 \,(\mathrm{mod}\, x), \;\; \alpha \text{ constant} \,\}. \tag{5}$$

Clearly, this set is a subgroup of Z_x^*. It is a variation of its counterpart used in the Solovay-Strassen primality test technique. There, H_x is defined such that the exponent α is $(x-1)/2$. In our "test in the dark" method, the verifier Bob is not given the modulus x, let alone does he know the relation between the exponent and the modulus. All the information Bob has is that the modulus is a factor of n, and that the exponent is a constant. (The result of Lemma 2 stipulates the constant be $(\log_g(A) - 1)/2$.)

Lemma 3 *Let x, y be as in Lemma 2, α, β be constant integers, and h be an element satisfying $(h \bmod x,\, x) = (h \bmod y,\, y) = 1$. If the following test*

$$\begin{cases} h^\alpha \equiv \pm 1 \,(\mathrm{mod}\, x) \\ h^\beta \equiv \mp 1 \,(\mathrm{mod}\, y) \end{cases}$$

is passed for k such h's chosen at random, then both x and y are prime powers. The probability for failing this does not exceed $1/2^k$.

Proof We prove the lemma by estimating the probability for x not being a prime power. A prime power can be written as p^r with p prime and $r \geq 1$. Suppose x is not a prime power. Then let $x = \xi\eta$ with $\xi > 1$, $\eta > 1$ and $(\xi, \eta) = 1$.

Obviously, either H_x is a proper subgroup of Z_x^*, or $H_x = Z_x^*$.

In the first case, $\#H_x$ is at most half of $\#Z_x^*$ (since the former must divide the latter), and thereby the probability for each h randomly picked from Z_x^* to fall in H_x cannot exceed $1/2$, which amounts to $1/2^k$ to bound the probability for k such h's to be so.

Now we consider $H_x = Z_x^*$. We claim that H_x will only contain elements satisfying

$$h^\alpha \equiv 1 \,(\mathrm{mod}\, x). \tag{6}$$

Suppose $H_x = Z_x^*$ while (6) is not true for some element in H_x. Let h be such an element. So $h^\alpha \equiv -1 \,(\mathrm{mod}\, x)$. Since ξ and η are relatively prime, by the Chinese remainder theorem, the system $f \equiv 1 \,(\mathrm{mod}\, \xi)$, $f \equiv h \,(\mathrm{mod}\, \eta)$ has a solution $f \in Z_x^*$. Obviously,

$$f^\alpha \equiv 1 \,(\mathrm{mod}\, \xi), \qquad f^\alpha \equiv -1 \,(\mathrm{mod}\, \eta),$$

yielding

$$f^{\alpha} \not\equiv \pm 1 \,(\mathrm{mod}\, x).$$

So $f \in Z_x^* \setminus H_x$, contradiction to $H_x = Z_x^*$.

So now we must consider $H_x = Z_x^*$ with all elements in H_x satisfying (6). This implies that for k randomly chosen h's with $(h \bmod y,\, y) = 1$, $h^{\beta} \equiv -1 \,(\mathrm{mod}\, y)$. Let z be a prime factor of y. Then we will also have $(h^{\beta} \bmod z,\, z) = 1$ and

$$h^{\beta} \equiv -1 \,(\mathrm{mod}\, z). \tag{7}$$

Since z is prime, by Fermat's Theorem we know $z - 1 \,|\, 2\beta$, i.e., β is a multiple of $(z-1)/2$. In Z_z^* there are exactly half the elements which are quadratic non-residues satisfying (7) (none of other elements can satisfy it). So the probability for this congruence to hold for k randomly chosen h's cannot exceed $1/2^k$. This value must also bound the probability for x not being a prime power.

By symmetry, y is also a prime power. □

Lemma 4 *Under the hypotheses of Lemma 3, $(x, y) = 1$. The probability for failing this does not exceed $1/2^k$.*

Proof Since x and y are both prime powers, if $(x, y) > 1$, we can assume without loss of generality that $x = p^r \,|\, y$. Using the result of Lemma 2 we can derive

$$\mp 1 \equiv h^{\beta} (\mathrm{mod}\, y) \equiv h^{\beta} (\mathrm{mod}\, x).$$

At the same time we have

$$h^{\alpha} \equiv \pm 1 \,(\mathrm{mod}\, x).$$

Thus,

$$h^{|\alpha - \beta|} \equiv -1 \,(\mathrm{mod}\, x) \equiv -1 \,(\mathrm{mod}\, p),$$

for all k instances of randomly-picked h with $(h, p) = 1$. Since p is prime, the above is only possible if $p - 1$ divides $2|\alpha - \beta|$ but not divides $|\alpha - \beta|$. So $|\alpha - \beta|$ is an odd multiple of $(p-1)/2$ which implies

$$h^{(p-1)/2} \equiv -1 \,(\mathrm{mod}\, p) \tag{8}$$

for all such $h \bmod p$. There are only half the elements in Z_p^* which are quadratic non-residues satisfying (8). Therefore the probability for (8) to hold for k time, i.e., for k random h's with $h \bmod p$ being quadratic non-residues will not exceed $1/2^k$. Since the congruence in (8) is derived from the assumption $(x, y) > 1$, the value $1/2^k$ also bounds the probability for $(x, y) > 1$. □

Lemma 5 *Under the hypotheses of Lemma 2, there exists integers a and b satisfying*

$$\log_g(A) = ax \leq 8n^{1/2}, \quad \log_g(B) = by \leq 8n^{1/2}.$$

Proof From the proof of Lemma 2 we know that A is generated from g. So its order y can only be reduced from $Ord_P(g)$ and thereby $y \mid Ord_P(g)$. We also know

$$0 \equiv \log_g(1) \equiv \log_g(A^y) \equiv y \log_g(A) (\mathrm{mod}\, Ord_P(g)).$$

This means

$$Ord_P(g) \mid y \log_g(A). \tag{9}$$

By symmetry, $x \mid Ord_P(g) \mid x \log_g(B)$. Then $xy \mid Ord_P(g)$ since $(x, y) = 1$ (Lemma 4). Combining this with (9), we have $x \mid \log_g(A)$. By symmetry we can also derive

$$Ord_P(g) \mid x \log_g(B), \tag{10}$$

and $y \mid \log_g(B)$. So we can write

$$\log_g(A) = ax, \quad \log_g(B) = by,$$

for some a and b.

In protocol step 5.1. Bob has checked that in both challenge cases, the responses r and s satisfy

$$\ell(r) \leq \lfloor \ell(n)/2 \rfloor + 2.$$

Since when the challenge is $c = 1$, $\ell(\log_g(A)) \leq \ell(2r + 1) = \ell(r) + 1$,

$$\ell(\log_g(A)) \leq \lfloor \ell(n)/2 \rfloor + 3.$$

This implies

$$\log_g(A) \leq 2^{\lfloor \ell(n)/2 \rfloor + 3} \leq 8n^{1/2}.$$

By symmetry, $by = \log_g(B) \leq 8n^{1/2}$. □

Now we can prove the soundness of our protocol.

Theorem 2 *Upon acceptance of a proof on running* Two_Prime_Product(n, g, A, B, P) *where* $n \geq 24^4$ *and is odd, Bob accepts that* $x = \log_g(A)$, $y = \log_g(B)$, *and they are distinct odd primes. The probability for failing this does not exceed* $\max(1/2^k, 24/n^{1/4})$ *where* k *is the number of iterations used in the proof.*

Proof We know $x \neq y$ since they are relatively prime to each other. Both are odd since both divide an odd number n. By symmetry, we only need to prove the case for $x = \log_g(A)$ to be a prime. We have already established $ax = \log_g(A)$ (Lemma 5) and $x = p^r$ with p being prime (Lemma 3). So to prove this theorem we need only to show $a = r = 1$. We shall establish the probability for Bob to accept the proof while assuming either $r > 1$, or $a > 1$. Using the method that we have used in the proof of Lemma 3, we shall reason that if any of these two cases is true, then either H_x (defined in (5)) should be a proper subgroup of Z_x^*, which will render $1/2^k$ to bound the probability for Bob to accept a proof of k iterations, or another event of a negligibly small probability should has occurred.

First, consider the case of $r > 1$.

There exists $h \in Z_{p^r}^*$ of the full order $(p-1)p^{r-1}$. This element cannot be in H_x since otherwise the first congruence established in Lemma 2 will imply

$$h^{ap^r-1} \equiv 1 \,(\mathrm{mod}\, p^r),$$

which yields

$$(p-1)p^{r-1} \,|\, ap^r - 1.$$

So there exists λ satisfying

$$ap^r - \lambda(p-1)p^{r-1} = 1.$$

This means p^{r-1} is relatively prime to p^r, impossible with $r > 1$. So H_x must be a proper subgroup of Z_x^*.

The remaining case is $a > 1$ and x prime.

There exists $h \in Z_x^*$ of full order $x - 1$. If h is not in H_x then H_x is a proper subgroup and we have done. Now suppose $h \in H_x$. The first congruence in Lemma 2 implies

$$h^{ax-1} \equiv 1 \,(\mathrm{mod}\, x),$$

which further implies $x - 1 \,|\, ax - 1 = a(x-1) + a - 1$. So $x - 1 \,|\, a - 1$. This is only possible if $x \leq a$. From Lemma 5, $ax \leq 8n^{1/2}$. So $x^2 \leq ax \leq 8n^{1/2}$, or $x < 3n^{1/4}$. Lemma 5 also requires $\log_g(B) = by \leq 8n^{1/2}$. These yield

$$x \log_g(B) = xby < 24n^{3/4}.$$

So, this case of $\log_g(A)$ requires $xby < 24n^{3/4}$. From (10), $Ord_P(g) \,|\, xby$. Also, $Ord_P(g) \,|\, n$. So $Ord_P(g) \,|\, (xby, n) \leq xby \leq n$ $(n \geq 24^4)$. Now we can apply Lemma 1 and obtain

$$Pr[\, Ord_P(g) \text{ divides } (xby, n)\,] = (xby, n)/n \leq xby/n < 24/n^{1/4}.$$

We have shown that if x is not a prime, or $x \neq \log_g(A)$, then the probabilities for Bob to accept the proof are bounded by either $1/2^k$, or $24/n^{1/4}$, whichever is larger. The latter value bounds the probability for Bob to have chosen g of such a small order. □

Remark In the proof of Theorem 2 and Lemma 3 we have used random elements in Z_x^*. We should point out that in the protocol Bob only picks h at random from Z_n^*, rather than from Z_x^*, since he does not know the factorisation of n. Also h is chosen to have the negative Jacobi symbol mod n. However, the mapping from such h in Z_n^* to h mod x in Z_x^* is *onto* (the mapping is accomplished by the double exponentiations checked in protocol step 5.3) and thereby results in uniformly distributed elements in Z_x^*.

Theorem 3 *Under the hypotheses of Theorem 2,* $n = \log_g(A)\log_g(B)$. *The probability for failing this does not exceed* $\max(1/2^k, 8/n^{1/4})$.

Proof In Theorem 2 we have proved $\log_g(A)\log_g(B) = xy = Ord_P(g) \,|\, n$ where x and y are distinct primes. Suppose $n = xyz$ for some integer z. We prove the theorem by estimating the probability for $z > 1$.

The congruence checked in protocol step 5.4 implies that each h that Bob chooses at random satisfies

$$Ord_n(h) \,|\, n - x - y + 1 \tag{11}$$

Define the following set as a subgroup of Z_n^*:

$$H = \{\, h \in Z_n^* \mid h^{(n-x-y+1)} \equiv 1 \,(\mathrm{mod}\, n) \,\}.$$

Since x, y are distinct primes, there exists $h \in Z_n^*$ of order $\max(x-1, y-1)$. If $h \notin H$ then H is a proper subgroup of Z_n^* and $\#H$ cannot exceed the half of $\#Z_n^*$. Thus, the probability for choosing k random elements from Z_n^* which also fall in H (to pass the congruence in step 5.4) will not exceed $1/2^k$.

Now suppose $h \in H$. Without loss of generality, let $x - 1 \geq y - 1$. Then from (11) we can derive

$$x - 1 = Ord_n(h) \,|\, z - 1.$$

This is only possible if $x \leq z$. Given $y \leq 8n^{1/2}$, the maximum possible value for $Ord_P(g) = xy = n/z$ can only be resulted from the maximum possible value of $x = z$, which renders

$$Ord_P(g) = xy \leq 8n^{3/4}.$$

Applying Lemma 1 we know that the probability for Bob having chosen g of such a small order does not exceed $8/n^{1/4}$.

Thus, we can use $\max(1/2^k, 8/n^{1/4})$ to bound the probability for $z > 1$. □

To this end we know that the two primes factors of n have roughly equal size since

$$\log_g(A) \leq 8n^{1/2}, \quad \log_g(B) \leq 8n^{1/2}.$$

As a concluding remark for our soundness analysis, we emphasize the importance of verifying the congruences in the protocol step 5.3. Besides their roles in the soundness proof that we have seen, they also exclude x and y from being certain pseudoprimes such as Carmichael numbers (see e.g., p.137 of [13]). Moreover, they prevent x and y from being methodically chosen in a cheating way that can pass a (flawed) protocol in [12] for proof of a required format for RSA moduli. (The required format is the same as what our protocol proves: n is the product of exactly two primes of roughly equal size.) That protocol first applies a square-root displaying protocol to prove that n is the product of two prime powers ([12] suggests to use the method of [9] for proof of Blum integers; we will discuss more on square-root displaying protocols in Section 4), and then verifies

$$h^{x+y} \equiv h^{n+1} (\mathrm{mod}\, n)$$

(equivalent to the congruence checked in our protocol step 5.4), plus checking the sizes of x and y. Below we reason that such verification does not suffice for proving the required format of n.

Let $n = xy$ with x, y being odd. It is easy to see that, as long as $\lambda(n)$ (Carmichael function of n, which is the lowest order of all elements in Z_n^*) divides $(x-1)(y-1)/2 = (n-1)/2 - [(x-1)+(y-1)]/2$, the congruence above will always pass. Alice can thus cheat as follows. She sets $x = p^r$ with p prime and $r > 1$ such that $y = 2p^{r-1}+1$ is prime and $\ell(x) \approx \ell(y)$. There are sufficiently many primes p such that $2p^{r-1}+1$ is also prime. So it will be easy for Alice to find p and y to satisfy what is required. Clearly, n is the product of two prime powers, and will therefore pass a Blum integer proof based on displaying square roots of challenges. The size checking on x and y will pass too. Moreover,

$$(x-1)(y-1) = (p^r - 1)2p^{r-1} = (p-1)(...)2p^{r-1},$$

and

$$\lambda(n) = \mathrm{lcm}(\phi(x),\, \phi(y)) = \mathrm{lcm}((p-1)p^{r-1},\, 2p^{r-1}) = (p-1)p^{r-1}.$$

So it always holds

$$\lambda(n) \,|\, (x-1)(y-1)/2.$$

Consequently, verification using $h^{x+y} \equiv h^{n+1} (\mathrm{mod}\, n)$ will pass for all $h \in Z_n^*$. But n is not the product of exactly two primes, and the sizes of its prime factors are not roughly equal ($\ell(p) \approx \ell(y)/r$).

4.3 Privacy

Theorem 4 *Assume the computational infeasibility of computing discrete logarithms to the base g, of factoring n, and of determining $\left(\frac{h}{x}\right)$ for x being a factor of n with given $g^x (\mathrm{mod}\, P)$. Then on inputing $n = pq$ with p and q being odd primes, and with challenge h satisfying $\left(\frac{h}{n}\right) = -1$, the protocol* Two_Prime_Product *is in honest verifier zero-knowledge.*

Proof First, we note that in step 2, Alice picks two random numbers u and v with $\ell(u) = \ell((p-1)/2)$ and $\ell(v) = \ell((q-1)/2)$. Since p and q are odd, it is clear that for the challenge case $c = 1$, the responses r and s do not disclose any information about p and q.

We now show that given n, each iteration in a protocol run (with an honest verifier) can be simulated by a simulator in polynomial time. This includes to simulate the values g, A, and B. The simulator can pick the following values at uniformly random: it picks g, A, B of order n (e.g., by picking g at random as Bob does and setting $A = g^a \bmod P$, $B = g^b \bmod P$ using random values a and b), and picks h and c exactly as (an honest) verifier does in the protocol:

$h \in Z_n^*$ satisfying $\left(\frac{h}{n}\right) = -1$;

$c \in \{0,1\}$; if $c = 1$ then further picks $d \in \{1,-1\}$;

r, s satisfying $\ell(r) \leq \lfloor \ell(n)/2 \rfloor + 2$, $\ell(s) \leq \lfloor \ell(n)/2 \rfloor + 2$.

Using these random values, it then computes the following values to construct a simulated view.

For $c = 0$:

$$U \leftarrow g^{2r} \bmod P, \quad V \leftarrow g^{2s} \bmod P,$$

$$H_U \leftarrow B^{(h^r \bmod n)} \bmod P, \quad H_V \leftarrow A^{(h^s \bmod n)} \bmod P,$$

$$H_{UV} \leftarrow h^r h^s \bmod n;$$

For $c = 1$:

$$U \leftarrow g^{2r+1}/A \bmod P, \quad V \leftarrow g^{2s+1}/B \bmod P,$$

$$H_U \leftarrow (B^{(h^r \bmod n)})^d \bmod P, \quad H_V \leftarrow (A^{(h^s \bmod n)})^{-d} \bmod P,$$

$$H_{UV} \leftarrow h^r h^s h^{(n-1)/2} \bmod n;$$

Under the hypotheses of the theorem, the distribution of the values U, V, H_U, H_V, H_{UV} computed above is computationally indistinguishable from those resulted from a true proof run. Consequently, Bob gains no additional information about p and q other than from n. (We point out that since g is a quadratic residue, for both cases of c, all values computed in mod P are quadratic residues mod P, without being distinguishable by exploiting quadratic residue information.) □

We should point out that the simulator described in Theorem 4 does not simulate a run with a dishonest verifier. Such a verifier can force a proof's view to have a distribution which is distinguishable from that of our simulation. For instance, a dishonest verifier can choose an h with the negative Jacobi symbol, and then uses h^i with i being random odd numbers as challenges in the rest of the proof iterations. The proof view will then show a fixed Jacobi information in verification step 5.3 (e.g., fixed H_U and H_V^{-1} in step 5.3 for all cases of $c = 1$). This indicates that our protocol is only up to honest verifier zero-knowledge.

Finally we point out that the security of the new protocol rests not only on the difficulty of factorisation, but also on the discrete logarithm problem since two constants A and B have been disclosed where $p \equiv \log_g(A) (\bmod\, n)$. We believe that the disclosure of these two constants will not form a degradation for the difficulty of factoring n since the original problem of factoring an RSA modulus has never been harder than computing the discrete logarithm modulo the modulus as illustrated below for most $h \in Z_n^*$

$$h^{p+q} \equiv h^{n+1} (\bmod\, n).$$

The disclosure of A and B merely adds two additional elements to the huge set of such elements already available.

5 Performance

The operations in the protocol mainly involve exponentiations modulo big integers and evaluation of Jacobi symbols. Because the cost of the latter is trivial in comparison to that of the former, we shall focus our attention on estimating the cost of modulo exponentiations.

We shall not consider the cost for Alice to generate n and the related prime $P = 2\alpha n + 1$ since these procedures are purely local to Alice (while a protocol involves communications). She can prepare these two numbers well in advance before running the protocol. However, the cost to Bob of testing the primality of P should be included in the cost for him to run the protocol.

Testing the primality of P using a Monte-Carlo method needs k testing iterations to achieve $1/2^k$ error probability (using k the same as that in the protocol to equalise the error probability). Each iteration mainly involves exponentiation mod P. So for this part, Bob performs k exponentiations mod P.

In the proof protocol, in each iteration Alice computes four exponentiations mod P and two of them mod n. Bob performs slightly more: four of them mod P and on average 2.5 of them mod n (2 for $c = 0$ and 3 for $c = 1$). Thus, with a proof of k iterations, Alice computes $4k$ exponentiations mod P and $2k$ of them mod n. For Bob's part, adding the cost of testing the primality of P, he should perform in total $5k$ exponentiations mod P and $2.5k$ of them mod n.

Notice the fact that $P = 2\alpha n + 1$ where α is small (at the level of $\ln(2n \ln n)$, see Section 3.2). We have

$$\log_2 P - \log_2 n \approx \log_2[2 \ln(2n \ln n)]. \tag{12}$$

This means that the size of P may exceed that of n by only a few bits (for instance, for any n of size less than 10,000 bits, $\log_2[2 \ln(2n \ln n)] < 5$, which is less than two percent of the size of n). Since (12) renders that the growth of the size difference between the two moduli is much slower than that of the moduli, we can claim that for n of any size larger than 512 bits (recommended least size for today), the size of P will not exceed that of n by two percent (of the size of n), namely

$$\log_2 P \leq 1.02 \log_2 n.$$

Since in bit operation, the cost for exponentiation mod P is measured in $O((\log_2 P)^3)$, i.e., $C(\log_2 P)^3$ for some constant C, we can use the following to relate the cost of exponentiation mod P to that mod n (of any size larger than 512 bits):

$$(\log_2 P)^3 \leq (1.02 \log_2 n)^3 \approx 1.062 (\log_2 n)^3.$$

That is, the cost of one exponentiation mod P will not exceed that of one mod n by seven percent. We nevertheless use a ten percent expansion and convert Bob's

workload of $5k$ exponentiations mod P into $5.5k$ exponentiations mod n. So in total Bob will need to compute no more than $8k = (5.5 + 2.5)k$ exponentiations mod n. Alice will compute no more than $7k$ of them. Since on average an exponentiation mod n amounts to $1.5 \log_2 n$ multiplications mod n, the total cost to Bob for running the protocol will be $12k \log_2 n$ multiplications of integer of size of n. We can also use this quantity to bound Alice's cost of running the protocol.

Theorem 5 *For n of size larger than 512 bits, the computational cost of proving and verifying that n is the product of two primes of roughly equal size using protocol* Two_Prime_Product *is $12k \log_2 n$ multiplications of integer of size of n. Both parties should perform this amount of operations.* □

Considering the fact that a Monte-Carlo primality test on a non-secret number mainly involves modulo exponentiation, Bob's verification cost is equal to eight such tests on non-secret numbers of size of n.

6 Conclusion

We have constructed an efficient knowledge proof protocol for demonstrating an integer being the product of two prime factors of roughly equal size. The new protocol is the first of its kind that proves such a structure with efficiency comparable to that of a Monte-Carlo method for primality testing of non-secret numbers of comparable sizes regardless of whether or not the number is a Blum integer. It can be regarded as a fast Monte-Carlo method for showing primality evidence "in the dark".

A further investigation could be to construct a proof that shows Miller-Rabin primality evidence "in the dark". For a non-Blum integer, Miller-Rabin primality test has a lower error probability of $1/4^k$ (assuming $1/4^k \gg 24/n^{1/4}$ for n of secure size in this day). The new method shows a particular applicability to proving two-prime-product structure for non-Blum integers. Therefore a proof protocol in Miller-Rabin version will be of interest in achieving an even better performance.

Acknowledgments It is a pleasure to thank my colleagues Simon Crouch, Kenny Paterson and Nigel Smart for interesting discussions and helpful comments on the subject of the paper.

References

[1] R. Berger, S. Kannan and R. Peralta. A framework for the study of cryptographic protocols, Advances in Cryptology — Proceedings of CRYPTO 85 (H.C. Williams ed.), Lecture Notes in Computer Science, Springer-Verlag 218 (1986), pp. 87–103.

[2] S.R. Blackburn and S.D. Galbraith. Certification of secure RSA keys, University of Waterloo Centre for Applied Cryptographic Research, Technical Report CORR 99-44, Available from `http://www.cacr.math.uwaterloo.ca/`

[3] M. Blum. Coin flipping by telephone: a protocol for solving impossible problems, Proceedings of *24th IEEE Computer Conference (CompCon)*, 1982, pp. 133–137.

[4] J. Boyar, K. Friedl and C. Lund. Practical zero-knowledge proofs: Giving hints and using deficiencies, Advances in Cryptology — Proceedings of EUROCRYPT 89 (J.-J. Quisquater and J. Vandewalle, eds.), Lecture Notes in Computer Science, Springer-Verlag 434 (1990), pp. 155–172.

[5] J. Camenisch and M. Michels. Proving in zero-knowledge that a number is the product of two safe primes, In Advances in Cryptology — EUROCRYPT 99, Lecture Notes in Computer Science, Springer-Verlag 1592 (1999), pp. 106–121.

[6] I.B. Damgård. Practical and provably secure release of a secret and exchange of signatures, Advances in Cryptology: Proceedings of EUROCRYPT 93 (T. Helleseth, ed.), Lecture Notes in Computer Science, Springer-Verlag, 765 (1994), pp. 201–217.

[7] Z. Galil, S. Haber and M. Yung. A private interactive test of a boolean predicate and minimum-knowledge public-key cryptosystems, *26th FOCS*, 1985, pp. 360–371.

[8] R. Gennaro, D. Miccianicio and T. Rabin. An efficient non-interactive statistical zero-knowledge proof system for quasi-safe prime products, In 5th ACM Conference on Computer and Communications Security, October 1998.

[9] J. van de Graaf and R. Peralta. A simple and secure way to show the validity of your public key, Advances in Cryptology — Proceedings of CRYPTO 87 (E. Pomerance, ed.), Lecture Notes in Computer Science, Springer-Verlag 293 (1988), pp. 128–134.

[10] ISO/IEC 9798-3. "Information technology – Security techniques – Entity authentication mechanisms – Part 3: Entity authentication using a public-key algorithm", International Organization for Standardization, Geneva, Switzerland, 1993 (first edition).

[11] E. Kranakis. *Primality and Cryptography*, Wiley-Teubner Series in Computer Science, John Wiley & Sons, 1986.

[12] M. Liskov and R.D. Silverman. A statistical limited-knowledge proof for secure RSA keys, IEEE P1363 Research Contributions, Available at `http://grouper.ieee.org/groups/1363/contributions/ifkeyval.ps`

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

[14] S. Micali. Fair public key cryptosystems, Advances in Cryptology — Proceedings of CRYPTO 92 (E.F. Brickell, ed.) Lecture Notes in Computer Science Springer-Verlag 740 (1993), pp. 113–138.

[15] R.L. Rivest, A. Shamir and L.M. Adleman. A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM* v.21, n.2, 1978, pp. 120–126.

[16] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality, *SIAM J. Comput*, vol. 6, no. 1, March 1977, pp. 84–85.