Security of 2^t-Root Identification and Signatures

C.P. Schnorr

Fachbereich Mathematik/Informatik
Universität Frankfurt
PSF 111932
60054 Frankfurt/Main, Germany
e-mail: schnorr@cs.uni-frankfurt.de

Abstract. Ong-Schnorr identification and signatures are variants of the Fiat-Shamir scheme with short and fast communication and signatures. This scheme uses secret keys that are 2^t-roots modulo N of the public keys, whereas Fiat-Shamir uses square roots modulo N. Security for particular cases has recently been proved by Micali [M94] and Shoup [Sh96].

We prove that identification and signatures are secure for *arbitrary* moduli $N = pq$ unless N can easily be factored. The proven security of identification against active impersonation attacks depends on the maximal 2-power 2^m that divides either $p-1$ or $q-1$. We show that signatures are secure against adaptive chosen-message attacks. This proves the security of a very efficient signature scheme.

Keywords: identification, signature, Fiat-Shamir scheme, active/passive impersonation attacks, adaptive chosen-message attack, random oracle model, factoring of integers.

1 Introduction and Summary

Fiat and Shamir [FS86] proposed a practical identification and signature scheme that is based on a zeroknowledge protocol of Goldwasser, Micali and Rackoff [GMR89] for proving quadratic residuosity. The GQ-protocol of Guillou and Quisquater [GQ88] and Ong-Schnorr identification and signatures [OS90] are variants of the Fiat-Shamir scheme which provide shorter communication and signatures than the Fiat-Shamir scheme. The Ong-Schnorr scheme is a direct generalization of the Fiat-Shamir scheme, where square roots modulo N are replaced by 2^t-roots. This compact variant of the Fiat-Shamir scheme is as fast, in the number of modular multiplications, as the original scheme. Until recently it was only known that Ong-Schnorr identification is secure provided that particular 2^t-roots modulo N are hard to compute [OS90]. Recently there has been surprising progress for the case of *Blum integers* N, i.e. $N = p \cdot q$ with primes p, q that are congruent 3 mod 4.

Previous results. Micali [M94] proves that Ong-Schnorr signatures are secure if the secret key is a 2^t-root of 4, 2 is a quadratic non-residue modulo N and N is difficult to factor. In the case considered by Micali, the secret key, the 2^t-root

N. Koblitz (Ed.): Advances in Cryptology - CRYPTO '96, LNCS 1109, pp. 143-156, 1996.

of 4, reveals the prime factors of N. Therefore, distinct users must have different moduli N, and N must be part of the secret key rather than a public parameter as in the Fiat-Shamir scheme and its extension by Ong-Schnorr. Micali assumes that the hash function used for signatures acts as a random oracle.

Shoup [Sh96] proves that Ong-Schnorr identification with Blum integers N is secure against active adversaries unless N is easy to factor. Shoup transforms, less efficiently than for the Fiat-Shamir scheme, active impersonation attacks into the factorization of N. Shoup's reduction is not entirely constructive, as it requires a priori knowledge on the adversary's probability of success.

Our results. We prove that Ong-Schnorr identification is secure for *arbitrary* moduli $N = pq$. This extends and improves the results of Shoup in various ways. It sheds new light on the prime factors p and q of the modulus N. The efficiency of our reduction from factoring $N = pq$ to impersonation attacks depends on the maximal 2-power 2^m that divides either $p - 1$ or $q - 1$. We distinguish active and of passive attacks. In an *active attack* the adversary poses, before the impersonation attempt, as verifier in a sequence of executions of the ID-protocol and asks questions of his choice using the legitimate user as oracle. In a *passive impersonation attack* the adversary is given the public key, but he cannot even listen in to executions of the ID-protocol.

The cases that $m \geq t$, respectively $m < t$, are quite different. For $m \geq t$ we transform active impersonation attacks into a factorization of N, as efficiently as for Fiat-Shamir ID. This factoring method only requires that the adversary's success rate is twice the success rate for guessing the exam posed by the verifier. Moduli N with $m \geq t$ provide optimal security against active/passive impersonation attacks provided that N is difficult to factor.

For the case $m < t$, we transform *passive* impersonation attacks into the factorization of N, as efficiently as for Fiat-Shamir ID. The factoring method uses public keys that are generated together with a pseudo-key which is independent of the secret key. Having only a pseudo-key complicates for small m the reduction from factoring to active impersonation attacks. It becomes difficult to simulate the ID-protocol, which is necessary to provide the information needed by the adversary for an active impersonation attack. This leads to a trade-offf which we describe in Theorem 8: either there is an additional time factor 2^{t-m} for factoring N or the minimal required success rate of the active adversary increases by the factor 2^{t-m}.

Security of signatures. The aforementioned results translate into corresponding security results for Ong-Schnorr signatures. We assume that the public hash function of the signature scheme acts as a *random oracle*. The random oracle assumption has already been used in [FS86] and is commonly accepted to be appropriate for hash functions without cryptographic weaknesses, see [BR93]. We consider the strongest type of attacks, *adaptive chosen-message attacks*. Here the adversary uses, before attempting to generate a valid signature-message pair, the legitimate signer as oracle to sign messages of his choice.

Pointcheval and Stern [PS96] show how to transform security proofs for dis-

crete logarithm identification schemes into security proofs for the corresponding signature scheme. Using similar arguments we transform security of Ong-Schnorr ID, against passive attacks, into security of the corresponding signature scheme, against adaptive chosen-message attacks. In Theorem 6, we prove that signatures cannot be produced faster by an adaptive chosen-message attack, than by random trials unless the modulus N can easily be factored. We get the same result for *arbitrary* keys and moduli N which Micali [M94] proves for *particular* keys and moduli N.

Generalizing the properties of Blum integers. Blum integers N are characterized by the property that squaring acts as a permutation on the set QR_N of quadratic residues modulo N. The cryptographic relevance of Blum integers relies on this property. One of our basic tool is a generalization of this property to arbitrary integers N based on Lemma 2.

2 Ong-Schnorr identification

Let N be a product of two large primes p and q. Assume that N is public with unknown factorization. Let $\mathbb{Z}_N^*$ denote the multiplicative group of integers modulo N. Let the prover A have the private key $s = (s_1, ..., s_k)$ with components $s_1, ..., s_k \in \mathbb{Z}_N^*$. The corresponding public key $v = (v_1, ..., v_k)$ has components v_j satisfying $1/v_j = s_j^{2^t}$ for $j = 1, ..., k$. We assume that the verifier B has access to A's public key v.

Ong-Schnorr ID-protocol (A, B) (A proves its identity to verifier B)

1. A picks a random $r \in_R \mathbb{Z}_N^*$ and sends $x := r^{2^t}$ to B.
2. B picks a random exam $e = (e_1, \ldots, e_k) \in_R [0, 2^t)^k$ and sends it to A.
3. A sends $y := r \prod_j s_j^{e_j}$ to B.
4. B checks that $x = y^{2^t} \prod_j v_j^{e_j}$.

Standard forgery. It is known that a fraudulent prover $\tilde{A}$ can cheat by guessing the exam e and sending the crooked proof $x := r^{2^t} \prod_j v_j^{e_j}$, $y := r$. The probability of success is 2^{-kt}. Our goal is to prove that this 2^{-kt} success rate cannot be much improved unless we can easily factorize N. As the security level is 2^{kt}, we are interested in parameters k, t where kt is approximately 72.

Ong-Schnorr signatures are obtained by replacing in the ID-protocol the verifier B by a public hash function h. To sign a message M the signer picks a random $r \in_R \mathbb{Z}_N^*$ forms $x := r^{2^t}$ and computes the hash value $e := h(x, M)$ in $[0, 2^t)^k$ as well as $y := r \prod_j s_j^{e_j}$. The signature of the message M is the pair (e, y). It is verified by checking that $h(y^{2^t} \prod_j v_j^{e_j}, M) = e$ holds true.

The length of signatures and communication. For a security level of 2^t steps we only need hash-values $e = h(x, M)$ that are t bits long. The length of signatures is the length of the modulus plus t bits. Fiat, Shamir [FS86] cautiously recommend hash-values with 128 bits. It has been argued against shorter hash

values that the signer can compromise his key by constructing distinct messages having the same signature. By the birthday paradox, the signer can generate such colliding messages in time $O(2^{t/2})$. However this attack is not relevant, as the legitimate signer can always corrupt his key by revealing his secret. On the other hand successful attacks, without using the secret key, require 2^t steps.

In the ID-scheme, A can send in step 1 a hash-value $h(x)$ instead of x. Then B checks in step 4 that $h(x) = h(y^{2^t} \prod_j v_j^{e_j})$. Even in case of the ID-scheme, it suffices that $h(x)$ is slightly longer than t bits, see Girault, Stern [GS94] for a thorough analysis. It is tempting to let $h(x)$ consist of some bits of x. Only rather weak attacks are known, see [GS94].

Efficiency. For Ong-Schnorr identification (resp. signatures), both the prover (resp. signer) A and the verifier B perform on the average $\frac{k+2}{2}t$ multiplications in $\mathbb{Z}_N^*$. For $k = 8$, $t = 9$, these are 45 multiplications. Further optimization is possible in the same way as for the Fiat-Shamir scheme, see [FS86], [MS88]. While very fast generation of signatures requires long multi-keys, signature generation is rather efficient even for single component keys. For $k = 1$, $t = 72$ generation of signatures requires only 108 modular multiplications whereas RSA, using a 1000 bit modulus, requires 1500 modular multiplications on the average.

Verification of signatures is very efficient if the public key components v_j are integers with only a few non-zero bits in their binary representation. The verifier performs only t squarings, for computing y^{2^t} in $\mathbb{Z}_N^*$, and a few additions, shifts and reductions modulo N. If the binary representation of v_j has w_j many ones, a multiplication by v_j requires w_j additions, shifts and reductions modulo N. The reductions modulo N can be dismissed if the v_j are small integers. Micali and Shamir [MS88] propose public keys consisting of small primes v_j. More generally, the v_j can be small integers that are relatively prime and have small Hamming-weight.

Previous protocols. The original Fiat-Shamir scheme is the case $t = 1$ of the Ong-Schnorr protocol, repeated several times. While Fiat-Shamir ID requires t sequential rounds for a security level 2^{kt}, the Ong-Schnorr scheme compacts t rounds of the Fiat-Shamir scheme into a single round. Fiat-Shamir ID is secure against passive and active attacks unless N can easily be factored. Fiat-Shamir signatures are secure in the random oracle model [FS86], [FFS88]. Attacks with a success rate that is at least twice the probability of guessing the exam e, can be transformed into the factorization of N.

The GQ-protocol [GQ88] is the case of single component keys, where 2^t-powers $x = r^{2^t}$ are replaced by u-powers $x = r^u$ for an arbitrary integer u less than N. The GQ-protocol consists of a single round with a large exam e. This greatly reduces the length of transmission and of signatures compared to the Fiat-Shamir scheme, at the expense of a slightly increased work load.

Notation. Let the fraudulent prover $\tilde{A}$ be an interactive, probabilistic Turing machine that is given the fixed inputs k, t, N (k, t are sometimes omitted). Let RA be the sequence of coin tosses of $\tilde{A}$. Define the success bit $S_{\tilde{A},v}(RA, e)$ to be 1

if $\tilde{A}$ succeeds with v, RA, e, N and 0 otherwise. Accordingly call the pair (RA, e) *successful/unsuccessful*. The *success rate* $S_{\tilde{A},v}$ of $\tilde{A}$ with v is the expected value of $S_{\tilde{A},v}(RA, e)$ for uniformly distributed pairs (RA, e). For simplicity, we assume that the time $T_{\tilde{A},v}(RA, e)$ of $\tilde{A}$ with v, RA, e is the same for all pairs (RA, e), i.e. $T_{\tilde{A},v}(RA, e) = T_{\tilde{A},v}$. This is no restriction since limiting the time to twice the average running time of successful pairs (RA, e) decreases the success rate $S_{\tilde{A},v}$ at most by a factor 2. For simplicity we assume that $T_{\tilde{A},v} = \Omega(k \cdot t(\log_2 N)^3)$ and thus $T_{\tilde{A},v}$ covers the time of the correct verifier B.

Theorem 1. [OS90] *There is a probabilistic algorithm AL which, given the attacker $\tilde{A}$, N and v, computes $(y, \bar{y}, e, \bar{e})$ such that $y, \bar{y} \in \mathbb{Z}_N^*$, $e, \bar{e} \in [0, 2^t)^k, e \neq \bar{e}$ and $(y/\bar{y})^{2^t} = \prod_j v_j^{\bar{e}_j - e_j}$. If $S_{\tilde{A},v} \geq 2^{-tk+1}$, then AL runs in expected time $O(T_{\tilde{A},v}/S_{\tilde{A},v})$.*

The proof of Theorem 1 is a straightforward extension of Lemma 4 in [FFS88]. Algorithm AL constructs a random pair (RA, e) with $S_{\tilde{A},v}(RA, e) = 1$ and produces a second random exam $\bar{e}$ for which $\tilde{A}_f$ succeeds with the same RA. AL outputs $e, \bar{e}$ together with the replies $y, \bar{y}$ of $\tilde{A}$ associated with RA.

Theorem 1 does not yet transform successful attacks into a factorization of N. Let the public key components v_j are generated as $v_j := s_j^{-2^t}$ from random $s_j \in_R \mathbb{Z}_N$. Denoting $Y := y/\bar{y}$ and $S := \prod_j s_j^{e_j - \bar{e}_j}$, we have $Y^{2^t} = S^{2^t}$. Unfortunately $S^{2^{t-1}}$ can be independent of the 2^t-roots s_j of v_j. Otherwise, the factorization $\{\gcd(\mathrm{Y} \pm \mathrm{S}, \mathrm{N})\} = \{\mathrm{p}, \mathrm{q}\}$ would hold at least with probability $\frac{1}{2}$.

As our security proofs are based on Theorem 1, it is convenient to introduce some notation for the entities of Theorem 1. We denote $Y := y/\bar{y}$, $\ell := \max\{i \mid e = \bar{e} \bmod 2^i\}$, $Z := \prod_j s_j^{(e_j - \bar{e}_j)/2^\ell}$. By the construction we have $Y^{2^t} = Z^{2^{t+\ell}}$.

We use the structure of the prime factors p, q of N. Let $p-1 = 2^{m_p}p'$, $q-1 = 2^{m_q}q'$ with p', q' odd. W.l.o.g. let $m_q \geq m_p$ and denote $m := m_q = \max(m_p, m_q)$. We have $m = 1$ iff both p and q are congruent 3 mod 4, i.e. if N is a Blum integer. For Blum integers squaring acts as a permutation on the subgroup QR_N of quadratic residues in $\mathbb{Z}_N^*$. This property characterizes the set of Blum integers. Lemma 2 extends this property to arbitrary cyclic groups. For a multiplicative group G let G^u denote the subgroup of u-powers in G, $G^u = \{g^u \mid g \in G\}$. Lemma 2 is obvious.

Lemma 2. *For any cyclic group G of order $|G| = 2^m\bar{m}$ with $\bar{m}$ odd, squaring $SQ : G^{2^i} \to G^{2^{i+1}}$, $x \mapsto x^2$ is a $2-1$ mapping for $i = 0, \ldots, m-1$ and is $1-1$ for $i \geq m$.*

Extension of the Blum integer property. Let N, $m_p \leq m_q = m$ be as above. $\mathbb{Z}_N^*$ is direct product of the cyclic groups $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$. Hence squaring $\mathrm{SQ} : \mathbb{Z}_N^{*2^i} \to \mathbb{Z}_N^{*2^{i+1}}$, $\mathrm{x} \mapsto \mathrm{x}^2$, acts as a 4–1 mapping for $i < m_p$, as a 2–1 mapping for $m_p \leq i < m_q$, and as a permutation for $i \geq m_q = m$. With this observation we can extend cryptographic applications from Blum integers to arbitrary moduli.

3 Passive impersonation attacks for $m \geq t$

We show that Ong-Schnorr ID is for $m \geq t$ as secure as Fiat-Shamir ID. We assume that k and t are given as input along with N, but m may be unknown.

Theorem 3. *There is a probabilistic algorithm which on input $\tilde{A}$, N generates a random public key $v \in_R (\mathbb{Z}_N^{*2^t})^k$, factorizes N with probability at least $1/2$, with respect to its coin tosses, and runs in expected time $O(T_{\tilde{A},v}/S_{\tilde{A},v})$ provided that $S_{\tilde{A},v} \geq 2^{-kt+1}$ and $t \leq m$.*

Proof. The factoring algorithm picks random $s_j \in_R \mathbb{Z}_N^*$, sets $1/v_j := s_j^{2^t}$ for $j = 1, \ldots, k$, runs algorithm AL of Theorem 1 on input $\tilde{A}, N, v$ producing the output $(y, \bar{y}, e, \bar{e})$, and computes the corresponding ℓ, Y, Z with $Y^{2^t} = Z^{2^{t+\ell}}$. Then, it checks whether

$$\{\gcd(Y^{2^i} \pm Z^{2^{i+\ell}}, N)\} = \{p, q\} \quad \text{holds for some} \quad i,\ 0 \leq i < t\ .$$

For the analysis we assume w.l.o.g. that $(e_1 - \bar{e}_1)/2^\ell$ is odd. The probability space consists of the coin tosses of AL including $s_j \in_R \mathbb{Z}_N^*$ for $j = 1, \ldots, k$. To simplify the analysis we fix Y, $Z (\mathrm{mod}\, p)$, $s_2 (\mathrm{mod}\, q), \ldots, s_k (\mathrm{mod}\, q)$ so that the probability space reduces to $s_1 (\mathrm{mod}\ q) \in_R \mathbb{Z}_q^*$. By Lemma 2 and since $t \leq m$ there are 2^t different 2^t-roots $s_1 (\mathrm{mod}\, q)$ of $1/v_1 = s_1^{2^t} (\mathrm{mod}\, q)$. They yield 2^t different values $Z (\mathrm{mod}\, q)$. Since $\ell < t \leq m$, we have for at least 2^{t-1} of these cases that $Y \neq \pm Z^{2^\ell}$ and that $Y^{2^{i+1}} = Z^{2^{i+1+\ell}}$ holds for the largest $i < t$ which satisfies $Y^{2^i} \neq \pm Z^{2^{i+\ell}}$. Hence for at least half of the cases we obtain square roots Y^{2^i} and $Z^{2^{i+\ell}}$ of the same square modulo N, that are distinct even when changing the sign, and thus $\{\gcd(Y^{2^i} \pm Z^{2^{i+\ell}}, N)\} = \{p, q\}$. This shows that the algorithm factorizes N with probability at least $1/2$.

The expected time of the factoring algorithm is that of algorithm AL of Theorem 1. The other steps are negligible due to the assumption $T_{\tilde{A},v} = \Omega(k \cdot t(\log_2 N)^3)$. □

A basic difficulty with the above factoring algorithm is that it requires $\ell < m$, while the construction only guarantees $\ell < t$. If $\ell \geq m$ it can happen that $Y = Z^{2^\ell}$ holds for all possible 2^t-roots s_j of $1/v_j$. In this case the factoring method breaks down completely.

Lemma 4. *Let $\bar{m}$ be an arbitrary integer with $1 \leq \bar{m} \leq t$. Algorithm AL of Theorem 1 produces on input $\tilde{A}, v$ an output $(y, \bar{y}, e, \bar{e})$ so that $e \neq \bar{e} \bmod 2^{\bar{m}}$ holds with probability $\geq 1/4$ provided that $S_{\tilde{A},v} \geq 2^{-k\bar{m}+2}$.*

The Lemma shows that the algorithm of Theorem 3 factorizes N with probability at least $1/8$ and runs in expected time $O(T_{\tilde{A},v}/S_{\tilde{A},v})$ provided that $S_{\tilde{A},v} \geq 2^{-k\bar{m}+2}$.

Proof. A coin tossing sequence RA of $\tilde{A}$ is called *$\bar{m}$-heavy* if $\sum_e S_{\tilde{A},v}(RA, e) \geq$

$2^{kt-k\bar{m}+1}$, i.e. if $\tilde{A}$ succeeds for at least a $2^{-k\bar{m}+1}$ fraction of the e. The claim follows from facts A and B.

Fact A. *If RA is $\bar{m}$-heavy and $S_{\tilde{A},v}(RA,e) = 1$ then $e \neq \bar{e} \bmod 2^{\bar{m}}$ holds for at least half of the $\bar{e}$ with $S_{\tilde{A},v}(RA,\bar{e}) = 1$.*

Proof. For every e we have $\#\{\bar{e} \mid e = \bar{e} \bmod 2^{\bar{m}}\} \leq 2^{kt-k\bar{m}}$ since $e_i = \bar{e}_i \bmod 2^{\bar{m}}$ holds for at most a $2^{-\bar{m}}$-fraction of the $\bar{e}_i$. Now the fact follows since RA is $\bar{m}$-heavy.

Fact B. *If $S_{\tilde{A},v} \geq 2^{-k\bar{m}+2}$ then RA is $\bar{m}$-heavy for at least half of the pairs (RA,e) with $S_{\tilde{A},v}(RA,e) = 1$.*

Proof. If RA is not $\bar{m}$-heavy at most a $2^{-k\bar{m}+1}$-fraction of the e satisfy $S_{\tilde{A},v}(RA,e) = 1$. On the other hand, since $S_{\tilde{A},v} \geq 2^{-k\bar{m}+2}$, at least a $2^{-k\bar{m}+2}$ fraction of the (RA,e) satisfy $S_{\tilde{A},v}(RA,e) = 1$.

Algorithm AL generates a random pair (RA,e) with $S_{\tilde{A},v}(RA,e) = 1$. By Fact B RA is $\bar{m}$-heavy with probability $\geq 1/2$. After fixing (RA,e) so that $S_{\tilde{A},v}(RA,e) = 1$, AL generates a random $\bar{e}$ with $S_{\tilde{A},v}(RA,\bar{e}) = 1$. By Fact A $e \neq \bar{e} \bmod 2^{\bar{m}}$ holds with probability $\geq 1/4$. □

Remark. The lower bound $S_{\tilde{A},v} > 2^{-k\bar{m}}$ is necessary in Lemma 4. It is possible to position a $2^{-k\bar{m}}$-fraction of successes so that $e = \bar{e} \bmod 2^m$ always holds.

4 Passive impersonation attacks for $m < t$

For $m < t$ we give another reduction from factoring to impersonation. The factoring algorithm generates a random public key v together with a pseudo-key $\tilde{s}$ which enables to transform successful attacks of a passive adversary $\tilde{A}$ into the factorization of N.

Theorem 5. *There is a prob. algorithm which, given $\tilde{A}$ and N, generates a random public key $v \in_R (\mathbb{Z}_N^{*2^t})^k$, factorizes N with probability $\geq 1/2$ with respect to its coin tosses, and runs in expected time $O(T_{\tilde{A},v}/S_{\tilde{A},v})$ provided that $S_{\tilde{A},v} \geq 2^{-kt+1}$ and $m < t$.*

Proof. Factoring algorithm

1. Pick random $\tilde{s}_j \in_R \mathbb{Z}_N^*$, set $1/v_j := \tilde{s}_j^{2^m}$ for $j = 1,\ldots,k$ (thus $v_j \in_R \mathbb{Z}_N^{*2^t}$).
2. According to Theorem 1 compute $AL : (\tilde{A},v) \mapsto (y,\bar{y},e,\bar{e})$ and set $\ell := \max\{i \mid e = e \bmod 2^i\}$, $Y := y/\bar{y}$, $\tilde{Z} := \prod_j \tilde{s}_j^{(e_j-\bar{e}_j)/2^\ell}$.
3. Test whether for some i, $\ell < i \leq t$: $\{\gcd(Y^{2^{t-i}} \pm \tilde{Z}^{2^{\ell+m-i}}, N)\} = \{p,q\}$.

By the construction we have $Y^{2^t} = \tilde{Z}^{2^{\ell+m}}$ and $\ell < t$. W.l.o.g. let $(e_1-\bar{e}_1)/2^\ell$ be odd. Arbitrarily fix $\tilde{Z} \bmod p$, $\tilde{s}_2 \bmod q, \ldots, \tilde{s}_k \bmod q$ and Y so that the probability space reduces to the 2^m solutions $\tilde{s}_1 \bmod q$ of $\tilde{s}_1^{2^m} = 1/v_1 \bmod q$. These 2^m solutions yield 2^m different values $\tilde{s}_1 \in \mathbb{Z}_N^*$ and, since $(e_1-\bar{e}_1)/2^\ell$ is odd, they generate 2^m different values $\tilde{Z} \in \mathbb{Z}_N^*$. For at least 2^{m-1} of these cases we have that $Y^{2^{t-\ell-1}} \neq \pm\tilde{Z}^{2^{m-1}}$ and that $Y^{2^{t-i+1}} = \tilde{Z}^{2^{\ell+m-i+1}}$ holds for the

largest i which satisfies $Y^{2^{t-i}} \neq \pm \tilde{Z}^{2^{t+m-i}}$. Hence for at least half of the cases we obtain square roots $Y^{2^{t-i}}$, $\tilde{Z}^{2^{t+m-i}}$ of the same square, that are distinct even when changing the sign, and thus $\{\gcd(Y^{2^{t-i}} \pm \tilde{Z}^{2^{t+m-i}}, N)\} = \{p, q\}$. This shows that the algorithm factorizes at least with probability $1/2$. □

The above proof establishes security of public keys v that are generated without a corresponding secret key s. We have generated v from a random pseudo-key $\tilde{s}$ so that $1/v_j = \tilde{s}_j^{2^m}$ holds for $j = 1, \ldots, k$. We cannot generate first a secret key s to produce a pseudo-key $\tilde{s}$ by squaring the components of s, as the components $\tilde{s}_j$ are, with probability $3/4$, quadratic non-residues. If we have v and s together with $\tilde{s}$, we can easily factor N.

5 Security of Ong-Schnorr signatures

We study the security of Ong-Schnorr signatures in the *random oracle model* where the hash function h is replaced by a random oracle. This is widely believed to be the appropriate model for hash functions without cryptographic weaknesses. This model has already been used in [FS86] and has been further developed in [BR93]. In the random oracle model, the hash function h produces for each query (x, M) a random value $h(x, M) \in_R [0, 2^t)^k$. If the same query is repeated, the same answer must be given.

We consider the most powerful attacks: adaptive chosen-message attacks as introduced by Goldwasser, Micali, Rivest [GMR88]. The adversary, before attempting to generate a new message-signature pair, uses the legitimate signer as an oracle to sign messages of his choice.

The strength of the adaptive chosen-message attack gets somewhat diluted by the random oracle assumption. The hash values $h(x, M)$ are random in $[0, 2^t)^k$ and independent for distinct pairs (x, M). The adversary cannot get anything from signatures (e, y) that are produced according to the protocol. Such signatures are random pairs in $[0, 2^t)^k \times \mathbb{Z}_N^*$. In the random oracle model, adaptive chosen-message attacks are not stronger than no-message attacks, where the attacker is merely given the public key.

For the next theorem, let $\tilde{A}_f$ be an attacker which, given N and the public key v, executes an adaptive chosen-message attack, where the oracle for the hash function h is queried at most f times, $f \geq 1$. Firstly, $\tilde{A}_f, v$ asks for signatures of messages of his choice, and then attempts to produce a new messgae-signature pair. Let $T_{\tilde{A}_f,v}$ be its expected time and $S_{\tilde{A}_f,v}$ its success rate with v.

Theorem 6. *There is a probabilistic algorithm which, given the attacker $\tilde{A}_f$ and N, generates a random $v \in_R (\mathbb{Z}_N^{*2^t})^k$, factorizes N with probability at least $1/2$ with respect to its coin tosses, and runs in expected time $O(f\, T_{\tilde{A}_f,v}/S_{\tilde{A}_f,v})$ provided that $S_{\tilde{A}_f,v} \geq f\, 2^{-kt+1}$.*

Compared to Theorem 3 there is an additional factor f both in the time bound for factoring as well as in the minimal required success rate of $\tilde{A}_f$. We

explain below that the second factor f is necessary. The first factor f comes in because the adversary cannot solicit a successful oracle query, one that results in a valid signature. It is open whether the factor f in the time bound is necessary. This is not a weakness of Ong-Schnorr signatures as this factor appears already for Fiat-Shamir signatures, see Lemma 7 of [FS86]. This Lemma claims the time bound $O(T^2 2^{kt})$ without giving a proof. We prove a stronger time bound for general Ong-Schnorr signatures.

Proof. Depending on whether $m \geq t$ or $m < t$, we mimic the factoring algorithms corresponding to Theorems 3 and 5. We first give an informal argument for the case $m \geq t$.

The factoring algorithm picks random $s_j \in_R \mathbb{Z}_N^*$, sets $1/v_j := s_j^{2^t}$ for $j = 1, \ldots, k$, and lets $\tilde{A}_f$ execute his attack on the public key v. For the signatures requested by $\tilde{A}_f$ it produces random pairs in $[0, 2^t)^k \times \mathbb{Z}_N^*$. Suppose that $\tilde{A}_f$ queries the h-oracle about (x_i, M_i) for $i = 1, \ldots, f$ and outputs the message-signature pair (M, e, y).

We can assume that $(y^{2^t} \prod_j v_j^{e_j}, M) = (x_i, M_i)$ holds for some $i \leq f$, since otherwise $e = h(y^{2^t} \prod_j v_j^{e_j}, M)$ holds with probability 2^{-kt}. If the adversary produces $x_i := y^{2^t} \prod_j v_j^{e_j}$ for some preselected e and y, the oracle returns the preselected e with probability 2^{-kt}. Each oracle query contributes at most 2^{-kt} to the success rate $S_{\tilde{A}_f, v}$. Hence at least with probability $S_{\tilde{A}_f, v} - f2^{-kt}$, the attacker $\tilde{A}_f$ is able to produce two distinct pairs (e, y) and $(\bar{e}, \bar{y})$ so that $e \neq \bar{e}$ and $y^{2^t} \prod_j v_j^{e_j} = \bar{y}^{2^t} \prod_j v_j^{\bar{e}_j} = x_i$. For these pairs we have $(y/\bar{y})^{2^t} = \prod_j v_j^{\bar{e}_j - e_j}$, and $(y, \bar{y}, e, \bar{e})$ has the same properties as the output of algorithm AL of Theorem 1. It yields the factorization of N with probability $1/2$ as described in Theorem 3.

The formal factoring algorithm constructs the above mentioned pairs (e, y), $(\bar{e}, \bar{y})$ employing a version of algorithm AL of Theorem 1. It simulates $\tilde{A}_f$ using statistically independent oracles for h.

Factoring algorithm

1. Pick random $s_j \in_R \mathbb{Z}_N^*$, set $1/v_j := s_j^{2^t}$ for $j = 1, \ldots, k$ and $u := 0$
2. Pick a random sequence of coin tosses RA for $\tilde{A}_f$.
3. (first signing attempt) Simulate the adversary $\tilde{A}_f$ with v, RA.
   For the message signature pairs requested by $\tilde{A}_f$ provide random signatures.
   Let the adversary query the h-oracle about (x_i, M_i) for $i = 1, \ldots, f$.
   If $\tilde{A}_f$ fabricates a signature (e, y) satisfying $y^{2^t} \prod_j v_j^{e_j} = x_i$ for some i (in this case we call the pair (RA, e) *successful* with x_i) then fix RA, i, x_i, M_i, e, y, set $u := 4uf$ and go to step 4. Otherwise, increase u by 1 and go back to step 2 undoing $\tilde{A}_f$'s computation.
4. (second signing attempt) Simulate the adversary $\tilde{A}_f$ with v, RA.
   Let the oracle answer the first $i - 1$ queries the same way as in step 3.
   Let it answer the other queries statistically independent from previous oracle outputs. (In particular, the oracle is repeatedly queried about the

(x_i, M_i) of step 3 providing statistically independent replies $\bar{e}$.)

If $\tilde{A}_f$ fabricates a second signature $(\bar{e}, \bar{y})$ satisfying $\bar{y}^{2^{\ell}} \prod_j v_j^{\bar{e}_j} = x_i$, then go to step 5. Otherwise, set $u := u - 1$, if $u > 0$ go back to step 4, if $u = 0$ go back to step 2 (undoing the computation of $\tilde{A}_f$ in either case).

5. Compute $Y := y/\bar{y}$, $\ell := \max\{i \mid e = \bar{e} \bmod 2^i\}$, $Z := \prod_j s_j^{(e_j - \bar{e}_j)/2^{\ell}}$ (hence $Y^{2^{\ell}} = Z^{2^{t+\ell}}$).
6. Test whether $\{\gcd(Y^{2^{t-\ell}} \pm Z^{2^{t+\ell-\ell}}, N)\} = \{p, q\}$ holds for some $i \leq t$.

Sketch of the analysis. On the average it takes $1/S_{\tilde{A}_f,v}$ many passes of steps 2 and 3 to find i, x_i, M_i, e, y. If $S_{\tilde{A}_f,v} > f\, 2^{-kt+1}$ the subsequent step 4 fabricates a second signature $(\bar{e}, \bar{y})$ with the same x_i at least with probability $\frac{1}{4}(1 - 2.7^{-1})$. For this we note: with probability at least $\frac{1}{4}$, step 3 probes at least $u \geq \frac{1}{2} S^{-1}_{\tilde{A}_f,v}$ many pairs (RA, e) and fixes some RA for which the fraction of successful pairs $(RA, \bar{e})$ is at least $\frac{1}{2} S^{-1}_{\tilde{A}_f,v}$. In this case, at least a $\frac{1}{2f} S^{-1}_{\tilde{A}_f,v}$-fraction of $\bar{e}$ succeeds in step 4 with the x_i of step 3. Since step 4 probes at least $2f S_{\tilde{A}_f,v}$ many random $\bar{e}$ it succeeds at least with probability $1 - 2.7^{-1}$. (The additional factor f for the number u of probes in step 4 compensates for the number of possibilities for successful queries. Only a second signature with the same i, x_i of the first signature can possibly factor N.) Finally, steps 5 and 6 factorize N at least with probability $1/2$.

In case that $m < t$, the factoring algorithm generates, as in the proof of Theorem 5, the public key from a random pseudo-key $\tilde{s}$ and factorizes N according to Theorem 5. □

6 Ong-Schnorr ID is secure against active impersonation

Theorem 7 extends the reduction of Theorem 3 from passive to active impersonation attacks. Theorem 8 presents, for arbitrary moduli $N = p \cdot q$ with $m \leq t$, a reduction from factoring to active impersonation attacks. The latter result extends and improves the reduction given by Shoup for the case of Blum integers N. The efficiency of the reduction depends in an interesting way on the parameter m. While the reduction is quite efficient for m close to t, it is less efficient for Blum integers, where $m = 1$. This deficiency of Blum integers was not apparent from Shoup's proof. Shoup's proof of security is not entirely constructive. It requires a priori knowledge about the success rate of the adversary $\tilde{A}_f$, given the knowledge from the f executions of the protocol $(A, \tilde{A}_f)$. We eliminate this a priori knowledge. In a way, Theorem 7 combines Shoup's argument with the proof of Lemma 4 [FFS88].

An active adversary, before the impersonation attempt, poses as B in a sequence of executions of the protocol (A, B), asking A questions of his choice without necessarily following the protocol of B. Then, he attempts to pose as A in the protocol (A, B). For short we let $\tilde{A}_f$ denote an active adversary who asks for f ID-proofs of A via $(A, \tilde{A}_f)$ and then attempts to impersonate A in protocol

$(\tilde{A}_f, B)$. Let $T_{\tilde{A}_f,v}$ denote the total running time for f consecutive executions of protocol $(A, \tilde{A}_f)$, followed by protocol $(\tilde{A}_f, B)$. The success rate $S_{\tilde{A}_f,v}$ of $\tilde{A}_f$ refers to the coin tosses of $\tilde{A}_f$, A, B in these $f+1$ protocol executions. We first show that Theorem 3 holds in case $m \geq t$ for any active adversary $\tilde{A}_f$.

Theorem 7. *There is a probabilistic algorithm, which given N and an active adversary $\tilde{A}_f$, generates a random public key $v \in_R (\mathbb{Z}_N^{*2^t})^k$, factorizes N with probability at least $1/2$ with respect to its coin tosses, and runs in expected time $O(T_{\tilde{A}_f,v}/S_{\tilde{A}_f,v})$ provided that $S_{\tilde{A}_f,v} \geq 2^{-kt+1}$ and $m \geq t$.*

Proof. The factoring algorithm picks $s_j \in_R \mathbb{Z}_N^*$ for $i = 1, \ldots, k$ and generates the public key v as $1/v_j := s_j^{2^t}$ for $j = 1, \ldots, k$. Using the private key $s = (s_1, \ldots, s_k)$, the algorithm executes the protocol $(A, \tilde{A}_f)$ f-times providing to $\tilde{A}_f$ the information necessary to impersonate A with success rate $S_{\tilde{A}_f,v}$.

A key observation is that the protocol $(A, \tilde{A}_f)$ is witness indistinguishable and witness hiding in the sense of [FS90]. The protocol $(A, \tilde{A}_f)$, executed using the secret key s, does not reveal to $\tilde{A}_f$ any information about which 2^t-root s_j of $1/v_j$ has been used by A. The same distribution of data is given to $\tilde{A}_f$ in protocol $(A, \tilde{A}_f)$, no matter which of the 2^t-roots s_j has been chosen by the factoring algorithm. For this we note that in step 1 of protocol $(A, \tilde{A}_f)$, A sends $x = r^{2^t}$, a random 2^t-power in $\mathbb{Z}_N^{*2^t}$. In step 3, A sends $y = r \cdot \prod_j s_j^{e_j}$, a random 2^t-root of $x/\prod_j v_j^{e_j}$ that is uniformly distributed among all possible 2^t-roots. This uniform distribution is based on the random choice of r and is independent of the 2^t-roots s_j of $1/v_j$.

Using the data transmitted within the f executions of protocol $(A, \tilde{A}_f)$, algorithm AL of Theorem 1 produces an output $(y, y, e, \bar{e})$ so that $Y^{2^t} = Z^{2^{t+\ell}}$ holds for $Y := y/\bar{y}$ and $Z := \prod_j s_j^{(e_j - \bar{e}_j)/2^\ell}$. The distribution of Y does not change if s_j is replaced by another 2^t-root of the same $1/v_j$. This holds even though y, $\bar{y}$ formally depend on s. On the other hand, $Z = \prod_j s_j^{(e_j - \bar{e}_j)/2^\ell}$ changes with the choice of the 2^t-roots s_j. Therefore the factoring method of Theorem 3 remains intact. With probability at least $1/2$, $\{\gcd(Y^{2^i} \pm Z^{2^{i+\ell}}, N)\} = \{p, q\}$ holds for some i with $0 \leq i < t$. □

Secure moduli. In view of Theorem 7, moduli N with $m \geq t$ provide optimal security against active impersonation attacks provided that N is difficult to factor. This raises the question on how the difficulty of factoring a random integer N depends on the parameter m. We are not aware of a factoring algorithm that makes a relevant difference for small values of m, say for $m \leq 10$, the relevant case for Ong-Schnorr ID.

The previous reductions cannot be easily extended to the case of active adversaries if $m < t$. At best, we can combine Lemma 4 with the use of pseudo-keys as in Theorem 5. The factoring method of Theorem 3 requires $\ell < m$ which in

turn necessitates a large success rate: $S_{\tilde{A}_f,v} > 2^{-km}$. Using a pseudo-key $\tilde{s}$, we can factorize N with smaller success rates $S_{\tilde{A}_f,v}$.

Suppose the pseudo-key $\tilde{s}$ satisfies $\tilde{s}_j^{2^{\bar{m}}} = 1/v_j$ for $j = 1,...,k$ with $m \leq \bar{m} \leq t$. Using such a pseudo-key the factoring method works iff $\ell < t+m-\bar{m}$. The drawback is that the factoring algorithm, without secret key, cannot easily simulate the protocol $(A, \tilde{A}_f)$ which is necessary to provide the information needed by the adversary for an active impersonation attack. Following Shoup [Sh96], we can simulate the protocol $(A, \tilde{A}_f)$ in zeroknowledge fashion by guessing the exam e partly. It is sufficient to guess e mod $2^{t-\bar{m}}$ since the $\lfloor 2^{\bar{m}-t}e_j \rfloor$-part of the exam can be answered using the pseudo-key $\tilde{s}$. To guess e mod $2^{t-\bar{m}}$, we need on the average $2^{k(t-\bar{m})}$ many trials. This causes a time factor $2^{k(t-\bar{m})}$ for the factoring algorithm.

Theorem 8 presents a trade-off in case of small m-values. We can either have an additional time factor $2^{k(t-\bar{m})}$ for factoring N, or else a required success rate $S_{\tilde{A}_f,v}$ that is $2^{k(\bar{m}-m)}$ times larger than the success rate required in case $m \geq t$.

Theorem 8. *There is a probabilistic algorithm which, given the active attacker $\tilde{A}_f$, N and $\bar{m}$ with $m \leq \bar{m} \leq t$, generates a random public key $v \in_R (\mathbb{Z}_N^{*2^t})^k$, factorizes N with probability at least $1/8$, with respect to its coin tosses,and runs in expected time $O(2^{k(t-\bar{m})}T_{\tilde{A}_f,v}/S_{\tilde{A}_f,v})$ provided that $S_{\tilde{A}_f,v} \geq 2^{-kt+k(\bar{m}-m)+2}$.*

This theorem contains the result of Shoup [Sh96] that active impersonation attacks can be transformed in polynomial time into the factorization of a Blum integer modulus N. If the success rate $S_{\tilde{A}_f,v}$ is at least $1/(\log(N))^c$ for some constant $c > 0$ and if we have a corresponding a priori lower bound for $S_{\tilde{A}_f,v}$, we apply Theorem 8 with the maximal $\bar{m}$ satisfying $2^{-kt+k(\bar{m}-m)+2} < S_{\tilde{A}_f,v}$. With this $\bar{m}$ the time factor $2^{k(t-\bar{m})}$ is polynomially bounded, and together with a polynomial time adversary $\tilde{A}_f$, the factoring algorithm becomes polynomial time. A priori knowledge of $\tilde{A}$'s success rate is not required since we can simply guess the optimal $\bar{m}$, which increases the factoring time by the small factor $\bar{m}$.

Proof. Factoring algorithm

1. Pick random $\tilde{s}_j \in_R \mathbb{Z}_N^*$, set $1/v_j := \tilde{s}_j^{2^m}$ for $j = 1,\ldots,k$ and $u := 0$
2. Pick a random sequence of coin tosses RA for $\tilde{A}_f$.
   To simulate f executions of $(A, \tilde{A}_f)$ using $\tilde{s}$, repeat steps 2.1, 2.2 f times.
   2.1 Pick $r \in_R \mathbb{Z}_N^*$, $e' = (e'_1,\ldots,e'_k) \in_R [0, 2^{t-\bar{m}})^k$ and set $x := r^{2^t} \prod_j v_j^{e'_j}$.
   2.2 Compute $e \in [0, 2^t)^k$ following $\tilde{A}_f$.
   If $e \neq e'$ mod $2^{t-\bar{m}}$ go back to step 2.1 undoing the computation of $\tilde{A}_f$.
   Otherwise set $y := r \cdot \prod_j \tilde{s}_j^{\lfloor 2^{m-t}e_j \rfloor}$ (An easy calculation shows that $y^{2^t} \prod_j v_j^{e_j} = x$. The f iterations of steps 2.1 and 2.2 provide to the adversary $\tilde{A}_f$ the information, needed for impersonation attacks.)
3. (first impersonation attempt) Pick $e \in_R [0, 2^t)^k$ and execute $(\tilde{A}_f, B)$ with exam e. If $S_{\tilde{A}_f,v}(RA, e) = 1$ set $u := 4u$ and go to step 4.

Otherwise set $u := u + 1$ and go back to step 2 undoing the computation of $\tilde{A}_f$.

4. (second impersonation attempt) Pick $e \in_R [0, 2^t)^k$ and execute $(\tilde{A}_f, B)$ with exam $\bar{e}$. If $S_{\tilde{A}_f,v}(RA, \bar{e}) = 1$ and $e \neq \bar{e}$, compute the replies $y, \bar{y}$ of $\tilde{A}_f$ with e, $\bar{e}$ and go to step 5. Otherwise set $u := u - 1$, if $u > 0$ go back to step 4, if $u = 0$ go back to step 2 (undoing the computation of $\tilde{A}_f$ in either case).
5. Compute $Y := y/\bar{y}$, $\ell := \max\{i \mid e = e \bmod 2^i\}$, $\tilde{Z} := \prod_j \tilde{s}_j^{(e_j - \bar{e}_j)/2^\ell}$ (hence $Y^{2^\ell} = \tilde{Z}^{2^{\bar{m}+\ell}}$).
6. Test whether $\{\gcd(Y^{2^{t-i}} \pm \tilde{Z}^{2^{\bar{m}+\ell-i}}, N)\} = \{p, q\}$ holds for some $i \leq \min(t, \bar{m} + \ell)$.

Analysis. Each evaluation of $S_{\tilde{A}_f,v}(RA, e)$ requires f executions of protocol $(A, \tilde{A}_f)$ followed by an execution of protocol $(\tilde{A}_f, B)$. Here $\tilde{A}_f$ is determined by its sequence of coin tosses RA while A and B follow the protocol (A, B) with independent coin flips.

The steps 2.1 and 2.2 simulate the protocol $(A, \tilde{A}_f)$ in zeroknowledge fashion using the pseudo-key $\tilde{s}$. This is possible by partially guessing the exam e.

Step 3 counts the number u of probed pairs (RA, e) until a successful pair is found. Then step 4 probes at most $4u$ pairs to find a second successful pair $(RA, \bar{e})$ for the same RA. Steps 2, 3, 4 are passed on the average at most $O(1/S_{\tilde{A}_f,v})$ times. This follows from the argument set forth in Lemma 4 of [FFS88].

In step 2.2, the equation $e = e' \bmod 2^{t-\bar{m}}$ holds with probability $2^{-k(t-\bar{m})}$. Guessing a correct e takes on the average $2^{k(t-\bar{m})}$ many trials causing a time factor $2^{k(t-\bar{m})}$. Hence the algorithm runs in expected time $O(2^{k(t-\bar{m})} T_{\tilde{A}_f,v}/S_{\tilde{A}_f,v})$.

By the construction we have $Y^{2^\ell} = \tilde{Z}^{2^{\bar{m}+\ell}}$. Therefore, the factorization attempt in step 6 succeeds with probability $\geq 1/2$ iff there exists i with $\ell + \bar{m} - m < i \leq \min(t, \bar{m} + \ell)$. This condition is satisfiable iff $\ell < t + m - \bar{m}$. By Lemma 4 and since $S_{\tilde{A}_f,v} \geq 2^{-kt+k(\bar{m}-m)+2}$, the inequality $\ell < t + m - \bar{m}$ holds at least with probability $\geq 1/4$. Hence the factoring of N succeeds at least with probability $1/8$.

The required lower bound on $S_{\tilde{A}_f,v}$ is nearly sharp, as the inequality $S_{\tilde{A}_f,v} > 2^{-kt+k(\bar{m}-m)}$ is necessary for the condition $\ell < t + m - \bar{m}$. □

Patent note. No patent has been filed for the Ong-Schnorr scheme. This makes it attractive to use instances of this scheme that do not fall under the FS- or the GQ-patent.

Acknowlwedgement. The author thanks V. Shoup for pointing out an error in a draft version and J.P. Seifert for his comments.

References

[BR93] M. Bellare and P. Rogaway. Random oracle are practical: a paradigma for designing efficient protocols. Proceedings of the 1st ACM Conference on Computer Communication Security, pages 62–73, 1993.

[DGB87] Y. Desmedt, C. Goutier, and S. Bengo. Special uses and abuses of the Fiat-Shamir passport protocol. Proceedings CRYPTO'87, Springer LNCS 293: pages 21-39, 1988.

[FS86] A. Fiat and A. Shamir. How to prove yourself: Practical Solution to Identification and Signature Problems. Proceedings of CRYPTO'86, Springer LNCS 263: pages 186-194, 1986.

[FFS88] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. J. Cryptology, 1: pages 77–94, 1988.

[FS90] U. Feige, A. Shamir. Witness indistinguishable and witness hiding protocols Proceedings 22rd STOC, pages 416–426, 1990.

[FS86] A. Fiat and A. Shamir. How to prove yourself: Practical Solution to Identification and Signature Problems. Proceedings of CRYPTO'86, Springer LNCS 263: pages 186–194, 1986.

[GS94] M. Girault and J. Stern. On the length of cryptographic hash-values used to identification schemes. Proceedings of CRYPTO'94, Springer LNCS 839: pages 202–215.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. SIAM J. Comput., 18: pages 186–208, 1989.

[GMR88] S. Goldwasser, S. Micali and R. Rivest. A digital signature secure against adaptive chosen-message attacks. Siam J. Computing 17: pages 281–308, 1988.

[GQ88] L. Guillou and J. Quisquater. A practical zero-knowledge protocol fitted to security microprocesors minimizing both transmission and memory. Proceedings of Eurocrypt'88, Springer LNCS 330: pages 123–128, 1988.

[M94] S. Micali. A secure and efficient digital signature algorithm. Technical Report, MIT/LCS/TM-501, 1994

[MS88] S. Micali and A. Shamir. An improvement of the Fiat-Shamir Identification Scheme. Proceedings CRYPTO'88, Springer LNCS 403: pages 244–247, 1990.

[O92] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. Proceedings of CRYPTO'92, Springer LNCS 740: pages 31–53, 1992.

[OS90] H. Ong and C.P. Schnorr. Fast signature generation with a Fiat Shamir-like scheme. Proceedings of Eurocrypt'90, Springer LNCS 473: pages 432–440, 1990.

[PS96] D. Pointcheval and J. Stern. Security proofs for signatures. Proceedings Eurocrypt'96, Springer LNCS 1070: pages 387–398, 1996.

[Sch91] C.P. Schnorr. Efficient signature generation by smart cards. J. Cryptology, 4 pages 161–174, 1991.

[Sh96] V. Shoup. On the security of a practical identification scheme. Proceedings of Eurocrypt'96, Springer LNCS 1070: pages 340–353, 1996.

Erratum

C.P. Schnorr: Security of 2^t–Root Identification and Signatures, Proceedings CRYPTO'96, Springer LNCS 1109, (1996), pp. 143–156
page 148, section 3, line 5 of the proof of Theorem 3.
Correction. The proposed factoring method

Check whether $\{\gcd(Y^{2^i} \pm Z^{2^{i+\ell}}, N)\} = \{p, q\}$ holds for some i with $0 \le i < t$

fails if $Y^{2^i} = -Z^{2^{i+\ell}}$ holds for some i with $0 \le i < t$, otherwise it factors N with probability $\frac{1}{2}$. In the first case continue the factoring algorithm as follows until it factors N with probability $\frac{1}{2}$:
Supplemental steps to the factoring algorithm. Repeat the entire algorithm using independent coin flips and construct independent pairs (Y, Z) with $Y^{2^t} = Z^{2^{t+\ell}} \bmod N$ until either of the following two cases arises.

Case I. $Y^{2^i} \neq -Z^{2^{i+\ell}}$ for all i with $0 \le i < t$ holds for some (Y, Z). Then terminate as the proposed factoring method succeeds using Y, Z with probability $\frac{1}{2}$.

Case II. $Y^{2^i} = -Z^{2^{i+\ell}}$ holds for two independent pairs $(Y, Z), (Y', Z')$. Then replace these pairs by $(Y_{\text{new}}, Z_{\text{new}})$ with $Y_{\text{new}} := YY'$, $Z_{\text{new}} := ZZ'$. If $Y_{\text{new}}^{2^{i_{\text{new}}}} = -Z_{\text{new}}^{2^{i_{\text{new}}+\ell}}$ holds for some i_{new} then we have $i_{\text{new}} < i$, otherwise terminate (as the proposed factoring method succeeds using $Y_{\text{new}}, Z_{\text{new}}$ with probability $\frac{1}{2}$).
Continue the repetitions of the entire algorithm using idependent coin flips and continue to decrease i until the algorithm either terminates in Case I or enters Case II with $i = 1$. In the latter case the proposed factoring method succeeds using $Y_{\text{new}}, Z_{\text{new}}$ with probability $\frac{1}{2}$, in particular $\{\gcd(Y_{\text{new}} \pm Z_{\text{new}}, N)\} = \{p, q\}$ holds with probability $\frac{1}{2}$.

With the supplemental steps the algorithm factorizes N with probability $\frac{1}{2}$. The supplemental steps increase the time bound for factoring by a factor $O(\ell)$. The correctness proof of the amended factoring method uses the following observation

We see from $Y^{2^t} = Z^{2^{t+\ell}} \bmod N$ that Z^{2^ℓ}/Y is a 2^t-root of $1 \bmod N$. This root is not necessarily uniformly distributed over all 2^t-roots of $1 \bmod N$. But it is uniformly distributed within certain cosets.

Fact. Let $Y = Y(Z^{2^t})$ be a function of Z^{2^t} that solves $Y^{2^t} = Z^{2^{t+\ell}} \bmod N$ with $\ell < t$. Then Z^{2^ℓ}/Y takes the roots in $c_0 R_N(2^t)^{2^\ell}$ with equal probability for all $c_0 \in R_N(2^t)$, where $R_N(2^t)$ denotes the group of 2^t-roots of $1 \bmod N$ and $R_N(2^t)^{2^\ell} \subset R_N(2^t)$ denotes the subgroup of 2^ℓ-powers.

All subsequent factoring algorithms in the paper have to be amended in the same way.