To Appear in *Designs, Codes and Cryptography.*

A Non-interactive Public-Key Distribution System [1]

Ueli M. Maurer [2]

Dept. of Computer Science
ETH Zurich
CH-8092 Zurich, Switzerland
maurer

Yacov Yacobi

Bellcore
445 South St.
Morristown, NJ 07962
yacov@bellcore.com

Abstract. An identity-based non-interactive public key distribution system is presented that is based on a novel trapdoor one-way function allowing a trusted authority to compute the discrete logarithms modulo a publicly known composite number m while this is infeasible for an adversary not knowing the factorization of m. Without interaction with a key distribution center or with the recipient of a given message, a user can generate a mutual secure cipher key based solely on the recipient's identity and his own secret key, and subsequently send the message, encrypted with the generated cipher used in a conventional cipher, over an insecure channel to the recipient. In contrast to previously proposed identity-based systems, no public keys, certificates for public keys or other information need to be exchanged and thus the system is suitable for certain applications such as electronic mail that do not allow for interaction. The paper solves an open problem proposed by Shamir in 1984.

1. Introduction

In their seminal 1976 paper, Diffie and Hellman [4] introduced the ingenious concept of public key cryptography and proposed the first public key distribution system, which is based on exponentiation in a finite field. The basic idea of a public key distribution system is briefly summarized in the following in order to point out the novelty in our scheme that allows it to be non-interactive. In an insecure communication network where all messages sent over a communication channel can be intercepted by an adversary, two

[1] The results of this paper have appeared in part in the proceedings of EUROCRYPT '91, Lecture Notes in Computer Science, Vol. 547, Springer Verlag, pp. 498-507, 1991.

parties not sharing any secret information initially can generate a secure cipher key (to be subsequently used together with a conventional symmetric cryptosystem) by each choosing a secret number, applying a one-way transformation to this number and exchanging the results of this transformation (the public keys) over an insecure channel. The one-way transformation has the property that given the result, it is infeasible to compute the argument. The one-way transformation proposed by Diffie and Hellman has the crucial additional property, which is due to the commutativity of multiplication, that each party can generate the same mutual secure cipher key from his own secret number and the other party's public key. Without knowing at least one of the secret numbers it is infeasible to generate the secure cipher key using present technology and algorithmic knowledge.

Public-key distribution systems and public-key cryptosystems suffer from the following well-known authentication problem. In order to prevent an adversary from fraudulently impersonating another user, it must be possible to verify that a received public key belongs to the user it is claimed to belong to. A commonly used solution to this authentication problem is the certification of public keys by a trusted authority which, after checking a user's identity, signs the concatenation of his name and public key using a digital signature scheme. Systems based on either the RSA [25] or the ElGamal [6] signature schemes have been proposed [8].

Shamir [27] suggested as a simple but ingenious method for solving the authentication problem in public-key cryptography to let each user's public key be his (publicly-known) identification information. Because it must be infeasible for users to compute the secret key corresponding to a given identity (including their own), the secret keys must be computed by a trusted authority who knows some secret trapdoor information. The security of such an identity-based system depends on the trusted authority in a more crucial way than the security of a public-key certification system because in the former the trusted authority knows all secret keys.

Because a user's identity can be assumed to be publicly known (the identity can be defined as that part of the identification information that *is* publicly known), the public keys of an identity-based public-key cryptosystem need not be transmitted. Therefore an identity-based system can be used in a completely non-interactive manner.

A simple way to set up an identity-based public-key cryptosystem would seem to be to use the RSA-system with a universal modulus where each user's public encryption exponent is his (odd, and relatively prime to $\varphi(m)$) identity and in which a trusted authority knowing the factorization of the modulus computes the secret decryption exponents for users. However, this system is insecure because knowledge of a matching (secret/public) key pair allows to easily factor the modulus.

While Shamir presented an identity-based signature scheme, he proposed as an open problem to find an identity-based public-key cryptosystem or public-key distribution system [27]. In the context of signature schemes, however, a non-interactive system is less advantageous than it would be in the context of a public-key cryptosystem: for any signature scheme, the signed message can be sent together with a certified public key in one message whereas for encryption, the sender must in a non-interactive system first obtain the recipient's public key.

Many previously proposed systems [8, 9, 21, 22, 30] have been called identity-based public-key distribution systems because they make use of Shamir's idea for self-authentication of public keys. However, none of these (with the exception of the quite impractical and also insecure version of a scheme discussed in [30]) is an identity-based system in Shamir's sense because the public key is a function not only of the identity but also of some random number selected either by the user or by the trusted authority. As a consequence, these systems are bound to be interactive. A major achievement of this paper is that it presents the first truly identity-based public-key distribution system. It should be mentioned that the key predistribution system of Matsumoto and Imai [15], which is based on a completely different approach, also achieves non-interactive key distribution.

The original Diffie-Hellman public key distribution system [4] with a prime modulus p cannot be used as an identity-based system in Shamir's sense because if the scheme is secure, that is when discrete logarithms modulo p are infeasible to compute, it is infeasible even for a trusted authority to compute the secret key corresponding to a given public key, i.e., a given identity. This comment applies to any public-key distribution system based on a one-way function without trapdoor. One of the achievements of this paper is that a method for building a trapdoor into the modular exponentiation one-way function is proposed which allows a trusted authority to feasibly compute discrete logarithms whereas this is completely infeasible for an adversary using present technology and algorithmic knowledge. This allows a trusted authority to set up a non-interactive public-key distribution system. Non-interactiveness may be crucial in some applications (e.g. electronic mail, some military applications) and in some other applications allows at least to simplify the protocols. The computational effort that the trusted authority must spend is considerable but the key distribution protocol itself is very efficient.

2. A Non-interactive Public Key Distribution System

From a protocol viewpoint, the difference between a public-key distribution system and a public-key cryptosystem is that in the former, both parties must receive the other party's public key whereas in the latter, only the sending party must receive the public key of the receiving party. Therefore, a public-key distribution system, when combined with a conventional symmetric cryptosystem used for encryption, cannot be used as a public-key cryptosystem. In contrast, a non-interactive public-key distribution system can be used as a public-key cryptosystem by sending as one message the sender's identity and the enciphered plaintext, where the cipher key is computed from the receiver's identity and the sender's secret key and where some agreed conventional cipher is used for encryption of the message.

Our non-interactive public key distribution system is based on a variant of the Diffie-Hellman system with composite modulus m. By choosing the prime factors of m appropriately such that discrete logarithms modulo each prime factor can feasibly be computed but such that computing discrete logarithms modulo m is nevertheless infeasible, a trusted authority can set up a public key distribution system based on exponentiation modulo m. One can show that computing discrete logarithms modulo a composite number is at least as difficult as factoring the modulus (cf. Section 3).

Two different ways of generating such a modulus m are presented below and in Section 4, respectively. To use a composite modulus $m = pq$ with p and q prime in the Diffie-Hellman scheme has previously been proposed by Shmuely [29] and McCurley [17] in order to exhibit a system which to break requires the ability both to factor m and to compute discrete logarithms modulo p and q.

Our approach to identity-based public key distribution differs in a crucial way from previous approaches [8, 9, 21, 22, 30] in that the public key consists entirely of public identity information (e.g. name, address, physical description), but does not depend on an additional random number selected either by the user or the trusted authority. This is the reason why our system can be used in a truly non-interactive manner. Clearly, the type and amount of information about a user that can be assumed to be publicly known depends on the application, but note that in most applications, at least part of the identification information is indeed publicly known. For instance, the receiver's address, which must be known in every communication system in order to send a message, can serve as his public key.

One problem that arises in the proposed system is that the multiplicative group Z_m^* is cyclic if and only if m is either 2, 4, a power of an odd prime or twice the power of an odd prime. When m is the product of distinct odd primes there hence exists no element that generates the entire group Z_m^*. Thus not every identity number that corresponds to some valid identification information is guaranteed to have a discrete logarithm with respect to some universal base α. This problem could be solved by adding the smallest offset to every identity number that makes the new number have a discrete logarithm. However, the resulting system would have to be interactive since the offsets must be exchanged between the users. Two different solutions to this problem are presented below and in Section 4, respectively. Both are computationally more efficient (for the trusted authority) than the offset method and at the same time allow to preserve the advantage of non-interactiveness of our scheme.

Let $m = p_1 \cdot p_2 \cdots p_r$ where the primes $p_1, \ldots, p_r$ are in the following assumed to be odd and distinct. The maximal order of an element of the multiplicative group Z_m^* is given by $\lambda(m) = \operatorname{lcm}(p_1 - 1, \ldots, p_r - 1)$, which is at most 2^{-r+1} times the group order $\varphi(m)$. $\lambda(m)$ is strictly less than $\varphi(m)/2^{r-1}$ unless the numbers $(p_1 - 1)/2, \ldots, (p_r - 1)/2$ are pairwise relatively prime. Let α be an element of Z_m^* that is primitive in each of the prime fields $GF(p_1), \ldots, GF(p_r)$, i.e., such that for $1 \le i \le r$, $p_i - 1$ is the smallest exponent t_i for which $\alpha^{t_i} \equiv 1 \pmod{p_i}$. Then α has maximal order $\lambda(m)$ in Z_m^*. The discrete logarithm of a number y modulo m to the base α is defined as the smallest non-negative integer x such that $\alpha^x \equiv y \pmod{m}$ (if such an x exists) and can, when the complete factorization of m is given, be obtained by computing for $i = 1, \ldots, r$ the discrete logarithm x_i of y to the base α modulo p_i, i.e., by computing x_i satisfying $\alpha^{x_i} \equiv y \pmod{p_i}$, and solving the system

$$
\begin{array}{rcl}
x & \equiv & x_1 \pmod{p_1 - 1}, \\
 & \cdot & \\
 & \cdot & \\
x & \equiv & x_r \pmod{p_r - 1}
\end{array}
$$

of r congruences for x by the Chinese remainder technique. It follows from Theorem 5.4.2 on page 155 of [28] that this system of concruences has a solution if and only for all distinct

i and j, $\gcd(p_i-1, p_j-1)$ divides $x_i - x_j$. In particular, the above system has no solution unless either all x_i are odd or all x_i are even.

For every prime q dividing at least 2 of the r numbers $p_i - 1$ $(1 \le i \le r)$, let q^{s_q} be the maximal power of q that divides at least 2 of these numbers. Let u be the product of all prime powers q^{s_q}, i.e., let u be the maximal integer for which every prime power dividing u also divides at least 2 of the numbers $p_i - 1$. The following lemma and in particular its corollary suggests an easy to compute publicly-known function that transforms, without use of the secret trapdoor, any identity number into a modified identity number that is guaranteed to have a discrete logarithm.

Lemma 1. *Let m, α and u be as defined above. An element y of Z_m^* that can be expressed as $y \equiv w^u \pmod{m}$ for some $w \in Z_m^*$ has a discrete logarithm x modulo m to the base α. Moreover, x is given by the system of congruences $x \equiv uz_i \pmod{p_i - 1}$ for $i = 1, \ldots, r$ where z_i is the discrete logarithm of w modulo p_i to the base α.*

Proof. A system of congruences has a solution if and only if all congruences are consistent when considered modulo every prime power contained in at least one of the moduli. This condition is equivalent to the condition that the congruences be pairwise consistent modulo the greatest common divisor of the corresponding pair of moduli. Hence the necessary and sufficient condition for the above system of congruences to have a solution is that for any i and j, $x_i \equiv x_j \pmod{\gcd(p_i-1, p_j-1)}$. Clearly, u is a multiple of $\gcd(p_i-1, p_j-1)$ for any $i \neq j$ and therefore $x_i \equiv x_j \equiv 0 \pmod{\gcd(p_i-1, p_j-1)}$, i.e., the system is pairwise consistent. □

Corollary. *Let m and α be as defined above where the numbers $(p_i-1)/2, \ldots, (p_r-1)/2$ are pairwise relatively prime. Then every square modulo m has a discrete logarithm modulo m to the base α.*

A complete description of the preferred version of the proposed non-interactive public key distribution system follows. The following three paragraphs describe the system set up by a trusted authority, the user registration phase and the user communication phase, respectively.

To set up the system we suggest that a trusted authority choose the primes p_i such that the numbers $(p_i-1)/2$ are odd and pairwise relatively prime [16]. Preferably, $(p_i-1)/2$ are chosen to be primes themselves. The primes p_i are chosen small enough such that computing discrete logarithms modulo each prime is feasible (though not trivial) using for instance the algorithm of [3] but such that factoring the product, even with the best known method for finding relatively small prime factors [12] of a number, is completely infeasible. The trusted authority then computes the product

$$m = p_1 \cdot p_2 \cdots p_r$$

of the selected primes, determines an element α of Z_m^* that is primitive in every of the prime fields $GF(p_i)$ and publishes m and α as system parameters. We refer to Section 3 for an analysis of the security versus the feasibility for different sizes of parameters. To choose 3 to 4 primes of between 60 and 70 decimal digits seems at present to be appropriate, but these figures can vary according to future progress in computer technology and number-theoretic algorithms. An alternative approach to making the discrete logarithm problem feasible other than by choosing the prime factors of m sufficiently small is described in Section 4.

When a user A wants to join the system she visits the trusted authority, presents her identification information ID_A together with an appropriate proof of her identity (e.g. a passport) and receives the secret key s_A corresponding to ID_A. The above corollary suggests as a first solution appears to be that the secret key s_A can be computed by the trusted authority as the discrete logarithm of ${ID_A}^2$ modulo m to the base α:

$$s_A \equiv \log_\alpha({ID_A}^2) \pmod{m}. \tag{1}$$

Due to the squaring of ID_A, s_A is guaranteed to exist. However, this solution is insecure because a square root modulo m of the squared identity ${I_A}^2$ can be obtained when given the secret key $S_A = \log_g({I_A}^2)$ by computing $g^{S_A/2}$ (note that S_A is even). If for at least one of the prime factors p of m,

$$\log_g I_A \pmod{p} < (p-1)/2$$

while for at least some other prime factor q of m,

$$\log_g I_A \pmod{q} \geq (q-1)/2,$$

then the obtained square root of ${I_A}^2$ is different from I_A and $-I_A$ and thus allows one to find a non-trivial factor of m. This condition is satisfied by a fraction $1 - 2^{-r+1} \geq 1/2$ of all identities, where r is the number of distinct (odd) prime factors of m.

The described problem with equation (1) can be solved by a slight modification. The trusted authority chooses, once and for all, a secret multiplier t at random from $\mathbf{Z}^*_{\varphi(m)}$. Instead of issuing the discrete logarithms of squared identities as users' secret keys, the trusted authority conceals these logarithms by multiplying them with t before issuing them to users. Hence

$$s_A \equiv t \cdot \log_g({ID_A}^2) \pmod{\varphi(m)}.$$

In order to send a message M securely to a user B without interaction, user A establishes the mutual secure cipher key K_{AB} shared with user B by computing

$$K_{AB} \equiv (ID_B)^{2s_A} \pmod{m}.$$

Note that $K_{AB} \equiv \alpha^{vs_As_B} \pmod{m}$ where $v \equiv t^{-1} \pmod{\varphi(m)}$. She then uses a conventional symmetric cryptosystem (e.g. DES) to encipher the message M using the cipher key K_{AB}, which results in the ciphertext C. User A then sends C together with her identity number ID_A to user B. In order to decipher the received ciphertext C, user B proceeds symmetrically and computes

$$K_{BA} \equiv (ID_A)^{2s_B} \equiv \alpha^{vs_Bs_A} \equiv K_{AB} \pmod{m}.$$

He then deciphers C using the conventional cryptosystem with the secret key K_{AB}, which results in the plaintext message M.

Note that the trusted authority is only required for the initial system set up and for user registration, but not in the user communication phase described above. In fact, the trusted authority could close itself down if no additional users need to be registered, thereby irreversibly erasing the factorization of m.

In the described system the secret key shared by two users is the same when the protocol is repeated several times. In those cases where this is undesirable user A can choose a random number R and use $f(K_{AB}, R)$ as the mutual cipher key, where f is a cryptographically secure hash function. R is sent to B together with the ciphertext C. In order to prevent an adversary knowing a previously used cipher key from impersonating at a later time, a time stamp can be used as an additional argument of the hash function. It is possible to build a dynamic key distribution system using no hash function, that is provably as hard (on the average) to break against a disruptive adversary as factoring the modulus [31].

Although in the proposed trapdoor one-way function the trapdoor is the factorization of the modulus as in the RSA trapdoor one-way function [25], the two functions are nevertheless entirely different. In the RSA function, the argument is the base and the exponent e is a constant whereas in our exponentiation trapdoor one-way function the argument is the exponent and the base α is a constant. Accordingly, the inverse operations are the extraction of the e-th root and the discrete logarithm to the base α, respectively, and are infeasible to compute without knowledge of the trapdoor.

3. Security and Feasibility Analysis

We first prove a previously observed but neither published nor widely known fact about the difficulty of computing discrete logarithms for a composite modulus (see also [2]).

Lemma 2. *Let m be the product of distinct odd primes $p_1, \ldots, p_r$ and let α be primitive in each of the prime fields $GF(p_i)$ for $1 \leq i \leq r$. Then computing discrete logarithms modulo m to the base α is at least as difficult as factoring m completely.*

Proof. In order to apply an algorithm computing logarithms modulo m to factor m, one can choose a number t that is larger than $\lambda(m)$, for example $t = m$, compute α^t and apply the discrete logarithm algorithm to obtain t' satisfying $\alpha^{t'} \equiv \alpha^t \pmod{m}$ and thus also satisfying $t' \equiv t \pmod{\lambda(m)}$. Hence $t - t'$ is a (small) multiple of $\lambda(m)$. If necessary, this computation can be repeated a few times for different choices of t, and $\lambda(m)$ can be computed as the greatest common divisor of the results. Using an idea of Miller [18], m can be factored when $\lambda(m)$ is known by choosing elements r of Z_m^* at random until $gcd(r^{\lambda(m)/2} - 1, m)$ is a non-trivial factor of m, which happens when r is a quadratic residue modulo some p_i and a quadratic non-residue modulo some other p_j, i.e., with probability $1 - 2^{-r+1}$. Applying this technique several times yields the complete factorization of m. $\square$

Note that for the proposed scheme, $\lambda(m) = \varphi(m)/2^{r-1}$ and thus $\lambda(m) < m/2^{r-1} < 2\lambda(m)$. Therefore $\lambda(m)$ can be determined by a single application of the discrete logarithm algorithm. Notice also that the results of Shmuely [29] and McCurley [17] are different from Lemma 2 since they hold only for special moduli, but on the other hand the equivalence of *breaking* their schemes and factoring is proved.

The function

$$L_x(a,b) \;=\; e^{b(\log x)^a(\log\log x)^{1-a}}$$

is commonly used to express the conjectured asymptotic running time of number-theoretic algorithms. The asymptotically fastest known algorithm for computing discrete logarithms in $GF(p)$ is the number field sieve which has asymptotic running time $L_p(1/3,1)$, but the fastest practical algorithm for the size of number considered here is described in [3] and has asymptotic running time $L_p(1/2,1)$. The largest primes for which this algorithm is at present feasible with massively parallel computation have between 110 and 120 decimal digits. For primes of up to 65-70 decimal digits the algorithm is feasible using a few processors. An important feature of this algorithm is that most of the running time is spent in a precomputation phase that is independent of actual elements for which the logarithm is to be computed. After the precomputation, individual logarithms can be computed much faster in asymptotic running time $L_p(1/2,1/2)$. The algorithm is well suited for a parallel implementation.

The largest general integers that can at present feasibly be factored using massively parallel computation have on the order of 130 decimal digits [1]. The factoring algorithm with the best conjectured asymptotic running time $L_m(1/3,c)$ for some constant $c<2$ is the number field sieve [13], but for the size of general integers m that can be factored within reasonable time a variant of the quadratic sieve with asymptotic running time $L_m(1/2,1)$ is more efficient [11]. The running time of both these algorithms is independent of the size of the factor that is found. The best known algorithm for finding factors of moderate size is the elliptic curve algorithm [12] which is with massively parallel computation successful for factors with up to 40 decimal digits [10, 20]. Its asymptotic running time is $L_p(1/2,\sqrt{2})$ where p is the factor to be found. It is the ratio $L_p(1/2,\sqrt{2})/L_p(1/2,1) = L_p(1/2,\sqrt{2}-1)$ of the running times of the elliptic curve factoring algorithm and the discrete logarithm algorithm [3] that provides a range for the size of the primes for which our public-key distribution system is both practical and secure.

It seems at present to be appropriate to choose 3 or 4 prime factors of between 70 and 80 decimal digits. To factor such a modulus is for all presently known factoring algorithms completely infeasible. The largest factor that has been found by the elliptic curve algorithm has 40 decimal digits [5]. Odlyzko [20] estimated that for a given computational effort the size of primes modulo which discrete logarithms can be computed is 10 to 15 digits smaller than the size of integers that can be factored. Hence with the same computational effort that was spent on the factorization of the 129-digit number of [1], one could compute discrete logarithms for 115-digit prime moduli. To find an 80 digit factor with the elliptic curve factoring algorithm takes roughly $L_{10^{80}}(1/2,\sqrt{2})/L_{10^{40}}(1/2,\sqrt{2}) \approx 3.2\cdot 10^6$ times longer than to find a 40 digit factor. On the other hand, computing discrete logarithms for a 70-digit or an 80-digit prime modulus is about $L_{10^{115}}(1/2,1)/L_{10^{80}}(1/2,1) \approx 18.000$ or $L_{10^{115}}(1/2,1)/L_{10^{70}}(1/2,1) \approx 1700$, respectively, times faster than for a 115-digit prime modulus.

We now give a brief asymptotic analysis of the work factor of our system. As mentioned above, the number field sieve is the asymptotically fastest known factoring algorithm. In order to ensure that the above mentioned asymptotic work factor $L_p(1/2,\sqrt{2}-1)$ is valid, i.e., that the elliptic curve factoring algorithm is faster than the number field sieve for finding a factor p of the modulus m, the number k of prime factors of m must be such that $L_{p^k}(1/3, c) > L_p(1/2, \sqrt{2})$. This condition is equivalent to

$$c(k \log p)^{1/3}(\log k + \log\log p)^{2/3} > \sqrt{2}(\log p)^{1/2}(\log\log p)^{1/2}$$

which is satisfied when $k = d\sqrt{\log p}$ for a constant $d > (\sqrt{2}/c)^3$. This analysis demonstates that future progress in computer technology is to the cryptographer's advantage (or to the cryptanalyst's disadvantage) and allows to increase the security of the system.

4. An Alternative Implementation

In the system discussed in Section 2, the trusted authority can feasibly compute discrete logarithms modulo the prime factors of m because these primes are chosen sufficiently small. As a consequence, several prime factors must be used to prevent feasible factorization of the modulus by a general-purpose factoring algorithm. In this section we suggest an alternative method for making the computation of discrete logarithms modulo the prime factors p_i of m feasible. There is no restriction on the size of the primes and therefore it is sufficient to use two primes of appropriate size.

There exists a discrete logarithm algorithm for $GF(p)$ due to Pohlig and Hellman [23] whose running time is proportional to the square root of the largest prime factor of $p-1$, if the factorization of $p-1$ is known. Hence the primes p_i can be chosen such that $(p_i-1)/2$ is the product of some primes of a certain relatively small size. Unfortunately, there also exists a special purpose factoring algorithm due to Pollard [24] that is particularly efficient for finding prime factors p for which $p-1$ has only relatively small prime factors. However, the running time of Pollard's algorithm is proportional to the largest prime factor of $p-1$ rather than its square root. Therefore there may exist a range for the size of the largest prime factors of $p_i - 1$ for which a system based on this idea is both practical and secure. A possible choice could be to let m be the product of 2 primes p_1 and p_2 of about 120 decimal digits each, where $(p_1-1)/2$ and $(p_2-1)/2$ both are the product of several 15-digit primes.

The choice $p_1 \equiv 3 \pmod 8$ and $p_2 \equiv 7 \pmod 8$ is particularly attractive because it implies that $(2/m) = -1$. Hence Alice's modified identity I'_A can therefore be defined as

$$I'_A = \begin{cases} I_A & \text{if } (I_A/m) = 1 \\ 2I_A & \text{if } (I_A/m) = -1 \end{cases}$$

which can be obtained easily from I_A without knowledge of the trapdoor and is guaranteed to have a discrete logarithm.

When the computational effort spent by the trusted authority is increased by a factor k, this forces an adversary to increase his computational effort by a factor k^2. Thus when k-fold faster computer hardware becomes available this system's security can also be increased by a factor of k. This system is asymptotically superior to the system of Section 2 for which the work factor could be increased only by a factor $k^{\sqrt{2}-1} = k^{.414}$.

5. Conclusions

A non-interactive public-key distribution system based on the Diffie-Hellman scheme with a composite modulus m has been proposed in which the modular exponentiation function contains as a trapdoor the factorization of the modulus. This solves an open problem suggested by Shamir in 1984 [27]. The trapdoor is known only to a mutually trusted authority setting up the system. The problem that the group Z_m^* is not cyclic and therefore not every element has a discrete logarithm is solved by squaring the identity number.

Clearly, the presented scheme can just as well be used in applications encorporating user interaction. In fact, the proposed scheme might be superior to previously proposed schemes in terms of the computational efficiency of the users' cipher key generation process because only a single modular exponentiation, but no signature verification, is required. A remarkable property of the system is that not only the cryptanalyst, but also the trusted authority must spend time super-polynomial in the input size. However, because the system is used for an appropriate fixed size of parameters, the trusted authority's computation is nevertheless feasible. Progress in computer technology can be exploited to increase the security of the system.

There may exist different approaches to making the discrete logarithm problem feasible only when given the factorization of the modulus. Any progress in the discrete logarithm problem not leading to a comparable progress in the factorization problem, especially when applicable to primes of a certain special form, has the potential of leading to an improvement of the presented system.

An interesting open question is whether it is possible to construct primes p of a special form containing a trapdoor such that computing discrete logarithms modulo p is feasible if and only if the trapdoor is known. One such method, which seems to be of little practical significance, was discovered by Odlyzko [20] who suggested to generate a prime p of the form $p = \sum_{i=1}^{d} a_i m^i$, where the a_i's are small numbers, and to exploit the fact that when m and d are carefully chosen and when the above representation of p is known, a faster number field sieve can be used for computing discrete logarithms than when this representation is not known. We hope that this paper stimulates research on special-purpose factoring and discrete logarithm algorithms.

Acknowledgements

We are grateful to Arjen Lenstra for helpful comments. We would also like to thank Tom Berson and Jim Massey for highly appreciated discussions. and K. Ohta for drawing our attention to the paper [19] written in Japanese, which describes a scheme having some similarities with the first scheme presented in this paper. The first author would like to thank Dr. P. Schmid and Martin Benninger of Omnisec AG for their comments and generous support of this work.

References

[1] D. Atkins, M. Graff, A.K. Lenstra, and P.C. Leyland, The magic words are squeamish ossifrage, to appear in Proc. of Asiacrypt '94, Wollongong, Australia, Nov. 28 - Dec. 1, 1994.

[2] E. Bach, Discrete logarithms and factoring, Technical Report UCB/CSD 84/186, Computer Science Division, University of California, Berkeley, June 1984.

[3] D. Coppersmith, A.M. Odlyzko and R. Schroeppel, Discrete Logarithms in $GF(p)$, *Algorithmica*, vol. 1, pp. 1-15, 1986.

[4] W. Diffie and M.E. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, vol. IT-22, pp. 664-654, Nov. 1976.

[5] B. Dixon and A.K. Lenstra, Massively parallel elliptic curve factoring, *Advances in Cryptology - EUROCRYPT '92*, Lecture Notes in Computer Science, Vol. 658, pp. 183-193, Berlin: Springer-Verlag, 1993.

[6] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions on Information Theory*, vol. IT-31, pp. 469-472, July 1985.

[7] M. Girault, Self-certified public keys, *Advances in Cryptology - EUROCRYPT '91*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, vol. 547, pp. 490-497, 1991.

[8] C.G. Günther, An identity-based key-exchange protocol, *Advances in Cryptology - EUROCRYPT '89*, Lecture Notes in Computer Science, vol. 434, Berlin: Springer Verlag, pp. 29-37, 1990.

[9] K. Koyama and K. Ohta, Identity-based conference key distribution systems, *Advances in Cryptology - CRYPTO '87*, Lecture Notes in Computer Science, vol. 293, Berlin: Springer Verlag, pp. 175-184, 1988.

[10] A.K. Lenstra, personal communication, 1991.

[11] A.K. Lenstra and M.S. Manasse, Factoring with two large primes, *Advances in Cryptology - EUROCRYPT '90*, Lecture Notes in Computer Science, vol. 473, Berlin: Springer Verlag, pp. 69-80, 1991.

[12] H.W. Lenstra, Factoring integers with elliptic curves, *Annals of Mathematics*, vol. 126, pp. 649-673, 1987.

[13] A.K. Lenstra, H.W. Lenstra, M.S. Manasse and J.M. Pollard, The number field sieve, *Proc. 22nd ACM Symposium on Theory of Computing*, pp. 564-572, 1990.

[14] A.K. Lenstra and M.S. Manasse, Factoring with electronic mail, *Advances in Cryptology - EUROCRYPT '89*, Lecture Notes in Computer Science, vol. 434, Berlin: Springer Verlag, pp. 355-371, 1990.

[15] T. Matsumoto and H. Imai, On the key predistribution system: a practical solution to the key distribution problem, *Advances in Cryptology - CRYPTO '87*, Lecture Notes in Computer Science, vol. 293, Berlin: Springer Verlag, pp. 185-193, 1988.

[16] U.M. Maurer, Fast generation of prime numbers and secure public-key cryptographic parameters, to appear in *Journal of Cryptology*, 1995.

[17] K.S. McCurley, A key distribution system equivalent to factoring, *Journal of Cryptology*, vol. 1, no. 2, pp. 95-106, 1988.

[18] G.L. Miller, *Riemann's hypothesis and tests for primality*, Journal of Computer and System Sciences, vol. 13, pp. 300-317, 1976.

[19] Y. Murakami and M. Kasahara, An ID-based key distribution system, Proc. of ISEC90, pp. 33-40, 1990 (in Japanese).

[20] A.M. Odlyzko, personal communications, 1991.

[21] T. Okamoto and K. Ohta, How to utilize the randomness of zero-knowledge proofs, *Advances in Cryptology - CRYPTO '90*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, vol. 537, pp. 456-475, 1991.

[22] E. Okamoto and K. Tanaka, Key distribution based on identification information, *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 4, pp. 481-485, May 1989.

[23] S.C. Pohlig and M.E. Hellman, An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance, *IEEE Transactions on Information Theory*, vol IT-24, pp. 106-110, Jan. 1978.

[24] J.M. Pollard, Theorems on factorization and primality testing, *Proc. Cambridge Philos. Society*, vol. 76, pp. 521-528, 1974.

[25] 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.

[26] R.J. Schoof, Elliptic curves over finite fields and the computation of square roots mod p, *Mathematics of Computation*, vol. 44, pp. 483-494, 1985.

[27] A. Shamir, Identity-based cryptosystems and signature schemes, *Advances in Cryptology - CRYPTO '84*, Lecture Notes in Computer Science, vol. 196, Berlin: Springer Verlag, pp. 47-53, 1985.

[28] H.N. Shapiro, *Introduction to the theory of numbers*, New York: Wiley, 1983.

[29] Z. Shmuely, Composite Diffie-Hellman public-key generating systems are hard to break, TR 356, CS Dept., Technion, Israel, Feb. 1985.

[30] S. Tsujii and T. Itoh, An ID-based cryptosystem based on the discrete logarithm problem, *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 4, pp. 467-473, May 1989.

[31] Y. Yacobi, A key distribution "paradox", *Advances in Cryptology - CRYPTO '90*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, vol. 537, pp. 268-273, 1991.