The Representation Problem Based on Factoring

Marc Fischlin and Roger Fischlin

Fachbereich Mathematik (AG 7.2)
Johann Wolfgang Goethe-Universität Frankfurt am Main
Postfach 111932
60054 Frankfurt/Main, Germany

{marc,fischlin}@mi.informatik.uni-frankfurt.de
http://www.mi.informatik.uni-frankfurt.de/

February 14, 2002

Abstract. We review the representation problem based on factoring and show that this problem gives rise to alternative solutions to a lot of cryptographic protocols in the literature. And, while the solutions so far usually either rely on the RSA problem or the intractability of factoring integers of a special form (e.g., Blum integers), the solutions here work with the most general factoring assumption. Protocols we discuss include identification schemes secure against parallel attacks, secure signatures, blind signatures and (non-malleable) commitments.

Keywords. Blind Signature, Commitment, Factoring, Identification, Non-Malleability, Representation Problem, Signature.

1 Introduction

The RSA representation problem deals with the problem of finding a decomposition of a value into an RSA-like representation. Specifically, given a modulus $N = pq$ of two secret primes p, q, an exponent e relatively prime to Euler's totient function $\varphi(N)$ and a value $g \in \mathbb{Z}_N^*$, find to $X \in \mathbb{Z}_N^*$ a representation $x \in \mathbb{Z}_e$ and $r \in \mathbb{Z}_N^*$ with $X = g^x r^e \bmod N$. It is well-known that given N, e, g coming up with some X and distinct representations $(x_1, r_1), (x_2, r_2)$ is as hard as the RSA problem [Ok92].

The RSA representation problem has a vast number of applications: for instance, Okamoto [Ok92] constructs an identification protocol secure against (parallel) active attacks which Pointcheval and Stern [PS00] subsequently turn into a secure signature scheme and a blind signature scheme. Fischlin and Fischlin [FF00] as well as Di Crescenzo et al. [CKOS01] use the RSA representation problem to derive efficient non-malleable commitment schemes based on RSA. Brands [B97] shows how to prove linear relations on committed values with an extended version of the RSA representation problem.

Interestingly, there is a seemingly less popular analogue to the RSA representation problem relying on the assumed hardness of factoring integers. In this

A preliminary version appears in: RSA Security 2002 Cryptographer's Track, Lecture Notes in Computer Science, Vol.2271, pp.96–113, © Springer-Verlag

case, a representation of X with respect to N, g and a number t is a pair $x \in \mathbb{Z}_{2^t}$ and $r \in \mathbb{Z}_N^*$ with $X = g^x r^{2^t} \bmod N$. Brassard et al. [BCC88] introduce this representation type for the special case $t = 1$. Damgård [D95] generalizes this to arbitrary $t \geq 1$ for Blum integers N where $p, q = 3 \bmod 4$. In order to advance to general moduli we introduce an "adjustment" parameter τ which depends on the prime factorization of N (and which equals 0 for Blum integers, for example), and we define a representation of X with respect to N, τ, g and t to be a pair $x \in \mathbb{Z}_{2^t}$ and $r \in \mathbb{Z}_N^*$ such that $X = g^x r^{2^{\tau+t}} \bmod N$. As we will elaborate, for appropriate choices of τ, g the task of finding a value X and different representations becomes equivalent to the factoring problem for *arbitrary moduli*.

One reason for the unpopularity of the factoring representation problem may stem from the fact that Okamoto's previously proposed identification scheme based on this problem is flawed. It is sufficient to solve the RSA problem to pass the identification scheme with constant probability, without necessarily being able to factor the modulus. We review this shortcoming in Appendix A. Fortunately, the bug in Okamoto's scheme is fixable, and we can indeed devise a secure identification scheme using the factoring representation problem. We show that for suitable parameters the protocol becomes provably secure under the factoring assumption.

Among other identification schemes provably secure as factoring, the presumably most popular are the Feige-Fiat-Shamir protocol [FFS88] and its variation due to Ong-Schnorr [OS90,S96] as well as Shoup's system [Sh99]. For these schemes there is a trade-off between the key size and security against parallel attacks. While the Feige-Fiat-Shamir protocol provides security against such parallel attacks, and therefore forms a fundament for secure resettable identification [BFGM01] and blind signatures with parallel signature generation [PS97,PS00], it also requires large secret and public keys. The Shoup and the Ong-Schnorr system, on the other hand, admit short keys but are conceivably not secure against parallel attacks.[1]

Our protocol supplements the known schemes and achieves security against parallel attacks and requires only short keys. With the techniques introduced in [PS00] we therefore obtain a secure signature scheme and a secure blind signature scheme withstanding up to poly-logarithmically many concurrent signature request, both in the random oracle model. Furthermore, we derive a secure resettable identification protocol by the general transformation presented in [BFGM01].

As for further applications, our result generalizes the result by Halevi [H99] that two-round commitment schemes does not only work with William integers

[1] Schnorr [S96,S97] claims that the Ong-Schnorr protocol with short keys is secure against parallel attacks for very special system parameters where a large power 2^m divides $p-1$ or $q-1$ (e.g., $m \geq 25$ for reasonable choices). Such primes form only a small subspace of all primes and may be much harder to find. Moreover, although we are not aware of any factoring method today taking advantage of this property, such moduli are in principle more vulnerable to improved factoring procedures.

but rather with any moduli. Also, plugging our result into the constructions of [FF00,CKOS01], we conclude that efficient non-malleable commitment schemes can be constructed under the assumption that factoring is hard. In fact, our variation of the protocols in [CKOS01] does not only base the security on a milder assumption, but also simplifies and improves the scheme concerning computational effort and communication complexity.

The paper is structured as follows. In Section 2 we formally state the representation problem based on factoring and prove equivalence to the intractability of factoring large numbers. Section 3 discusses applications of the representation problem to identification and (blind) signatures. In Section 4 we deal with commitments and show how to construct efficient non-malleable commitment schemes based on the factoring representation problem.

2 Representation Problem

We state the RSA and factoring representation problems formally in Sections 2.1 and 2.2, respectively. In Section 2.3 we prove the equivalence of the factoring representation problem to the factoring problem.

2.1 RSA Representation Problem

An RSA modulus $N = pq$ is the product of two distinct primes p, q. A corresponding RSA exponent $e \neq \pm 1 \bmod \varphi(N)$ is relatively prime to Euler's totient function $\varphi(N) = (p-1)(q-1)$ [RSA78]. We say that N is an n-bit modulus if n bits are sufficient and necessary for the binary representation of N, that is, if $2^{n-1} \leq N < 2^n$.

We presume that there is an efficient index generator $\mathsf{RSAIndex}$ for the representation problem which, on input 1^n, returns an n-bit RSA modulus N, a corresponding RSA exponent e and a random element $g \in_{\mathrm{R}} \mathbb{Z}_N^*$. Let $(N, e, g) \leftarrow \mathsf{RSAIndex}(1^n)$ denote the sampling process. An *RSA representation* for a value $X \in \mathbb{Z}_N^*$ with respect to a tuple (N, e, g) is a pair (x, r) with $x \in \mathbb{Z}_e$ and $r \in \mathbb{Z}_N^*$ such that

$$X = g^x r^e \bmod N.$$

Every $X \in \mathbb{Z}_N^*$ has exactly e representations with respect to (N, e, g), because for each $x \in \mathbb{Z}_e$ there is a unique $r \in \mathbb{Z}_N^*$ such that $r^e = X g^{-x} \bmod N$. We usually omit mentioning the reference to (N, e, g) if it is clear from the context, and simply say that (x, r) is a representation of X.

Definition 1 (RSA Representation Problem). *Given* $(N, e, g) \leftarrow \mathsf{RSAIndex}(1^n)$ *return some* $X \in \mathbb{Z}_N^*$ *as well as two different representations* $(x_1, r_1), (x_2, r_2) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$ *of* X*.*

In contrast, the ordinary RSA problem asks to compute the e-th root $g^{1/e} \bmod N$ given $(N, e, g) \leftarrow \mathsf{RSAIndex}(1^n)$. This task is widely assumed to be intractable, i.e., no polynomial-time algorithm solves the RSA problem with more than negligible success probability. This implies that factoring N, too, is believed to be intractable. Yet, it is an open problem if RSA is indeed equally hard as factoring (see also [BV98] for a discussion).

Provided one can solve the RSA problem, then the RSA representation problem becomes tractable, e.g., for any $r \in \mathbb{Z}_N^*$ both $(0, r)$ and $(1, rg^{-1/e} \bmod N)$ are representations of $X = r^e \bmod N$. The converse holds as well [Ok92], and the equivalence reveals that both problems can be solved with the same success/running time characteristics, neglecting minor extra computations (in the sequel we keep on disregarding the effort for such additional minor computations).

2.2 Factoring Representation Problem

We next address the factoring representation problem. We replace the RSA exponent e by some power of 2. Namely, we substitute e by $2^{\tau+t}$ where t describes the bit length of $x \in \mathbb{Z}_{2^t}$ and the integer τ depends on the prime factorization of the modulus N; we will explain the choice and role of this adjustment parameter τ later. Then a representation for $X \in \mathbb{Z}_N^*$ with respect to $N = pq$, $g \in \mathbb{Z}_N^*$ and $\tau \geq 0$, $t \geq 1$ is a pair $(x, r) \in \mathbb{Z}_{2^t} \times \mathbb{Z}_N^*$ such that

$$X = g^x r^{2^{\tau+t}} \bmod N.$$

Apparently, given the factorization of N one can easily come up with two different representations. The converse does not hold in general: for example (x, r) and $(x, -r)$ represent the same X. Since we are mainly interested in finding distinct x-components we therefore call representations (x_1, r_1) and (x_2, r_2) *different* or *distinct* if and only if $x_1 \neq x_2$. Observe that this subsumes the RSA case where distinct x-components imply different r's and vice versa.

Basically, the RSA and the factoring representation problem diverge concerning the equivalence to the underlying number-theoretic assumption because of the number of preimages of r^e and $r^{2^{\tau+t}}$, respectively. For RSA parameters the mapping $r \mapsto r^e \bmod N$ constitutes a permutation on $\mathbb{Z}_N^*$. Squaring on $\mathbb{Z}_N^*$, however, is a 4:1 mapping for $N = pq$. Restricting the modulus to a Blum integer where $p, q = 3 \bmod 4$ squaring becomes a permutation on the subgroup of quadratic residues QR_N. More generally, for any odd modulus N with prime factorization $N = \prod_{i=1}^{r} p_i^{e_i}$ where $p_1, p_2, \ldots, p_r$ are distinct odd primes and $e_1, e_2, \ldots, e_r \geq 1$, let η denote the smallest integer such that $2^{\eta+1}$ does not divide any $\varphi(p_i^{e_i})$. Then squaring is a permutation on the subgroup

$$\mathrm{HQR}_N := \{x^{2^\eta} \mid x \in \mathbb{Z}_N^*\} = \{x \in \mathbb{Z}_N^* \mid \mathrm{ord}_N(x) \text{ is odd}\}$$

of the "highest quadratic" residues, namely the 2^η-th powers (see, for example, [S96,H99]):

Proposition 1. *For any odd modulus N squaring is a permutation on* HQR_N.

Squaring permutes the 2^k-th powers for any $k \geq \eta$ for any odd n-bit modulus N. In other words, as long as $k \geq \eta$, the set of the 2^k-th powers of the elements in $\mathbb{Z}_N^*$ is the subgroup of elements with odd order. Since $\eta \leq n$, even without knowledge of η the set HQR_N is efficiently samplable by taking a random element from $\mathbb{Z}_N^*$ and raising it to its 2^n-th power.

With the similarity of Blum integers and QR_N to general moduli and HQR_N we are ready to state the factoring representation problem turning out to be equivalent to the factoring problem. But before, some words of clarification about the parameter τ follow. Recall that a representation for X is a pair (x, r) with $X = g^x r^{2^{\tau+t}} \bmod N$. In the following we demand that $\tau \geq \eta - 1$ and thus τ may reveal some information about the factors of N. But because $1 \leq \eta \leq n$ we can easily guess this information with probability $\frac{1}{n}$, or, in case of Blum moduli for instance, the fact $\eta = 1$ is publicly known anyway. In particular, for Blum integers we may set $\tau = 0$ and the representation problem in this case equals the one stated by Damgård [D95].

Let $\mathsf{FactIndex}$ denote an efficient index generator that outputs an n-bit RSA modulus N, $\tau \geq \eta - 1$, $t \geq 1$ and an independently chosen element $g \in_{\mathrm{R}} \mathrm{HQR}_N$ for input 1^n, and write $(N, \tau, t, g) \leftarrow \mathsf{FactIndex}(1^n)$ for the sampling process.

Definition 2 (Factoring Representation Problem). *Given* $(N, \tau, t, g) \leftarrow \mathsf{FactIndex}(1^n)$ *return some* $X \in \mathbb{Z}_N^*$ *as well as two different representations* $(x_1, r_1), (x_2, r_2) \in \mathbb{Z}_{2^t} \times \mathbb{Z}_N^*$ *of* X*, i.e., with* $x_1 \neq x_2$*.*

An important observation for our identification and commitment protocols is that each $X \in \mathrm{HQR}_N$ has exactly 2^t different representations. It follows that for a random representation (x, r) the value $X := g^x r^{2^{\tau+t}} \bmod N$ does not reveal anything about the specific x.

2.3 Factoring Representation Problem and Factoring

Given the factorization of N it is easy to compute a $2^{\tau+t}$-th root of $g \in \mathrm{HQR}_N$ and the corresponding representation problem becomes tractable. On the other hand, by solving the representation problem one efficiently determines the prime factors of N. Before we prove this we present a technical lemma:

Lemma 1. *If a probabilistic algorithm solves the factoring representation problem* $(N, \tau, t, g) \leftarrow \mathsf{FactIndex}(1^n)$*, then a* $2^{\tau+1}$*-th root* $b \in \mathbb{Z}_N^*$ *of* g *can be computed within the same time bound and same success probability.*

Proof. Given two different representations (x_1, r_1) and (x_2, r_2) of some $X \in \mathbb{Z}_N^*$, let $\Delta x := x_1 - x_2$ and $r := r_2 r_1^{-1} \bmod N$ where $0 < |\Delta x| < 2^t$. Then

$$g^{\Delta x} = g^{x_1 - x_2} = r_2^{2^{\tau+t}} r_1^{-2^{\tau+t}} = r^{2^{\tau+t}} \bmod N. \tag{1}$$

Notice that the exponents Δx and $2^{\tau+t}$ may not be relatively prime. So suppose $2^k = \gcd(\Delta x, 2^{\tau+t})$ where $0 \leq k < t$. Computing $u, v \in \mathbb{Z}$ subject to $u\Delta x + v2^{\tau+t} = 2^k$ by applying the extended Euclidean algorithm we derive

$$g^{2^k} = g^{u\Delta x + v2^{\tau+t}} = (g^{\Delta x})^u \cdot (g^v)^{2^{\tau+t}} = (r^u g^v)^{2^{\tau+t}} \bmod N.$$

Let $b := (r^u g^v)^{2^{t-k-1}} \bmod N$. Then

$$g^{2^k} = (b^{2^{\tau+1}})^{2^k} \pmod N.$$

We have $g = b^{2^{\tau+1}} \pmod N$ since $g, b^{2^{\tau+1}} \in \mathrm{HQR}_N$ and squaring permutes HQR_N. □

We next prove that factoring is reducible to the factoring representation problem:

Theorem 1. *If a probabilistic algorithm solves the factoring representation problem $(N, \tau, t, g) \leftarrow \mathsf{FactIndex}(1^n)$ with probability ϵ, then N can be factored within the same time bound and success probability at least $\frac{1}{2}\epsilon$.*

Proof. Pick a random $a \in_{\mathrm{R}} \mathbb{Z}_N^*$ and set $g := a^{2^{\tau+1}} \bmod N$ which is uniformly distributed in HQR_N. Based on Lemma 1 compute some $2^{\tau+1}$-th root b of g. Let $c := ab^{-1} \bmod N$. Then:

$$c^{2^{\tau+1}} = 1 = c^{2^\eta} \bmod N.$$

The second equation follows from $\tau + 1 \geq \eta$ and since squaring is a permutation on HQR_N. We next consider the equation modulo the prime factors p, q of N. Suppose $p - 1 = 2^{\eta_p} p'$ and $q - 1 = 2^{\eta_q} q'$ for odd p', q', and therefore $\eta = \max\{\eta_p, \eta_q\}$. Wlog. let $\eta = \eta_p$. Because of $\eta_q \leq \eta_p = \eta$ we have

$$c^{2^\eta} = 1 \bmod p \qquad c^{2^{\eta-1}} = \sigma_p \bmod p$$
$$c^{2^\eta} = 1 \bmod q \qquad c^{2^{\eta-1}} = \sigma_q \bmod q$$

for some $\sigma_p, \sigma_q \in \{\pm 1\}$.[2] To complete the proof we show that $\sigma_p \sigma_q = -1$ holds with probability $\frac{1}{2}$, because in that case one of the GCD computations $\gcd\bigl(c^{2^{\eta-1}} \pm 1, N\bigr)$ yields the factorization of N.

Note that c is uniformly distributed among the 2^η-th roots of unity, because g does not reveal any information about the random root a we have actually chosen, thus the element b determined by the representation finder's output is independent of a. Hence, $c \bmod p$ and $c \bmod q$ are independently and uniformly distributed among the 2^η-th roots of unity modulo p and modulo q. Consequently, σ_p and σ_q are independent.

For half of the 2^η-th roots w of 1 modulo p we have $w^{2^{\eta-1}} = 1 \bmod p$ and otherwise $w^{2^{\eta-1}} = -1 \bmod p$. Since $c \bmod p$ is a random 2^η-th root of unity modulo p, the value σ_p is uniformly distributed in $\{\pm 1\}$. As σ_p does not depend on σ_q we have $\sigma_p \sigma_q = -1$ with probability $\frac{1}{2}$. □

Figure 1 illustrates the proof idea of Theorem 1. The root of each tree is labeled with $+1$. Descending from one node to the successors corresponds to taking a square root modulo the prime p or q, e.g., in the left tree the tree's root $+1$ has the successors $+1$ and -1 as squaring is still a 2:1-mapping modulo p, whereas in the right tree $\eta_q < \eta_p$ and squaring permutes HQR_q, implying that the tree's root $+1$ only has the square root $+1$. Hence, the leaves in each tree represent all 2^{η_p} and 2^{η_q} many 2^η-th roots of 1 modulo p and q, respectively.

[2] While both values for σ_p may occur, if $\eta_q < \eta$ then we always have $\sigma_q = +1$ as squaring is one-to-one on HQR_q and $+1$ is the unique square root.

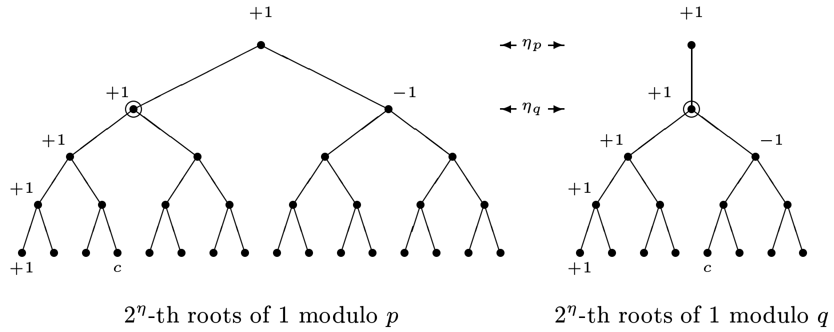

Fig. 1. Factoring via 2^η-th root of 1

The path to the leftmost leaf in each tree represents the 2^η-th root 1 of 1 modulo p and modulo q (1-*path*). In the proof we use the representation finder to derive a random 2^η-th root c of 1 in $\mathbb{Z}_N^*$. Thus, the values $c \bmod p$ and $c \bmod q$ each describe a random path to some leaf in the corresponding tree (c-*path*), and each path is independent of the other one. We are able to find the prime factors of N if and only if at some level k one of the c-paths branches from the 1-path whilst the other one still follows the 1-path. For example, in Figure 1 this happens in the marked nodes for $k = \eta_p - 1$: there we have $c^{2^{k-1}} = 1 \bmod p$ but $c^{2^{k-1}} = -1 \bmod q$ and a GCD computation yields the prime factors of N. In fact, in the proof of Theorem 1 we only check the divergence of the paths for $k = \eta_p = \eta$. Therefore, except for Blum integers, the probability of retrieving the factors is actually higher than $\frac{1}{2}\epsilon$.

Theorem 1 even holds for fixed $g \in \mathrm{HQR}_N$, given that $\tau \geq \eta$ and some 2^τ-th root $a \notin \mathrm{QR}_N$ of g with $-a \notin \mathrm{QR}_N$ is publicly known. Besides that variant the factoring representation problem gives rise to other modifications and generalizations:

1. One may substitute the RSA modulus by an arbitrary odd integer N. Then the algorithm of Theorem 1 retrieves a non-trivial factor of N.
2. The problem can be relaxed such t is not given as part of the output of FactIndex, but the representation finder rather gets the freedom to select an arbitrary $t \geq 1$ on its own after seeing (N, τ, g). Given two representations (x_1, r_1, t_1) and (x_2, r_2, t_2) where wlog. $t_1 \geq t_2$ use $r := r_2 r_1^{-2^{t_1 - t_2}}$ for the proof of Lemma 1.
3. If $e = \mathcal{O}(\log n)$ divides $\varphi(N)$, then one may replace $2^{\tau+t}$ by $e^{\tau+t}$. In this case, η denotes the smallest integer such that $e^{\eta+1}$ neither divides $p-1$ nor $q-1$ and use $\{x^{e^\eta} \mid x \in \mathbb{Z}_N^*\}$ instead of HQR_N. The hardness is also based on factoring as Ohto and Okamoto [OO88] have shown that taking e-th roots in this case is equivalent to factoring N.

3 Identification and Signature Schemes

In this section we show how to repair Okamoto's identification protocol [Ok92] obtaining a provably secure identification scheme withstanding parallel active attacks. Exploiting the relationship to signature schemes via the Fiat-Shamir heuristic [FS86], we then show that this identification protocol can be used for ordinary as well as blind signatures.

3.1 Identification Scheme

Our identification protocol in Figure 2 follows the framework of Okamoto [Ok92] for the RSA setting, which in turn is an extension of the Guillou-Quisquater setup [GQ88]. The values $(N, \tau, t, g) \leftarrow \mathsf{FactIndex}(1^n)$ are public parameters. The public key of a user is $X \in_{\mathrm{R}} \mathrm{HQR}_N$ and the corresponding secret key is a random representation $(x, r) \in \mathbb{Z}_{2^t} \times \mathbb{Z}_N^*$ of X. The user is not required to be aware of the factorization of N and several users may share the same public parameters N, τ, g (even with different t's). The prover $\mathcal{P}$ tries to convince the verifier $\mathcal{V}$ that

Fig. 2. Identification Scheme using Factoring Representation Problem

Prover $\mathcal{P}$	(N, τ, t, g)	**Verifier** $\mathcal{V}$
representation x, r of X	$X = g^x r^{2^{\tau+t}} \bmod N$	
pick $y \in_{\mathrm{R}} \mathbb{Z}_{2^{\tau+t}}$, $s \in_{\mathrm{R}} \mathbb{Z}_N^*$ $Y := g^y s^{2^{\tau+t}} \bmod N$	$\xrightarrow{\quad Y \quad}$	
	$\xleftarrow{\quad c \quad}$	pick $c \in_{\mathrm{R}} [0, 2^t)$
$z := y + cx \bmod 2^{\tau+t}$ $W := sr^c g^{\lfloor (y+cx)/2^{\tau+t} \rfloor} \bmod N$	$\xrightarrow{\quad W, z \quad}$	$YX^c \overset{!}{=} W^{2^{\tau+t}} g^z$

$\mathcal{P}$ knows a representation of X with respect to (N, τ, t, g). In the first step $\mathcal{P}$ sends an initial commitment Y to $\mathcal{V}$ who answers with a challenge $c \in_{\mathrm{R}} [0, 2^t)$, and $\mathcal{P}$ finally hands the response W, z to $\mathcal{V}$ which determines acceptance or rejection.

Obviously, this protocol is complete in the sense that the honest prover $\mathcal{P}$ always passes the examination of honest verifiers $\mathcal{V}$. We show that this identification scheme is secure against active attacks, i.e., where the adversary $\mathcal{A}$ may run executions with the honest prover before trying to impersonate. But first we start with passive attacks in which the adversary tries to intrude given the public key only:

Lemma 2. *If a passive adversary $\mathcal{A}$ passes the identification scheme in Figure 2 with time bound T and success probability ϵ, then the modulus N can be factored in expected time $\mathcal{O}(T)$ with probability at least $\frac{1}{4}(\epsilon - 2^{-t})$.*

Proof. The proofs follows the one in [Ok92] for RSA. Let $N = pq$, τ, t and a random $g \in_{\mathrm{R}} \mathrm{HQR}_N$ be given, i.e., $(N, \tau, t, g) \leftarrow \mathsf{FactIndex}(1^n)$. We show how to compute with probability $\frac{1}{2}(\epsilon - 2^{-t})$ some $2^{\tau+1}$-th root of g with the help of $\mathcal{A}$. As a result, the claim is a consequence of Theorem 1.

Pick $x \in_{\mathrm{R}} \mathbb{Z}_{2^t}$ and $r \in_{\mathrm{R}} \mathbb{Z}_N^*$. Next, simulate an attack of $\mathcal{A}$ for $N, t, g, X := g^x r^{2^{\tau+t}} \bmod N$. After $\mathcal{A}$ has sent W, z rewind to the situation where $\mathcal{A}$ faces the challenge. By this, we obtain in expected time $\mathcal{O}(T)$ with probability $\epsilon - 2^{-t}$ two successful intrusion attempts in which $\mathcal{A}$ has sent the same Y but has answered with W, z and W', z' to different challenges c, c'. Then

$$X^{-c} W^{2^{\tau+t}} g^z = Y = X^{-c'} (W')^{2^{\tau+t}} g^{z'} \bmod N$$

or, rewritten,

$$g^{z-z'+x(c'-c)} = (r^{c-c'} W^{-1} W')^{2^{\tau+t}} \bmod N \tag{2}$$

Let $\Delta z := z - z'$ and $\Delta c := c' - c$. Now we have an equation similar to Equation (1) in the proof of Lemma 1. If $\gcd(\Delta z + x\Delta c, 2^{\tau+t}) = 2^k$ for some $k < t$, then we are able to retrieve some $2^{\tau+1}$-th root of g. To complete the proof it thus suffices to give an upper bound of $\frac{1}{2}$ for the probability that the GCD exceeds 2^{t-1}. Obviously,

$$\gcd(\Delta z + x\Delta c, 2^{\tau+t}) \geq 2^t \qquad \Longleftrightarrow \qquad x \cdot \Delta c = -\Delta z \bmod 2^t.$$

Whenever this modular equation is solvable, then for fixed $\Delta c, \Delta z$ the number of solutions for x equals $2^j := \gcd(\Delta c, 2^t)$ where $0 \leq j < t$ because $0 < |\Delta c| < 2^t$. Observe that in the actual protocol execution the selection of the parameters $\Delta c, \Delta z$ for the equation is done *after* the variable x has been chosen. But Δc is distributed independently of x because the challenges are simply picked at random, and the distribution of the adversary's choice z, z' for Δz does not depend on x either since the public key X does not reveal anything about the specific choice of x. Therefore, we can view the process as first fixing $\Delta c, \Delta z$ and then picking $x \in \mathbb{Z}_{2^t}$ at random. But then the probability that the random x matches the equation is bounded above by 2^{j-t}. From $j < t$ it follows that this probability is at most $\frac{1}{2}$. □

Note that this approach factors N but unlike the corresponding RSA based scheme it does not extract a representation of the prover. Hence, once more we have a secure identification protocol which does not constitute a proof of knowledge in the sense of Bellare and Goldreich [BG92]. See [OS90,S96,Sh99] for other examples.

In order to prove security against active adversaries, we follow the approach in [Ok92] and show that even executions with the prover before the intrusion attempt do not disclose any information about x (called witness-indistinguishability [FS90]):

Lemma 3. *The protocol in Figure 2 is perfectly witness-indistinguishable.*

Proof. We have to justify that even in the case of a dishonest verifier $\mathcal{V}$ the view (i.e., the distribution of the communication Y, c, W, z) is independent of the representation actually known by the prover $\mathcal{P}$. We show that for any communication (Y, c, W, z) of an execution of $\mathcal{V}$ with $\mathcal{P}$, another prover $\mathcal{P}'$ knowing another representation (x', r') of X generates this communication with the same probability in an execution with $\mathcal{V}$.

Let $\Delta x := x' - x$ and $\Delta r = r/r' \bmod N$. Since $r^{2^{\tau+t}} = Xg^{-x}$ we have $\Delta r^{2^{\tau+t}} = g^{\Delta x}$. Assume that (Y, c, W, z) is a transcript of a communication with $\mathcal{P}$ having chosen y, s at the outset. The probability that $\mathcal{P}'$ picks

$$\begin{aligned} y' &:= y - c \cdot \Delta x \bmod 2^{\tau+t} \\ s' &:= s \cdot \Delta r^{c} \bmod N \end{aligned}$$

in the first step is exactly the same as for $\mathcal{P}$ choosing y, s; both times the values are uniformly distributed. For this choice of y', z' we have $Y' = Y$, and therefore $\mathcal{V}$ returns the challenge $c' = c$ with equal probability in both executions. Now, W', z' and W, z are deterministically determined by the secret key, the challenge and the random values from the first step, and it is easily shown that $(W', z') = (W, z)$ here. Hence, the probability that a run with $\mathcal{P}'$ generates (Y, c, W, z) equals the one for $\mathcal{P}$. This completes the proof. $\square$

It follows that the identification scheme is also secure against active attacks:

Theorem 2. *If an active adversary $\mathcal{A}$ passes the identification scheme in Figure 2 with time bound T and success probability ϵ, then the modulus N can be factored in expected time $\mathcal{O}(T)$ with probability at least $\frac{1}{4}(\epsilon - 2^{-t})$.*

Proof. Given N, t, g pick a random secret key (x, r) and simulate an attack $\mathcal{A}$ on N, t, g and the public key $X := g^x r^{2^{\tau+t}} \bmod N$. This includes several interactions of $\mathcal{A}$ with the prover before trying to fool the verifier. But we can easily run these prover-adversary executions as we know the secret key. Due to the witness-indistinguishability, these executions still hide x perfectly, and the argument of Lemma 2 applies. $\square$

The proposition even holds if the adversary is allowed to run concurrent executions with the prover. Hence, the scheme can be turned into one secure against reset attacks under the factoring assumption; for details see [BFGM01].

3.2 Signature Schemes

The identification scheme in Figure 2 gives rise to a signature scheme secure against chosen-message attacks [GMR88] using the Fiat-Shamir heuristic. The challenge is generated by applying a hash function H to the message and the initial commitment of the prover. More specifically, publish (N, τ, t, g) as public parameter, X as public key and use the representation (x, r) as the secret key of the signer $\mathcal{S}$. In order to sign a message m pick $y \in_{\mathrm{R}} \mathbb{Z}_{2^{\tau+t}}$ and $s \in_{\mathrm{R}} \mathbb{Z}_N^*$ at random, calculate $Y := g^y s^{2^{\tau+t}} \bmod N$, $c := H(Y, m)$ and compute z, W as in the case of the identification scheme. The signature to m becomes $\sigma(m) := (Y, W, z)$. Verification is straightforward.

Provided the hash function H behaves like a random function, then even with the help of an signature oracle any adversary fails to come up with a valid signature for a new message of his own choice [PS00, Sec. 3.2]:

Proposition 2. *In the Random Oracle Model the signature scheme based on the factoring representation problem is secure against existential forgery under adaptive chosen-message attacks relative to the hardness of factoring.*

Fig. 3. Blind Signature Scheme using Factoring Representation Problem

Signer $\mathcal{S}$	(N, τ, t, g)	**User $\mathcal{U}$**
representation x, r of X	$X = g^x r^{2^{\tau+t}} \bmod N$	
pick $y \in_{\mathrm{R}} \mathbb{Z}_{2^{\tau+t}}$, $s \in_{\mathrm{R}} \mathbb{Z}_N^*$ $Y := g^y s^{2^{\tau+t}} \bmod N$	$\xrightarrow{\quad Y \quad}$	pick $\alpha, \gamma \in_{\mathrm{R}} \mathbb{Z}_{2^{\tau+t}}$ pick $\beta \in_{\mathrm{R}} \mathbb{Z}_N^*$ $Y' := Y g^{\alpha} \beta^{2^{\tau+t}} X^{\gamma}$ $c := H(Y', m) \in \mathbb{Z}_{2^t}$ $c' := c + \gamma \bmod 2^{\tau+t}$ set c'' such that
	$\xleftarrow{\quad c' \quad}$	$c + \gamma = c' + 2^{\tau+t} c''$
$z' := y + c'x \bmod 2^{\tau+t}$ $W' := s r^{c'} g^{\lfloor (y+c'x)/2^{\tau+t} \rfloor} \ (N)$	$\xrightarrow{\quad W', z' \quad}$	$Y X^{c'} \overset{!}{=} (W')^{2^{\tau+t}} g^{z'}$ $z := z' - \alpha \bmod 2^{\tau+t}$ set z'' such that $z' + \alpha = z - 2^{\tau+t} z''$ $W := W' \beta g^{-z''} X^{c''} \ (N)$
	Then $Y' X^c = W^{2^{\tau+t}} g^z$	

An important variant of signatures schemes are blind signatures. In this case, the user $\mathcal{U}$ blinds the actual message m and requests a signature from the signer $\mathcal{S}$, which $\mathcal{U}$ later turns into a valid signature for the message m while the signer $\mathcal{S}$ cannot infer something about m. In a "one-more" forgery the adversary $\mathcal{A}$ tries to generate one more signed message than $\mathcal{A}$ originally requested from the signer $\mathcal{S}$ [PS00]. For example, in the ecash setting where messages signed by the bank represent anonymous digital coins, $\mathcal{U}$ cannot spent more money than $\mathcal{U}$ has actually withdrawn from the bank.

The blind signature scheme based on the factoring problem is given in Figure 3; it is heavily influenced by the discrete-log and RSA protocols of Pointcheval and Stern [PS00]. In the first step, the signer $\mathcal{S}$ commits to Y. Then the user $\mathcal{U}$ blinds Y by multiplying with $g^{\alpha}\beta^{2^{\tau+t}}X^{\gamma}$ for random values α, β, γ. The actual challenge $c \in \mathbb{Z}_{2^t}$ is hidden by $c' := c + \gamma \in \mathbb{Z}_{2^{\tau+t}}$. $\mathcal{S}$ replies the challenge by sending W', z' subject to

$$YX^{c'} = (W')^{2^{\tau+t}} g^{z'}.$$

Now, $\mathcal{U}$ undoes the blinding and finally retrieves the signature $\sigma(m) := (Y', W, z)$:

$$\begin{aligned} W^{2^{\tau+t}} g^{z} &= (W'\beta g^{-z''} X^{c''})^{2^{\tau+t}} g^{z'} g^{z-z'} \\ &= YX^{c'} (\beta g^{-z''} X^{c''})^{2^{\tau+t}} g^{z-z'} \\ &= YX^{c'+2^{\tau+t}c''} \beta^{2^{\tau+t}} g^{z-z'-2^{\tau+t}z''} \\ &= YX^{c+\gamma} \beta^{2^{\tau+t}} g^{\alpha} \\ &= Y'X^{c}. \end{aligned}$$

The scheme is perfectly blind as (Y, c', W', z') and (Y', c, W, z) are independently distributed. Security follows as in [PS00]:

Theorem 3. *In the Random Oracle Model the blind signature scheme based on the factoring representation problem is secure against a "one-more" forgery under a parallel attack (where up to poly-logarithmic signature generations are executed concurrently) relative to the hardness of factoring.*

Note that this scheme is provable secure against interleaving attacks meanwhile the one based on the Ong-Schnorr identification is only known to be secure against sequential attacks [PS97].

4 Commitment Schemes

A commitment scheme is a protocol of three stages (initialization, commitment and decommitment) between to parties called the sender $\mathcal{S}$ and the receiver $\mathcal{R}$. In the commitment stage $\mathcal{S}$ binds himself to a message m by sending a commitment meanwhile the receiver $\mathcal{R}$ cannot deduce any information about m. Later, the sender $\mathcal{S}$ reveals m and $\mathcal{R}$ checks whether this message indeed matches the commitment.

4.1 Non-Interactive Commitment Scheme

In this section we set up a commitment scheme based on the factoring representation problem following the well-known scheme derived from the RSA representation problem and generalizing Halevi's scheme [H99].

Assume for the moment that a trusted third party selects a valid instance $(N, \tau, t, g) \leftarrow \mathsf{FactIndex}(1^n)$ for the factoring representation problem and publishes it; we afterwards discuss how to delegate this task to the receiver. In any case, $\mathcal{S}$ must not know the factorization of N. To commit to a message $m \in \mathbb{Z}_{2^t}$, the sender $\mathcal{S}$ picks a random $r \in_{\mathrm{R}} \mathbb{Z}_N^*$ and sends

$$\mathrm{com}(m, r) := g^m r^{2^{\tau+t}} \bmod N \tag{3}$$

to the receiver $\mathcal{R}$. For the decommitment, $\mathcal{S}$ reveals the commited message m and the random value r. The receiver $\mathcal{R}$ verifies that (m, r) is indeed a representation of $\mathrm{com}(m, r)$.

If we let the receiver instead choose $(N, \tau, t, g) \leftarrow \mathsf{FactIndex}(1^n)$ and send it to $\mathcal{S}$ in the first step, then there is no guarantee that a malicious receiver does not select inproper values like $g \notin \mathrm{HQR}_N$ or $\tau < \eta - 1$. To prevent this we take $\tau := n$ and use a method suggested in [H99] to make sure that g really is an element from HQR_N, even if N is not the product of two primes. Namely, let the sender verify that N is odd and raise g to the 2^n-th power first. $\mathcal{S}$ then transmits

$$\mathrm{com}(m, r) := \left(g^{2^n}\right)^m \left(r^{2^n}\right)^{2^{t+n}} = g^{m2^n} r^{2^{2n+t}} \bmod N \tag{4}$$

Given factoring N is intractable, then Theorem 1 implies that $\mathcal{S}$ cannot come up the a different representation of $\mathrm{com}(m, r)$, in either case (3) or (4). Hence, a commitment is computationally binding and $\mathcal{S}$ cannot ambiguously open the commitment. On the other hand, the distribution of $\mathrm{com}(m, r) \in \mathrm{HQR}_N$ is independent of the message m, that is, even a computationally unbounded malicious receiver $\mathcal{R}$ is unable to deduce any information about m given only the commitment. To summarize:

Proposition 3. *The factoring representation commitment scheme* (3) *respectively* (4) *has the following properties:*

1. *Computational unambiguity relative to the hardness of factoring.*
2. *Perfect privacy.*

We compare this commitment scheme with the one introduced by Halevi [H99]. To commit to a message $m \in \mathbb{Z}_{2^t}$ with a trusted setup mechanism providing a correct N pick at random $r \in_{\mathrm{R}} \mathbb{Z}_N^*$ and publish

$$\mathrm{com}(m, r) := 4^m r^{2^{t+1}} \bmod N \tag{5}$$

for a William integer N, i.e., an RSA modulus $N = pq$ with $p = 3 \bmod 4$ and $q = 7 \bmod 8$. The binding property relative to the hardness of factoring N can be proven in a direct way [H99]. Alternatively one may apply Theorem 1. We have $\eta = 1$, $\tau := \eta$ and $4 \in \mathrm{HQR}_N$ because its square root $(+2, -2) \in \mathrm{QR}_p \times \mathrm{QR}_q$ is a square, too. As $\pm 2 \notin \mathrm{QR}_N$, the adversary has to compute some other square root of 4 yielding the factorization of N.

4.2 Non-Malleable Commitment Scheme

Roughly speaking a commitment scheme is non-malleable if for any adversary $\mathcal{A}$ seeing the commitment of an honest sender $\mathcal{S}$ to an unknown message m it is infeasible to commit to a related (but different) message m^*. Depending on the level of security, the adversary may also be obliged to provide a valid decommitment after having learned the decommitment of $\mathcal{S}$ (called non-malleability with respect to opening). See [DDN00,FF00] for details.

Fischlin and Fischlin [FF00] and Di Crescenzo et al. [CKOS01] present efficient non-malleable commitment schemes based either on the discrete-log or the RSA assumption. All protocols work in the public parameter model, where public data like an RSA modulus N and a value $g \in \mathbb{Z}_N^*$ are published by a trusted party. Also, both solutions apply so-called trapdoor or equivocable commitments: knowledge of a secret information, the trapdoor, enables to open a given commitment with any message later on. For instance, for RSA an e-th root of g allows to fake commitments. Here, in case of the factoring representation commitment scheme (3), a $2^{\tau+t}$-th root h of $g \in \mathrm{HQR}_N$ provides a trapdoor, because a commitment $g^m r^{2^{\tau+t}}$ can be opened for m' by transmitting m' and $r' := h^{m-m'} r \bmod N$.

Fig. 4. Non-Interactive Non-Malleable Commitment Scheme

Sender $\mathcal{S}$	$N, \tau, t, H(\cdot)$ $g, h_0, h_1 \in \mathrm{HQR}_N$	**Receiver $\mathcal{R}$**
message $m \in \mathbb{Z}_{2^t}$		
<u>*commitment:*</u>		
pick $r, s \in_{\mathrm{R}} \mathbb{Z}_N^*$ $X := g^m r^{2^{\tau+t}} \bmod N$ $x := H(X)$ $M := (h_0^x h_1)^m s^{2^{\tau+2t}} \bmod N$	$\xrightarrow{x, M}$	
<u>*decommitment:*</u>		
	$\xrightarrow{m, r, s}$	$X := g^m r^{2^{\tau+t}} \bmod N$, check $x \stackrel{!}{=} H(X)$, $M \stackrel{!}{=} (h_0^x h_1)^m s^{2^{\tau+2t}} \bmod N$

We discuss how to modify the non-malleable commitments schemes based on the RSA representation problem [FF00,CKOS01] to derive non-malleable commitments schemes as secure as factoring. Fischlin and Fischlin [FF00] present interactive schemes that work with the RSA representation problem, one time using standard proofs of knowledge, the other time with a more sophisticated variant based on the Chinese Remainder Theorem. We cannot plug in the factoring representation problem into the proof-of-knowledge-based approach as the protocol given in Figure 2 does not constitute a proof of knowledge. However, we can use the Chinese Remainder Theorem protocol with the factoring representation problem instead of the RSA problem. By this, we obtain a non-malleable commitment scheme with statistical privacy. Details are omitted.

The non-interactive scheme in [CKOS01] achieves a weaker notion of non-malleability than the one in [FF00], where the adversary does not have any side information about the message m of $\mathcal{S}$. In Figure 4 we present a modification of their RSA protocol which is based on the factoring representation problem. Surprisingly, although this modification works under a potentially weaker assumption than RSA, it is even more efficient than the RSA protocol in [CKOS01]. In fact, transferring our protocol to the RSA or discrete-log representation problem setting also improves these protocols in [CKOS01] with respect to computational and communication complexity.

Basically, we let the sender commit twofold to the message m: one time by X with the standard factoring representation problem, the other time by M with a base where the hash value of the former commitment enters (and for which we use $\tau + 2t$ rather than $\tau + t$, see below). For this, let $H : \mathrm{HQR}_N \to \mathbb{Z}_{2^t}$ be some universal one-way hash function [NY90] with which we hash down $X \in \mathbb{Z}_N^*$ to $x \in \mathbb{Z}_{2^t}$. In case of $t \geq n$ one may eliminate the hash function by using X as exponent x.

Theorem 4. *There exists (efficient) commitments schemes with the following properties relative to the hardness of factoring:*

1. *Non-malleable with respect to opening.*
2. *Computationally binding.*
3. *Statistical privacy (and perfect privacy for the scheme in Figure 4).*

We outline the non-malleability proof for the non-interactive commitment scheme given in Figure 4. The definition of non-malleability essentially requires that for any adversary that is given a commitment of the sender and generates another commitment for which it is also able to adapt the sender's opening to one of a related message, there is a simulator that is almost as successful but without interacting with the sender at all.

We briefly recall the proof method in [CKOS01]. There, the simulator prepares a commitment on behalf the original sender which includes a trapdoor. The simulator submits it to the adversary who answers with its commitment. Then the simulator samples a sufficient number of random messages and sequentially opens the trapdoor commitment (by adapting the decommitment with the trapdoor accordingly). By this, the adversary reveals with sufficiently high probability a valid opening for its commitment to *some* message. The probability that the adversary finds different valid openings is negligible under the discrete-log or RSA assumption, hence, the simulator extracts *the* message of the adversary that is related to the original message of the sender.

In our case, given a commitment (x, M) for some unknown message m, the adversary $\mathcal{A}$ tries to commit to a related but different message m^* by sending (x^*, M^*). We first condition on the event that the adversary selects $x^* \neq x$. Assume towards contradiction that the adversary succeeds in an actual attack by sending $x^* = x$ with noticeable probability. For simplicity, we presume that the sender's value $X := g^m r^{2^{\tau+t}} \bmod N$ equals the adversary's choice $X^* := g^{m^*}(r^*)^{2^{\tau+t}} \bmod N$; otherwise we find a collision for the universal one-way hash function H. But then the decommitment step yields distinct representations of X and this allows to efficiently solve the factoring representation problem with noticeable success. We may thus consider only the adversary's success on values $x^* \neq x$ without sacrificing more than a negligible probability.

It remains to describe the trapdoor in our scheme to apply the technique of [CKOS01]. Given $(N, \tau, t, h_0) \leftarrow \mathsf{FactIndex}(1^n)$ select a universal one-way hash function H and define $M := s^{2^{\tau+t}}$, $X := r^{2^{\tau+t}}$, $g := u^{2^{\tau+t}}$, $h_1 := h_0^{-x} v^{2^{\tau+2t}}$ for random $r, s, u, v \in_{\mathrm{R}} \mathbb{Z}_N^*$ and $x := H(X)$. Take $(N, \tau, t, H, g, h_0, h_1)$ as public parameters and send (x, M) on behalf of the honest sender.

For the data in the simulation we know a $2^{\tau+2t}$-root v of $h_0^x h_1 = v^{2^{\tau+2t}}$. This, together with the $2^{\tau+t}$-th root of X, enables us to correctly open the commitment (x, M) with any message later. In contrast, even if we know the trapdoor, the adversary will not be able to find distinct openings for its commitment since, by assumption, $x^* \neq x$. The reason for this is that any valid decommitments of the adversary including (m_1^*, r_1^*), (m_2^*, r_2^*) for M^* imply that

$$(h_0^{x^*} h_1)^{m_1^*}(r_1^*)^{2^{\tau+2t}} = M^* = (h_0^{x^*} h_1)^{m_2^*}(r_2^*)^{2^{\tau+2t}}$$

and, substituting $h_1 = h_0^{-x} v^{2^{\tau+2t}}$,

$$h_0^{(x^*-x)m_1^*}(v^{m_1^*} r_j)^{2^{\tau+2t}} = M^* = h_0^{(x^*-x)m_2^*}(v^{m_2^*} r_2^*)^{2^{\tau+2t}}.$$

Since $x^* - x \neq 0$ and both products with $m_1^* \neq m_2^*$ are less than 2^{2t} this results in different representations with x-components $(x^* - x)m_1^*$, $(x^* - x)m_2^* \in \mathbb{Z}_{2^{2t}}$ for M^*. The probability that this happens is therefore negligible under the factoring assumption. With these preliminaries the rest of the proof is the same as in [CKOS01].

5 Acknowledgments

This work has been stimulated by discussions with Stefan Brands at Crypto 2000 about the factoring based representation problem mentioned in his Eurocrypt '97 paper. We also thank the anonymous reviewers for their comments.

References

[BFGM01] M. Bellare, M. Fischlin, S. Goldwasser and S. Micali: *Identification Protocols Secure Against Reset Attacks*, Advances in Cryptology — Proceedings Eurocrypt 2001, Lecture Notes in Computer Science, vol. 2045, pp. 495–511, Springer Verlag, 2001.

[BG92] M. Bellare and O. Goldreich: *On Defining Proofs of Knowledge*, Advances in Cryptology — Proceedings Crypto '92, Lecture Notes in Computer Science, vol. 740, pp. 390–420, Springer Verlag, 1993.

[BR93] M. Bellare and P. Rogaway: *Random Oracles are Practical: a Paradigm for Designing Efficient Protocols*, First ACM Conference on Computer and Communication Security, ACM Press, pp. 62–73, 1993.

[B97] S. Brands: *Rapid Demonstration of Linear Relations Connected by Boolean Operators*, Advances in Cryptology — Proceedings Eurocrypt '97, Lecture Notes in Computer Science, vol. 1233, pp. 318–333, Springer-Verlag, 1997.

[BCC88] G. Brassard, D. Chaum and C. Crépeau: *Minimum Disclosure Proofs of Knowledge*, Journal Computing System Science, vol. 37(2), pp. 156–189, 1988.

[BV98] D. Boneh and R. Venkatesan: *Breaking RSA may Not be Equivalent to Factoring*, Advances in Cryptology — Proceedings Eurocrypt '98, Lecture Notes in Computer Science, vol. 1403, pp. 59–71, Springer Verlag, 1998.

[CKOS01] G. Di Crescenzo, J. Katz, R. Ostrovsky and A. Smith: *Efficient And Non-Interactive Non-Malleable Commitment*, Advances in Cryptology — Proceedings Eurocrypt 2001, Lecture Notes in Computer Science, vol. 2045, pp. 40–59, Springer Verlag, 2001.

[D95] I. Damgård: *Practical and Provable Secure Release of a Secret and Exchange of Signature*, Journal of Cryptology, vol. 8, pp. 201–222, 1995.

[DDN00] D. Dolev, C. Dwork and M. Naor: *Nonmalleable Cryptography*, SIAM Journal on Computing, vol. 30(2), pp. 391-437, 2000.

[FFS88] U. Feige,A. Fiat and A. Shamir: *Zero-Knowledge Proofs of Identity*, Journal of Cryptology, vol. 1(2), pp. 77–94, 1988.

[FS86] A. Fiat and A. Shamir: *How to Prove Yourself: Practical Solutions to Identification and Signature Schemes*, Advances in Cryptology — Proceedings Crypto '86, Lecture Notes in Computer Science, vol. 263, Springer-Verlag, pp. 186–194, 1986.

[FS90] A. Fiat and A. Shamir: *Witness Indistinguishable and Witness Hiding Protocols*, Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing (STOC), pp. 416–426, ACM Press, 1990.

[FF00] M. Fischlin and R. Fischlin: *Efficient Non-Malleable Commitment Schemes*, Advances in Cryptology — Proceedings Crypto 2000, Lecture Notes in Computer Science, vol. 1880, pp. 414–432, Springer Verlag, 2000.

[GQ88] L.C. Guillou and J.-J. Quisquater: *A Practical Zero-Knowledge Protocol Fitted to Security Microprocessors Minimizing Both Transmission and Memory*, Advances in Cryptology — Proceedings Eurocrypt '88, Lecture Notes in Computer Science, vol. 330, pp. 123–129, Springer Verlag, 1988.

[GMR88] S. Goldwasser, S. Micali and R.L. Rivest: *A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks*, SIAM Journal of Computing, vol. 17(2), pp. 281–308, 1988.

[H99] S. Halevi: *Efficient Commitment Schemes with Bounded Sender and Unbounded Receiver*, Journal of Cryptology, vol. 12(2), pp. 77–90, 1999.

[NY90] M. Naor and M. Yung: *Universal Oneway Hash Functions and Their Cryptographic Applications*, Proceedings of the 21st Annual ACM Symposium on the Theory of Computing (STOC), pp. 33–43, ACM Press, 1989.

[OS90] H. ONG and C.P. SCHNORR: *Fast Signature Generation with as Fiat-Shamir-Like Scheme*, Advances in Cryptology — Proceedings Eurocrypt '90, Lecture Notes in Computer Science, vol. 473, pp. 432–440, Springer Verlag, 1991.

[OO88] K. OHTA and T. OKAMOTO: *A Modification of the Fiat-Shamir Scheme*, Advances in Cryptology — Proceedings Crypto '88, Lecture Notes in Computer Science, vol. 403, pp. 232–243, Springer Verlag, 1989.

[Ok92] T. OKAMOTO: *Provable Secure and Practical Identification Schemes and Corresponding Signature Schemes*, Advances in Cryptology — Proceedings Crypto '92, Lecture Notes in Computer Science, vol. 740, pp. 31–53, Springer Verlag, 1993.

[PS97] D. POINTCHEVAL and J. STERN: *New Blind Signatures Equivalent to Factorization*, Proceedings of the 4^{th} ACM Conference on Computer and Communications Security (CCS) '97, pp. 92-99, ACM Press, 1997.

[PS00] D. POINTCHEVAL and J. STERN: *Security Arguments for Digital Signatures and Blind Signatures*, Journal of Cryptology, vol. 13(3), pp. 361–396, 2000.

[RSA78] R.L. RIVEST, A. SHAMIR and L. ADLEMAN: *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM, vol. 21, pp. 120–126, 1978.

[S96] C.P. SCHNORR: *Security of 2^t-Root Identification and Signatures*, Advances in Cryptology — Proceedings Crypto '96, Lecture Notes in Computer Science, vol. 1109, pp. 143–156, Springer Verlag, 1996.

[S97] C.P. SCHNORR: *Erratum: Security of 2^t-Root Identification and Signatures*, in Advances in Cryptology — Proceedings Crypto '97, Lecture Notes in Computer Science, vol 1294, page 540, Springer Verlag, 1997.

[Sh99] V. SHOUP: *On the Security of a Practical Identification Scheme*, Journal of Cryptology, vol. 12, pp. 247–260, 1999.

A On Okamoto's Identification Scheme

Okamoto [Ok92] presents witness-indistinguishable identification schemes based on the hardness of discrete log and RSA. In the same paper he also suggests the modified RSA scheme given in Figure 5. Compared to the RSA based scheme the prime RSA exponent is replaced by $2e$ for some prime e.

Okamoto claims that the security is based on the hardness of factoring the modulus N. However, we show that the security de facto relies on the RSA problem rather than on the factoring problem. Namely, we show that computing e-th roots enables an adversary to pass the protocol with probability $\frac{1}{2}$.

Suppose we are given $g \in \mathrm{QR}_N$ and the public key X. Hence, $X \in \mathrm{QR}_N$. Now, pick $x \in_{\mathrm{R}} \mathbb{Z}_{2e}$ and compute

$$r^2 = (Xg^{-x})^{\frac{1}{e}}$$

by solving the RSA problem. Apparently, (x, r) is a representation for $X = g^x r^{2e}$ but we are just aware of x and the square r^2. Nevertheless, we are able to compute $z := y + cx \bmod 2e$ and whenever the challenge c is even then knowledge of r^2 suffices to determine W:

$$W = sr^c g^{\lfloor (y+cx)/2e \rfloor} = s(r^2)^{\frac{c}{2}} g^{\lfloor (y+cx)/2e \rfloor}.$$

Fig. 5. Okamoto's Identification Scheme

Prover $\mathcal{P}$	N, e, g	**Verifier** $\mathcal{V}$
representation x, r of X	$X = g^x r^{2e} \bmod N$	
pick $y \in \mathbb{Z}_{2e}$, $s \in \mathbb{Z}_N^*$ $Y := g^y s^{2e} \bmod N$	$\xrightarrow{\quad Y \quad}$	
	$\xleftarrow{\quad c \quad}$	pick $c \in_{\mathrm{R}} [0, 2e)$
$z := y + cx \bmod (2e)$ $W := s r^c g^{\lfloor (y+cx)/2e \rfloor} \bmod N$	$\xrightarrow{\quad W, z \quad}$	$Y X^c \stackrel{!}{=} W^{2e} g^z$

Thus, solving the RSA problem allows to pass the protocol with probability $\frac{1}{2}$ since the challenge is even with this probability. Whenever $g \notin \mathrm{QR}_N$, one computes $r^4 = (Xg^{-x})^{\frac{2}{e}}$ and succeeds if the challenge satisfies $c = 0 \bmod 4$.

It is tempting to restrict the challenge c to odd values. Still, we are not aware of any proof in this case (e.g., we were unable to modify the proof of Lemma 2 about security against passive adversaries to work in this case).