14

Hidden signature schemes based on the discrete logarithm problem and related concepts

Patrick Horster · Markus Michels · Holger Petersen
Theoretical Computer Science and Information Security
University of Technology Chemnitz-Zwickau
Straße der Nationen 62, D-09111 Chemnitz, Germany
E-mail: {pho,mmi,hpe}@informatik.tu-chemnitz.de

Abstract

One year ago, four classes of blind signature schemes have been introduced: the *hidden*, the *weak blind*, the *interactive blind* and the *strong blind* signature schemes. In this paper we present several hidden and weak blind signature schemes based on the Meta-ElGamal signature scheme and a hidden signature scheme based on the Meta-Message recovery signature scheme. All these schemes can be used in many applications like pseudonymous access control for credentials or for obtaining self-certified public keys. Therefore they can also be used for authentication and authentic key exchange schemes. Most of the new variants are very efficient, as they take advantage of the efficient variants of the Meta signature schemes.

Keywords

Cryptography, digital signatures, hidden signatures, blind signatures

1 INTRODUCTION

The concept of blind signature schemes has been proposed by David Chaum (1982). Since then there have been many efforts to construct blind signature schemes. They can be used in many applications like pseudonymous credentials, electronic cash or anonymous access control (Brands, 1993, Camenisch, Piveteau and Stadler, 1994, Horster, Michels and Petersen, 1994).

In all cases the security of the protocols is considered at the moment when the notary signs the document. It isn't necessary to check, whether the signature keeps it's anonymity when it is presented later to the notary who can store the signature parameters of all signed documents, because the signatures are untraceable. This new aspect leads to a classification in this paper, because all of our proposed schemes are only anonymous during signature generation but they are not at the time of signature verification by the notary. This characteristic can be useful in many applications. Some of them are discussed at the

R. Posch (ed.), *Communications and Multimedia Security*

end of this paper. Additionally it is also possible to blind only the signature parameters but not the message, which can be very useful for self-certified public keys and the related authentication and authentic key exchange protocols (Horster and Knobloch 1991, Horster and Petersen, 1994, Horster, Michels and Petersen, 1994).

The security of all protocols is based on the discrete logarithm problem, which has the advantage compared with the factorization problem, that all users have a common modul and that several structures like GF(p), GF(p^n), elliptic curves or Lucas functions are known to implement cryptosystems based on this problem.

In section two we present our classification of blind signature schemes. Section three gives a brief introduction into the Meta signature scheme with appendix, which is necessary to develop the hidden signatures in section four. Sections five gives an example for weak blind signatures with appendix. In section six the Meta-Message recovery signature scheme is briefly reviewed. A hidden and a weak blind variant of this scheme are presented in the next two sections. Further we present some important applications of all schemes.

2 CLASSIFICATION OF BLIND SIGNATURES

There are three parties in a blind signature protocol: The *owner* of the signature *Alice*, who chooses the message, the *signer* or *notary Nancy*, who generates the blind signature on it and the *verifier Bob*, who checks the validity of the (unblinded) signature. The notary and the verifier can be one person. We use the notation established by ElGamal (1994).

Depending on the strength of anonymity given by the signature, we can distinguish four classes of blind signatures (Horster and Petersen, 1994):

1. **Hidden signatures**
   This kind of signatures can be distinguished into two subtypes in case of ElGamal like signature schemes:

   (a) *Message hidden* signatures, in which the notary doesn't know the document he signs, but he knows the signature parameters. If he stores them, he can recognize the signature later by comparing them with a given signature.
   (b) *Parameter hidden* signatures, in which the notary doesn't know the signature parameter s but he knows the message m and the parameter r.

2. **Weak blind signatures**
   The notary doesn't know the message and the signature parameters, but he can recognize them later, as there exists a relationship between the blinded signature parameters and the unblinded parameters.
3. **Interactive blind signatures**
   The notary doesn't know the message and the signature parameter r. By demonstrating the knowledge of the signature parameter s with an interactive proof, the signer doesn't get any knowledge about the relation between the blinded and the unblinded signature parameters. So he doesn't get any information about the relationship between a given document and his stored parameters. If it's necessary, the owner can be forced in case

of a complaint to show the signature parameter s which effects that the notary can discover the relationship.

4. **Strong blind signatures**
   The signer couldn't see a relation between any of his stored parameters and the shown signature parameters, so that the signature is really anonymous and can't be disclosed at any time.

In the following we will only consider the first three classes. Examples of the strong blind signature can be found in Okamoto (1992), Camenisch, Piveteau and Stadler (1994) and Horster, Michels and Petersen (1994).

3 THE META SIGNATURE SCHEME WITH APPENDIX

The Meta signature scheme with appendix has been proposed by Horster, Michels and Petersen (1994). First we briefly describe the basic ElGamal signature scheme (ElGamal, 1984) and then we introduce the Meta signature scheme with appendix for one message block.

The basic ElGamal signature scheme

For an ElGamal signature (ElGamal, 1984) the trusted authority chooses a large prime p and a generator $\alpha \in \mathbf{Z}_p^*$ with order $p-1$. p and α are public system parameters and authentically known to all users. The signer *Alice* chooses a random number $x_A \in \mathbf{Z}_{p-1}$ and computes $y_A := \alpha^{x_A} \pmod p$. She publishes y_A and keeps x_A secret. These values are constant for all messages to be signed. To sign a message $m \in \mathbf{Z}_{p-1}$ Alice chooses a random number $k \in \mathbf{Z}_{p-1}^*$. She computes $r := \alpha^k \pmod p$ and solves the congruence

$$m \equiv x_A \cdot r + k \cdot s \pmod{p-1} \tag{1}$$

for the parameter s. The triple $(m; r, s)$ is the signed message. It can be verified by checking the congruence

$$\alpha^m \equiv y_A^r \cdot r^s \pmod p. \tag{2}$$

The Meta signature scheme

Instead of signature generation by the equation (1) we can also choose the general equation

$$A \equiv x_A \cdot B + k \cdot C \pmod q \tag{3}$$

with $q \in \mathbf{P}$, $q|(p-1)$, and choose A, B, C as general functions $e, f, g : \mathbf{Z}_q{}^3 \to \mathbf{Z}_q$ with arguments m, r and s. As $m \in \mathbf{Z}_{p-1}$ we imply that m is reduced modulo q before it is used as an argument but in the following description we omit this for the sake of clarity.

The parameter s should either be used as argument in only one of the three functions or the functions have to be chosen carefully, such that the signature equation can be solved. Also all of the parameters m, r, s have to occur at least once. If two or three functions use exactly the same arguments, then they should be chosen as different operations. The occurrence of the insecure $rs-$ and $ms-$variants (Horster, Michels and Petersen, 1994),

where the parameters r and s (m and s) occur exactly in one of the three functions e, f and g together but neither r nor s (m nor s) occurs in one of the two other, should be avoided. All four conditions apply also for equivalent variants, in which the signature equations can be transformed into each other. Furthermore none of the three functions should be equal to zero. To get efficient variants, the functions should be chosen, such that s can be easily extracted (e.g. without inversions). It's also an advantage to choose one of the functions equal to one, to obtain an efficient signature verification. This verification is done by checking the equation

$$\alpha^A \equiv y_A^B \cdot r^C \pmod{p}. \tag{4}$$

Additionally we can generalize the computation of the parameter r by choosing $r' := \alpha^k \pmod{p}$ and computing $r := d(r', m)$ with a suitable function $d : \mathbf{Z}_p{}^2 \to \mathbf{Z}_p$. In this case, the verification equation (4) modifies to

$$r = d\left(\alpha^{A \cdot C^{-1}} \cdot y_N^{-B \cdot C^{-1}} \pmod{p}, m\right) \tag{5}$$

It also possible to vary the mode of operation that determines the group orders and the length of the parameters (Horster, Michels and Petersen, 1994):

- XL: ElGamal mode with $|p| = |q| = 512$,
- L: Schnorr mode (Schnorr, 1989) with $|p| = 512, |q| = 160$,
- M: DSA mode (NIST, 1991) with $|p| = 512, |q| = 160$, r reduced modulo q, and
- S: small mode (Schnorr, 1989, Knobloch, 1994) with $|p| = 512, |q| = 160$ and a q_1 bit number $h(r)$ $(50 \leq |q_1| \leq 160)$ reduced by any hash function h.

These modes are implicitely defined by the choice of q which specifies the order of the subgroup and the function d, wich fixes the size of the parameter r.

As there are numerous variants, in the following we will only consider some efficient special cases of permutations, namely to choose A, B, C as a permutation of one of the following five types EG I – EG V, which have been analyzed in detail by Horster, Michels and Petersen (1994):

$$\text{EG I: } (m, r, s), \text{EG II: } (f(m,r), s, 1), \text{EG III: } (f(m,r), g(m,s), 1),$$
$$\text{EG IV: } (f(m,r), g(r,s), 1), \text{EG V: } (f(m,s), g(r,s), 1).$$

The functions $f, g : \mathbf{Z}_q{}^2 \to \mathbf{Z}_q$ have to be invertible in the argument s to guarantee the solubility of the general signature equation (3) for the signature parameter s. For every type we get one of the following six permutations of the coefficients, which are enumerated by *No.* 1 – 6:

$$1: (a,b,c) \quad 2: (a,c,b) \quad 3: (c,b,a)$$
$$4: (c,a,b) \quad 5: (b,c,a) \quad 6: (b,a,c)$$

For example $(a, b, c) = (m, r, s)$ in *Type* EG I and $(a, b, c) = (f(m, r), s, 1)$ in *Type* EG II.

Combining the described variations we get a description of the Meta signature scheme with appendix which can be written as

$$MEG = (Mode.Type.No, d, e, f, g).$$

The parameters are chosen in the following way:

- $Mode \in \{\text{XL, L, M, S}\}$ specifies the mode of operation,
- $No \in \{1, 2, 3, 4, 5, 6\}$ gives the number of the permutation,
- $Type \in \{\text{EG I, EG II, ..., EG XI}\}$ gives the type of permutation,
- $d : \mathbf{Z}_p{}^2 \rightarrow \mathbf{Z}_p$ specifies the computation of r,
- $e, f, g : \mathbf{Z}_q{}^3 \rightarrow \mathbf{Z}_q$ invertible in the argument s.

In a simplified (non-redundant) manner, we can also describe the Meta scheme by the tuple $(Mode, d, e, f, g)$ but then we loose useful structural information for the security analysis. Therefore we prefer the first notation even if it contains redundancy.

4 HIDDEN SIGNATURES WITH APPENDIX

The following protocols are based on the ideas of the testimonial scheme (Horster and Knobloch, 1991), the hidden signature scheme (Horster and Petersen, 1994) and the Meta signature scheme with appendix. We have to decide, whether the parameter s or the message m is covered. This results in a classification into parameter hidden and message hidden signatures.

First we give a brief review on the first parameter hidden signature scheme, the *testimonial scheme*, then we present a general approach for blinding the parameter C of the Meta signature scheme with appendix.

The testimonial scheme

For a parameter hidden signature, the notary Nancy chooses x_N and y_N like the signer does in the ElGamal signature scheme. The owner Alice, who wants to get the hidden signature on the message $m \in \mathbf{Z}_p$, chooses a random $h \in \mathbf{Z}_{p-1}^*$, computes $\beta = \alpha^h \pmod p$ and passes β, m to the notary. Now, the notary chooses a random $k \in \mathbf{Z}_{p-1}^*$ and computes $r := \beta^k \pmod p$. The parameters k and h should never be reused for another signature. The notary solves the congruence $x_N \cdot r + \tilde{s} \cdot k \equiv m \pmod{p-1}$ for the parameter $\tilde{s}$ and passes the signature $(r, \tilde{s})$ to the owner A, who computes $s = \tilde{s} \cdot h^{-1} \pmod{p-1}$. The triple $(m; r, s)$ is the signed message. The signature can be verified by checking the equation $\alpha^m \equiv y_N^r \cdot r^s \pmod p$.

Reconstructing $s \in \mathbf{Z}_{p-1}^*$ by the notary is equivalent to compute the discrete logarithm $\log_\alpha(\alpha^h) = h$ since $h \equiv s^{-1} \cdot \tilde{s} \pmod{p-1}$.

General approach

To develop a hidden signature scheme from the Meta signature scheme with appendix, we use the following general approach:

1. Make sure, that the hidden parameter s or the hidden message m appears only as argument in the coefficient C but not in A and B.
2. Write the coefficient C as a product of the hidden parameter and the remaining part of C.
3. Alice uses a random blinding factor $h \in \mathbf{Z}_q^*$.
4. Nancy chooses the generator $\beta := \alpha^h \pmod q$ for signature generation.
5. She solves the signature equation and sends the hidden parameter to Alice, who computes the unblinded parameter by multiplicating it with h or h^{-1}.

4.1 Message hidden signatures

The trusted authority chooses large primes p, q with $q|(p-1)$ and a generator $\alpha \in \mathbf{Z}_p^*$ of a multiplicative subgroup of order q. To get a message hidden signature on the message m, Alice and Nancy carry out the following steps:

1. Alice chooses a random number $h \in \mathbf{Z}_q^*$, computes $\beta := \alpha^h \pmod p$ and $\tilde{m} := m \cdot h \pmod q$.
2. She transmits $\tilde{m}$ and β to Nancy.
3. Nancy chooses a random $k \in \mathbf{Z}_q^*$, computes $r := \beta^k \pmod p$ and signs the message $\tilde{m}$ by solving the congruence (6) for the parameter s.

$$A \equiv x_N \cdot B + k \cdot \tilde{m} \cdot C \pmod q \tag{6}$$

   A, B and C are chosen as general functions e, f, g with arguments r and s.
4. Nancy transmits the signature (r, s) to Alice.

The tuple (r, s) is a message hidden signature on the message m. It can be verified by the following congruence:

$$\alpha^A \equiv y_N^B \cdot r^{m \cdot C} \pmod p \tag{7}$$

This congruence holds due to the following equation:

$$y_N^B \cdot r^{mC} \equiv \alpha^{x_N B} \cdot \beta^{kmC} \equiv \alpha^{x_N B} \cdot \alpha^{hkmC} \equiv \alpha^{x_N B + k\tilde{m}C} \equiv \alpha^A \pmod p.$$

Hence the Meta-Message hidden signature scheme can be written as

$$MMH = (Mode.Type.No.e.f.g)$$

with $Type \in \{\text{MH I, MH II, MH III, MH IV, MH, V}\}$ and the related choices for $Mode$ and No (see section 3). Note, that the function d must be chosen as $d(r', m) = r'$, as the message should appear only as argument of C. Table 1 gives a survey of the most efficient variants of the message hidden signature scheme.

Theorem 1 *The signature scheme described in steps 1.-4. above is a message hidden signature scheme.*

Proof. As the signature parameters (r, s) are not blinded, this is only a hidden signature. The notary doesn't know the message m during signature generation which is blinded by the parameter h. Reconstructing m from $\tilde{m}$ is equivalent to the discrete logarithm problem $\log_\alpha(\beta) := h \pmod p$ since $h \equiv \tilde{m} \cdot m^{-1} \pmod q$. □

Efficiency considerations

The efficient variants have been proposed in table 1. Because in all of them, one of the coefficients A, B, C is equal to one, they have efficient signature verification with only two exponentiations. Among these variants, those are most efficient, in which the parameter s is an argument of the coefficient A and doesn't occur elsewhere, because in these cases Nancy needs no inversion for signature generation. Namely these are the variants MH I.3, MH II.4, MH III.4 and MH IV.2.

Var.	A	B	C	signature generation	verification
MH I.3	s	r	1	$s \equiv x_N r + k\tilde{m}$	$\alpha^s \equiv y_N^r \cdot r^m$
MH I.5	r	s	1	$r \equiv x_N s + k\tilde{m}$	$\alpha^r \equiv y_N^s \cdot r^m$
MH II.2	1	s	$f(r)$	$1 \equiv x_N s + k\tilde{m} f(r)$	$\alpha \equiv y_N^s \cdot r^{mf(r)}$
MH II.4	s	1	$f(r)$	$s \equiv x_N + k\tilde{m} f(r)$	$\alpha^s \equiv y_N \cdot r^{mf(r)}$
MH III.1	1	$f(m,r)$	$g(s)$	$1 \equiv x_N f(m,r) + k\tilde{m} g(s)$	$\alpha \equiv y_N^{f(m,r)} \cdot r^{mg(s)}$
MH III.2	1	$g(m,s)$	$f(r)$	$1 \equiv x_N g(m,s) + k\tilde{m} f(r)$	$\alpha \equiv y_N^{g(m,s)} \cdot r^{mf(r)}$
MH III.4	$g(m,s)$	1	$f(r)$	$g(m,s) \equiv x_N + k\tilde{m} f(r)$	$\alpha^{g(m,s)} \equiv y_N \cdot r^{mf(r)}$
MH III.6	$f(m,r)$	1	$g(s)$	$f(m,r) \equiv x_N + k\tilde{m} g(s)$	$\alpha^{f(m,r)} \equiv y_N \cdot r^{mg(s)}$
MH IV.2	1	$g(r,s)$	$f(r)$	$1 \equiv x_N g(r,s) + k\tilde{m} f(r)$	$\alpha \equiv y_N^{g(r,s)} \cdot r^{mf(r)}$
MH IV.4	$g(r,s)$	1	$f(r)$	$g(r,s) \equiv x_N + k\tilde{m} f(r)$	$\alpha^{g(r,s)} \equiv y_N \cdot r^{mf(r)}$
MH V.2	1	$g(r,s)$	$f(s)$	$1 \equiv x_N g(r,s) + k\tilde{m} f(s)$	$\alpha \equiv y_N^{g(r,s)} \cdot r^{mf(s)}$
MH V.4	$g(r,s)$	1	$f(s)$	$g(r,s) \equiv x_N + k\tilde{m} f(s)$	$\alpha^{g(r,s)} \equiv y_N \cdot r^{mf(s)}$

Table 1 Variants of the Meta-Message hidden signature scheme

Security considerations

Total break of the scheme

To avoid a total break of the scheme, which means that an attacker can compute the secret key x_N of the notary Nancy, Nancy should be aware that she doesn't sign a hidden message $\tilde{m}$ where m is equal to zero or $(p-1)/2$ in *Mode* XL.

As already described by Horster, Michels and Petersen (1994), the variants of the El-Gamal signature scheme can be totally broken, if m is chosen equal to $(p-1)/2 \pmod p$ in *Mode* XL (or equal to $0 \pmod q$ in modes L, M and S). In these cases every verifier can compute the secret key x_N. The corresponding uncovered signature equation is $A \equiv x_N \cdot B + k \cdot m \cdot C \pmod{p-1}$. If $m = (p-1)/2$ then this equation simplifies to $A \equiv x_N \cdot B \pmod{p-1}$ if $k \cdot C$ is even and to $A \equiv x_N \cdot B + (p-1)/2 \pmod{p-1}$ if $k \cdot C$ is odd. In both cases x_N can be computed if $\gcd(B, p-1) = 1$. In the case of $m \equiv 0 \pmod q$ the equation simplifies to $A \equiv x_N \cdot B \pmod q$, which can always be solved for x_N if $B \neq 0$.

To avoid this kind of attack, it is necessary, that m can't be chosen equal to $(p-1)/2$ or 0 without knowledge of the notary. We see in equation (6) that if $m \equiv 0$ then the hidden message $\tilde{m} := m \cdot h \equiv 0 \pmod q$, such that Nancy won't sign it. If $m \equiv (p-1)/2$, then $\tilde{m}$ is either equal to zero or $(p-1)/2$, depending on the parity of h. So the notary shouldn't sign any hidden message $\tilde{m} := (p-1)/2$.

Universal and existential forgery

There are three different persons with different views who are able to cheat:

1. The notary Nancy knows the hidden message and perhaps later the uncovered message. She wants to find out some relationship between the hidden and uncovered parameters at the time of signature generation and needs not to follow the protocol.

Owner *Alice*	Channel	Notary *Nancy*
		$k \in_R \mathbf{Z}_p^*$
r'	$\longleftarrow$	$r' := \alpha^k \pmod p$
$r := d(r', m)$	$\longrightarrow$	r
s	$\longleftarrow$	$s := \rho(x_N, k, r)$

Table 2 An alternative hidden signature scheme

If $m \in \mathbf{Z}_q$, to reconstruct m by the notary is equivalent to compute the discrete logarithm $\log_\alpha(\alpha^h) = h$ since $h \equiv m^{-1} \cdot \tilde{m} \pmod q$.

2. The verifier Bob knows the message. He tries to forge a signature without influencing the protocol. This case can be reduced to the Meta signature scheme. The possibility of cheating has already been analyzed by Horster, Michels and Petersen (1994) with the result, that there are no security flaws known today.
3. The owner Alice knows the related hidden and uncovered message. Her aim is to get valid signatures on arbitrarily chosen messages. She needn't follow the protocol. She obtains the signature (r, s) for which she knows the equations $\beta^A \equiv y_N^{hB} \cdot r^{\tilde{m}C} \pmod p$ and the verification equation $\alpha^A \equiv y_N^B \cdot r^{mC} \pmod p$. As these equations are simply related by exponentiation with h, she has no more information and abilities to cheat than the verifier Bob.

An alternative approach

An alternative approach for hidden signature schemes can be obtained using the ideas of Schnorr's signature scheme (Schnorr, 1989) and has already been described in the literature (e.g. see Horster and Petersen (1994)). We choose the function d in the Meta signature scheme as an one way hash function h with arguments r' and m. The parameter $r := d(r', m)$ is computed by the owner of the message, such that the notary Nancy doesn't learn anything about the contents of the message. Nancy signs the message by choosing the general signature equation

$$A \equiv x_N \cdot B + k \cdot C \pmod q \tag{8}$$

where A, B and C are chosen as suitable general functions with arguments r and s. Suppose, that the parameter s can be extracted from equation (8) using the function $s := \rho(x_N, k, r)$. Then we get the hidden signature scheme given in table 2. The signature on the message m is given by (r, s). Its verification is done by checking the equation (9).

$$r = d\left(\alpha^{AC^{-1}} \cdot y_N^{-BC^{-1}} \pmod p, m\right). \tag{9}$$

As the parameters r and s are not covered to the notary in this scheme, we obtain only a hidden signature from this protocol. The approach is not suitable for signature schemes giving message recovery, as the function d has to be invertible in that case (Horster, Michels and Petersen, 1994).

4.2 Parameter hidden signatures

To get a parameter hidden signature from Nancy on the message m, Alice and Nancy carry out the following steps:

1. Alice chooses a random number $h \in \mathbf{Z}_q^*$ and computes $\beta := \alpha^h \pmod p$.
2. She transmits m and β to Nancy.
3. Nancy chooses a random $k \in \mathbf{Z}_q^*$, computes $r' := \beta^k \pmod p$, $r := d(r', m)$ and signs the message m by solving the congruence (10) for the parameter $\tilde{s}$.

$$A \equiv x_N \cdot B + k \cdot \tilde{s} \cdot C \pmod q \tag{10}$$

   The coefficients A, B, C are chosen as suitable general functions e, f, g with arguments m and r.
4. Nancy transmits the signature $(r, \tilde{s})$ to Alice.
5. Alice computes $s := \tilde{s} \cdot h^{-1} \pmod q$.

The tuple (r, s) is a parameter hidden signature on the message m. It can be verified by the congruence (11) if $d(r', m) = r'$ and by the congruence (12) in all other cases.

$$\alpha^A \equiv y_N^B \cdot r^{s \cdot C} \pmod p \tag{11}$$

$$r = d\left(\alpha^{A \cdot (s \cdot C)^{-1}} \cdot y_N^{-B \cdot (s \cdot C)^{-1}} \pmod p, m\right) \tag{12}$$

The congruence (11) is true as the following equation holds:

$$y_N^B \cdot r^{sC} \equiv \alpha^{x_N B} \cdot \beta^{ksC} \equiv \alpha^{x_N B} \cdot \alpha^{hk\tilde{C}h^{-1}s} \equiv \alpha^{x_N B + ks\tilde{C}} \equiv \alpha^A \pmod p.$$

Congruence (12) can be verified in a similar manner.

The Meta-parameter hidden signature scheme can be written as

$$MPH = (Mode.Type.No, d, e, f, g)$$

with $Type \in \{$ PH I, PH II, PH III, PH IV $\}$ and the related choices for the parameters $Mode$ and No (see section 3).

Theorem 2 *The signature scheme described in steps 1.-5. above is a parameter hidden signature scheme.*

Proof. As the message m and the signature parameter r are not blinded, this is only a hidden signature. The notary doesn't know the signature parameter s after signature generation,as it is blinded by the parameter h. Reconstructing s from $\tilde{s}$ is equivalent to solve the discrete logarithm problem $\log_\alpha(\beta) := h \pmod p$ since $h \equiv \tilde{s} \cdot s^{-1} \pmod q$.

Table 3 gives an overview about the most efficient variants of the Meta-parameter hidden signature scheme which are obtained from the Meta signature scheme in section 3.

No.	A	B	C	signature generation	verification
PH I.1	m	r	1	$m \equiv x_N r + k\tilde{s}$	$\alpha^m \equiv y_N^r \cdot r^s$
PH I.6	r	m	1	$r \equiv x_N m + k\tilde{s}$	$\alpha^r \equiv y_N^m \cdot r^s$
PH II.1	1	$f(m,r)$	1	$1 \equiv x_N f(m,r) + k\tilde{s}$	$\alpha \equiv y_N^{f(r,m)} \cdot r^s$
PH II.6	$f(m,r)$	1	1	$f(m,r) \equiv x_N + k\tilde{s}$	$\alpha^{f(m,r)} \equiv y_N \cdot r^s$
PH III.1	1	$f(m,r)$	$g(m)$	$1 \equiv x_N f(m,r) + k\tilde{s}g(m)$	$\alpha \equiv y_N^{f(m,r)} \cdot r^{g(m)s}$
PH III.6	$f(m,r)$	1	$g(m)$	$f(m,r) \equiv x_N + k\tilde{s}g(m)$	$\alpha^{f(m,r)} \equiv y_N \cdot r^{g(m)s}$
PH IV.1	1	$f(m,r)$	$g(r)$	$1 \equiv x_N f(m,r) + k\tilde{s}g(r)$	$\alpha \equiv y_N^{f(m,r)} \cdot r^{sg(r)}$
PH IV.6	$f(m,r)$	1	$g(r)$	$f(m,r) \equiv x_N + k\tilde{s}g(r)$	$\alpha^{f(m,r)} \equiv y_N \cdot r^{sg(r)}$

Table 3 Efficient variants of the Meta-parameter hidden signature scheme

Security considerations

The security of the parameter hidden signature scheme is comparable to the security of the message hidden signature scheme, except that the owner can't choose any hidden parameter in advance and thus the attack for total break described in the last subsection can't be applied.

Efficiency considerations

The efficient variants in which one of the coefficients A, B, C is equal to one and thus the verification is possible with only two exponentiations is given in table 3. Because the parameter $\tilde{s}$ has to occur in the coefficient C, they all need one inversion for signature generation.

5 WEAK BLIND SIGNATURES WITH APPENDIX

The weak blind signatures with appendix can be obtained from the Meta blind signature schemes with appendix presented by Horster, Michels and Petersen (1994), by simply choosing the random number b equal to zero. For the sake of clarity the function d is chosen as $d(r,m) := r$ and we only focus on the *Mode* L. The adoption to the other modes and other choices of the function d is straightforward (see Horster, Michels and Petersen (1994) for details).

The idea is that notary Nancy chooses the blinded parameter $\tilde{r} := \alpha^{\tilde{k}} \pmod p$ herself (with a random $\tilde{k} \in \mathbf{Z}_q{}^*$) and the owner Alice chooses the unblinded $r := \tilde{r}^a \pmod p$ (with random $a \in \mathbf{Z}_q{}^*$). Nancy signs the blinded message $\tilde{m}$ using the equation

$$\tilde{A} \equiv x_N \cdot \tilde{B} + \tilde{k} \cdot \tilde{C} \pmod q. \tag{13}$$

The coefficients $\tilde{A}, \tilde{B}$ and $\tilde{C}$ are chosen as suitable general functions e, f, g with arguments

Owner *Alice*	Channel	Notary *Nancy*
$a \in_R \mathbf{Z}_q^*$		$\tilde{k} \in_R \mathbf{Z}_p{}^*$
$\tilde{r}$	$\longleftarrow$	$\tilde{r} := \alpha^{\tilde{k}} \pmod p$
$r := \tilde{r}^a \pmod p$		
$\tilde{m} := \psi(a, m, r, \tilde{r})$	$\longrightarrow$	$\tilde{m}$
$\tilde{s}$	$\longleftarrow$	$\tilde{s} := \rho(x_N, \tilde{k}, \tilde{m}, \tilde{r})$
$s := \theta(a, m, r, \tilde{m}, \tilde{r}, \tilde{s})$		

Table 4 Meta-weak blind signature scheme with appendix

$\tilde{m}, \tilde{r}$ and $\tilde{s}$. The unblinded signed message is given by (m, r, s). Its validity is checked by the congruence (14).

$$\alpha^A \equiv y_N^B \cdot r^C \pmod p. \tag{14}$$

To guarantee the correctness of the signature scheme, this equation must be satisfied. Hence we get

$$y_N^B \cdot r^C \equiv \alpha^{x_N B} \cdot \alpha^{a\tilde{k}C} \equiv \alpha^{x_N B + aC\tilde{C}^{-1}(\tilde{A} - x_N \tilde{B})} \equiv y_N^{(B - a\tilde{C}^{-1}\tilde{B}C)} \cdot \alpha^{a\tilde{C}^{-1}\tilde{A}C} \stackrel{!}{\equiv} \alpha^A \pmod p.$$

Therefore we get the equations

$$A = a\tilde{A}C\tilde{C}^{-1} \pmod q, \tag{15}$$

$$B = a\tilde{B}C\tilde{C}^{-1} \pmod q. \tag{16}$$

If the value s does not appear in C then it is possible to transform these two equations to get $\tilde{m} := \psi(a, m, r, \tilde{r})$ and $s := \theta(a, m, r, \tilde{m}, \tilde{r}, \tilde{s})$. Note that s and $\tilde{s}$ are not allowed in the equation for $\tilde{m}$. Furthermore we can transform the signature equation (13) to get $\tilde{s} := \rho(x_N, \tilde{k}, \tilde{m}, \tilde{r})$. This results in the Meta-weak blind signature scheme (MWB) given in table 4. The signature on the message m is given by (r, s). Its verification can be done by checking the equation (14).

The Meta-weak blind signature scheme with appendix can be written as

$$MWB = (Mode.Type.No, d, e, f, g)$$

with $Type \in$ {WB I, WB II, WB III, WB IV, WB V} and the related choices for the parameters $Mode$ and No (see section 3).

We illustrate the Meta signature scheme by giving equations for some efficient variants in table 5. Note that for those schemes in which the parameter s appears in C we can't get weak blind signature schemes for general functions f and g, because s and $\tilde{s}$ are not allowed as arguments in the function ψ. Thus we can't get a weak blind signature scheme using the basic ElGamal signature scheme.

Theorem 3 *The signature scheme given in table 4 is a weak blind signature scheme.*

Proof. On the one side, the notary can't find a relationship between the covered and

No.	$\psi(a, m, r, \tilde{r})$	signature generation	$\theta(a, m, r, \tilde{m}, \tilde{r}, \tilde{s})$
WB I.2	$\tilde{m} := (ar)^{-1} m \tilde{r}$	$\tilde{s} := x_N^{-1}(\tilde{m} - \tilde{k}\tilde{r})$	$s := a\tilde{s}r\tilde{r}^{-1}$
WB I.3	$\tilde{m} := am\tilde{r}r^{-1}$	$\tilde{s} := x_N\tilde{r} + \tilde{k}\tilde{m}$	$s := a\tilde{s}m\tilde{m}^{-1}$
WB I.4	$\tilde{m} := (ar)^{-1} m \tilde{r}$	$\tilde{s} := x_N\tilde{m} + \tilde{k}\tilde{r}$	$s := a\tilde{s}r\tilde{r}^{-1}$
WB I.5	$\tilde{m} := am\tilde{r}r^{-1}$	$\tilde{s} := x_N^{-1}(\tilde{r} - \tilde{k}\tilde{m})$	$s := a\tilde{s}m\tilde{m}^{-1}$
WB II.3	$\tilde{m} := (m + r)a^{-1} - \tilde{r}$	$\tilde{s} := x_N(\tilde{r} + \tilde{m}) + \tilde{k}$	$s := a\tilde{s}$

Table 5 Efficient variants of the Meta-weak blind signature schemes

uncovered values during signature generation as he doesn't know the random value a which is involved in the computation of $\tilde{m}$. On the other side, there exists a relationship between the blinded and unblinded signature parameters, which can be verified by the notary using the congruence (17).

$$A \cdot B^{-1} \equiv \tilde{A} \cdot \tilde{B}^{-1} \pmod{q} \Longleftrightarrow A \cdot \tilde{B} \equiv \tilde{A} \cdot B \pmod{q}. \tag{17}$$

This congruence is satisfied, as we can see using the equations (15) and (16):

$$A \cdot B^{-1} \equiv a\tilde{A}C\tilde{C}^{-1} \cdot (a\tilde{B}C\tilde{C}^{1})^{-1} \equiv a\tilde{A}C\tilde{C}^{-1} \cdot (a^{-1}\tilde{B}^{-1}C^{-1}\tilde{C}) \equiv \tilde{A}\tilde{B}^{-1} \pmod{q}.$$

Note, that those variants, in which A and B are a permutation of $e(s)$ and 1 and $C := f(m, r)$ don't result in weak blind signature schemes, as s is revealed by the notary in these cases. This can easily be seen by the invariant in equation (17). So the notary can rediscover the signature using the parameter s as an indicator.

Alternative designs

Instead of computing the parameter r by the equation (A1) $r := (\tilde{r})^a \equiv \alpha^{\tilde{k}a} \pmod{p}$ we can also use (A2) $r := \tilde{r}y_N^{-a} \equiv \alpha^{\tilde{k}-x_N a} \pmod{p}$ as suggested similar by Okamoto (1992), (A3) $r := (\tilde{r})\alpha^a \equiv \alpha^{\tilde{k}+a} \pmod{p}$ or (A4) $r := (\tilde{r})^a y_N \equiv \alpha^{\tilde{k}a+x_N} \pmod{p}$ which have been proposed by Horster, Michels and Petersen (1994) or (A5) $r := (\tilde{r} \cdot y_N)^a \equiv \alpha^{(\tilde{k}+x_N)a} \pmod{p}$. This leads to slightly modified general equations (15), (16) from which we obtain many additional efficient variants. We get the following equations:

$(A2)$ $A = \tilde{A}C\tilde{C}^{-1} \pmod{q}$ (18)

$B = \tilde{B}C\tilde{C}^{-1} + aC \pmod{q}$ (19)

$(A3)$ $A = \tilde{A}C\tilde{C}^{-1} + aC \pmod{q}$ (20)

$B = \tilde{B}C\tilde{C}^{-1} \pmod{q}$ (21)

$(A4)$ $A = a\tilde{A}C\tilde{C}^{-1} \pmod{q}$ (22)

$B = a\tilde{B}C\tilde{C}^{-1} - 1 \pmod{q}$ (23)

$$(A5) \quad A = a\tilde{A}C\tilde{C}^{-1} \pmod{q} \tag{24}$$

$$B = a\tilde{B}C\tilde{C}^{-1} - aC \pmod{q} \tag{25}$$

This gives four additional designs together with the multiple variants of the Meta-weak blind signature scheme with appendix. We have to choose the coefficients A, B and C carefully, to avoid variants, in which one of the parameters can be revealed by the notary.

6 SIGNATURE SCHEME GIVING MESSAGE RECOVERY

This scheme has been proposed by Nyberg and Rueppel (1993). Let p and q be primes and q is a large integer factor of $p-1$. Let $\alpha \in \mathbf{Z}_p^*$ be an element of order q.

The signer *Alice* chooses x_A and y_A as in the ElGamal scheme. To sign the message $m \in \mathbf{Z}_p$, she chooses a random $k \in \mathbf{Z}_q$, computes $r := \alpha^{-k} \cdot m \pmod{p}$ and solves the congruence $s \equiv k - x_A \cdot r \pmod{q}$ for the parameter s. The tuple (r, s) builds the signature. The message can be recovered by computing $m := \alpha^s \cdot y_A^r \cdot r \pmod{p}$. The correctness of the signature is verified implicitly by checking the redundancy scheme of the message. The security of this scheme seems to be similar to the ElGamal scheme, although an equivalence couldn't be proved.

6.1 Meta signature scheme giving message recovery

We can apply the Meta signature scheme with appendix to obtain the Meta-Message recovery scheme (Horster, Michels and Petersen, 1994). The general signature equation is of the form

$$A \equiv x_A \cdot B + k \cdot C \pmod{q} \tag{26}$$

where A, B, C are permutations of general functions $e, f, g : \mathbf{Z}_q{}^2 \rightarrow \mathbf{Z}_q$ with arguments r and s. For the sake of clarity we describe the case in which the parameter r is computed as $r := \alpha^{-k} \cdot m \pmod{p}$. This is the choice for the function d proposed by Nyberg and Rueppel (1993). From the signature (r, s) the message can be recovered by the equation

$$m := \alpha^{A \cdot C^{-1}} \cdot y_A^{-B \cdot C^{-1}} \cdot r \pmod{p}. \tag{27}$$

If we look carefully on the necessary conditions on the functions e, f, g described in section 3, we get ten types of permutations which are listed in table 6.

Among these only the types MR I, III and V are solvable for all possible choices of the functions e, f and g. The most efficient types are MR I – IV if we choose the parameter $C = 1$, because we need no inversion for message recovery. In *Type* MR II, IV, VI – X we have to choose suitable functions e, f, g to guarantee the solvability for the parameter s. In *Type* MR IV, we have to choose different functions f, g without homomorphic properties to guarantee the security of the signature scheme. Furthermore it is argued by Horster, Michels and Petersen (1994) that *Mode* L is best suited for message recovering schemes.

Type	(A, B, C) permutation of		
MR I	e(1)	$f(r)$	$g(s)$
MR II	e(1)	$f(s)$	$g(r, s)$
MR III	e(1)	$f(r)$	$g(r, s)$
MR IV	e(1)	$f(r, s)$	$g(r, s)$
MR V	$e(r)$	$f(r)$	$g(r, s)$
MR VI	$e(r)$	$f(s)$	$g(r, s)$
MR VII	$e(s)$	$f(s)$	$g(r, s)$
MR VIII	$e(r)$	$f(r, s)$	$g(r, s)$
MR IX	$e(s)$	$f(r, s)$	$g(r, s)$
MR X	$e(r, s)$	$f(r, s)$	$g(r, s)$

Table 6 Permutations of the Message recovery scheme

7 HIDDEN SIGNATURES GIVING MESSAGE RECOVERY

In the signature schemes giving message recovery it is only possible to hide the message m and at least one of the parameters simultaneously, because the message is embedded in the parameter r.

The method for blinding is developed from the existential forgery of some of the variants of the Meta-Message recovery scheme described by Horster, Michels and Petersen (1994). We will refer to it as *message hidden signature*, because the parameter r is unchanged.

Message hidden signature

As the parameter s can be embedded in the coefficient A or B we get two variants of this kind of signature. In the following we describe both variants.

First variant:
To get a message hidden signature from the notary *Nancy* on the message m, Alice and Nancy carry out the following steps:

1. Alice chooses a random number $h \in \mathbf{Z}_q^*$ and computes $\tilde{m} := m \cdot \alpha^{-h} \pmod{p} \Leftrightarrow m \equiv \tilde{m} \cdot \alpha^h \pmod{p}$.
2. She transmits $\tilde{m}$ to Nancy.
3. Nancy chooses a random $k \in \mathbf{Z}_q^*$, computes $r := \alpha^{-k} \cdot \tilde{m} \pmod{p}$ and signs the message $\tilde{m}$ with variant MR I.3 of the Meta-Message recovery scheme, that is she solves the congruence (28) for the parameter $\tilde{s}$.

$$\tilde{s} \equiv x_N \cdot f(r) + k \pmod{q}. \tag{28}$$

4. Nancy transmits the signature $(r, \tilde{s})$ to Alice.
5. Alice computes $s := \tilde{s} + h \pmod{q}$.

The tuple (r, s) is a message hidden signature on the message m. The message can be recovered by the following congruence:

$$m := \alpha^s \cdot y_N^{-f(r)} \cdot r \pmod{p} \tag{29}$$

It can be verified by checking the underlying redundancy scheme. This congruence is true because of the following equation:

$$\alpha^s \cdot y_N^{-f(r)} \cdot r \equiv \alpha^{(\tilde{s}+h)} \cdot y_N^{-f(r)} \cdot r \equiv \alpha^{\tilde{s}-x_N f(r)} \cdot \alpha^h \cdot r \equiv \alpha^k \cdot \alpha^h \cdot \alpha^{-k} \cdot \tilde{m} \equiv \alpha^h \cdot \tilde{m} \equiv m \pmod{p}.$$

Second variant:
The same type of signature is possible, if Alice computes $\tilde{m} := m \cdot y_N^{-h}$ and Nancy solves the equation $f(r) \equiv x_N \cdot \tilde{s} + k \pmod{q}$ for the parameter $\tilde{s}$, which is variant MR I.5 of the Meta-Message recovery scheme. The tuple (r, s) is a message hidden signature on the message m which can be recovered by the congruence $m \equiv \alpha^{f(r)} \cdot y_N^{-s} \cdot r \pmod{p}$. The correctness is also checked implicitly by the underlying redundancy scheme. This congruence is true because the following equation holds:

$$\alpha^{f(r)} \cdot y_N^{-s} \cdot r \equiv \alpha^{f(r)} \cdot y_N^{-(\tilde{s}+h)} \cdot r \equiv \alpha^{f(r)-x_N\tilde{s}} \cdot y_N^h \cdot r \equiv \alpha^k \cdot y_N^h \cdot \alpha^{-k} \cdot \tilde{m} \equiv y_N^h \cdot \tilde{m} \equiv m \pmod{p}.$$

Security considerations

The security of the hidden signature scheme with message recovery corresponds to the security of the variants of the Meta-Message recovery scheme (Nyberg and Rueppel, 1993, Horster, Michels and Petersen, 1994). It must only be shown, that the notary Nancy couldn't get any information about the message m at the time of signature generation. This is true because the message is blinded with a random parameter α^{-h} $[y_A^{-h}]$, which she can't compute without knowledge of the random number h.

Efficiency considerations

The first variant is more efficient because the signature equation can be solved without computing the inverse of x_A.

8 WEAK BLIND SIGNATURES GIVING MESSAGE RECOVERY

Following the design criterias of the weak blind signature scheme with appendix in section 5 and the Meta-Message recovery blind signature scheme proposed by Horster, Michels and Petersen (1994) we can obtain the weak blind signatures giving message recovery. For the sake of clarity, we focus on the *Mode* L, the adoption to the other modes is straightforward.

As in the weak blind signature scheme with appendix, Nancy chooses the blinded parameter $\tilde{r}' := \alpha^{\tilde{k}} \pmod{p}$ which is unblinded by Alice as $r' := (\tilde{r}')^a \pmod{p}$ $(a, k$ random in $\mathbf{Z}_q^*)$. Alice computes $r := d(r', m)$ and the blinded parameter $\tilde{r}$. Nancy signs $\tilde{r}$ using the equation

$$\tilde{A} \equiv x_N \cdot \tilde{B} + \tilde{k} \cdot \tilde{C} \pmod{q}. \tag{30}$$

The coefficients $\tilde{A}, \tilde{B}$ and $\tilde{C}$ are chosen as suitable general functions e, f, g with arguments

Owner *Alice*	Channel	Notary *Nancy*
$a \in_R \mathbf{Z}_q^*$		$\tilde{k} \in_R \mathbf{Z}_p{}^*$
$\tilde{r}'$	$\longleftarrow$	$\tilde{r}' := \alpha^{\tilde{k}} \pmod p$
$r' := \tilde{r}'^a \pmod p$		
$r := d(r', m)$		
$\tilde{r} := \psi(a, r)$	$\longrightarrow$	$\tilde{r}$
$\tilde{s}$	$\longleftarrow$	$\tilde{s} := \rho(x_N, \tilde{k}, \tilde{r})$
$s := \theta(a, r, \tilde{r}, \tilde{s})$		

Table 7 Meta-weak blind signature scheme giving message recovery

$\tilde{r}$ and $\tilde{s}$. The signature on the unblinded message m is given by (r, s). Its validity is checked by computing the message using equation (31) and checking if m satisfies the given redundancy scheme.

$$m := d^{-1}\left(r, \alpha^{AC^{-1}} \cdot y_N^{-BC^{-1}} \pmod p\right) \tag{31}$$

To guarantee the correctness of the signature scheme, this equation must be satisfied. As $m := d^{-1}(r, r')$ by definition, we get the equation

$$r' \equiv (\tilde{r}')^a \equiv \alpha^{\tilde{k}a} \equiv \alpha^{a\tilde{C}^{-1}(\tilde{A}-x-n\tilde{B})} \equiv \alpha^{a\tilde{A}\tilde{C}^{-1}} \cdot y_N^{-a\tilde{C}^{-1}\tilde{B}} \stackrel{!}{\equiv} \alpha^{AC^{-1}} \cdot y_N^{-BC^{-1}} \pmod p.$$

Therefore we get the equations

$$A = a\tilde{A}C\tilde{C}^{-1} \pmod q, \tag{32}$$
$$B = a\tilde{B}C\tilde{C}^{-1} \pmod q, \tag{33}$$

which are the same as in the case of weak blind signatures with appendix. If the value s does not appear in C then it is possible to transform these two equations to get $\tilde{r} := \psi(a, r)$ and $s := \theta(a, r, \tilde{r}, \tilde{s})$. Note that s and $\tilde{s}$ are not allowed in the equation for $\tilde{r}$. Furthermore we can transform the signature equation (30) to get $\tilde{s} := \rho(x_N, \tilde{k}, \tilde{r})$. This results in the Meta-weak blind signature scheme giving message recovery (MWM) given in table 7. The signature on the message m is given by (r, s). Message recovery is done by the equation (31).

Note, that those variants, in which A and B are a permutation of 1 and $e(s)$ and $C := f(r)$, don't result in weak blind signature schemes, as s can be revealed by the notary in these cases. This can easily be seen by the invariant in equation (17). So the notary can rediscover the signature using the parameter s as an indicator.

The Meta-weak blind signature scheme giving message recovery can be written as

$$MWM = (Mode.Type.No, d, e, f, g)$$

with $Type \in \{\text{WM I, WM II, WM III, WM IV, WM V}\}$ and the related choices for the parameters *Mode* and *No* (see section 3).

We illustrate the Meta signature scheme by giving equations for some efficient variants in table 8. The proof that the signature scheme given in table 7 is a weak blind signature scheme is analogue to theorem 3.

No.	$\psi(a,r)$	signature generation	$\theta(a,r,\tilde{r},\tilde{s})$
WM I.3	$\tilde{r} := a\tilde{r}$	$\tilde{s} := x_N\tilde{r} + \tilde{k}$	$s := a\tilde{s}$
WM I.5	$\tilde{r} := a\tilde{r}$	$\tilde{s} := x_N^{-1}(\tilde{r} - \tilde{k})$	$s := a\tilde{s}$
WM II.1	$\tilde{r} := a^{-1}r$	$\tilde{s} := x_N(\tilde{r} + \tilde{s}) + \tilde{k}$	$s := a\tilde{s}$

Table 8 Some efficient variants of the signature scheme in table 7

9 APPLICATIONS

In all applications for *message hidden* signatures the notary is not obliged to know the message but the owner has in spite of this a special interest in the anonymity of the message. This might be the case in a testament application in which the notary signs the last will of Alice without knowing the content of it during Alice's lifetime. Later the inheritance Bob can verify that the testament has been signed by the notary and it is possible to check that the notary has signed the testament even if the signature scheme is broken in meantime. The notary just looks in his list of signature parameters and compares the given signature parameters with his stored ones. This kind of application is also possible with weak blind signatures but *not* with strong signature schemes, because in the strong blind schemes the notary can't find any relationship between the given and the stored parameters. Other applications of the hidden and weak blind signature scheme are pseudonymous credentials or anonymous access control.

An important application for *parameter hidden* signature schemes are self-certified public keys. They can be obtained by the approach described by Horster, Michels and Petersen (1994) which can also be adopted to the Meta signature scheme with appendix. Using the hidden signature schemes, the problem that the notary Nancy always knows the secret key of the user Alice can be solved as pointed out in (Horster and Knobloch 1991, Horster, Michels and Petersen, 1994, Horster and Petersen, 1994). The self-certified public keys are mainly used for authentication and authentic key-exchange protocols (Horster, Michels and Petersen, 1994, Horster and Petersen, 1994). Weak blind signatures can be used for similar applications.

10 CONCLUSION

We have presented several hidden and weak blind signature schemes. They have the property that the blinding of the parameters is only weak, such that the blinded message (or parameters) and the unblinded message can be related by the notary later. This property is useful in many applications and so this kind of signatures have their use and importance among the strong blind, untraceable signatures and the conventional uncovered signatures.

REFERENCES

S.Brands, (1993), Untraceable off-line cash in wallets with observers, Lecture Notes in Computer Science 773, *Advances in Cryptology: Proc. Crypto '93*, Berlin: Springer Verlag, 1994, pp. 302 – 18.

J.L.Camenisch, J.-M.Piveteau, M.A.Stadler, (1994), Blind signature schemes based on the discrete logarithm problem, Preprint, Presented at *Rump session of Eurocrypt '94*, Perugia, Italy, 5 pages.

D.Chaum, (1982), Blind signatures for untraceable payments, *Advances in Cryptology: Proc. Crypto '82*, New York: Plenum, 1983, pp. 199 – 203.

T.ElGamal, (1984), Cryptography and logarithms over finite fields, Stanford University, CA., UMI Order No. DA 8420519, 119 pages.

L.Harn, (1994), New digital signature scheme based on discrete logarithm, *Electronics Letters*, Vol. 30, No. 5, pp. 396 – 8.

P.Horster, H.-J.Knobloch, (1991), Discrete Logarithm based protocols, Lecture Notes in Computer Science 547, *Advances in Cryptology: Proc. Eurocrypt '91,* Berlin: Springer Verlag, 1992, pp. 399–408.

P.Horster, M.Michels, H.Petersen, (1994), Meta-ElGamal signature schemes, *Proc. 2nd ACM Conference on Computer and Communications Security*, Fairfax, Virginia, Nov. 2–4, 1994, pp. 96 – 107.

P.Horster, M.Michels, H.Petersen, (1994), Meta-Message recovery and Meta-blind signature schemes based on the discrete logarithm problem and their applications, Lecture Notes in Computer Science 917, *Advances in Cryptology: Proc. Asiacrypt '94*, Berlin: Springer Verlag, 1995, pp. 224 – 37.

P.Horster, H.Petersen, (1994), Verallgemeinerte ElGamal Signaturen, Sicherheit in Informationssystemen, *Proceedings of SIS '94*, Verlag der Fachvereine Zürich, pp. 89 – 106.

P.Horster, H.Petersen, (1994), Signature and authentication schemes based on the discrete logarithm (in German), *Internal Report 94 – 9*, RWTH Aachen, ISSN 0935-3232, March, 1994, 96 pages.

P.Horster, H.Petersen, (1994), Classification of blind signature schemes and examples of hidden and weak blind signatures, *Rump Session of Eurocrypt '94*, Perugia, Italy, Technical Report TR-94-1, University of Technology Chemnitz-Zwickau, 6 pages.

H.-J.Knobloch, (1994), A remark on the size of ElGamal-type digital signatures, *E.I.S.S.-Report 94-1*, University of Karlsruhe, Germany, 5 pages.

National Institute of Standards and Technology, (1991), *Federal Information Processing Standard*, FIPS Pub XX: Digital Signature Standard (DSS).

K.Nyberg, R.Rueppel, (1993), A new signature scheme based on the DSA giving message recovery, *1st ACM Conference on Computer and Communications Security*, Fairfax, Virginia, Nov. 3–5., 1993, pp. 58 – 61.

K.Nyberg, R.Rueppel, (1994), Message recovery for signature schemes based on the discrete logarithm problem, *Pre-proceedings of Eurocrypt '94*, pp. 175 – 90.

T.Okamoto, (1992), Provable secure and practical identification schemes and corresponding signature schemes, Lecture Notes in Computer Science 740, *Advances in Cryptology: Proc. Crypto '92*, Berlin: Springer Verlag, 1993, pp. 31 – 53.

C.P.Schnorr, (1989) Efficient identification and signatures for smart cards, Lecture Notes in Computer Science 435, *Advances in Cryptology: Proc. Crypto '89,* Berlin: Springer Verlag, 1990, pp. 239 – 51.