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

March 6, 1996

Abstract

The GQ-protocol of Guillou and Quisquater and Ong-Schnorr identification and signatures are variants of the Fiat-Shamir scheme that provide short and fast communication and signatures. Let $N = pq$ be an arbitrary product of two primes that is difficult to factor. The Ong-Schnorr scheme uses secret keys that are $2^t$-roots modulo $N$ of the public keys, whereas Fiat-Shamir use square roots modulo $N$. The Ong-Schnorr scheme is quite efficient, in particular in its multi-key version. Under the assumption that the module $N$ is a Blum integer that is difficult to factor, security for the Ong-Schnorr scheme has recently been proved for particular cases. Micali proves security of the signature scheme for particular keys and modules $N$. Shoup proves that the identification scheme is secure against active adversaries.

We prove for arbitrary modules $N = pq$ that Ong-Schnorr identification and signatures are secure unless $N$ can easily be factored. The proven security of Ong-Schnorr identification against active impersonation attacks depends in an interesting way on the maximal $2$-power $2^m$ that divides either $p-1$ or $q-1$. For $m \geq t$ we give a reduction from factoring $N$ to active impersonation attacks that is as efficient as the one known for Fiat-Shamir identification. For $m < t$ we give an equally efficient reduction from factoring $N$ to passive impersonation attacks and a less efficient reduction to active impersonation attacks. As these security results depend on the parameter $m$ the question arises on how the difficulty of factoring $N$ depends on $m$.

We show that Ong-Schnorr signatures with arbitrary module $N$ are secure against adaptive chosen-message attacks unless the module $N$ can easily be factored. Unlike to the security of identification against active adversaries, the parameter $m$ is irrelevant for the security of the signature scheme in the random oracle model.

Keywords

identification scheme, signature scheme, Fiat-Shamir scheme, active/passive impersonation attacks, adaptive chosen-message attack, random oracle model.
1 Introduction and Summary

Fiat and Shamir proposed a practical identification/signature scheme that is based on a zero-knowledge protocol of Goldwasser, Micali and Rackoff (1989) for proving quadratic residuosity. The GQ-protocol of Guillou and Quisquer and Ong-Schnorr identification and signatures are variants of the Fiat-Shamir scheme which provide shorter communication and signatures than the Fiat-Shamir scheme. Ong-Schnorr identification and signatures are direct extensions of the Fiat-Shamir scheme replacing square roots modulo $N$ by $2^t$-roots. Moreover Ong-Schnorr identification and signatures are as fast, in the number of modular multiplications, as Fiat-Shamir. 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 $N$ ($N = pq$ is called a Blum integer if $p$, $q$ are primes that are congruent 3 mod 4).

Previous results. Micali [M94] proves security of Ong-Schnorr signatures for the case that the secret key is a $2^t$-root of 4 and that 2 is a quadratic non-residue modulo $N$. Micali assumes that the hash function used for signatures acts as a random oracle. He shows that any algorithm which produces, without secret key, a valid signature faster than by random trials immediately leads to the factorization on $N$. This surprising result requires that the secret key, the $2^t$-root of 4, already reveals the prime factors $p$ and $q$ of $N$. Therefore distinct users must have different modules $N$, and $N$ is part of the secret key rather than a public parameter as in the Fiat-Shamir scheme and its extension by Ong-Schnorr.

Shoup [Sh95] proves that Ong-Schnorr identification with Blum integers $N$ is secure against active adversaries unless $N$ is easy to factor. Shoup gives a reduction from factoring $N$ to active impersonation attacks that is less efficient than the one known for the Fiat-Shamir scheme. Also his reduction is not entirely constructive as it requires apriori knowledge on the adversary’s probability of success.

Our results. We present security proofs for Ong-Schnorr identification for arbitrary modules $N = pq$. This extends and improves the results of Shoup in various ways. It sheds new light on the prime factors $p$ and $q$ of the module $N$. The efficiency of our reduction from factoring $N = pq$ to impersonation attacks depends in an interesting way on the maximal 2-power $2^m$ that divides either $p - 1$ or $q - 1$. We distinguish the cases of active and of passive attacks. In an active attack, before the impersonation attempt, the adversary poses 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 executions of the ID-protocol.

The cases that $m \geq t$, respectively $m < t$, are quite different. For $m \geq t$ we present a reduction from factoring $N$ to active impersonation that is as efficient as the one known for Fiat-Shamir ID. It only requires that the adversary’s success rate is twice the success rate for guessing the exam posed by the verifier. Thus modules $N$ with $m \geq t$ provide optimal security against active/passive impersonation attacks unless they can easily be factored.

For the case $m < t$ we give a reduction from factoring to (only) passive impersonation that is as efficient as the one known for Fiat-Shamir ID. The reduction works for public
keys that are generated together with a pseudo-key (independent from the secret key) which enables to transform successful passive impersonations into the factorization of \( N \). Having only a pseudo-key complicates for small \( m \) the reduction from factoring to active impersonation attacks as it becomes difficult to simulate the ID-protocol which is necessary to provide the information which the adversary needs for an active impersonation attack. This leads to a trade-off which we describe in Theorem 8. We either have an additional time factor \( 2^{t-m} \) for factoring \( N \) or the required probability of success of the active adversary increases by the factor \( 2^{t-m} \).

**Security of signatures.** The above 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. This random oracle assumption has already been used in [FS86] and is commonly accepted to be appropriate for hash functions without cryptographic weaknesses, see also [BR93]. We consider the strongest type of attacks, adaptive chosen-message attacks. Here the adversary, before attempting to generate a valid signature-message pair, uses the legitimate signer as oracle to sign messages of his choice.

Pointcheval and Stern [PS96] show how to transform security proofs for discrete logarithm identification schemes into security proofs for the corresponding signature scheme. Using similar arguments we transform security against passive attacks for Ong-Schnorr ID into security against adaptive chosen-message attacks for the corresponding signature scheme. In Theorem 6 we prove the following. Ong-Schnorr signatures cannot be produced by an adaptive chosen-message attack faster than by random trials unless the module \( N \) can easily be factored. We get the same result for arbitrary keys and modules \( N \) which Micali [M94] proves for particular keys and modules \( N \).

**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 tools is a generalization of this property for arbitrary \( N \), following Lemma 2.

## 2 Ong-Schnorr identification

Let \( N \) be product of two large primes \( p,q \). Assume that \( N \) is public but the factorization is completely unknown. Let \( \mathbb{Z}_N^* \) denote the multiplicative group of integers modulo \( N \). Let the prover \( A \) have the private key \( s = (s_1,\ldots,s_k) \) with components \( s_1,\ldots,s_k \in \mathbb{Z}_N^* \). The corresponding public key \( v = (v_1,\ldots,v_k) \) has components \( v_j \) satisfying \( 1/v_j = s_j^2 \) for \( j = 1,\ldots,k \). We assume that the verifier \( B \) has access to \( A \)'s public key \( v \).

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

1. \( A \) picks a random \( r \in_R \mathbb{Z}_N^* \) and sends \( x := r^{3t} \) 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 $x := r^2 \prod v_j^{e_j}, \ y := r$. The probability of success is $2^{-kt}$. The 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 $k, t$ with $kt$ about 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 $r \in \mathbb{Z}_N^*$ forms $x := r^d$ and computes the hash value $e := h(x, M)$ in $[0, 2^k)^k$ and $y := r \prod v_j^{e_j}$. The signature of the message $M$ is the pair $(e, y)$. It is verified by checking that $h(y^{2t} \prod v_j^{e_j}, M) = e$ holds.

Efficiency. For Ong-Schnorr ID (resp. signatures) both prover (resp. signer) $A$ and 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 the same way as for the Fiat-Shamir scheme [FS86]. If the public key components $v_j$ are integers having only a few non-zero bits in their binary representation, the work load of the verifier reduces to only $t$ squarings in $\mathbb{Z}_N^*$ and a few additions, shifts and reductions modulo $N$ with integers of the order $N$. If $v_j = \sum v_{j,i} 2^{i}$ has $w_j$ 1-bits $v_{j,i} = 1$, a multiplication by $v_j$ can be done by $w_j$ additions, shifts and reductions modulo $N$. Thus the verifier needs only to perform $t$ squarings, for computing $y^{2t}$, and on the average $\frac{1}{t} \sum w_j$ additions, shifts and reductions modulo $N$. Moreover the reductions modulo $N$ are needless if the $v_j$ are small integers.

Previous protocols. The original Fiat-Shamir scheme is the case $t = 1$ of the Ong-Schnorr protocol, repeated several times. While the Fiat-Shamir scheme requires many rounds to become secure, the Ong-Schnorr scheme executes a single round. Fiat-Shamir ID is secure against passive and active attacks unless $N$ can easily be factored. Moreover Fiat-Shamir signatures are secure in the random oracle model [FS86], [FFS88]. Attacks with a success rate that is twice the probability for guessing the examine $e$ can be transformed into the factorization of $N$.

The GQ-protocol [GQ88] is the case of single component keys $k = 1$, where $2^t$-powers $x = r^2t$ are replaced by $u$-powers $x = r^u$ for an arbitrary integer $u$ of order $N$. The GQ-protocol consists of a single round with a large exam $e$. This greatly reduces the length of transmission and signatures of 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 $A$. Define the success bit $S_{A,v}(RA, e)$ to be 1 if $A$ succeeds with $v, RA, e, N$ and 0 otherwise; accordingly call the pair $(RA, e)$ successful/unsuccessful. The success rate $S_{A,v}$ of $A$ with $v$ is the expected value of $S_{A,v}(RA, e)$ for uniformly distributed pairs $(RA, e)$. For simplicity, we assume that the time $T_{A,v}(RA, e)$ of $A$ 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 for successful pairs \((RA,e)\) decreases the success rate \(S_{\tilde{A},v}\) at most by a factor 2. We assume that \(T_{\tilde{A},v} = \Omega(k \cdot t(\log_2 N)^3)\) and thus \(T_{\tilde{A},v}\) majorizes the time of \(B\) in the protocol \((\tilde{A}, B)\).

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

The proof is a straightforward extension of Lemma 4 in Feige, Fiat, Shamir (1988). Algorithm \(AL\) constructs a random pair \((RA,e)\) with \(S_{\tilde{A},v}(RA,e) = 1\) and produces a second random exam \(\tilde{e}\) for which \(\tilde{A}_{f}\) succeeds with the same \(RA\), i.e. \(e \neq \tilde{e}\) and \(S_{\tilde{A},v}(RA, \tilde{e}) = 1\). \(AL\) outputs \(e, \tilde{e}\) and the replies \(y, \tilde{y}\) of \(\tilde{A}\) with coin tosses \(RA\) to the exams \(e, \tilde{e}\).

For the entities of Theorem 1 we denote \(X := y/\tilde{y}\), \(\ell := \max\{i \mid e = \tilde{e} \mod 2^i\}\), \(Z := \Pi_j v_j^{(2^\ell-\varepsilon_j)/2^t}\). By the construction we have \(X^{2^t} = Z^{2^t+\ell}\). The goal is to derive from \(X, Z\) two statistically independent square roots of the same square modulo \(N\), so that we can factorize \(N\) with prob. \(\geq 1/2\).

We use the structure of the prime factors \(p, q\) of \(N = p \cdot q\). 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 \(G^n\) denote the subgroup of \(u\)-powers in \(G\), \(G^n = \{g^n \mid g \in G\}\). Lemma 2 is obvious.

**Lemma 2.** For any cyclic group \(G\) of order \(|G| = 2^m m'\) with \(m'\) odd, squaring \(SQ : G^{2^t} \rightarrow G^{2^{t+1}}, x \mapsto x^2\) 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 \(SQ : \mathbb{Z}_N^{2^t} \rightarrow \mathbb{Z}_N^{2^{t+1}}, x \mapsto x^2\) 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 modules \(N\).

### 3 Passive impersonation attacks for \(m \geq t\)

We show that Ong-Schnorr ID in case \(m \geq t\) is 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 \(v \in \mathbb{Z}_N^{2^t}\), 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 \( 1/v_j := s_j^{2^j} \) for \( j = 1, \ldots, k \), runs algorithm \( AL \) of Theorem 1 on input \( A, N, v \) to produce \( (y, \widehat{y}, e, \widehat{e}) \) and computes the corresponding \( \ell, X, Z \) with \( X^{2^\ell} = Z^{2 + \ell} \). Then, it checks whether

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

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 arbitrarily fix \( X, (Z \bmod p), s_2 \bmod q, \ldots, s_k \bmod q \) so that the probability space reduces to \( s_1 \bmod q \in R \mathbb{Z}_q^* \). By Lemma 2 and since \( t \leq m \) there are \( 2^t \) many \( 2^t \)-roots \( s_1 \bmod q \) of \( 1/v_1 = s_1^{2^t} \bmod q \). They yield \( 2^t \) many values \( Z \bmod q \). Since \( \ell < t \leq m \) we have \( X \neq \pm Z^{2^t} \) for at least half of these \( 2^t \) cases. If \( X \neq \pm Z^{2^t} \) take the largest \( i < t \) with \( X^{2^i} \neq \pm Z^{2 + i} \). Then \( X^{2^i}, Z^{2 + i} \) are square roots of the same square modulo \( N \), they are distinct even when changing the sign. Hence \( \{\gcd(X^{2^i} \pm Z^{2 + i}, N)\} = \{p, q\} \). This shows that the algorithm factorizes \( N \) with probability at least \( 1/2 \).

The expected time of the factoring algorithm is that of algorithm \( AL \). By the assumption \( T_{\hat{A}, v} = \Omega(k \cdot t(\log_2 N)^3) \) this covers all other steps. \( \square \)

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

**Lemma 4.** For any \( m' \) with \( 1 \leq m' \leq t \) algorithm \( AL \) of Theorem 1 produces on input \( \hat{A}, v \) an output \( (y, \widehat{y}, e, \widehat{e}) \) so that \( e \neq \widehat{e} \bmod 2^m' \) holds with probability \( \geq 1/4 \) provided that \( S_{\hat{A}, v} \geq 2^{-k m' + 2} \).

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

**Proof.** We call a coin tossing sequence \( RA \) of \( \hat{A} \) \( m' \)-heavy if \( \sum_e S_{\hat{A}, v}(RA, e) \geq 2^{k m' + 1} \), i.e., if \( \hat{A} \) succeeds for at least a \( 2^{-k m' + 1} \) fraction of the \( e \). The claim follows from facts A and B.

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

**Proof.** For every \( e \) we have \( \# \{ e \mid e = \bar{e} \bmod 2^m' \} \leq 2^{k t - k m'} \) since \( e_i = \bar{e}_i \bmod 2^m' \) holds for at most a \( 2^{-m'} \) fraction of the \( \bar{e}_i \). Now the fact follows since \( RA \) is \( m' \)-heavy.

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

**Proof.** If \( RA \) is not \( m' \)-heavy at most a \( 2^{-k m' + 1} \) fraction of the \( e \) satisfy \( S_{\hat{A}, v}(RA, e) = 1 \). Therefore at most a \( 2^{-k m' + 1} \) fraction of pairs \( (RA, e) \) satisfy \( S_{\hat{A}, v}(RA, e) = 1 \) without that \( RA \) is \( m' \)-heavy. On the other hand, since \( S_{\hat{A}, v} \geq 2^{-k m' + 2} \), at least a \( 2^{-k m' + 2} \) fraction of the \( (RA, e) \) satisfy \( S_{\hat{A}, v}(RA, e) = 1 \).
Algorithm $AL$ generates a random pair $(RA, e)$ with $S_{A,v}(RA, e) = 1$. By Fact A $RA$ is $m'$-heavy with probability $≥ 1/2$. After fixing $(RA, e)$ with $S_{A,v}(RA, e) = 1$ $AL$ generates a random $\bar{e}$ with $S_{\bar{A},v}(RA, \bar{e}) = 1$. By Fact B $e ≠ \bar{e}$ mod $2^{m'}$ holds with probability $≥ 1/4$. □

Remark. The lower bound $S_{\bar{A},v} > 2^{-km'}$ is necessary in Lemma 4. It is possible to position a $2^{-km'}$-fraction of successes so that $e = \bar{e}$ mod $2^{m'}$ 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 $\bar{s}$ which enables to transform successful attacks of a passive adversary $A$ into the factorization of $N$.

Theorem 5 There is a prob. algorithm which on input $A,N$ generates a random public key $v ∈ R\left(Z_N^{*2^k}\right)$, factorizes $N$ with probability $≥ 1/2$ with respect to its coin losses, and runs in expected time $O(T_{\bar{A},v}/S_{\bar{A},v})$ provided that $S_{\bar{A},v} ≥ 2^{-k+i+1}$ and $m < t$.

Proof. Factoring algorithm

1. Pick random $\bar{s}_j ∈ R\left(Z_N^{*2^k}\right)$ and set $1/v_j = \bar{s}_j^{2^m}$ for $j = 1, \ldots, k$ (we have $v ∈ R\left(Z_N^{*2^k}\right)$).
2. According to Theorem 1 compute $AL : (\bar{A}, v) ↦ (y, \bar{y}, e, \bar{e})$ and set

   $\ell := \max\{i \mid e = \bar{e} \mod 2^i\}$, $X := y/\bar{y}$, $\bar{Z} := \Xi \bar{s}_j^{(e_j - \bar{e}_j)/2^i}$.
3. Test whether for some $i$, $\ell < i ≤ t$: $\{\gcd(X^{2^{i-1}} ± \bar{Z}^{2^{t+m-i}}, N)\} = \{p, q\}$.

By the construction we have $X^{2^{i}} = \bar{Z}^{2^{t+m}}$ and $\ell < t$. W.l.o.g. let $(e_1 - \bar{e}_1)/2^\ell$ be odd. Arbitrarily fix $\bar{Z}(\mod p)$, $\bar{s}_2(\mod q), \ldots, \bar{s}_k \mod q$ and $X$ so that the probability space reduces to the $2^m$ solutions $\bar{s}_1(\mod q)$ of $\bar{s}_1^{2^m} = 1/v_1 \mod q$. These $2^m$ solutions yield $2^m$ many values $\bar{s}_1 ∈ Z_N^{*}$ and, since $(e_1 - \bar{e}_1)/2^\ell$ is odd, they generate $2^m$ many values $\bar{Z} ∈ Z_N^{*}$. Note that $X^{2^{i-1}} ± \bar{Z}^{2^{t+m-i}}$ holds for at least $2^{m-1}$ many $\bar{Z}$-values. (By Lemma 2 and since $\bar{Z}(\mod p)$ is fixed we have $X^{2^{i-1}} = ± \bar{Z}^{2^{m-1}}$ for at most $2^{m-1}$ of these $\bar{Z}$-values.) For such $\bar{Z}$ consider the smallest $i > 0$ with $X^{2^{i-1}} ± \bar{Z}^{2^{t+m-i}}$. Then $X^{2^{i-1}}$, $\bar{Z}^{2^{t+m-i}}$ are square roots of the same square in $Z_N^*$. These square roots are distinct even if we change signs. Hence $\gcd(X^{2^{i-1}} ± \bar{Z}^{2^{t+m-i}}, N) = \{p, q\}$. 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 $\bar{s}$ so that $1/v_j = \bar{s}_j^{2^m}$ for $j = 1, \ldots, k$. We cannot generate first a secret key $s$ to produce a pseudo-key $\bar{s}$ by squaring the components of $s$. The components $\bar{s}_j$ must be random in $Z_N^{*}$, and thus $\bar{s}_j$ is a quadratic non-residue with probability $3/4$. In fact we cannot have $v, s$ together with $\bar{s}$ unless we can easily factor $N$.
5 Security of Ong-Schnorr signatures

We study the security in the random oracle model where the hash function $h$ is replaced by a random oracle. This assumption has already been made in [FS86] and has been further developed in [BR93]. Under this assumption the hash function $h$ produces for each query $(x, M)$ a random value $h(x, M) \in_R [0, 2^r]^k$. If a query is repeated the same answer is given.

We consider most powerful attacks, adaptive chosen-message attacks as introduced by Goldwasser, Micali, Rivest in [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 $[0, 2^r]^k$ and independent for distinct pairs $(x, M)$. The adversary cannot get anything from correct signatures $(e, y)$ since these are random pairs in $[0, 2^r]^k \times \mathbb{Z}_N^*$ that can easily be produced, with the same probability distribution, by anybody. In the random oracle model, adaptive chosen-message attacks on Ong-Schnorr signatures 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 executes an adaptive chosen-message attack on $N$ and public key $v$ so that the oracle for the hash function $h$ is queried at most $f$ times, $f \geq 1$. Let $T_{\tilde{A}_f,v}$ be its expected time and $S_{\tilde{A}_f,v}$ its probability of success with $v$.

**Theorem 6.** There is a probabilistic algorithm which on input $\tilde{A}_f, N$ generates a random $v \in_R ([\mathbb{Z}_N^*]^k)^k$, 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 $S_{\tilde{A}_f,v} \geq f 2^{-k^{1+\epsilon}}$.

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

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

We can assume that $(y^2 \Pi_j v_j^{\epsilon_i}, M) = (x_i, M_i)$ holds for some $i \leq f$ since otherwise $e = h(y^2 \Pi_j v_j^{\epsilon_j}, M)$ holds with prob. $2^{-kt}$. If the adversary produces this $x_i$ as $x_i := y^2 \Pi_j v_j^{\epsilon_j}$ for some preselected $e$ and $y$, the oracle returns the preselected $e$ with prob. $2^{-kt}$. Thus, each such oracle query can at most add $2^{-kt}$ to the success rate $S_{\tilde{A}_f,v}$. Hence, at least with probability $(S_{\tilde{A}_f,v} - f 2^{-kt})$ the attacker $\tilde{A}_f$ is able to produce two distinct pairs $(e, y)$ and $(\bar{e}, \bar{y})$ with $e \neq \bar{e}$ satisfying $y^2 \Pi_j v_j^{\epsilon_j} = \bar{y}^2 \Pi_j v_j^{\bar{\epsilon}_j} = x_i$. For these pairs we have $(y/\bar{y})^2 = \Pi_j v_j^{\epsilon_j - \bar{\epsilon}_j}$ 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 prob. 1/2 as described in Theorem 3.

The formal factoring algorithm employs a version of algorithm AL of Theorem 1 to construct $(e, y, \bar{e}, \bar{y})$. It simulates $\tilde{A}_f$ using statistically independent oracles for $h$. 

7
Factoring algorithm

1. Pick random $s_j \in_R \mathbb{Z}_k^*$, 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 $A_f$.
3. (first signing attempt) Simulate the adversary $A_f$ with $v, RA$.
   For the message signature pairs requested by $A_f$ provide random pairs.
   Let the adversary query the oracle for $h$ about $(x_i, M_i)$ for $i = 1, \ldots, f$.
   If $A_f$ fabricates a signature $(e, y)$ satisfying $y^{2^t} \Pi_j v_j^{e_j} = x_i$ for some $i$ (in this case we call the pair $(RA, e)$ successful with $i$) 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 $A_f$'s computation.
4. (second signing attempt) Simulate the adversary $A_f$ 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.
   If $A_f$ fabricates a signature $(\bar{e}, \bar{y})$ with $e \neq \bar{e}$ satisfying $\bar{y}^{2^t} \Pi_j v_j^{\bar{e}_j} = x_i$ for the $x_i$ fixed in step 3 and the new oracle reply $\bar{e}$ for $(x_i, M_i)$, then go to step 5.
   Otherwise, if $u > 0$ set $u := u - 1$ and go back to step 4,
   if $u = 0$ go back to step 2 (undoing the computation of $A_f$ in either case).
5. Compute $X := y/\bar{y}$, $\ell := \max\{i \mid e = \bar{e} \mod 2^i\}$, $Z := \Pi_j u_j^{2^t}$
   (hence $X^{2^t} = Z^{2^{t+i}}$).
6. Test whether $\{\gcd(X^{2^t-i} \pm Z^{2^{t+i-i}}, N) = \{p, q\}$ holds for some $i \leq t$.

Sketch of the analysis. On the average it takes $1/S_{\bar{A}_f,v}$ many passes of steps 2 and 3 to find $i, x_i, M_i, e, y$. If $S_{\bar{A}_f,v} > f \cdot 2^{-kt+1}$ the subsequent step 4 succeeds to find $(\bar{e}, \bar{y})$ with $e \neq \bar{e}$ at least with probability $\frac{1}{4}(1 - 2.7^{-1})$. For this we note that, with probability at least $\frac{1}{4}$, step 3 probes at least $u \geq \frac{1}{2^t} S_{\bar{A}_f,v}$ many pairs $(RA, e)$ before fixing some $RA$ for which the fraction of successful pairs $(RA, \bar{e})$ is at least $\frac{1}{2} S_{\bar{A}_f,v}^{-1}$. In this case at least a $\frac{1}{2^t} S_{\bar{A}_f,v}^{-1}$-fraction of $\bar{e}$ succeeds in step 4 with the $i$ fixed in step 3. Since step 4 probes at least $2fS_{\bar{A}_f,v}$ many random $\bar{e}$, step 4 succeeds at least with probability $1 - 2.7^{-1}$. 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 $\delta$ and factorizes $N$ according to Theorem 5. \qed

6 Ong-Schnorr ID is secure against active impersonation

In Theorem 7 we extend the reduction of Theorem 3 from passive to active impersonation attacks. In Theorem 8 we present a reduction from factoring to active impersonation attacks for arbitrary modules $N = p \cdot q$ with $m \leq t$. 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 this reduction is quite efficient if \( m \) is close to \( t \) it is less efficient for Blum integers, i.e. for \( m = 1 \). 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 on the probability of success of the adversary \( \hat{A}_f \), given the knowledge from the \( f \) executions of the protocol \((A, \tilde{A}_f)\). We eliminate this a priori knowledge. We only use \( \hat{A}_f \)'s overall success rate \( S_{\hat{A}_f,v} \) depending on the coin tosses of the entire sequence of \( f \) executions of protocol \((A, \tilde{A}_f)\) followed by \((\hat{A}_f, B)\).

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

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

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

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

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

9
Secure modules. In view of Theorem 7, modules $N$ with $m \geq t$ provide optimal security against active impersonation attacks unless $N$ can easily be factored. 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 $m \leq 10$, which are most interesting for Ong-Schnorr ID.

The previous reductions cannot be easily extended to the case of active adversaries if $m < t$. The best we can do is to 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 probability of success, $S_{A_f,v} > 2^{-km}$. Using a pseudo-key $\tilde{s}$ we can factorize $N$ with smaller success rates.

Suppose the pseudo-key $\tilde{s}$ satisfies $s_j^{2m'} = 1/v_j$ for $j = 1, \ldots, k$ with $m \leq m' \leq t$. Using such a pseudo-key the factoring method works iff $\ell < t + m - m'$. The drawback is that the factoring algorithm, without secret key, cannot easily simulate the protocol $(A, A_f)$ which is necessary to provide the information which the adversary needs for an active impersonation attempt. Following Shoup [Sh95] we can simulate the protocol $(A, A_f)$ in zero knowledge fashion by guessing the exams $e$ partly. It is sufficient to guess $e$ mod $2^{t-m'}$ since the $[2^{m'-e_j}]$-part of the exam can be answered using the pseudo-key $\tilde{s}$. To guess $e$ mod $2^{t-m'}$ we need on the average $2^{k(t-m')}$ many trials. This causes a time factor $2^{k(t-m')}$ for the factoring algorithm.

Thus we have a trade-off in case of small $m$-values. We can have an additional time factor $2^{k(t-m')}$ for factoring $N$ or a required success rate $S_{A_f,v}$ that is $2^{k(m'-m)}$ times larger than the success rate required in case $m \geq t$. The trade-off is expressed in the following theorem:

**Theorem 8.** There is a probabilistic algorithm which on input $\hat{A}_f, N$, $m'$ with $m \leq m' \leq t$ generates a random public key $v \in R (\mathcal{Z}_N^{2m'})^k$, factorizes $N$ with probability at least $1/8$ with respect to its coin tosses and runs in expected time $O(2^{k(t-m')})_{A_f,v}/S_{A_f,v}$ provided that $S_{A_f,v} \geq 2^{-k/2+k(2^{t-m})}$.

For Blum integers this theorem contains the result of Shoup [Sh95] that factoring $N$ is polynomial time reducible to active impersonation attacks. If the success rate $S_{A_f,v}$ is at least $1/(\log(N))^c$ for some fixed $c > 0$ and we have a corresponding a priori lower bound for $S_{A_f,v}$ we apply Theorem 8 with the maximal $m'$ satisfying $2^{-k/2+k(2^{t-m})+2} < S_{A_f,v}$. With this $m'$ the time factor $2^{k(t-m')}$ is polynomially bounded and together with a polynomial time adversary $A_f$ the factoring algorithm becomes polynomial time.

**Proof.** Factoring algorithm

1. Pick random $\tilde{s}_j \in_R \mathcal{Z}_N^*$, set $1/v_j = \tilde{s}_j^{2m'}$ for $j = 1, \ldots, k$ and $u := 0$
2. Pick a random sequence of coin tosses $RA$ for $\hat{A}_f$.

To simulate $f$ executions of $(A, \hat{A}_f)$ using $\tilde{s}$, repeat steps 2.1, 2.2 $f$ times.

2.1 Pick $r \in R \mathcal{Z}_N^*$, $e' = (e'_1, \ldots, e'_k) \in R [0, 2^{t-m'})^k$ and set $x := r^{2^f} \Pi_j v_j^{e'_j}$. 

10
2.2 Compute $e \in [0, 2^t)^k$ following $\tilde{A}_f$.

If $e \neq e'$ mod $2^{t-m'}$, go back to step 2.1 undoing the computation of $\tilde{A}_f$.

Otherwise set $y := r \cdot \Pi_j s_j^{[m'-e_j]}$ (an easy calculation shows that $y^{2t} \Pi j v_j^{e_j} = x$).

(By the $f$ iterations of steps 2.1 and 2.2 the adversary $\tilde{A}_f$ gets the necessary information for impersonation attempts.)

3. (first impersonation attempt) Pick $e \in [0, 2^t)^k$ 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 $\overline{e} \in [0, 2^t)^k$ and execute $(\tilde{A}_f, B)$ with exam $\overline{e}$.

If $S_{\tilde{A}_f,v}(RA, \overline{e}) = 1$ and $e \neq \overline{e}$, compute the replies $y, \overline{y}$ of $\tilde{A}_f$ with $e, \overline{e}$ 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 $X := y/\overline{y}$, $\ell := \max\{i \mid e = \overline{e} \mod 2^i\}$, $\tilde{Z} := \Pi_j s_j^{[e_j-\overline{e}_j]} / 2^{t}$

(hence $X^{2^t} = \tilde{Z}^{2^{m'+t}}$).

6. Test whether $\{\gcd(X^{2^t-i} \pm \tilde{Z}^{2^{m'+t-i}}, N) = \{p, q\}$ holds for some $i \leq \min(t, m' + \ell)$.

**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 $(A_f, B)$. Here $\tilde{A}_f$ is determined by its 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 zero knowledge fashion using the pseudo-key $\tilde{s}$. This is possible by partially guessing the exams $e$.

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 $(RA, \overline{e})$ for the same $RA$. This way 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 by Feige, Fiat, Shamir in Lemma 4 of [FFS88].

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

By the construction we have $X^{2^t} = \tilde{Z}^{2^{m'+t}}$. Therefore the factorization attempt in step 6 succeeds with probability $\geq 1/2$ iff there exists $i$ with $\ell + m' - m < i \leq \min(t, m' + \ell)$. This condition is satisfied iff $\ell < t + m - m'$. By Lemma 4 and since $S_{\tilde{A}_f,v} \geq 2^{-k(t+2(m'-m)+2}$ the inequality $\ell < t + m - m'$ 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 $S_{\tilde{A}_f,v} > 2^{-kt+k(m'-m)}$ is necessary for the condition $\ell < t + m - m'$.

**Acknowledgement.** The author thanks V. Shoup for pointing out an error in a draft version and J.P. Seifert for his comment.
References


12