An Efficient Non-Interactive Statistical Zero-Knowledge Proof System for Quasi-Safe Prime Products

Rosario Gennaro Daniele Micciancio Tal Rabin

January 28, 1998

Abstract

We present efficient zero-knowledge proof systems for quasi-safe prime products and other related languages. Quasi-safe primes are a relaxation of safe primes, a class of prime numbers useful in many cryptographic applications.

Our proof systems achieve higher security and better efficiency than all previously known ones. In particular, all our proof systems are perfect or statistical zero-knowledge, meaning that even a computationally unbounded adversary cannot extract any information from the proofs. Moreover, our proof systems are extremely efficient because they do not use general reductions to NP-complete problems, can be easily parallelized preserving zero-knowledge, and are non-interactive for computationally unbounded provers. The prover can also be efficiently implemented given some trapdoor information and using very little interaction.

We demonstrate the applicability of quasi-safe primes by showing how they can be effectively used in the context of RSA based undeniable signatures to enforce the use of "good" public keys, i.e., keys such that if a signer can convince a recipient of the validity of a signature, then he won't be able to subsequently deny the same signature in case of a dispute.

Keywords: safe primes, zero-knowledge, non-interactive proofs, RSA, undeniable signatures.

1 Introduction

An odd prime $P = 2p+1$ is called *safe* if p is prime. Safe primes are an important class of prime numbers useful in many cryptographic applications. For example, the use of safe primes has been recommended in the choice of RSA moduli because prime products $N = PQ$ are believed to be harder to factor when P and

Q are safe. The undeniable signature scheme in [4] requires the RSA modulus to be the product of safe primes in order to achieve undeniability. [5] uses safe prime products to prove modular polynomial relations in zero-knowledge.

In general, safe prime products have proven particularly useful in cryptographic application due to the special structure of Z^*_{PQ} when P and Q are safe primes. However, restricting P and Q to be safe also raises the following problem: how to prove that $N = PQ$ is the product of two safe primes without giving out the factorization of N. In particular we would like to be able to prove that a number is a safe prime product in zero-knowledge [6]. Certifying a number $N = PQ$ as the product of two safe primes is of critical importance in applications such as the undeniable signature scheme in [4], where if P and Q are not safe the signer can convince somebody of the validity of a signature, and subsequently deny the same signature in case of a dispute.

Although zero-knowledge proofs for the language of safe prime products can be constructed using general results on NP languages [7, 2, 3, 8, 1], these general solutions are not efficient and achieve only computational zero-knowledge (i.e., no computationally bounded adversary can extract information from the proof).

It is therefore both a theoretically interesting and practically important question whether safe prime products have proof systems which are more efficient than those guaranteed by general constructions and achieve stronger notions of security (e.g., perfect or statistical zero-knowledge).

Both questions for safe prime products are still open. In this paper we introduce a relaxation of safe primes, called quasi-safe primes, and show that, for a wide class of quasi-safe prime products, membership can be efficiently proved in statistical zero-knowledge. Namely, an odd prime $P = 2\hat{p} + 1$ is quasi-safe if $\hat{p}$ is a prime power, i.e., $\hat{p} = p^{\alpha}$ for some prime p. We give an efficient one-sided error non-interactive statistical zero-knowledge (NIZK [2]) proof system for the language of quasi-safe prime products $N = PQ$ such that $P = 2p^{\alpha} + 1, Q = 2q^{\beta} + 1, p, q$ are distinct odd primes satisfying $P, Q, p, q \not\equiv 1 \pmod 8$, $P \not\equiv Q \pmod 8$ and $p \not\equiv q \pmod 8$. Quasi-safe primes have in common enough properties with safe primes, to be useful in many applications designed to work with safe primes.

The rest of the paper is organized as follows. In section 2 we introduce some basic definitions and notation regarding number theory and zero-knowledge proof systems. Our proof system for quasi-safe prime products is presented in section 3 where we assume the prover is computationally unbounded. In section 4 we show how the prover can be efficiently implemented using some trapdoor information. Finally in section 5 we demonstrate the applicability of quasi-safe primes by discussing their use in the context of RSA-based undeniable signature.

2 Definitions

In this section we review some definition and notation from number theory and zero-knowledge proof systems that will be use in the rest of the paper.

2.1 Number Theory

Let a, b, n be integers. We say that n *divides* a (written $n|a$) if $a = nb$ for some b, a and b are *congruent* modulo n (written $a \equiv b \pmod{n}$) if n divides $a - b$. A *safe prime* is an odd prime $P = 2p+1$ such that p is an odd prime. A *quasi-safe prime* is an odd prime $P = 2\hat{p} + 1$ such that $\hat{p} = p^\alpha$ is an odd prime power. A number N is *square free* if m^2 does not divide N for any $m > 1$. For any N, let $\mathrm{odd}(N)$ be the greatest odd number m such that m divides N. Let P and Q be two primes. We say that P and Q are *disjoint* if $\mathrm{odd}(\gcd(P-1, Q-1)) = 1$, i.e., $P - 1$ and $Q - 1$ have no odd common factors.

A number N is a *quasi-safe prime product* if $N = PQ$ where $P = 2p^\alpha+1$, $Q = 2q^\beta + 1$, p and q are distinct odd primes. For technical reasons we will restrict our attention to quasi-safe prime products such that $P, Q, p, q \not\equiv 1 \pmod{8}$, $P \not\equiv Q \pmod{8}$ and $p \not\equiv q \pmod{8}$.

2.2 Languages

In section 3 we will give proof systems for various combinations of the following languages:

- ODD: The set of odd numbers. ODD can be easily decided in the obvious way.
- FP (fermat primes): The set of prime numbers of the form $2^k + 1$. FP is also easily decidable due to the following fact: $n = 2^k + 1$ is prime iff k is a power of two and either $n = 5$ or $5^{2^{k/2}} \equiv -1 \pmod{n}$.
- SF (square free): The set of all square free integers.
- SF': The same as SF with the additional requirement that for any two primes P, Q dividing N, it must be $P \not| (Q - 1)$.
- PPP (prime power product): The set of all N with at most two distinct odd prime factors.
- PPP': The same as PPP with the additional requirement that the two odd prime factors P, Q satisfy $P, Q \not\equiv 1 \pmod{8}$ and $P \not\equiv Q \pmod{8}$.
- PP (prime product): The set of all N which are the product of at most two distinct primes. Notice that $\mathsf{PP} = \mathsf{SF} \cap \mathsf{PPP}$.

- DPP (disjoint prime product): The set of all $N = PQ$ which are the product of two disjoint primes, i.e., $P-1$ and $Q-1$ have no odd common factors.
- ASPP (almost safe prime product): The set of all $N = PQ$ such that $P = 2^a p^\alpha + 1, Q = 2^b q^\beta + 1, p, q$ are distinct primes satisfying $p, q \not\equiv 1 \pmod 8$ and $p \not\equiv q \pmod 8$.
- QSPP (quasi safe prime product): The set of all $N = PQ$ such that $P = 2p^\alpha + 1, Q = 2q^\beta + 1, p, q$ are distinct primes satisfying $P, Q, p, q \not\equiv 1 \pmod 8$ and $P \not\equiv Q \pmod 8$ and $p \not\equiv q \pmod 8$.

2.3 Non-Interactive Zero-Knowledge

In this section we briefly review the definition of *non-interactive zero-knowledge proof system.* The reader is referred to [1] for a complete description.

In a non-interactive proof system, a prover wants to convince a verifier that a string x (the common input) belongs to some language L. The interaction between the prover and the verifier is minimal: both the prover and the verifier have access, in addition to the common input x, to a common random string r which can be thought as provided by a trusted third party. On input x and r, the prover $P(x,r)$ computes a string π (a purported proof of membership of x in L with respect to random string r). Subsequently, the verifier $V(x,r,\pi)$ either accepts or rejects the string π produced by the prover as a valid proof of $x \in L$ with respect to the same random string r. (P,V) is a *non-interactive proof system* for a language L if V is polynomial time and the following two conditions hold

- Completeness: For every $x \in L$, $\Pr\{V(x,r,P(x,r)) = 1\} > 1 - \delta$
- Soundness: For every $x \notin L$, $\Pr\{\exists \pi . V(x,r,\pi) = 1\} < \epsilon$

where the probabilities are computed with respect to the choice of the common random string r and the coin tosses of algorithm P (V is a deterministic machine.)

The values δ and ϵ are called the completeness and soundness error and are usually set to $1/3$. The proof system is said *one-sided error* if the completeness error is zero, i.e., $V(x,r,P(x,r))$ for all $x \in L$ and all r. In the soundness condition π is often restricted to strings computable (not necessarily in polynomial time) from x and R.

The random string r is usually chosen uniformly at random from the set $\{0,1\}^{p(|x|)}$ of all string of some fixed length $p(|x|)$ polynomial in the size of the common input. We will consider a variant to the model in which the random string r is chosen from a set $R_x \subseteq \{0,1\}^*$ which may depend on the string x. It is easy to see that if R_x is efficiently samplable, the two models are equivalent (indeed given random input r uniformly distributed in $\{0,1\}^{p(|x|)}$, to generate a string r' according to distribution R_x, just run the sampling algorithm with coin tosses r.)

A non-interactive proof system (P, V) is *perfect zero-knowledge* if there exists a probabilistic polynomial time algorithm S (the simulator) such that for all $x \in L$, the output of $S(x)$ is distributed identically to $(R_x, P(x, R_x))$. It is *statistical zero-knowledge* if the statistical difference between $S(x)$ and $(R_x, P(x, R_x))$ is negligible in $|x|$, i.e., for all polynomial $p(\cdot)$ and for all long enough x,

$$\sum_{r \in R_x} \sum_{\pi \in \{0,1\}^*} |\Pr\{S(x) = (r, \pi)\} - \Pr\{R_x = r\} \Pr\{P(x, r) = \pi\}| < \frac{1}{p(|x|)}.$$

Let $L \subseteq L'$ be two languages. We say that (P, V) is a non-interactive zero-knowledge proof system for L in L', if the soundness condition is required to hold only for those x in $L' \setminus L$. This correspond to having already proved that $x \in L'$ and using the proof system (P, V) to prove some additional property about x (namely $x \in L$).

3 Proving Quasi-Safe Prime Products

We give a non-interactive one-sided error statistical zero-knowledge proof system for the language QSPP of quasi-safe prime products.

We break the proof system in several stages for clarity of exposition and because some stages could be of independent interest by themselves. Moreover, all but the last stage we are going to describe are *perfect* zero-knowledge, and only the last stage is actually *statistical* zero-knowledge.

Notice however that since all stages are non-interactive they can be composed in parallel resulting in a single non-interactive statistical zero-knowledge proof system.

For clarity of exposition, we will first assume that the prover is computationally unbounded as it is customary in statistical zero-knowledge proofs. In a later section we will show how the prover can be efficiently implemented using some trapdoor information and any commitment scheme.

3.1 Stage 1: Square Free

First of all, we give a proof system for the language SF' of all square free N such that for any distinct primes P, Q dividing N, it holds $P \not| (Q-1)$.

Theorem 1 *The non-interactive proof system defined by*

- Common Input: N
- Random Input: $x \in Z_N^*$
- Prover: *compute* $M = N^{-1} \bmod \phi(N)$ *and output* $y = x^M \bmod N$

- Verifier: *accept iff* $y^N = x \bmod N$.

is one-sided error perfect zero-knowledge with soundness error $1/d$ *for the language* SF', *where* d *is the smallest factor of* N.

Proof: We have to prove that the above proof system is complete, sound and zero-knowledge.

- **Completeness:** If $N \in \mathsf{SF}'$ then $\gcd(N, \phi(N)) = 1$ and N has a multiplicative inverse M modulo $\phi(N)$. Therefore $y^N \equiv (x^M)^N \equiv x^{MN} \equiv x^1 \pmod N$ and the verifier always accepts.
- **Soundness:** If $N \notin \mathsf{SF}'$ then $\gcd(N, \phi(N)) = d > 1$ and $|\{x^N \mid x \in Z_N^*\}| = |Z_N^*|/d$. Therefore given a random $x \in Z_N^*$ the probability that $x \equiv y^N \pmod N$ for some y is at most $1/d$.
- **Zero-Knowledge:** The simulator chooses $y \in_R Z_N^*$ and outputs $(y^N \bmod N, y)$. Notice that if $N \in \mathsf{SF}'$ then $\gcd(N, \phi(N)) = 1$ and $x = y^N \bmod N$ is uniformly distributed over Z_N^*.

3.2 Stage 2: Prime Power Product

We give a proof system for the language PPP' of all prime power products $N = P^\alpha Q^\beta$ such that $P, Q \not\equiv 1 \pmod 8$ and $P \not\equiv Q \pmod 8$.

Theorem 2 *The non-interactive proof system defined by*

- Common Input: $N \in \mathsf{ODD}$
- Random Input: $x \in Z_N^*$
- Prover: *output a square root* r *modulo* N *of one of* $\pm x, \pm 2x$.
- Verifier: *accept iff* r^2 *is congruent to* $\pm x$ *or* $\pm 2x$ *modulo* N

is one-sided error perfect zero-knowledge with soundness error $1/2$, *for the language* PPP'.

Proof: We have to prove that the above proof system is complete, sound and zero-knowledge.

- **Completeness:** Assume $N \in \mathsf{PPP}'$, i.e., $N = P^\alpha Q^\beta$ for primes P and Q such that $P, Q \not\equiv 1 \pmod 8$ and $P \not\equiv Q \pmod 8$. Notice that -1 is a square modulo P (resp. Q) iff $P \equiv 1 \pmod 4$ (resp. $Q \equiv 1 \pmod 4$) and 2 is a square modulo P (resp. Q) iff $P \equiv \pm 1 \pmod 8$ (resp. $Q \equiv \pm 1 \pmod 8$). It follows that for any $x \in Z_N^*$ one and only one of $\pm x, \pm 2x$ is a square modulo PQ. Therefore the prover can always extract the square root modulo $N = P^\alpha Q^\beta$ of one of $\pm x, \pm 2x$ and the verifier always accepts.

- **Soundness:** If N is not in PPP' then no prover can convince the verifier with probability better than $1/2$. We show this by cases: if $N \notin \mathsf{PPP}$, then N has more than two odd prime factors and the probability that a random $x \in Z_N^*$ is a square is at most $1/8$. Therefore with probability at least $1/2$ none of $\pm x$ and $\pm 2x$ has a square root modulo N and the verifier will reject with probability $1/2$. If $N \in \mathsf{PPP}$, but $N \notin \mathsf{PPP}'$ then N has at most two prime factors P and Q, but either $P \equiv 1 \pmod 8$, $Q \equiv 1 \pmod 8$ or $P \equiv Q \pmod 8$. Then either $-1, 2$ or -2 is a square and the probability that for a random x the set $\{x, -x, 2x, -2x\}$ contains a square is $1/2$ ($1/4$ if both $P \equiv Q \equiv 1 \pmod 8$).

- **Zero-Knowledge:** The simulator chooses $r \in_R Z_N^*$ and $\alpha \in_R \{\pm 1, \pm 2\}$ at random, computes $x = r^2/\alpha \bmod N$ and outputs (x, r).

3.3 Stage 3: Disjoint Prime Product

Assume we already proved that $N \in \mathsf{PP} = \mathsf{SF} \cap \mathsf{PPP}$, i.e., N is the product of at most two primes both with exponent 1. We want to prove that $N \in \mathsf{DPP}$, i.e., $N = PQ$ is the product of exactly two primes, and these two primes P and Q are disjoint.

Theorem 3 *The non-interactive proof system defined by*

- Common Input: $N \in \mathsf{PP}$

- Random input: $x \in Z_N^*$

- Prover: *output* $y = x^M \bmod N$*, where* M *is the inverse of* $odd(N-1)^{-1}$ *modulo* $\phi(N)$*.*

- Verifier: *check that* $N \notin \mathsf{FP}$ *and if so accept iff* $y^{odd(N-1)} = x \bmod N$*.*

is one-sided error perfect zero-knowledge with soundness error $1/d$ *(*d *the smallest factor of* $odd(N-1)$*) for the language* DPP *in* PP*.*

Proof: Assume $N \in \mathsf{PP}$, i.e., N is either a prime or the product of two distinct primes PQ. Notice that if N is a prime, $\gcd(\mathrm{odd}(N-1), \phi(N)) = \mathrm{odd}(N-1) = 1$ iff $N \in \mathsf{FP}$, but such a possibility is ruled out by the deterministic test the verifier carries on N (see Section 2.2). If $N = PQ$ is the product of two distinct primes $\gcd(\mathrm{odd}(N-1), \phi(N)) = 1$ iff $N \in \mathsf{DPP}$. Therefore assumed $N \in \mathsf{PP} - \mathsf{FP}$, $N \in \mathsf{DPP}$ iff $\gcd(\mathrm{odd}(N-1), \phi(N)) = 1$.

- **Completeness:** If $\gcd(\mathrm{odd}(N-1), \phi(N)) = 1$, then $\mathrm{odd}(N-1)$ has a multiplicative inverse M modulo $\phi(N)$ and the verifier always accepts.

- **Soundness:** If $\gcd(\mathrm{odd}(N-1), \phi(N)) = d > 1$, then given a random $x \in Z_N^*$ the probability that $x \equiv y^{\mathrm{odd}(N-1)} \pmod N$ for some $y \in Z_N^*$ is at most $1/d$.
- **Zero-Knowledge:** The simulator chooses $y \in_R Z_N^*$, computes $x = y^{\mathrm{odd}(N-1)} \bmod N$ and outputs (x, y).

3.4 Stage 4: Almost Safe Prime Product

Assume N is in DPP, i.e., $N = PQ$ where P and Q are two odd primes such that $\mathrm{odd}(\gcd(P-1, Q-1)) = 1$. We want to prove that $N \in \mathsf{ASPP}$, i.e., $\hat{p} = \mathrm{odd}(P-1) = p^\alpha$ and $\hat{q} = \mathrm{odd}(Q-1) = q^\beta$ are two prime powers such that $p, q \not\equiv 1 \pmod 8$ and $p \not\equiv q \pmod 8$.

Theorem 4 *The non-interactive proof system defined by*

- Common Input: $N \in \mathsf{DPP}$
- Random Input: $g \in Z_N^*, y \in \langle g \rangle$
- Prover: *computes* $x = \log_g y$ *and outputs a square root* $r \bmod \mathit{odd}(\phi(N))$ *of one of* $\pm x, \pm x/2$.
- Verifier: *let* $\gamma = 2^{|N|}$ *and accept iff* $y^\gamma \bmod N$ *is equal to* $g^{\pm\gamma r^2} \bmod N$ *or* $g^{\pm 2\gamma r^2} \bmod N$.

is one-sided error statistical zero-knowledge for ASPP *in* DPP *with soundness error* $9/10$.

Proof:

- **Completeness:** If $N \in \mathsf{ASPP}$, then $\mathrm{odd}(\phi(N)) = p^\alpha q^\beta$ has exactly two prime factors such that $p, q \not\equiv 1 \pmod 8$ and $p \not\equiv q \pmod 8$. Therefore at least one of $\pm x$ and $\pm x/2$ is a square modulo $\mathrm{odd}(\phi(N))$. Let r^2 be congruent to $\pm x$ or $\pm x/2$ modulo $\mathrm{odd}(\phi(N))$. Then γx will be congruent to $\pm\gamma r^2$ or $\pm 2\gamma r^2$ modulo $\gamma \mathrm{odd}(\phi(N))$. Since $\phi(N)/\mathrm{odd}(\phi(N))$ divides 2^γ, the last congruence holds also modulo $\phi(N)$, and therefore $y^\gamma = g^{\gamma x}$ is congruent to $g^{\pm\gamma r^2}$ or $g^{\pm 2\gamma r^2}$ modulo N and the verifier always accepts.
- **Soundness:** Assume $N \in \mathsf{DPP}$ but $N \notin \mathsf{ASPP}$, i.e., $N = PQ$ and $\gcd(\hat{p}, \hat{q}) = 1$ (where $\hat{p} = \mathrm{odd}(P-1)$, $\hat{q} = \mathrm{odd}(Q-1)$), but either $\mathrm{odd}(\phi(N)) = \hat{p}\hat{q}$ has more then two odd prime factors, or it has only two prime factors p and q, but either $p \equiv 1 \pmod 8$ or $q \equiv 1 \pmod 8$ or $p \equiv q \pmod 8$.

If $\phi(N)$ has more than 2 odd prime factors, with probability at least $(2/3)(4/5)(6/7) = 16/35$ the order of g has at least 3 odd prime factors and the probability that x is invertible modulo these three factors is also at least $16/35$. Therefore with probability at least $(16/35)^2/2 > 1/10$ none of $\pm x$ or $\pm x/2$ is a square modulo the order of g. Now assume $\text{odd}(\phi(N)) = p^\alpha q^\beta$ but either $p \equiv 1 \pmod 8$, $q \equiv 1 \pmod 8$ or $p \equiv q \pmod 8$. Then either $-1, 1/2$ or $-1/2$ is a square modulo pq. The probability that the order of g is divided by pq, and that x is invertible modulo pq are both at least $(2/3)(4/5) = 8/15$. Consequently, with probability at least $(8/15)^2/2 > 1/10$ none of $\pm x$ or $\pm x/2$ will be a square modulo the order of g.

- **Zero-Knowledge:** The simulator chooses $g \in Z_N$, $r \in \{0, \ldots, N^2\}$ and $\alpha \in \{1, -1, 2, -2\}$ at random, computes $y = g^{\alpha r^2} \bmod N$ and outputs (g, y, r). Notice that $r \bmod \phi(N)$ is statistically close to uniform and therefore the value y computed by the simulator is distributed almost uniformly in $\langle g \rangle$.

3.5 Combining All Stages: QSPP

It is easy to see that in order to get a NIZK proof system for QSPP it is enough to run the four stages outlined before. The four stages can be run simultaneously.

If the desired soundness error probability is ϵ, each stage must be repeated enough times to make the error probability of each stage smaller than $\epsilon/4$. This can be efficiently done in parallel given the parallel composition properties of NIZK proofs.

The soundness error can be easily bounded in stages 2 and 4 where the error probability is a constant. In stage 4, the soundness error can be reduced if we assume $p, q > d$ for some constant d. In particular, for $d = 3$ this can be easily checked by the verifier as follows. Since $\gcd(\hat{p}, \hat{q}) = 1$, $p, q \neq 3$ iff $N \equiv 1 \pmod 3$. This will reduce the soundness error in stage 4 to $1 - (1/2)((4/5)(6/7)(10/11))^2 \approx 4/5$. More complicate techniques can be used to lower the soundness error to almost $1/2$.

In step 1 and 3 instead the soundness error is bounded by $1/d$ where d is the smallest factor of N and $\text{odd}(N-1)$ respectively. Therefore, if we restrict P, Q, p and q to be large enough, say not smaller then 1024, (which can be deterministically checked by the verifier by trial division), the soundness error will be 2^{-10}, and the error probability can be lowered to 2^{-100} by only ten parallel executions of stages 1 and 3. Notice also that the two stages can be combined in a single proof system where the prover given a random $y \in Z_N^*$ outputs a x such that $x^{\text{odd}(N \cdot (N-1))} = y \pmod N$.

4 Implementing the Prover

In the previous section we assumed a computationally unbounded prover and the availability of a common random input.

We now show how the selection of the random input and the operations of the prover can be efficiently implemented using any commitment scheme and some trapdoor information.

In stage 1 and 3 the prover needs to compute the inverses modulo $\phi(N)$ of N and $\mathrm{odd}(N-1)$. This inverses can either be given as trapdoor information, or can be computed directly by the prover given $\phi(N)$ or the factorization of N.

In stages 2 and 4 the prover needs to compute square roots modulo N and $\phi(N)$. This can be efficiently done if the prover knows the factorization of N and $\phi(N)$. In particular, if $N = PQ$ is a quasi-safe prime product and the prover knows P and Q, square roots can be efficiently be computed.

We still need to show how to randomly sample Z_N^* or the subgroup generated by some element $g \in Z_N^*$ in such a way that the prover can compute discrete logarithms base g of the samples.

It is important to notice that in order to implement this step with a polynomial time prover, we do need interaction between the prover and the verifier. We stress that such interaction is not necessary when the prover is unbounded.

4.1 Sampling Z_N^*

The prover and the verifier want to choose a z is such a way that if either of them is honest, the result is uniformly distributed in Z_N^*.

1. The verifier randomly chooses $x \in Z_N$ and send $\mathrm{commit}(x)$ to the prover
2. The prover randomly chooses $y \in Z_N$ and send it to the verifier
3. The verifier decommits x.
4. Output: Let $z = x + y \bmod N$. If $z \in Z_N^*$ output z, otherwise output $\perp$.

If either the prover or the verifier is honest, the conditional distribution of the above protocol, given $z \neq \perp$, is uniform over Z_N^*.

4.2 Sampling $\{(g, g^i) \mid g \in Z_N^*, i \in Z_{o(g)}\}$

In stage 4, the random input is a pair g, y where $g \in Z_N^*$ and $y \in \langle g \rangle$ are uniformly distributed. Moreover, the prover needs to know the discrete logarithm base g of y.

The base g can be randomly chosen from Z_N^* using the protocol in the previous section. However, since stage 4 is zero-knowledge for any value of $g \in Z_N^*$, we can more simply let the verifier choose g at random (there is no need to protect the prover ¿from the verifier choosing g not at random). Ideally the verifier would like g to be a generator of the order $(P-1)(Q-1)/2$ subgroup of Z_N^*, but not knowing how to find such a generator, the verifier can still achieve a bounded soundness error by choosing g at random.

Once g has been chosen, the following protocol can be used to select an element of $\langle g \rangle$ in such a way that the prover knows its discrete logarithm.

1. The verifier randomly chooses $x \in Z_{N^2}$ and send commit(x) to the prover
2. The prover randomly chooses $y \in Z_{N^2}$ and send $g^y \pmod N$ to the verifier
3. The verifier decommits x.
4. Output: $z = g^{x+y} \pmod{N-1}$

Notice that at the end of the interaction the prover knows $\log_g z = x + y$. The above protocol does not guarantee that $z \in \langle g \rangle$ if the prover is cheating. However, in our proof system the prover has only to loose by not following the above protocol because if $x \notin \langle g \rangle$ the verifier will certainly reject the input in stage 4 of the proof system.

5 Applications: Undeniable Signatures

In [4] a new undeniable signature scheme was presented based on the RSA signature algorithm. The scheme is proven secure under the assumption that the composite modulus N is the product of two safe primes P and Q.

This assumption is needed in order to bound the probability of cheating by the prover during the confirmation and denial protocol. While in general one trusts the prover to choose a "safe" N as it appears to be in his own interest, the undeniable signature scheme of [4] is an example of a scheme where the prover might have an interest in choosing N differently as that could give him the possibility to deny valid signatures. A problem left open in [4] was how to make sure that the signer had chosen N correctly[1].

In [4] the proven bound on the probability of cheating is $\frac{O(1)}{p}$ where $N = PQ$ and $P = 2p+1$ and $Q = 2q+1$, $p < q$.

It turns out that one can relax the condition on P and Q to be quasi-safe primes, and the proof still carries through[2]. That is the probability of cheating for the prover remains $\frac{O(1)}{p}$ where $N = PQ$ and $P = 2p^\alpha + 1$, $Q = 2q^\beta + 1$, $p < q$.

[1] In [4] two partial solutions are presented but neither of them is fully satisfactory either in security or in efficiency

[2] This is an easy exercise for the reader, the final paper will present the full details of this proof

However in [4] one could say that $|p| = O(|N|)$ and from that establish that the probability of error was negligible. What can we say in the case of quasi-safe primes?

The point is to notice that if the prover chooses p so that the cheating probability is non-negligible, i.e. a small prime, then he is also allowing an adversary to easily factor N. Indeed if $\frac{O(1)}{p}$ is non-negligible in a security parameter k then p is polynomially large in k, which means that one could guess p among a polynomial set of candidates and then test if $2p^\alpha + 1$ divides N for all $\alpha < \lg N$.

Thus effectively quasi-safe primes are sufficient for the security of the [4] undeniable signature scheme.

References

[1] M. Blum, A. De Santis, S. Micali, G. Persiano, "Non-Interactive Zero-Knowledge," in *SIAM J. Comput.*, Vol.20, No.6, pp.1084–1118.

[2] M. Blum, P. Feldman, S. Micali, "Noninteractive Zero-Knowledge and its Applications," *Proc. of 20th ACM Symposium on Theory of Computing,* 1988, pp. 103-112.

[3] A. De Santis, S. Micali, G. Persiano, "Non-Interactive Zero-Knowledge Proof Systems," in *Advances in Cryptology - CRYPTO87,* Lecture Notes in Computer Science 293, Springer-Verlag, 1987, pp. 52-72.

[4] R. Gennaro, H. Krawczyk, T. Rabin, "RSA-Based Undeniable Signature," in *Advances in Cryptology - CRYPTO97,* Lecture Notes in Computer Science 1294, Springer-Verlag, 1997, pp. 132-149. Final version available from http://www.research.ibm.com/security/papers1997.html

[5] E. Fujisaki, T. Okamoto, "Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations," in *Advances in Cryptology - CRYPTO97,* Lecture Notes in Computer Science 1294, Springer-Verlag, 1997, pp. 16-30.

[6] S. Goldwasser, S. Micali, C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems," *SIAM J. Comput.* Vol. 18, No. 1, pp. 186-208, February 1989.

[7] O. Goldreich, S. Micali, A. Wigderson, "Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems," *Journal of ACM,* Vol. 38, No. 1, July 1991, pp. 691-729.

[8] U. Feige, D. Lapidot, A. Shamir, "Multiple Non-Interactive Zero Knowledge Proofs Under General Assumptions," *Proc. of 31st Symp. on Foundations of Computer Science,* 1990, pp. 308-317.