An Interactive Identification Scheme Based on Discrete Logarithms and Factoring*

(Extended Abstract)

Ernest F. Brickell Kevin S. McCurley

Sandia National Laboratories
Albuquerque, NM 87185

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. The modification that we present here will remain secure if either of two computational problems is infeasible, namely factoring a large integer and computing a discrete logarithm. For this enhanced security we require somewhat more communication and computational power, but the requirements remain quite modest, so that the scheme is well suited for use in smart cards.*

1 Introduction

In this note we describe an interactive identification scheme that is a variation of a scheme presented by Schnorr at Crypto '89 [9]. Schnorr's scheme has several features that make it advantageous for use in smart cards or other environments with limited computing power. Its security is based on the difficulty of the discrete logarithm problem in a subgroup of $\mathbb{Z}_p^*$. In this paper we shall describe a variation with the property that a successful attack on the scheme requires the ability to solve an instance of the discrete logarithm problem, and in addition to factor an integer that is divisible by two large primes.

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

*This work was performed under U. S. Department of Energy contract number DE-AC04-76DP00789

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 [7], where a key distribution scheme with this property was given. The key distribution scheme of [7] 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 computing a discrete logarithm in a subgroup of $\mathbb{Z}_p^*$, and factoring $p-1$.

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.

2 Schnorr's Identification Scheme

We begin by describing the original Schnorr authentication scheme in terms 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 $q \mid p-1$, $q \geq 2^{140}$, and $p \geq 2^{512}$,
- α of order q in the group $\mathbb{Z}_p^*$,
- its own private and public keys.

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 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 S of (I, v), and x. V then checks P's identification by verifying the signature S, chooses a random $e \in \{0, \ldots, 2^t - 1\}$, 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.

Schnorr's scheme may be regarded as a practical refinement of the zero-knowledge protocols of Chaum et.al. [3], [2] for demonstrating possession of a discrete logarithm. In [3], 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. In the modified scheme, each user will have his own prime p and base element α, and these will need to be transmitted along with v during each identification session. Once again the KAC serves only to sign the public keys of each user, but now these include p and α. Rather than the single security parameter t, we describe the scheme in terms of the parameters k, t, and u.

When a user wishes to join the system, he chooses primes q and w with $q < w$, $2^{k-1} < q < 2^k$, and $qw > 2^{512}$. The user further chooses a prime $p \equiv 1 \pmod{qw}$, an element $\alpha \in \mathbb{Z}_p^*$ of order q, and a random number $s \in \{1, \ldots, q\}$. The user then computes $v \equiv \alpha^{-s} \pmod{p}$, and presents p, v, and α to the KAC along with some form of identification, but keeps q, w, and s secret. The KAC verifies the user's identity, generates an identification string I, and produces a signature S of the quadruple (I, v, p, α), 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 a random number $r \in \{1, \ldots, q\}$ and computes $x \equiv \alpha^r \pmod{p}$. Then P sends to V the identification string I, its public keys v, p, and α, the KAC's signature S, and x. V checks P's identification by verifying the signature S of (I, v, p, α). If the keys are authentic, then V chooses a random $e \in \{0, \ldots, 2^t - 1\}$ and a random

$f \in \{0, \ldots, 2^u - 1\}$, and transmits the pair (e, f) to P. P then computes an integer y such that $y \equiv r + se \pmod{q}$ and $2^k f \leq y < 2^k(f+1)$, and sends y to V. V checks that $x \equiv \alpha^y v^e \pmod{p}$ and $2^k f \leq y < 2^k(f+1)$, and accepts P's proof of identity if these conditions are satisfied.

The parameters u and t can be adjusted to suit specific needs, but we suggest using $u = t = 20$. With this choice, there are 2^{40} possible challenges (e, f), and the probability of guessing the challenge ahead of time is therefore 2^{-40}. If an impostor somehow discovers the secret prime q, then a precomputed pair y, x that satisfies $\alpha^y v^e \equiv x \pmod{p}$ can always have the y adjusted to fit any challenge f, but the probability of guessing the e ahead of time is still only 2^{-20}. Similarly if an impostor knows a discrete logarithm of v to the base α, then the probability of success in guessing ahead of time is also 2^{-20}. We regard this as being acceptably low for use in an identification scheme.

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 $k > 140$ 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. The construction of p should be relatively easy, since heuristic evidence (see [10]) 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 [6] and Lenstra et. al. [5] 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 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 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 S. We have used a value of $k = 140$ in the original and $k = 200$ in the modified scheme.

Original Scheme	
I	100
v	512
S	100
x	512
e	40
y	140
total	1404

Modified Scheme	
I	100
v	512
p	512
α	512
S	100
x	512
(e, f)	40
y	220
total	2508

The modified scheme therefore pays a penalty of an extra 1104 bits in communication, and possibly more if error correction is included. On the other hand, this is still well within the realm of possibility using present technology.

We now compare the computational requirements of the two schemes. In both the original Schnorr scheme and the modified scheme, numerous refinements can be devised to improve peformance. No matter what we do, however, the amount of arithmetic required in the new scheme appears to impose a slight penalty on speed. Part of the penalty comes from the fact that the prime q is larger for the modified scheme. Both schemes can use a 512 bit modular exponentiation with an exponent r of at most 140 bits in the preprocessing stage.

In the original Schnorr scheme, the prover is required to compute $y \equiv r + se \pmod{q}$, and the most obvious way to do this requires a multiplication, an addition, and a division by q. In the modified scheme, we require in addition a multiplication by q and an extra addition.

This does not however take into account any optimization. We now discuss a method for speeding the computations in both the original Schnorr scheme and the modified scheme. The idea here is to replace the divisions by q with multiplications (using shorter integers). This can be done by precomputing (only once, when the initial keys are selected) an approximation Q of s/q. If $0 < s/q - Q < 2^{-t-1}$, then

$$r - q < r + se - q[[Qe]] < r + q,$$

where $[[x]]$ denotes the nearest integer to x. Hence after computing $r + se - q[[Qe]]$, at most one subtraction or addition of q will be required to reduce $r + se$ modulo q. The overall improvement from performing the precomputation is to replace the division by q with a multiplication of Q and e (both of which are only t bits) followed by multiplication of q and a t bit integer, followed by at most two subtractions or additions involving k bit integers. Depending on the implementation, this may result in a significant speedup by eliminating the multiple precision division.

In the modified scheme, we can employ a similar approach. For the modified scheme we need to compute y so that $y \equiv r + se \pmod{q}$ and $2^k f \leq y < 2^k(f + 1)$. To do this, we precompute two sufficiently good approximations Q_1 and Q_2 of s/q

and $2^k/q$ respectively. We then compute $r + se + q[[Q_2 f - Q_1 e]]$, and if necessary adjust the result with at most one addition or subtraction of q.

Using these division-free algorithms for the computations, the only extra work required in the modified scheme is for an additional multiplication and subtraction, on numbers of approximately t bits. This should have a negligible effect on the overall computation speed. As we shall see in the next section, this slight degradation in performance brings in return the promise of an extra measure of security that cannot be achieved by simply increasing the key size.

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 [8] 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 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. [4] 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 who (falsely) proves his identity at another location.

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 uniquely identify 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 that 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 a passive eavesdropper. There are several basic attacks that can be mounted by a passive eavesdropper against identification schemes. For example, in the original Schnorr scheme, one kind of attack would be to try to construct a pair (I, α^{-s}) and a legitimate signature S of this pair for later use in identification. This would however require a successful attack on the signature scheme of the KAC. Another attack would involve observing a user identify himself several times, collecting a set of the tuples (x, e, f, y). It can be shown that a reasonable number of such tuples cannot provide any useful information, since the attacker could himself construct such tuples from a distribution that is very close to the legitimate user's distribution by first choosing y, then f, then e, and then x.

A more serious attack would involve observing a user going through the identification process, and for the pair (I, v) that is observed, try to later produce an x for which there is a reasonable chance of being able to answer the challenge by finding a suitable y. Schnorr proved that an attack of this kind for the original scheme would require the ability to compute the discrete logarithm of v. In the same spirit, we shall prove in Theorem 1 that an attack of this kind on the modified scheme would require the ability to factor $p-1$ *and* the ability to find the discrete logarithm of v.

We should be careful to observe that an attack on the system has not been proved to be completely equivalent to the problem of simply factoring $p-1$. While a successful attack requires the ability to factor $p-1$, a cryptanalyst will be in possession of some side information. The most obvious information available is the knowledge of an element α whose order is the unknown factor q of $p-1$. Whether this information can be used to discover the factor q is unknown.

Theorem 1. *Let p and α be as described in Section 3. Let $A = A_{p,\alpha,v,x}$ be an algorithm with running time bounded by T that receives an input (e, f), 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 $\epsilon 2^{u+t}$ of the possible challenge pairs (e, f) (where $\epsilon \geq \max(2^{1-t}, 2^{1-u})$), then there exists a probabilistic algorithm that with at least a constant probability, will compute the prime factor q of $p-1$ and a discrete logarithm of v in $O(\log^3 p + \frac{T}{\epsilon})$ bit operations.*

Proof. We first describe an algorithm for computing a discrete logarithm of v. The idea is to construct correct triples (e_1, f_1, y_1) and (e_2, f_2, y_2) with $e_1 \neq e_2$. We first choose random pairs (e_1, f_1) until one is found for which A gives a correct output y_1. We then choose random pairs (e_2, f_2) with $e_2 \neq e_1$ until we find one for which A gives a correct output y_2. We now have $\alpha^{y_1 - y_2} \equiv v^{e_2 - e_1} \pmod{p}$. We use the Euclidean algorithm to compute $d = \gcd(e_2 - e_1, p-1)$. Assume first that

$d = 1$. Then the extended Euclidean algorithm gives an integer ℓ with $(e_2 - e_1)\ell \equiv 1 \pmod{p-1}$, so that $\alpha^{(y_1-y_2)\ell} \equiv v \pmod{p}$. Hence $(y_1 - y_2)\ell$ is a discrete logarithm of v to the base α.

Suppose now that we found $d > 1$. In this case we let $d_1 = d$, $m_1 = p - 1$, and for $i = 2, \ldots$, we compute $m_i = m_{i-1}/d_{i-1}$ and $d_i = \gcd(e_2 - e_1, m_i)$. Since $|e_2 - e_1| < q < w$, we will eventually arrive at $d_i = 1$ and $q \mid m_i$. Applying the extended Euclidean algorithm, we then obtain an integer ℓ such that $\ell(e_2 - e_1) \equiv 1 \pmod{m_i}$, and it follows that $(y_1 - y_2)\ell$ is a discrete logarithm of v.

Clearly, after examining $O(1/\epsilon)$ pairs (e_1, f_1) we have a probability of at least $1/2$ of getting an output from A. Even if all pairs (e_1, f) for $1 \le f \le 2^u$ are in the set of pairs on which A produces a correct output, the probability is still at least $\epsilon - 2^{-t}$ that a pair (e_2, f_2) with $e_1 \neq e_2$ will yield a correct output from A, so we have again a probability at least $1/2$ of success after we examine $O(1/(\epsilon - 2^{-t})) = O(1/\epsilon)$ pairs (e_2, f_2).

We now describe the algorithm for recovering the factor q. From the previous discussion, we may assume without loss of generality that we are already in possession of an integer L such that $\alpha^L \equiv v \pmod{p}$. We begin by choosing random (e_1, f_1) until a pair is found for which A produces a correct output y_1. After this we search for a second pair (e_2, f_2) for which A produces a correct output. Since $\alpha^{y_1-y_2} \equiv v^{e_2-e_1} \pmod{p}$, we have $y_1 - y_2 \equiv (e_2 - e_1)L \pmod{q}$. If it happens that $y_1 - y_2 \not\equiv (e_2 - e_1)L \pmod{w}$, then $\gcd(y_1 - y_2 - (e_2 - e_1)L, p-1)$ will give a splitting of qw. On the other hand, for each e_2, the congruence $y_1 - y_2 \equiv (e_2 - e_1)L \pmod{w}$ has only one solution y_2 in the interval $[1, w]$, so there is at most one f_2 for each e_2 that can give such a solution y_2. Hence the number of pairs (e_2, f_2) that do not lead to a splitting of qw is at most 2^t, and therefore the probability of success in finding a pair (e_2, f_2) that will split qw is at least $\epsilon - 2^{-u}$. Hence we expect to split q and w after examining $O(1/(\epsilon - 2^{-u}) = O(1/\epsilon)$ pairs (e_2, f_2), and this completes the proof. □

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.

References

[1] Thomas Beth, "Efficient zero-knowledge identification scheme for smart cards," *Advances in Cryptology (Proceedings of Eurocrypt '88)*, Lecture Notes in Computer Science **330** (1989), 77–84.

[2] David Chaum, Jan-Hendrik Evertse, Jeroen van de Graaf, and René Peralta, "Demonstrating possession of a discrete logarithm without revealing it," *Ad-*

vances in Cryptology (Proceedings of Eurocrypt '86) Lecture Notes in Computer Science **263** (1987), 200–212.

[3] David Chaum, Jan-Hendrik Evertse, and Jeroen 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 **304** (1988), 127–141.

[4] Yvo Desmedt, Claude Goutier, and Samy Bengio, "Special uses and abuses of the Fiat-Shamir passport protocol," *Advances in Cryptology (Proceedings of Crypto '87)* Lecture Notes in Computer Science **293** (1988), 21–39.

[5] 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, 564–572.

[6] Arjen K. Lenstra and Mark S. Manasse, Factoring by Electronic Mail, *Proceedings of Eurocrypt '89, Lecture Notes in Computer Science*, to appear.

[7] Kevin S. McCurley, A Key Distribution System Equivalent to Factoring, *Journal of Cryptology* **1** (1988), 95–105.

[8] J. M. Pollard, "Monte Carlo Methods for Index Computation mod p," *Mathematics of Computation* **32** (1978), 918–924.

[9] C.P. Schnorr, Efficient Identification and Signatures for Smart Cards, *Proceedings of Crypto '89, Lecture Notes in Computer Science*, to appear.

[10] Samuel S. Wagstaff, Jr., *Greatest of the Least Primes in Arithmetic Progressions Having a Given Modulus*, Mathematics of Computation **33** (1979), 1073–1080.