Group Signatures

David Chaum
Eugène van Heyst

CWI Centre for Mathematics and Computer Science,
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands.

Abstract. In this paper we present a new type of signature for a group of persons, called a group signature, which has the following properties:

(i) only members of the group can sign messages;
(ii) the receiver can verify that it is a valid group signature, but cannot discover which group member made it;
(iii) if necessary, the signature can be "opened", so that the person who signed the message is revealed.

These group signatures are a "generalization" of the credential/ membership authentication schemes, in which one person proves that he belongs to a certain group.
We present four schemes that satisfy the properties above. Not all these schemes are based on the same cryptographic assumption. In some of the schemes a trusted centre is only needed during the setup; and in other schemes, each person can create the group he belongs to.

1. Introduction

In this paper we present a new type of signature, which will be illustrated with the following example:

A company has several computers, each connected to the local network. Each department of that company has its own printer (also connected to the network) and only persons of that department are allowed to use their department's printer. Before printing, therefore, the printer must be convinced that the user is working in that department. At the same time, the company wants privacy: the user's name may not be revealed. If, however, someone discovers at the end of the day that a printer has been used too often, the director must be able to discover who misused that printer, to send him a bill.

More formally: a group of persons wants to create a signature scheme, which we will call a *group signature scheme*, that has the following three properties:

(i) only members of the group can sign messages;
(ii) the receiver of the signature can verify that it is a valid signature of that group, but cannot discover which member of the group made it;
(iii) in case of dispute later on, the signature can be "opened" (with or without the help of the group members) to reveal the identity of the signer.

Group signatures are a "generalization" of *credential mechanisms* ([Ch85]) and of *membership (authentication) schemes* (cf. [OOK90], [SKI90]), in which a group member can convince a verifier that

he belongs to a certain group, without revealing his identity. In [OOK90] and [SKI90], several of these schemes are proposed in which the same secret key is given to each group member. We define the following assumptions.

Assumption 1. *For each person it is unfeasible to compute RSA roots (hence it is unfeasible to split numbers that are the product of some large primes; and it is unfeasible to compute discrete logarithms modulo a large composite number).*

Assumption 2. *For each person it is unfeasible to compute the discrete logarithm modulo a large prime number.*

In this paper, only one group of persons will be considered (the hierarchical situation will not be treated here); and four different group signature schemes are presented. These schemes are compared.

Cryptographic assumption. In the first scheme every public key system can be used; the other schemes are based on Assumption 1 or 2. In all schemes (except in some modifications of the first scheme), the privacy of the signer is protected computationally. Not even a person from the group can determine who made a certain signature (except of course for the person who made that signature). Care must of course be taken in the selection of the exponents used in order to protect the anonymity of the signer. See Section 6.

Trusted authority. Let Z be a trusted authority, which sets the group signature scheme (it may be possible to distribute the power of Z). Except for the first scheme, Z is no longer needed after the setup. In the last scheme, a group signature scheme can be created from a "normal" setup, without a trusted authority.

Creation of the group. In the first two schemes the group of persons is fixed in advance. In the last two schemes, it is assumed that there is already a setup, based on RSA or discrete logarithm. If in these schemes someone wants to sign a message without revealing his name, then *at that moment* he creates some group of persons (for instance by picking them from a Trusted Public Directory of public keys) and proves that he belongs to that group. In case of dispute later on, the other "group members" are able to deny that signature.

Type of signature. In the last three schemes, the signatures made by the group members are undeniable signatures, but it is possible to make digital signatures. This can be realized as in [FS86], by doing the k iterations of the confirmation protocol in parallel and let the recipient choose the challenge vector not randomly, but as the outcome of a one-way-function on the received numbers. Because this protocol is no longer zero-knowledge, the signature and the confirmation protocol together will be a digital signature. Still to be proven is that this parallel protocol gives "no useful knowledge" to the recipient.

Costs. In all schemes the length of the public key (i.e., the number of bits of the group's public key) is linear in the number of group members. The numbers of bits and of computations are only compared in the case of the confirmation protocol, because in one disavowal protocol, these numbers are independent of the number of group members. We have not taken into account the looking-up of some public keys in a Trusted Public Directory.

Scheme number	Based on assumption	$\mathcal{Z}$ needed to open a signature	Group fixed in advance	Type of signature	Length of the group's public key	Number of computations during conf. pr.	Number of bits transmitted during conf. pr.
1	Any	Yes	Yes	Any type	Linear	Independent	Independent
2	1	No	Yes	Undeniable	Linear	Linear	Independent
3	1	No	No	Undeniable	Linear	Linear	Independent
4	2	No	No	Undeniable	Linear	Linear	Linear

Fig. 1. Comparison of the four group signature schemes presented in this paper. "Independent, linear" means that the number is independent respectively linear in the number of group members.

2. First group signature scheme

$\mathcal{Z}$ chooses a public key system, gives each person a list of secret keys (these lists are all disjunct) and publishes the complete list of corresponding public keys (in random order) in a Trusted Public Directory.

Each person can sign a message with a secret key from his list, and the recipient can verify this signature with the corresponding public key from the public list. Each key will be used only once, otherwise signatures created with that key are linked. $\mathcal{Z}$ knows all the lists of secret keys, so that in case of dispute, he knows who made the disputed signature. Hence $\mathcal{Z}$ is needed for the setup and for "opening" a signature.

If each person gets the same number of secret keys, then the length of the public key of this group signature scheme (i.e. the length of the Trusted Public Directory) is linear in the number of persons; but the number of messages a person can sign is fixed.

A problem with this scheme is that $\mathcal{Z}$ knows all the secret keys of the group members and can therefore also create signatures. This can be prevented by using *blinded public keys*. Let the public key system used be based on Assumption 2: for instance the ElGamal scheme [ElG85] or the undeniable signature scheme [CvA89]. Let g be a generator of the multiplicative group $\mathbb{Z}_p^*$, where p is a prime. Group member i creates his own secret key s_i and gives $g^{s_i} \pmod p$ to $\mathcal{Z}$. Thus $\mathcal{Z}$ has a list of all these public keys together with the group member's name. Each week $\mathcal{Z}$ gives each group member i a randomly chosen number $r_i \in \{1,\ldots,p-1\}$ and publishes the list of all the blinded public keys $(g^{s_i})^{r_i}$. During this week group member i will use $s_i r_i \pmod{p-1}$ as secret key.

The advantages of this modification are that $\mathcal{Z}$ cannot fake signatures, and that each group member only has to have one "really secret key" (for instance in a smart card), which can be blinded in order to make other secret keys. Only the one week's signatures can be linked, so that each group member can have only a few secret keys in his smart card to prevent this linking. If an r_i is accidentally revealed, still no more information about the secret key s_i is revealed.

In another modification, no trusted authority is needed: each user untraceably sends one (or more) public keys to a public list, which will be the public key of the group. But only group members must be able to send public keys to that list.

3. Second group signature scheme

$\mathcal{Z}$ chooses two different large primes p,q together with a one-way-function f of which the outcome may be assumed to be coprime with $N=pq$. $\mathcal{Z}$ gives person i of the group a secret key s_i, which is a large

prime randomly chosen from the set $\Phi=\{\lceil\sqrt{N}\rceil,\ldots,\lfloor 2\sqrt{N}\rfloor-1\}$, computes $v=\prod s_i$, and publishes N,v and f. If group member i wants to sign message n, his signature will be

$$(f(n))^{s_i} \bmod N,$$

and he has to prove to the recipient that $s_i|v$ and that $s_i\in\Phi$, without revealing anything more about s_i (see Section 3.1). In case of dispute later on, the recipient can perform a confirmation/disavowal protocol with each group member, without the help of $\mathcal{Z}$ (see Section 3.2). To prove the security of these schemes we need Assumption 1.

3.1. Confirmation protocol

We first consider the following instance, which is solved by [BCDvdG87] by using Protocol 1, which uses computationally secure blobs $\mathcal{B}$.

$\mathcal{P}$'s secret	:	c.
public	:	N,x,y,Ω: $x,y\in\mathbb{Z}_N^*,\Omega=\{\alpha,\ldots,\alpha+\beta\}\subset\mathbb{N}$.
prove to $\mathcal{V}$	:	$x^c\equiv y\pmod N\ \wedge\ c\in\Omega$.

Instance 1.

If this protocol is iterated k times, $\mathcal{V}$ will be convinced (with probability $1-2^{-k}$) that $c\in\tilde{\Omega}=\{\alpha-\beta,\ldots,\alpha+2\beta\}$, but $\mathcal{V}$ will receive no knowledge other than the fact that $c\in\Omega=\{\alpha,\ldots,\alpha+\beta\}$[†].

Protocol 1. (for Instance 1)

(1) *$\mathcal{P}$ chooses $r\in\{0,\ldots,\beta\}$. He computes blobs on $z_1\equiv x^r\pmod N$ and $z_2\equiv x^{r-\beta}\pmod N$, and sends the unordered pair $\{\mathcal{B}(z_1),\mathcal{B}(z_2)\}$ to $\mathcal{V}$.*

(2) *$\mathcal{V}$ chooses randomly $b\in\{0,1\}$ and sends it to $\mathcal{P}$.*

(3) *$\mathcal{P}$ sends $\mathcal{V}$ in case*

- *b=0: r and opens both blobs.*
- *b=1: $\tilde{r}$ which is $(c+r)$ or $(c+r-\beta)$, whichever is in the set Ω, and opens respectively the blob on z_1 or z_2 (which is called $\tilde{z}$).*

(4) *$\mathcal{V}$ verifies in case*

- *b=0: that $r\in\{0,\ldots,\beta\}$ and that the blobs contain x^r and $x^{r-\beta}$ in some order.*
- *b=1: that $\tilde{r}\in\Omega$, that one of the blobs contains $\tilde{z}$ and that $\tilde{z}$ satisfies $x^{\tilde{r}}\equiv\tilde{z}y$.*

If $c\in\Omega$, then the distribution of $\tilde{r}$ is uniform over Ω and is thus independent of c. With this protocol we will create a confirmation protocol, so let $\mathcal{P}$ be a fixed group member who wants to convince the recipient (verifier $\mathcal{V}$) that he gave him a valid signature S. So the following instance (in which we write m in stead of $f(n)$) has to be solved:

[†] Hence, by using $\Omega=\{1,\ldots,N\}$, one can prove that he knows a discrete logarithm modulo N, without knowing $\varphi(N)$.

$\mathcal{P}$'s secret	:	s.
public	:	N, v, m, S, Φ; $m, S \in \mathbb{Z}_N^*$.
prove to $\mathcal{V}$	:	$S \equiv m^s \pmod N \wedge s \in \Phi \wedge s \mid v$.

Instance 2.

Protocol 2. (for Instance 2)

Step 1. *Prove the knowledge of s such that $S \equiv m^s \pmod N$ and that $s \in \Phi$ with Protocol 1, iterated k times (take $\Omega=\Phi$, $x=m$, $y=S$ and $c=s$).*

Step 2. *Prove that $s \mid v$ as follows*

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
	$r \equiv S^r$ →	chooses $r \in \{1,\ldots,N\}$
$b \equiv a^{v/s}$	$\mathcal{B}(b)$ →	
verifies a	r ←	
	open blob →	verifies opening and that $b \equiv m^{vr}$

Note that for all x the probability distributions of $x^r \pmod N$ where $r \in \{1,\ldots,\varphi(N)\}$ or $r \in \{1,\ldots,N\}$ are polynomially indistinguishable [CEvdG87]. Step 1 of this protocol was already proven to be sound, complete, and zero-knowledge. Step 2 is trivially complete and zero-knowledge (the blobs $\mathcal{B}$ are computationally secure zero-knowledge). Because in Step 1 it is proved that $S \equiv m^s$, one can easily see that it is feasible to compute $b \equiv (m^r)^v$ from $\{s, v, a \equiv (m^r)^s\}$ if and only if $s \mid v$ (under the assumption that it is unfeasible to compute RSA roots, so here we use that N is not a prime). Hence Step 2 is also sound. ❑

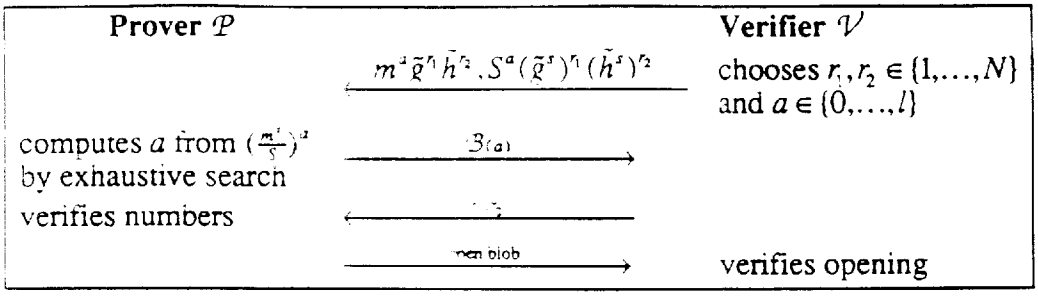
3.2. Disavowal protocol

If $\mathcal{P}$ wants to prove to $\mathcal{V}$ that S is not his signature on m, the following instance has to be solved:

$\mathcal{P}$'s secret	:	s.
public	:	N, v, m, S, Φ; $m, S \in \mathbb{Z}_N^*$.
prove to $\mathcal{V}$	:	$S \not\equiv m^s \pmod N \wedge s \in \Phi \wedge s \mid v$.

Instance 3.

There are no zero-knowledge disavowal protocols to prove that $\alpha^x \not\equiv \beta^x \pmod N$, for given $\{N, \alpha, \beta, \alpha^x\}$, where $\varphi(N)$ is unknown. Therefore we use the following modification of [Ch90] to solve Instance 3. $\mathcal{Z}$ publishes $< \tilde{g}, \tilde{h} >$, which generates the whole group $\mathbb{Z}_N^*$ (see the next section how to construct $\tilde{g}$ and $\tilde{h}$), together with a Trusted Public Directory containing {name group member, $\tilde{g}^s, \tilde{h}^s$}. Let l be a very small constant such that exhaustive search over $\{0,\ldots,l\}$ is feasible. Note that if $S \equiv m^s$, then $\mathcal{P}$ can not compute a from $(\frac{m^i}{S})^a$, because $(\frac{m^i}{S})^a \equiv 1$. So he has to guess a.



Protocol 3. (for Instance 3)

3.3. Some remarks on this group signature scheme

If all group members except one conspire, the secret key of that one person is revealed. This threat can be easily eliminated if the authority $\mathcal{Z}$ makes himself a member of the group, i.e., if $\mathcal{Z}$ computes v as $v = s_{\mathcal{Z}} \cdot \prod s_i$, where $s_{\mathcal{Z}}$ is a secret key only known to $\mathcal{Z}$. With this trick, the group can also consist of two members.

The number of bits of the public key v is linear in the number of persons, so raising a number to the power v will take a time linear in the number of group members.

The set Φ can also be chosen in other ways, but it must satisfy the following conditions. If $\Phi = \{\varphi_1,\ldots,\varphi_1+\varphi_2\} \subset \mathbb{N}$, then $1, N, \varphi_1 \notin \tilde{\Phi} = \{\varphi_1-\varphi_2,\ldots,\varphi_1+2\varphi_2\}$. The first condition is necessary to avoid the use of $s=1$. According to the last condition the following conspiracy attack is avoided: if two group members, say i and j, conspire, they can create signatures $S \equiv m^{s_i s_j}$, which they can both disavow later. But $s_i s_j \notin \tilde{\Phi}$, so this signature will not be accepted in Step 1 of Protocol 2. Hence also the choice of v/s or v as exponent in the signature is avoided.

The blob $\mathcal{B}$ can be implemented in the following way: $\mathcal{Z}$ chooses generators g_p and h_q of $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$ respectively, and constructs with the use of the Chinese Remainder Theorem $g \equiv \begin{cases} g_p \bmod p \\ 1 \bmod q \end{cases}$ and $h \equiv \begin{cases} 1 \bmod p \\ h_q \bmod q \end{cases}$. So $<g,h>$ generates $\mathbb{Z}_N^*$ uniformly, but it reveals the factorization of N. Therefore he chooses integers a_1, a_2, b_1, b_2 satisfying $\gcd(a_1, b_1, p-1) = \gcd(a_2, b_2, q-1) = \gcd(a_1 b_2 - a_2 b_1, \frac{\varphi(N)}{\lambda(N)}) = 1$ and publishes $\tilde{g} \equiv g^{a_1} h^{a_2}$ and $\tilde{h} \equiv g^{b_1} h^{b_2}$. It is not difficult to see also that $<\tilde{g}, \tilde{h}>$ generates the whole group $\mathbb{Z}_N^*$ uniformly, if the exponents are chosen from $\{1,\ldots,\varphi(N)\}$. Hence, in order for $\mathcal{P}$ to make $\mathcal{B}(y)$, he chooses $r_1, r_2 \in \{1,\ldots,N\}$ and creates $\mathcal{B}(y)$ as $y g^{r_1} h^{r_2} \pmod N$.

Another method of implementation is the following: $\mathcal{P}$ chooses randomly k numbers $g_1,\ldots,g_k$ from $\{1,\ldots,N\}$. Then with high probability $<g_1,\ldots,g_k>$ generates $\mathbb{Z}_N^*$ nearly uniformly, for k sufficiently large. In this case no trusted centre is needed [Ch87].

4. Third group signature scheme

For the security of this scheme we need Assumption 1, and we assume that there is a Trusted Public Directory in which each person's RSA modulus is listed (the public RSA exponent is not needed in this group signature scheme).

The secret key of group member i will be the factorization of his own RSA modulus $N_i = p_i q_i$.

During the setup, $\mathcal{Z}$ chooses an RSA-modulus N, which is independent of all the N_i's. Let M be a public integer such that $p_i \in \Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 2\sqrt{M}\rfloor - 1\}$ and $q_i > 4\sqrt{M}$ (for all i). If person i wants to sign message n, he chooses randomly some set Γ of persons (including himself); his signature will be

$$\Gamma, \quad f(n))^{p_i} \bmod N,$$

and he has to give a zero knowledge proof that the used exponent $p_i \in \Phi$ and that p_i is a divisor of the product of the RSA moduli of the persons of Γ. This can be done with Protocol 2 (with $\Omega=\Phi$), because $N_i > q_i > 4\sqrt{M}$ and thus all moduli, every product of two prime divisors of different moduli and each q_i are no elements of $\tilde{\Phi} = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 3\sqrt{M}\rfloor\}$. Hence the exponent used in the signature must be p_i. If a group member wants to deny a signature, he can use Protocol 3.

5. Fourth group signature scheme

The fourth group signature scheme is based on Assumption 2. Let p be a large public prime and let g,h be public generators of $\mathbb{Z}_p^*$. Person i has a secret key s_i and a public key $k_i \equiv g^{s_i} \pmod p$. If person i wants to sign message $m=f(n)$, he randomly chooses some set Γ of persons (including himself); and his signature will be

$$\Gamma, m^{s_i} \pmod p,$$

and he has to give a zero-knowledge proof that the secret exponent used in that signature is also used in the public key of somebody of the group Γ, i.e. the protocol has to solve the following instance:

$\mathcal{P}$'s secret	:	s_i.
public	:	p, g, h, S, Γ.
to prove to $\mathcal{V}$	:	$S \equiv m^{s_i} \pmod p \wedge g^{s_i} \in \{k_j \mid j \in \Gamma\}$.

Instance 4.

To prove this, $\mathcal{P}$ uses the following protocol, which gives no additional information about i and s_i. We have compressed the proofs that S is of the correct form, that the exponents used in S and in some public key are the same, and that the public key is used by somebody in Γ into one protocol.

Protocol 4. (for Instance 4)

(1) *$\mathcal{P}$ chooses numbers $r_1, \ldots, r_{|\Gamma|}, t_1, t_2, t_3 \in \{1, \ldots, p-1\}$ and a permutation τ of Γ. He sends $\mathcal{V}$ the numbers:* $x \equiv \left(\frac{g}{m}\right)^{t_1} h^{t_2} \pmod p$, $y \equiv m^{t_3} \pmod p$ *and* $z_{\tau(j)} \equiv k_j h^{r_j} \pmod p$ *(for all $j \in \Gamma$).*

(2) *$\mathcal{V}$ chooses $b \in \{0,1\}$ and sends b to $\mathcal{P}$.*

(3) *$\mathcal{P}$ sends $\mathcal{V}$ in case*

$b=0$: $r_1, \ldots, r_{|\Gamma|}, t_1, t_2, t_3$ *and* τ.

$b=1$: $t_1+s_i, t_2+r_i, t_3+s_i$ *and index* $\tau(i)$.

(4) *$\mathcal{V}$ verifies in case*

$b=0$: *that the numbers $x, y, z_1, \ldots, z_{|\Gamma|}$ are formed correctly.*

$b=1$: *that* $yS \equiv m^{t_3+s_i}$ $z_{\tau(i)} \equiv (g/m)^{t_1+s_i} \pmod p$.

If $\mathcal{P}$ can answer both questions, then he knows s_i; it is easy to see that this s_i satisfies $S \equiv m^{s_i}$ and $k_i \equiv g^{s_i}$. Hence if this protocol is iterated k times, then $\mathcal{V}$ will be convinced with confidence $1-2^{-k}$. This protocol is also zero-knowledge because it can be simulated (with the same probability distributions) by:

(1) *Choose a permutation* τ *of* Γ, *numbers* $r_1,\ldots,r_{|\Gamma|},t_1,t_2,t_3 \in \{1,\ldots,p-1\}$ *and* $e \in \{0,1\}$.
Compute and send the numbers: $z_{\tau(j)} \equiv k_j h^{r_j} \pmod p$ $(j \in \Gamma)$, $y \equiv m^{t_3} / S^e \pmod p$ *and* $x \equiv \left(\frac{g}{m}\right)^{t_1} h^{t_2} (S / z_{\tau(i)})^e \pmod p$.
(2) *Receive* $b \in \{0,1\}$.
(3) *Send in case*
$e=b=0$: $r_1,\ldots,r_{|\Gamma|},t_1,t_2,t_3$ *and* τ.
$e=b=1$: *index* $\tau(i)$ *and* t_1, t_2, t_3.
$e \neq b$: *restart this algorithm.*

If a person wants to deny a group signature, he can for instance use the disavowal protocol of [Ch90].

6. Some open problems

We have presented several group signature schemes, in which to open a signature the recipient asks Z or he performs a disavowal protocol with each group member. Is it possible to create other situations, such as: a majority of the group members can open a signature?

Is it possible to make digital group signatures other than by using [FS86] on undeniable signatures?

Can the results of [SS90] and [Per85] be applied to show that specific choices of the exponents in the schemes of Sections 2-4 and 5, respectively, protect anonymity in ways equivalent to known computational problems?

Is it possible to modify the fourth group signature system in such a way that the number of transmitted bits during the confirmation protocol is independent of the number of group members?

Acknowledgements

We would like to thank Jurjen Bos very much for his patience in listening to all the earlier schemes we proposed, and for his enthusiasm in breaking those; we also thank Gilles Brassard, Adam Cornford, Matthijs Coster, Jan-Hendrik Evertse, Maarten van der Ham, and Thijs Veugen for their help.

References

[BCDvdG87] Ernest Brickell, David Chaum, Ivan Damgård and Jeroen van de Graaf, Gradual and verifiable release of a secret, *Advances in Cryptology -CRYPTO 87*, C. Pomerance ed., Lecture Notes in Computer Science 293, Springer-Verlag, pp. 156-166.

[Ch85] David Chaum, Showing credentials without identification, *Advances in Cryptology - EUROCRYPT 85*, F. Pichler ed., Lecture Notes in Computer Science 219, Springer-Verlag, pp. 241-244.

[Ch87] David Chaum, Blinding for unanticipated signatures, *Advances in Cryptology -*

EUROCRYPT 87, D. Chaum, W. Price eds., Lecture Notes in Computer Science 304, Springer-Verlag, pp. 227-233.

[Ch90] David Chaum, Zero-knowledge undeniable signatures, *Advances in Cryptology - EUROCRYPT 90*, I. Damgård ed., Lecture Notes in Computer Science 473, Springer-Verlag, pp. 458-464.

[CvA89] David Chaum and Hans van Antwerpen, Undeniable signatures, *Advances in Cryptology - CRYPTO 89*, G. Brassard ed., Lecture Notes in Computer Science 435, Springer-Verlag, pp. 212-216.

[CEvdG87] David Chaum, Jan-Hendrik Evertse and Jeroen van de Graaf, An improved protocol for demonstrating possession of discrete logarithms and some generalizations, *Advances in Cryptology -EUROCRYPT 87*, D. Chaum, W. Price eds., Lecture Notes in Computer Science 304, Springer-Verlag, pp. 127-141.

[ElG85] Taher ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithm, *IEEE IT* **31** (1985), pp. 469-472.

[FS86] Amos Fiat and Adi Shamir, How to prove yourself: practical solution to identification and signature problems, *Advances in Cryptology -CRYPTO 86*, A.M. Odlyzko ed., Lecture Notes in Computer Science 263, Springer-Verlag, pp. 186-194.

[OOK90] Kazuo Ohta, Tatsuaki Okamoto and Kenji Koyama, Membership authentication for hierarchical multigroup using the extended Fiat-Shamir scheme, *Advances in Cryptology -EUROCRYPT 90*, I. Damgård ed., Lecture Notes in Computer Science 473, Springer-Verlag, pp. 446-457.

[Per85] René Peralta, Simultaneous security of bits in the discrete log, *Advances in Cryptology - EUROCRYPT 85*, F. Pichler ed., Lecture Notes in Computer Science 219, Springer-Verlag, pp. 62-72.

[SKI90] Hiroki Shizuya, Kenji Koyama and Toshiya Itoh, Demonstrating possession without revealing factors and its applications, *Advances in Cryptology AUSCRYPT 90*, J. Seberry and J. Pieprzyk eds., Lecture Notes in Computer Science 453, Springer-Verlag, pp. 273-293.

[SS90] Schrift and Shamir, The discrete log is very discreet, *Proc. 22nd STOC 1990*, pp. 405-415.