Minimum-Knowledge Schemes for low-power, low-memory Devices

T. Newe & T. Coffey
Data Communications Security Group,
Department of Electronic & Computer Engineering,
University of Limerick,
Ireland.
E-Mail: (Thomas.Newe@UL.ie) (Tom.Coffey@UL.ie)
Fax: 353-61-338176

Abstract

Minimum-knowledge identification/signature schemes enable principals prove their identities or the authenticity of their messages using an interactive challenge-response process. Due to their simplicity, strong security and relatively low computational requirements minimum-knowledge schemes are ideally suited for microprocessor based devices where the processing power and storage capacity is limited.

This paper discusses the operation of three landmark minimum-knowledge schemes, namely, the *Fiat-Shamir, Ohta-Okamoto and Guillou-Quisquater* minimum-knowledge identification and signature schemes. The relationship between the principals' storage and data exchange requirements and the computational complexity of the schemes is tabularised.

Key-Words: Minimum-Knowledge, Authentication, Signature, Ohta-Okamoto, Fiat-Shamir, Guillou-Quisquater, mobile.

1. Introduction

Minimum-knowledge identification schemes provide a means by which a claimant can prove its identity to a verifier. This process involves a series of random challenges issued by the verifier and the associated claimant responses. This interaction provides proof to a verifier that a claimant possesses a secret, without divulging the secret itself.

Minimum-knowledge identification schemes are only useful against external threats when both the claimant and verifier co-operate [1]. Using a minimum-knowledge identification scheme it is conceivable that a verifier could forge a credible transcript of an imaginary communication with a claimant by carefully choosing both the questions and answers in the dialogue. With signature schemes only real communication with a claimant can generate a credible transcript, as the signature for a message is generated entirely by the claimant and not by dialogue with the verifier.

The security of a minimum-knowledge scheme, generally represented as 2^{-x}, refers to the probability of success that a forger has to forge an identity. Taking a security level of 2^{-30} for example, a forger has a probability of success of $0.93 * 10^{-9}$, or 1 chance in every 10^9 identification procedural attempts. Even a patient adversary with an unlimited budget, who tries to misrepresent himself 1000 times daily, is expected to succeed only once every 3000 years. With signature schemes a forger will know in advance if the signature will be accepted as valid, since the signature is generated entirely by the claimant. If the forger experiments with 2^x random values it is possible to find a signature that will be accepted by a verifier. Thus, a large value of x must be chosen to make this type of attack infeasible.

When considering the implementation of minimum-knowledge schemes on low-power/low-memory devices, a number of features need to be analysed: the claimants' computational requirements, the claimants' storage requirements, the data exchange requirements, and the security of the scheme.

This paper outlines the operation of the Fiat-Shamir, Ohta-Okamoto and Guillou-Quisquater minimum-knowledge identification/signature schemes. An analysis of the schemes is presented ranging over the claimants' storage demands, data exchange requirements, computational complexity and security.

2. The Fiat-Shamir Scheme

The Fiat-Shamir scheme [1] is based on the difficulty of extracting modular square roots, where the factors of the public modulus n are unknown. The public modulus n is the product of two large secret prime numbers, p and q ($n = p * q$, where p and q ≥ 256 bits). The modulus n is common to all principals in the scheme. Extracting modular roots is as difficult as factoring the public modulus n [4]. Factoring n ($\geq$ 512 bits) is considered to be beyond the capabilities of current factoring algorithms [5, 6, 7, 8].

The Fiat-Shamir scheme requires the use of a trusted centre to configure the claimants authentication device. Each device contains a public modulus n which is common to all principals, an identification string I and the claimant's secrets $s_1, s_2,..,s_k$. The public modulus n is the product of two large secret prime numbers, p and q ($\geq$ 256 bits). The identification string I is not secret, so it can be stored in a non-secure area of the claimant's device. The claimant's secrets $s_1, s_2,...,s_k$ are calculated as follows:

1. Compute the values $v_j = f(I, j)$, for small values of j. I is concatenated with j and then hashed using a public one-way hash function f.
2. Pick k distinct values of j for which v_j is a Quadratic Residue (mod n). A Quadratic Residue is defined as a number less than n that has modular square roots. Then compute the smallest square root s_j of v_j^{-1} (mod n) $\Rightarrow s_j = \sqrt{1/v_j}$ (mod n).

2.1 Fiat-Shamir identification scheme

A claimant's proof of identity is confirmed if a verifier establishes that the claimant has knowledge of its secrets $s_1, s_2,..,s_k$. The procedure is outlined in Fig. 1.

1. The claimant transmits its identification string I to the verifier.
2. The verifier generates $v_j = f(I , j)$, for j = 1,..., k

Steps 3 to 6 inclusive are repeated t times.

3. The claimant transmits $X_i = r_i^2$ (mod n) to the verifier, where and r_i is a random number, $r_i \in [0, n)$.
4. The verifier challenges the claimant with a random binary vector $(a_{i1} ,..., a_{ik})$.
5. The claimant responds to the verifier's challenge with $Y_i = r_i \prod_{aij=1} s_j \pmod{n}$.
6. The verifier checks that: $X_i = Y_i^2 \prod_{aij=1} v_j \pmod{n}$.

   This establishes, if true for all i, that the claimant posses the secret S_j's, since:

$$Y_i^2 \prod_{aij=1} v_j \pmod{n}$$

$$= r_i^2 \prod_{aij=1} s_j^2 v_j \pmod{n}$$

$$= r_i^2 \pmod{n} = X_i$$

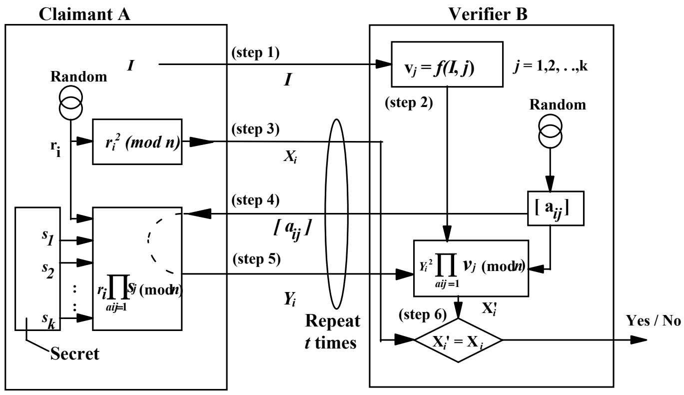

Fig.1: Fiat-Shamir identification scheme

2.2 Fiat-Shamir signature scheme

A claimant generates a digital signature for message m as follows:

1. The claimant picks random $r_1,\ldots,r_t \in [0, n)$ and computes $x_i = r_i^2$ (mod n).
2. The claimant computes $f(m, x_1,\ldots, x_t)$ and uses its first kt bits as a_{ij} values where $(1 \leq i \leq t, 1 \leq j \leq k)$, and f is a public one-way hash function.
3. The claimant computes : $Y_i = r_i \prod_{a_{ij}=1} s_j \pmod{n}$

   for i = 1, …, t and sends I, m, the a_{ij} vector and y_i to the verifier.

The signing procedure is illustrated in Fig. 2.

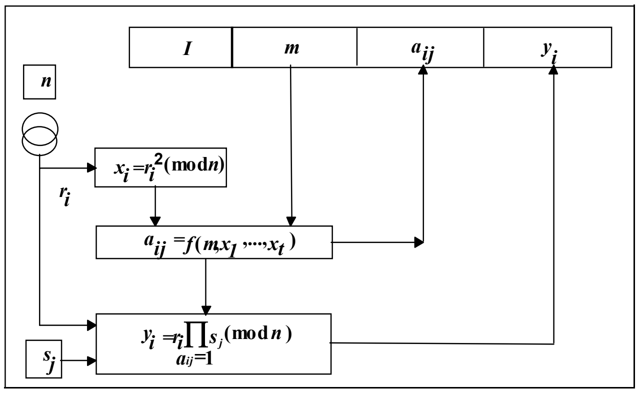

Fig. 2: Signature generation in the Fiat-Shamir scheme

A verifier authenticates a claimant's signature on message m as follows:

1. The verifier computes $v_j = f(I, j)$, for $j = 1,\ldots, k$.
2. The verifier computes: $z_i = y_i^2 \prod_{a_{ij}=1} v_j \pmod{n}$, for $i = 1,\ldots, t$.
3. The verifier establishes that the first kt bits of $f(m, z_1, \ldots, z_t)$ are equal to a_{ij}.
4. If $z_i = x_i$ then the claimant must have signed message m.

Proof: $z_i = y_i^2 \prod_{aij=1} v_j \pmod{n}$

$= r_i^2 \prod_{aij=1} s_j^2 v_j \pmod{n}$

$= r_i^2 \pmod{n} = x_i$

The signature verification procedure is illustrated in Fig. 3.

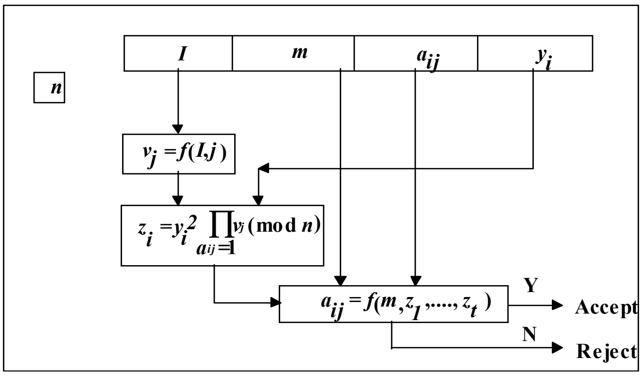

Fig. 3: Signature verification in the Fiat-Shamir Scheme

2.3 Security of the Fiat-Shamir scheme

The security level of the Fiat-Shamir scheme is represented as 2^{-kt}, where the product kt typically ranges from 30 to 72. Parameter t controls the number of iterations of the algorithm, while k specifies the number of secrets stored on the claimant's smart card. The values chosen for k and t depend on the desired compromise between algorithm speed and claimant storage. In the signature scheme, in order to make it infeasible for an adversary to forge a signature, the one-way hash function f is assumed to be non invertible and the product kt of the order 72.

3. The Ohta-Okamoto Scheme

The Ohta-Okamoto scheme [2] is based on the difficulty of extracting the L^{th} roots mod n when the factors of n are unknown. The scheme is particularly suitable for use with low memory devices, since it minimises the claimant's storage requirements and the amount of data exchanged during a verification [9].

In the Ohta-Okamoto scheme a trusted centre generates a public modulus n, which is the product of two secret prime numbers p and q (≥ 256 bits), and a public integer L. The value of L is chosen as a compromise between storage and security and is generally ≥ 30 bits in size. Parameters n and L are common to all principals in the domain of the trusted centre.

The claimant generates and publishes its own identity string I, where $I^{-1} = S^L \pmod{n}$, and S is the claimant's secret random integer, where $S \in [0,n)$. S is stored in secure memory. Since parameters L and n are publicly know, they can be stored in non-secure memory.

3.1 Ohta-Okamoto identification scheme

The Ohta-Okamoto identity verification procedure is outlined in steps 1-4.

1. The claimant generates a random number $R \in Z_n$, and sends $X = R^L \pmod{n}$ to the verifier.
2. The verifier challenges the claimant with a random number $E \in Z_L$.
3. The claimant responds to the verifiers challenge with Y, where $Y = R * S^E \pmod{n}$.
4. The verifier checks that $Y^L * I^E \equiv X \pmod{n}$.

If all t checks return a positive answer then the claimant has been verified. These checks establish that the claimant knows the secret S, since:

$$Y = R * S^E \pmod{n}$$

$$Y^L = R^L * S^{LE} \pmod{n}$$

$$= X * S^{LE} \pmod{n}$$

$$= X * I^{-E} \pmod{n}$$

$$\therefore \quad Y^L * I^E = X \pmod{n}$$

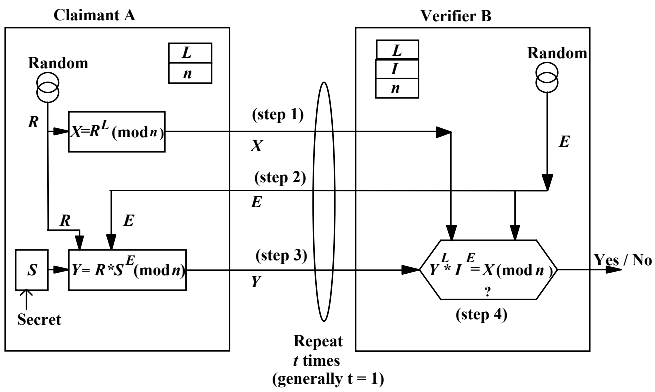

Fig. 4: Ohta-Okamoto identification scheme

If L is sufficiently large (≥ 30 bits) then t and $k = 1$, where t represents the number of cycles required and k represents the number of secrets which must be stored.

3.2 Ohta-Okamoto signature scheme

Signing a message M requires the availability of the public integer L, a random number R ($1<R < n$), and the claimant's secret S. A digitally signed message comprises a triplet M, E, Y, where M is a message, $E = f(M, X) \in Z_L$, f is a public one-way hash function, and $Y = R * S^E \pmod n$.

The Ohta-Okamoto signature generation procedure is illustrated in Fig. 5.

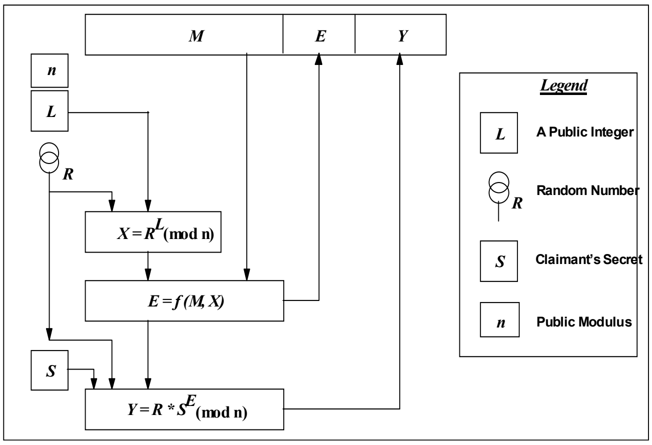

Fig. 5: Signature generation in the Ohta-Okamoto Scheme

A digitally signed message is verified by ascertaining that $f(M, Y^L * I^E \pmod n)) \equiv f(M, X)$.

The signature verification procedure is illustrated in Fig. 6.

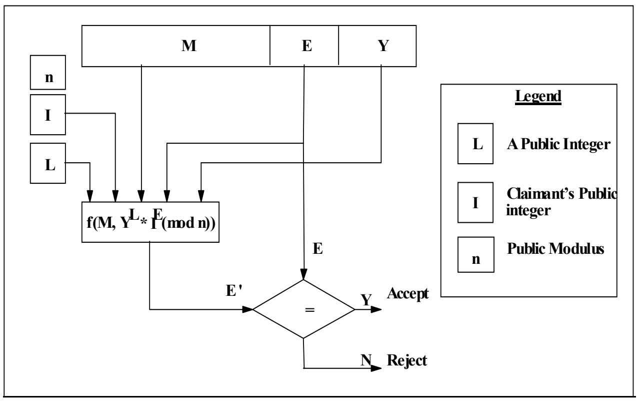

Fig. 6: Signature verification in the Ohta-Okamoto Scheme

3.3 Security of the Ohta-Okamoto scheme

The security of the Ohta-Okamoto scheme is represented as L^{-kt}, where t is the number of challenge-response cycles and k is the number of claimant secrets. If L is chosen to be large ($L \geq 2^{72}$), it makes it infeasible for an adversary to forge a signature and also with this size of L it is still possible to have $t = k = 1$. This reduces the communication and storage requirements of the scheme to a minimal level.

4. The Guillou-Quisquater Scheme

The Guillou-Quisquater scheme [3] uses a single challenge cycle. The scheme requires the storage of only one authentication secret on the claimant's device. While the scheme has a low memory requirement and a single verification cycle, its computational complexity is increased by a factor of 2 or 3, compared to the Fiat-Shamir scheme.

The scheme requires a trusted authority to generate the following information:

(i) A common public modulus n where n is the product of two large secret prime numbers, p and q (both ≥ 256 bits).

(ii) A public system constant v. The size of v is generally about 30 bits, which is determined as a compromise between speed and security.

(iii) An identification string I. The string is public and can be stored in the open memory of the authentication device, together with n and v.

(iv) A single secret B is calculated for each claimant from the following congruence: $J * B^v \pmod n = 1$, where $n = p * q$, and $J = f(I\|I)$.

This congruence can be solved using Euler's function, provided p and q is known. J is the shadowed identity of I. The shadowed identity is a hash of ($I\|I$) (I concatenated with I). The length of the string J, depends on the hashing algorithm used.

4.1 Guillou-Quisquater identity verification

The Guillou-Quisquater identity verification procedure is outlined in steps 1-5.

1. The claimant transmits its identification string I and a test value T to the verifier, where $T = r^v \pmod n$ and $r \in [1, n-1]$.
2. The verifier challenges the claimant with a random number $d \in [0, v-1]$.
3. The claimant responds to the verifiers challenge with $D = r * B^d \pmod n$.
4. The verifier computes its own test value $T' = J^d * D^v \pmod n$, where $J = f(I\|I)$.

5. If $T' = T$ the identity of the claimant has been verified.

The verifier has now established that the claimant posses the secret *B*, since:

$$\begin{aligned} T' &= J^d * D^v \pmod n \\ &= J^d (r * B^d)^v \pmod n \\ &= (J * B^v)^d * r^v \pmod n \\ &= (1)^d * r^v \pmod n \\ &= T \end{aligned}$$

The Guillou-Quisquater identity verification procedure is illustrated in Fig. 7.

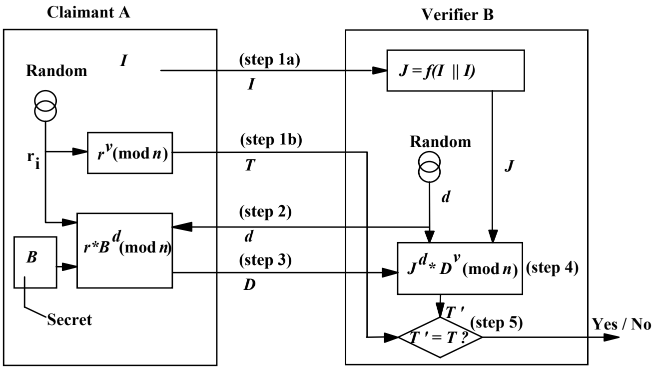

Fig. 7: Guillou-Quisquater identification scheme

4.2 Guillou-Quisquater signature scheme

A claimant generates a digital signature for message *M* by following steps 1-4:

1. The claimant generates $T = r^v \pmod n$, where $r \in [1, n-1]$.
2. The claimant computes $d = f(M, T)$. *M* is the message to be signed, and *f* is a public one-way hash function.
3. The claimant computes $D = r * B^d \pmod n$.
4. The claimant transmits *M*, *I*, *d*, and *D* to the verifier.

The signing procedure is illustrated in Fig. 8.

A verifier authenticates a claimant's signature on message *M* by following steps 1-3:

1. The verifier generates its own test value $T' = J^d * D^v \pmod n$, where $J = f(I \| I)$.
2. The verifier computes $d' = f(M, T')$.
3. If $d' = d$ then a verifier has authenticated a claimants signature on message *M*.

The signature verification procedure is illustrated in Fig. 9.

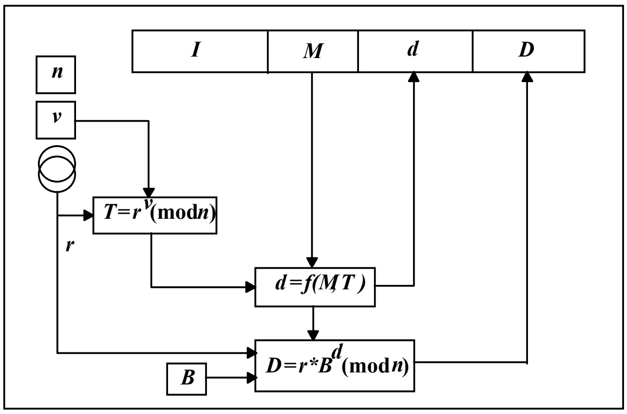

Fig. 8: Signature generation in the Guillou-Quisquater Scheme

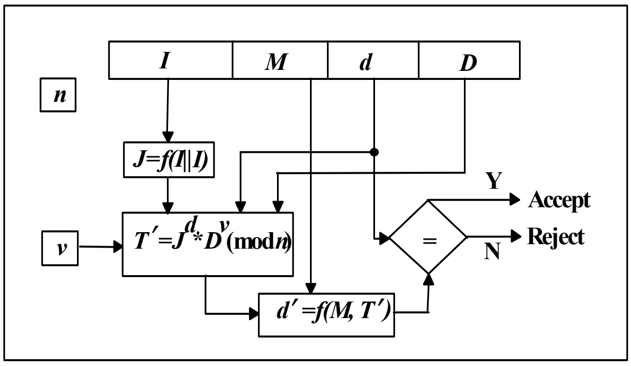

Fig. 9: Signature verification in the Guillou-Quisquater Scheme

4.2 Security of the Guillou-Quisquater scheme

In the Guillou-Quisquater scheme the security level is represented by 2^{-v}. The security of the scheme increases as *v* increases. The security of the signature scheme assumes that *f* is a true one-way hash function. If *f* is weak, then forgery of a signed document may be possible.

5. Conclusion

This paper discussed three of the most popular minimum-knowledge schemes suitable for claimant identification and digital signature generation and verification with low-power and low-memory devices. The operation of the schemes and the relationships between the claimants' storage, data exchange requirements and computational complexity was investigated. Tables 1 and 2 summarise these relationships for security levels of 2^{-30} and 2^{-72} respectively. All secrets and public moduli are assumed 512 bits in size.

The 2^{-72} security level is appropriate for both the identification and signature schemes of all the discussed schemes. These relationships are important in determining the suitability of minimum-knowledge schemes for use on low-power, low-memory devices such as mobile phones.

	Fiat-Shamir id. scheme	Fiat-Shamir sig. scheme	Guillou-Quisquater id. scheme	Guillou-Quisquater sig. scheme	Ohta-Okamoto id. scheme	Ohta-Okamoto sig. scheme
t	5	5	1	1	1	1
k	6	6	-	-	1	1
v (bits)	-	-	30	30	-	-
***l* (bits)**	-	-	-	-	30	30
EEPROM req. (bytes)	650	650	332	332	196	196
No. of bytes exchanged	850	730	332	468	200	400
Average # modular multiplies	**t(k+2)/2** = 20	**t(k+2)/2** = 20	**(5v+2)/2** = 76	**(5v+2)/2** = 76	**(5*l*+2)/2** = 76	**(5*l*+2)/2** = 76
Security level	2^{-kt}	2^{-kt}	2^{-v}	2^{-v}	L^{-kt}	L^{-kt}

Table 1. Comparison of minimum-knowledge schemes for a security level of 2^{-30}

t	8	8	1	1	1	1
k	9	9	-	-	1	1
v (bits)	-	-	72	72	-	-
***l* (bits)**	-	-	-	-	72	72
Memory req. (bytes)	850	850	337	337	201	201
No. of bytes exchanged	1242	930	337	473	210	410
Average # modular multiplies	**t(k+2)/2** = 44	**t(k+2)/2** = 44	**(5v+2)/2** = 181	**(5v+2)/2** = 181	**(5*l*+2)/2** = 181	**(5*l*+2)/2** = 181
Security level	2^{-kt}	2^{-kt}	2^{-v}	2^{-v}	L^{-kt}	L^{-kt}

Table 2. Comparison of minimum-knowledge schemes for a security level of 2^{-72}

References

1 Fiat, A. and Shamir, A.: 'How to prove yourself : Practical Solutions to Identification and Signature Problems'. Proceedings Crypto `86, Lecture notes in Computer Science, Vol. 263, Springer Verlag, 1987

2 Ohta, K. and Okamoto, T.: 'A modification of the Fiat-Shamir scheme'. Proceedings Crypto `88, Lecture notes in Computer Science, Vol. 403, Springer Verlag, 1987

3 Guillou, L.C. and Quisquater, J.J.: 'A practical Zero-Knowledge protocol fitted to security microprocessors minimising both transmission and memory'. Proceedings Eurocrypt `88, Lecture notes in Computer Science, Vol. 330, Springer Verlag, 1988

4 Alexi, W., Chor, B., Goldreich, O., and Schnorr, C.P.: 'RSA and Rabin Functions: Certain Parts are as Hard as the Whole'. SIAM Journal on Computing' vol 17, Apr 1988

5 Atkins, D., Graff, M., Lenstra, A.K. and Leyland, P.C.: 'The magic words are squeamish ossifrage'. Advances in Cryptology - Asiacrypt '94, Springer-Verlag, 1995

6 Buhler, J.P., Lenstra, H.W., and Pomerance, C.: 'The development of the number field sieve'. Volume 1554 of Lecture Notes in Computer Science, Springer-Verlag, 1994

7 Buchmann, J., Loho, J., and Zayer, J.: 'An implementation of the general number field sieve'. Advances in Cryptology - Crypto '93, pages 159-166, Springer-Verlag, 1994

8 Dodson, B. and Lenstra, A.K.: 'NFS with four large primes: An explosive experiment'. Advances in Cryptology - Crypto '95, pages 372-385, Springer-Verlag, 1995

9 Coffey, T. and Newe, T.: 'Realisation of a Minimum-Knowledge identification and signature scheme'. Computers & Security, Elsevier, Vol 17, No 3, 1998