J. Cryptology (1992) 5: 29–39

Journal of Cryptology

An Interactive Identification Scheme Based on Discrete Logarithms and Factoring[1]

Ernest F. Brickell and Kevin S. McCurley
Sandia National Laboratories, Albuquerque, NM 87185, U.S.A.

Abstract. We describe a modification of an interactive identification scheme of Schnorr intended for use by smart cards. Schnorr's original scheme had its security based on the difficulty of computing discrete logarithms in a subgroup of GF(p) given some side information. We prove that our modification will be witness hiding, which is a more rigid security condition than Schnorr proved for his scheme, if factoring a large integer with some side information is computationally infeasible. In addition, even if the large integer can be factored, then our scheme is still as secure as Schnorr's scheme. For this enhanced security we require only slightly more communication and about a factor of a 3.6 increase in computational power, but the requirements remain quite modest, so that the scheme is well suited for use in smart cards.

Key words. Interactive identification, Digital signatures, Witness hiding.

1. Introduction

In this paper we describe an interactive identification scheme that is a variation of a scheme presented by Schnorr at Crypto '89 [17]. Schnorr's scheme has several features that make it advantageous for use in smart cards or other environments with limited computing power. Its security, more specifically, the soundness of the protocol, is based on the difficulty of the discrete logarithm problem in a subgroup of $\mathbb{Z}_p^*$.

Due to the current state of complexity theory, cryptographic schemes whose security is based on the difficulty of solving a specific computational problem are exposed to the danger that a fast algorithm may be found for the underlying computational problem. It therefore seems desirable to design systems with the property that breaking them requires the ability to solve two apparently dissimilar computational problems, both of which appear to be hard. An example of such a scheme was given in [13], where a key distribution scheme with this property was

[1] Date received: December 20, 1990. Date revised: September 27, 1991. A preliminary version of this paper was presented at Eurocrypt '90, May 21–24, Århus, Denmark, and has appeared in the proceedings, pp. 63–71. This work was performed under U.S. Department of Energy contract number DE-AC04-76DP00789.

given. The key distribution scheme of [13] uses arithmetic modulo a number n that is a product of two primes. Breaking the system requires the factorization of n and the ability to solve the Diffie–Hellman problem modulo the prime factors of n. In the present paper we take a slightly different tack, by using arithmetic modulo a prime p. We choose p with the property that $p-1$ has at least two large prime factors, so that the factorization of $p-1$ is hard to recover. We then construct the system in such a way that breaking it requires both factoring $p-1$ and computing a discrete logarithm in a subgroup of $\mathbb{Z}_p^*$.

The extra security gained in this scheme extracts a penalty both in the computation time and the communication time, but the scheme still carries the advantage of allowing preprocessing of most of the computation, and should still be quite feasible for use in smart cards. The relative merits of the schemes will be discussed later, after we first present the schemes in detail.

The scheme that we present in this paper is simpler than the one presented in our Eurocrypt '90 paper [2]. There are also changes in what we can prove about the security. We can prove that the new scheme is witness hiding if factoring $p-1$ is hard. For the scheme in [2] and for Schnorr's scheme [17], nothing has been proved about witness hiding. Also, the scheme in [2] is sound if either factoring $p-1$ or computing a discrete logarithm in a subgroup of $\mathbb{Z}_p^*$ is hard. However, the new scheme is sound only if computing a discrete logarithm in a subgroup of $\mathbb{Z}_p^*$ is hard.

2. Schnorr's Identification Scheme

We begin by describing the original Schnorr authentication scheme in terms of a security parameter t. In this scheme, each person who wishes to use the scheme to prove his identity will visit a key authentication center (KAC) and register his or her public key. When the KAC is originally set up, it chooses

primes p and q such that $q|p-1$, $q \geq 2^{140}$, and $p \geq 2^{512}$,
α of order q in the group $\mathbb{Z}_p^*$,
its own private and public keys for a signature scheme.

The KAC publishes p, q, α, and its public key. When a user comes to the KAC for registration, the user chooses a secret $s \in \{1, \ldots, q\}$, computes $v \equiv \alpha^{-s} \pmod{p}$, and submits v to the KAC along with some form of identification. The KAC verifies the user's identity, generates an identification string I, and also generates a signature $\mathcal{S}$ of the pair (I, v). The KAC can use any secure digital signature scheme whatsoever for generating this signature.

We now describe the procedure by which party P (the prover) can prove its identity to V (the verifier). In a preprocessing phase, P should first have chosen a random number $r \in \{1, \ldots, q\}$ and computed $x \equiv \alpha^r \pmod{p}$. In the identification procedure, P first sends to V its identification string I, its public key v, the KAC's signature $\mathcal{S}$ of (I, v), and x. V then checks the validity of P's public key by verifying the signature $\mathcal{S}$, chooses a random $e \in \{1, \ldots, 2^t\}$, and transmits e to P. P sends to V the value $y := r + se \pmod{q}$. Finally, V checks that $x \equiv \alpha^y v^e \pmod{p}$ and accepts P's proof of identity if this holds.

Schnorr suggests using $t = 72$, although this can be reduced substantially for use in the identification scheme (Schnorr also proposed a companion signature scheme

which requires the larger t). The parameter t is used to control the probability that an impostor will be able to guess a correct response to a challenge e. For use in an identification scheme, we need only choose t so large that the probability 2^{-t} of guessing the challenge e is negligible.

This scheme has a number of novel features. First of all, much of the arithmetic to be done by the prover can be done in a preprocessing phase, using idle time of the processor. This is well suited to the case of a smart card, where the processing power is relatively small. Second, the number of bits that must be communicated is considerably reduced over other schemes such as RSA or Fiat–Shamir. There is also a signature scheme based on the same choice of keys, but we shall not discuss it here in great detail.

Schnorr's scheme may be regarded as a practical refinement of the zero-knowledge protocols of Chaum *et al.* [4] and [3] for demonstrating possession of a discrete logarithm. In [4], the challenge e was either a zero or a one, and the basic protocol was repeated several times (requiring the prover to perform multiple exponentiations). Yet another interesting identification scheme based on discrete logarithms was proposed by Beth [1]. The security of the latter scheme is however more closely related to the ElGamal signature scheme.

3. The Modified Scheme

In this section we shall describe the modification of Schnorr's scheme. The essential differences are that s is chosen and y is computed modulo $p-1$ rather than modulo q, and that q is secret. Rather than the single security parameter t, we describe the scheme in terms of the parameters k and t, with $t < k$. The KAC is used in the same manner as before. In the set-up phase, the KAC chooses primes p, q, and w such that $qw \mid p-1$, $q^2 \nmid p-1$, $q, w \geq 2^k$, and $qw \geq 2^{512}$. The KAC also chooses α of order q in the group $\mathbb{Z}_p^*$. The KAC publishes p, α, and its public key, but not q or w.

When a user wishes to join the system, he chooses a random number $s \in \{1, \ldots, p-1\}$. The user then computes $v \equiv \alpha^{-s} \pmod{p}$, and presents v to the KAC along with some form of identification, but keeps s secret. The KAC verifies the user's identity, checks that $v^q \equiv 1 \pmod{p}$, generates an identification string I, and produces a signature $\mathcal{S}$ of the pair (I, v), which it provides to the user. Once again the KAC can use any digital signature scheme whatsoever.

In the identification procedure, P once again has a preprocessing phase, where P chooses from the uniform distribution a random number $r \in \{1, \ldots, p-1\}$ and computes $x \equiv \alpha^r \pmod{p}$. Then P sends to V the identification string I, its public key v, the KAC's signature $\mathcal{S}$, and x. V checks the authenticity of P's public key by verifying the signature $\mathcal{S}$ of (I, v). If the keys are authentic, then V chooses a random $e \in \{1, \ldots, 2^t\}$ and transmits e to P. P then computes an integer y such that $y \equiv r + se \pmod{p-1}$ and sends y to V. V checks that $x \equiv \alpha^y v^e \pmod{p}$ and accepts P's proof of identity if this condition is satisfied.

The parameter t can be adjusted to suit specific needs, but we suggest using $t = 40$. With this choice, there are 2^{40} possible challenges e, and the probability of guessing the challenge ahead of time is therefore 2^{-40}.

Some care should be exercised in choosing the primes q and w, and in particular

we should try to choose them in such a way as to thwart any known algorithms for factoring qw. The choice of $k \approx 140$ is probably marginal in avoiding a determined implementation of the elliptic curve method of H. W. Lenstra, Jr., but may suffice for applications of a commercial nature. At present the record for the largest factor found by the elliptic curve method has 38 decimal digits, or about 127 binary digits (this factor was found by Robert Silverman). On the other hand, choosing $k > 200$ will probably be safe against any conceivable implementation, and in any case the modified scheme imposes no performance penalty for choosing q larger, since all arithmetic is done modulo p or $p - 1$ anyway. The construction of p should be relatively easy, since heuristic evidence (see [19]) suggests that we should expect a prime $p \equiv 1 \pmod{qw}$ can be found with $p \leq qw \log^2(qw)$.

The recent results of Lenstra and Manasse [11] and Lenstra *et al.* [12] have raised a question about how long a 512-bit modulus will remain safe from attack by current factorization methods. We suspect, however, that by the time anyone will have at their disposal enough computational power to factor a general 512-bit modulus, the smart card technology will probably have advanced enough to allow easy use of a 1024-bit modulus. Moreover, the best known attack for breaking the scheme we present here requires in addition the computation of a discrete logarithm modulo a 512-bit prime, and current algorithms will probably have a much more difficult time with this problem.

4. Performance Analysis of the Modified Scheme

It is evident that the modified scheme suffers from a slight disadvantage in the number of bits that must be communicated. The following tables show the number of bits to be communicated in the two schemes, using the security parameters mentioned above. For the sake of comparison, we have assumed that 100 bits suffice for each of I and $\mathcal{S}$. We have used a value of $k = 140$ in the original scheme.

Original scheme	
I	100
v	512
$\mathcal{S}$	100
x	512
e	40
y	140
Total bits	1404

Modified scheme	
I	100
v	512
$\mathcal{S}$	100
x	512
e	40
y	512
Total bits	1776

The modified scheme therefore pays a penalty of an extra 372 bits in communication, and possibly more if error correction is included. On the other hand, this is still well within the realm of possiblity using present technology.

We now compare the computational requirements of the two schemes. We first consider the off-line computation, where in both schemes the prover computes $\alpha^r \pmod{p}$. In the Schnorr scheme, r is chosen uniformly between 1 and q, while in our

scheme, r is chosen uniformly between 1 and $p-1$. Hence our scheme requires about $\log_2 p/\log_2 q$ more off-line computation. In the real time computation, the prover is required to compute $y \equiv r + se \pmod{q}$ in the Schnorr scheme, and $y \equiv r + se \pmod{p-1}$ in our scheme. Using standard algorithms [9, Section 4.3.1], Schnorr's scheme would use about $\log_2 q + ct \log_2 q$ bit operations, whereas the modified scheme takes about $\log_2 p + ct \log_2 p$ bit operations. Hence both the on-line and off-line portions of the computation require about a factor of $\log_2 p/\log_2 q$ more bit operations. For the parameters that were suggested, this is about a factor of $512/140 \approx 3.6$ more computation, but the on-line portion of the computation is still considerably less than in the Fiat–Shamir scheme.

So far we have only discussed the computational requirements of the prover, for which the new scheme shifts much of the burden to a preprocessing stage. We should point out that the computational requirements of the verifier are significantly greater for our scheme than for the Fiat–Shamir scheme, because in our scheme the verifier needs to do a full modular exponentiation. For a situation in which the verifier has more power (as in the case in a smart card talking to a host, or a mobile station talking to a mainframe computer), this is an advantage. For situations in which there is a need for bilateral identification, our scheme should perhaps be replaced by one more suited to a weak verifier.

We close this section with a final comment on the original Schnorr scheme. In that scheme, y is reduced modulo q before transmission. At first sight it may appear advantageous to remove the reduction of y modulo q in the original Schnorr scheme and thus gain a significant computational advantage in the on-line portion of the computation. In fact, this would be disastrous because if we know $r + se$ and e, then we can construct an interval of length approximately q/e containing s. An algorithm of Pollard [15] can then be used to compute s in only about $\sqrt{q/e}$ operations. For the parameters suggested by Schnorr, the expected value of this is only 2^{35}.

5. Security of the Modified Scheme

Like all cryptographic schemes, identification schemes can be attacked in a variety of ways. The purpose of introducing *interaction* to identification schemes is to protect against passive eavesdroppers and cheating verifiers recovering secret information that they can later use to impersonate the legitimate user. In this section, we will give evidence which indicates that our scheme does provide such protection. However, there are other kinds of attacks that might arise in applications that are not protected against by using an interactive identification scheme by itself.

In particular, Desmedt *et al.* [5] have pointed out that an interactive identification scheme offers no protection against the situation in which the verifier cheats by passing on information provided to him by the prover to another cheating prover

Furthermore, an interactive identification scheme does not offer any protection against a prover who gives away his secret information to another so that they may impersonate him, or against a prover who chooses weak secret keys that anyone can guess. A variant of this point was discussed by Burmester in the rump session at Eurocrypt '90.

Both of these attacks can be protected against if the system uses physical characteristic information to identify uniquely an individual. If the identification by physical characteristics offers perfect security, then there is no security gained by using an interactive identification scheme instead of simply using a digital signature (issued by the KAC) of the physical characteristics. However, if the identification by physical characteristics offers less than perfect security, then using an interactive identification scheme can in some cases result in increased total security of the system. For example, if two people share the same physical characteristics, then a digital signature of these characteristics could be transferred by a cheating verifier between these two people. With the use of interaction this will be impossible without the cooperation of the legitimate prover.

In the remainder of this section, we will consider only the security provided by the system against passive eavesdroppers and cheating verifiers recovering secret information that they can later use to impersonate the legitimate user. As in the original Schnorr scheme, one kind of attack would be to try to construct a pair (I, α^{-s}) and a legitimate signature $\mathscr{S}$ of this pair for later use in identification. This would however require a successful attack on the signature scheme of the KAC. For this paper, we will assume that the signature scheme of the KAC is secure.

To demonstrate the security of our identification scheme, it remains for us to show two things:

1. A cheating prover $\tilde{P}$ should not be able to convince a verifier that he $(\tilde{P})$ knows a discrete logarithm of v when this is not the case.
2. A (possibly cheating) verifier should not be able to obtain any information that would later be useful to an imposter.

This first condition is commonly referred to as the *soundness* [7], [18] of the protocol. Schnorr proved the soundness of his protocol [17], and with a slight modification of his proof, we prove the soundness of our protocol in Theorem 1.

The second condition is the property that has inspired the definitions of zero-knowledge proofs [8] and witness hiding protocols [6]. Neither of these conditions have been estabished for the Schnorr scheme. We will use the model of witness indistinguishable and witness hiding to argue that our scheme is secure. In this paper, we will state only the informal definitions of these concepts, since our theorems will give statements specific to our protocol. The concepts are described informally and defined formally in [7]. Informally, a protocol is witness indistinguishable if the verifier cannot tell which witness the prover is using, and a protocol is witness hiding if participating in the protocol does not help the verifier to compute any new witnesses which he did not know at the beginning of the protocol. Theorem 2 shows that our protocol is witness indistinguishable and Corollary 3 shows that it is witness hiding. Feige and Shamir [7] have shown that witness indistinguishability implies witness hiding under certain conditions and these conditions are met by our protocol. However, by proving witness hiding directly instead of using their general theorem, we are able to describe the exact connection between the security of our protocol and the difficulty of precise computational problems.

We should perhaps clarify the claim that we are basing the security on two different problems. We are in fact basing the security of our scheme on the difficulty

of the following problem:

Given an integer α and a prime p, find discrete logarithms modulo p to the base α.

The point is that if one could solve such a problem, then one could also solve the following two problems:

Given p, α, and q, with α of order q modulo p, find discrete logarithms modulo p to the base α.
Given p and α, find the order of α modulo p.

These two problems are then the actual problems that we base the security on. The dependence on factoring comes from the second problem, but note that while a successful attack on the scheme requires the ability to solve the second problem (and therefore to factor $p-1$), a cryptanalyst will be in possession of some side information, namely, the knowledge of an element α whose order is the unknown factor q of $p-1$. Whether this information can be used to factor $p-1$ faster than current general purpose factoring methods is unknown. For further information on the current state of the art in factoring, see [12] or [16], and for information on computing discrete logarithms, see [10] and [14].

This next theorem establishes the soundness of the identification scheme.

Theorem 1. *Let p and α be as described in Section 3. Let $x \equiv \alpha^r \pmod p$ for some integer r. Let $A = A_{p,\alpha,v,x}$ be an algorithm with running time bounded by T that receives an input e, and attempts to compute an integer y such that $\alpha^y v^e \equiv x \pmod p$. If A will produce a correct output for at least $\varepsilon 2^t$ of the possible challenges e (where $\varepsilon \geq 2^{1-t}$), then there exists a probabilistic algorithm that with at least a constant probability, will compute a discrete logarithm of v in $O(\log^3 p + T/\varepsilon)$ bit operations.*

Proof. This proof is similar to the proof given by Schnorr for Proposition 2.1 in [17]. Choose random e's until an e_1 and e_2 are found for which A gives the correct outputs y_1 and y_2. Such a pair e_1, e_2 exists since $\varepsilon \geq 2^{1-t}$. The expected time for this is $O(T/\varepsilon)$. Then $\alpha^{y_1-y_2} \equiv v^{e_2-e_1} \pmod p$. If $(e_2-e_1, p-1)=1$, then we use the Euclidean algorithm to compute $f \equiv (e_2-e_1)^{-1} \pmod{p-1}$. It then follows that $\alpha^{(y_1-y_2)f} \equiv v \pmod p$, so that $(y_1-y_2)f$ is the desired discrete logarithm.

Suppose now that $d = \gcd(e_2-e_1, p-1) > 1$. In this case, we set $d_1 = d$, $m_1 = p-1$, and for $i = 2, \ldots$, we compute $m_i = m_{i-1}/d_{i-1}$ and $d_i = \gcd(e_2-e_1, m_i)$. The m_i's will quickly decrease until we come to a point where $d_i = 1$, and we will still have $q | m_i$ since $|e_2-e_1| < q < w$. Applying the extended Euclidean algorithm, we then obtain an integer l such that $l(e_2-e_1) \equiv 1 \pmod{m_i}$, and it follows that $(y_1-y_2)l$ is a discrete logarithm of v. It is easy to see that these computations can be done in $O(\log^3 p)$ bit operations using standard algorithms. □

A *conversation* between a prover P and a verifier V consists of the public information, p, v, α, I, and a triple (x, e, y) where $x = \alpha^r \pmod p$ for some integer r chosen by P from the uniform distribution on the integers in $[1, p-1]$, e is an integer, $e \in [1, 2^t]$ is chosen by V, and y is an integer satisfying $\alpha^y v^e = x$ which is computed by P as $y \equiv r + se \pmod{p-1}$. A *tape of conversations* between a prover P and a

verifier V is a sequence of conversations between P and V. In the definition of conversation, we made no assumption about how the verifier chose e. Therefore, in a tape of conversations, the verifier is free to use any method in choosing the e's and can use any auxiliary input, h, that he has. We use the notation $a \in_r A$ to mean that a is an element of A chosen at random from the uniform distribution on elements of A.

We will now proceed with the proof that this identification scheme is witness hiding unless $p - 1$ can be factored.

Theorem 2. *The distribution of a tape of conversations between* P *and* V *does not depend on which discrete log of v is known by* P.

Proof. Let h be any auxiliary input that V has. The prover has an s such that $\alpha^{-s} \equiv v \pmod{p}$. Let $s' \equiv s \pmod{q}$ and $s'' \equiv s \pmod{(p-1)/q}$. s' is uniquely determined by v, but there are $(p-1)/q$ distinct choices of s'' for each v, corresponding to the $(p-1)/q$ different discrete logarithms of v. It suffices to show that the distribution of a tape of conversations between P and V does not depend on s''. To do this, we will show that each conversation does not depend on s'' and, furthermore, that the distribution of each conversation, given h and all of the previous conversations on the tape, does not depend on s''. The proof will be by induction on the number of conversations.

To initiate a conversation, P will pick $r \in_r [1, p-1]$. Let $r' \equiv r \pmod{q}$ and $r'' \equiv r \pmod{(p-1)/q}$. The distribution of r' is uniform on the set of equivalence classes modulo q, and the distribution of r'' is uniform on the set of equivalence classes modulo $(p-1)/q$. It follows that the distribution of the $x \equiv \alpha^{r'} \pmod{p}$ that P produces is uniform on the subgroup of residue classes generated by α. It is also easy to see that x does not depend on either h, r'', s'' or on previous conversations on the tape.

The e's that V picks to send to P can depend on h, v, α, x, and all of the previous information already on the tape of conversations, but since (by induction) everything on the tape up to that point did not depend on s'', and since V does not have access to s'', the distribution of e, given h and all of the previous conversations on the tape cannot depend on s'' either. Moreover, the e that V chooses cannot depend on r'', since V has not yet seen anything that contains any information about it. Since P chose r randomly, it follows also that the value of r'' does not depend on e, h, and the previous information on the tape.

When P receives e from V, he computes $y \equiv r + se \pmod{p-1}$. Let $y' \equiv y \pmod{q}$ and $y'' \equiv y \pmod{(p-1)/q}$. Clearly y' depends on r', s', e, and q, but does not depend on s''. On the other hand, $y'' \equiv r'' + s''e \pmod{(p-1)/q}$, but r'' does not depend on any of the previous communication. Therefore, r'' completely masks the value of s'', and the distribution of y'' is uniform on the equivalence classes modulo $(p-1)/q$. Thus, the distributions of both y' and y'' (even given h and all of the previous conversations on the tape) do not depend on s'', which implies that the same is true for the distribution of y, from which the conclusion of the theorem follows. □

We will now use these two theorems to show that if an impostor has a reasonable chance of success at passing himself off as P, then P together with the impostor could factor $p - 1$.

Corollary 3. *Let A be an algorithm that has access to a tape of conversations between* P *and any* V. *Suppose that A selects x, then receives an input e, and tries to produce an output y such that $\alpha^y v^e \equiv x \pmod{p}$. If A has running time T and will produce a correct output for at least $\varepsilon 2^t$ of the possible challanges $e \in [1, 2^t]$, where $\varepsilon \geq 2^{1-t}$, then there exists a probabilistic algorithm that uses A and P that will discover a nontrivial factor of $p - 1$ in time $O(\log^3 p + T/\varepsilon)$ with probability at least $1 - 2/w$, where w is the largest prime factor of $(p - 1)/q$.*

Proof. It follows from Theorem 1 that in the expected time of $O(\log^3 p + T/\varepsilon)$ bit operations, A will compute a discrete logarithm a of v. Since the tape of conversations does not depend on which discrete logarithm P knows, a does not depend on s''. Furthermore, P initially chose s randomly, and hence $a + s$ is a random integer multiple of q, with $(a + s)/q$ uniformly distributed on $[(a + 1)/q, (a + p - 1)/q]$. Let $d = (a + s, p - 1)$. Then $q|d$. If w is a prime factor of $(p - 1)/q$, then the probability that w divides $a + s$ is exactly

$$\frac{\#\left\{n: w|n, n \in \left[\frac{a+1}{q}, \frac{a+p-1}{q}\right]\right\}}{\#\left\{n: n \in \left[\frac{a+1}{q}, \frac{a+p-1}{q}\right]\right\}} \leq \frac{2}{w}.$$

Therefore, $\Pr((d, w) = 1) \geq 1 - 2/w$, and if $(d, w) = 1$, then d is a nontrivial factor of $(p - 1)/q$. □

6. Comments

For our modification of Schnorr's scheme, we have proved that if an impostor has a reasonable probability of success, then there is an efficient algorithm for factoring $p - 1$. Furthermore, it is easy to see that even if $p - 1$ is factored, then the security of our scheme becomes the same as the security of the Schnorr scheme.

Our identification scheme can be converted into a signature scheme using the same techniques that were introduced by Feige *et al.* [6] and also used by Schnorr [17]. To be more precise, let f be a one-way hash function. To sign a message, m, the prover selects x as in the identification scheme. Instead of the verifier choosing e, P computes $e = f(x, m)$. The remainder of the signature scheme is the same as the identification scheme.

An interesting modification to our scheme is to choose α to be a generator of the multiplicative group (mod p), i.e., an element of order $p - 1$. The rest of the protocol would work as before. In one sense, this appears to be more secure since we are no longer revealing an element of order q. We were able to modify the proof of Schnorr [17] to prove that if an impostor could be successful, then he would have learned

the discrete logarithm of v (if 2 is the only prime factor of $p-1$ that is smaller than 2^t). However, as with the Schnorr scheme, we could not prove that a verifier could not learn something about the discrete logarithm of v. Therefore, it is not clear whether choosing α to be an element of order $p-1$ increases or decreases the security of our scheme.

Acknowledgment

We would like to thank Jim Davis, John DeLaurentis, Peter Montgomery, Judy Moore, and C. P. Schnorr for helpful conversations during the course of this research. We would also like to express our thanks to the anonymous referees, who made critical comments that greatly improved the presentation.

References

[1] T. Beth, Efficient zero-knowledge identification scheme for smart cards, *Advances in Cryptology—Proceedings of Eurocrypt '88*, Lecture Notes in Computer Science, vol. 330, Springer-Verlag, Berlin, 1989, pp. 77–84.

[2] E. F. Brickell and K. S. McCurley, An interactive identification scheme based on discrete logarithms and factoring, *Advances in Cryptology—Proceedings of Eurocrypt '90* (to appear).

[3] D. Chaum, J.-H. Evertse, J. van de Graaf, and R. Peralta, Demonstrating possession of a discrete logarithm without revealing it, *Advances in Cryptology—Proceedings of Eurocrypt '86*, Lecture Notes in Computer Science, vol. 263, Springer-Verlag, Berlin, 1987, pp. 200–212.

[4] D. Chaum, J.-H. Evertse, and J. van de Graaf, An improved protocol for demonstrating possession of discrete logarithms and some generalizations, *Advances in Cryptology—Proceedings of Eurocrypt '87*, Lecture Notes in Computer Science, vol. 304, Springer-Verlag, Berlin, 1988, pp. 127–141.

[5] Y. Desmedt, C. Goutier, and S. Bengio, Special uses and abuses of the Fiat–Shamir passport protocol, *Advances in Cryptology Proceedings of Crypto '87*, Lecture Notes in Computer Science, vol. 293, Springer-Verlag, Berlin, 1988, pp. 21–39.

[6] U. Feige, A. Fiat, and A. Shamir, Zero-knowledge proofs of identify, *Journal of Cryptology* **1** (1988), 77–94.

[7] U. Feige and A. Shamir, Witness indistinguishable and witness hiding protocols, *Proceedings of the 22nd ACM Symposium on Theory of Computing*, Association for Computing Machinery, New York, 1990, pp. 416–424.

[8] S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof systems, *SIAM Journal on Computing* **18**, No. 1 (1989), 186–208.

[9] D. E. Knuth, *The Art of Computer Programming*, Vol. 2: *Seminumerical Algorithms*, 2nd ed., Addison-Wesley, Reading, MA, 1981.

[10] B. LaMacchia and A. Odlyzko, Computation of discrete logarithms in prime finite fields, *Advances in Cryptology—Proceedings of Crypto '90*, Lecture Notes in Computer Science (to appear).

[11] A. K. Lenstra and M. S. Manasse, Factoring by electronic mail, *Advances in Cryptology—Proceedings of Eurocrypt '89*, Lecture Notes in Computer Science, vol. 434, Springer-Verlag, Berlin, 1990, pp. 355–371.

[12] A. K. Lenstra, H. W. Lenstra, Jr., M.S. Manasse, and J. M. Pollard, The number field sieve, *Proceedings of the 22nd ACM Symposium on Theory of Computing*, Association for Computing Machinery, New York, 1990, pp. 564–572.

[13] K. S. McCurley, A key distribution system equivalent to factoring, *Journal of Cryptology* **1** (1988), 95–105.

[14] K. S. McCurley, The discrete logarithm problem, *Cryptology and Computational Number Theory, Proceedings of Symposia in Applied Mathematics*, vol. 42, American Mathematical Society, Providence, 1990, pp. 49–74.

[15] J. M. Pollard, Monte Carlo methods for index computation mod p, *Mathematics of Computation* **32** (1978), 918–924.

[16] C. Pomerance, Factoring, *Cryptology and Computational Number Theory, Proceedings of Symposia in Applied Mathematics*, vol. 42, American Mathematical Society, Providence, RI, 1990, pp. 27–48.

[17] C. P. Schnorr, Efficient identification and signatures for smart cards, *Advances in Cryptology—Proceedings of Crypto '89*, Lecture Notes in Computer Science, vol. 435, Springer-Verlag, Berlin, 1990, pp. 239–252.

[18] M. Tompa and H. Woll, Random self-reducibility and zero knowledge interactive proofs of possession of information, *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, IEEE, Washington, D.C., 1987, pp. 472–482.

[19] S. S. Wagstaff, Jr., Greatest of the least primes in arithmetic progressions having a given modulus, *Mathematics of Computation* **33** (1979), 1073–1080.