New Group Signature Schemes

(Extended Abstract)

L. Chen and T. P. Pedersen*

Aarhus University, Denmark

Abstract. Group signatures, introduced by Chaum and van Heijst, allow individual members of a group to sign messages on behalf of the group. The identity of the signer is kept secret except that a group authority can identify the signer if needed. This note presents a new group signature scheme, which hides the identity of the signer unconditionally and (unlike previous similar suggestions) allows new members to join the group. Simplifying this scheme a somewhat more efficient scheme giving computational anonymity is obtained. The group authority identifies the signer using a general method. This method can also be used to simplify three of the schemes suggested by Chaum and van Heijst. Finally, the schemes suggested here can be used to solve an open problem posed by Chaum and van Heijst.

1 Introduction

Group signatures as introduced in [CvH91] allow members of a group to sign messages on behalf of the group in such a way that

- the recipient of the signature can verify that it is a valid signature of that group, but can not discover which member of the group created it;
- in case of dispute later on either the group members together or a trusted authority can identify the signer.

Such a signature scheme can for example be used in invitations to submit tenders. All companies submitting a tender then form a group and each company signs its tender anonymously using the group signature. Later when the preferred tender has been selected the signer can be identified, whereas the signers of all other tenders will remain anonymous. A tender signed this way is binding as the identity of the signer can be computed without his cooperation (e.g., if the signer regrets his tender).

1.1 Related Work

Group signatures should not be confused with the related notion of group oriented signatures where certain subsets of a group of people are allowed to sign

* Funded by Carlsbergfondet

on behalf of the group. Such schemes do not provide a method for identifying the signers (see [D93] for a survey of such schemes). Another related concept is that of multi-signatures which require a digital signature by many persons (see [O88] and [OO93]).

To the best of our knowledge group signatures have previously only been described in [CvH91] (and again in [H92]), which presents four such schemes. One of these protects the anonymity of the signer unconditionally whereas the anonymity in the remaining three schemes depends on the difficulty of either factoring or computing discrete logarithms. The schemes also vary with respect to

- Identification of the signer by the authority:
  The group authority should be able to identify the signer based on the signature, the public key and some auxiliary, secret information. This is satisfied for the scheme giving unconditional anonymity, but not for the schemes giving computational anonymity. These schemes require that the authority contacts each group member until the signer is found.
- New group members:
  It should be possible to change the group dynamically in the sense that adding a new member to the group should only require that the public key and the auxiliary information of the authority be changed. This requirement is only satisfied for two of the schemes (both giving computational anonymity).

Thus, none of the four schemes perform optimally with respect to both of these properties.

[CvH91] also states the following as an open problem: is it possible to construct an efficient scheme in which certain subsets of the group members (e.g., the majority) can identify the signer?

1.2 Results

This paper presents two group signature schemes: one gives unconditional anonymity, whereas the other requires an assumption related to the discrete logarithm assumption. Both schemes allow the group to be changed dynamically, and the group authority can identify the signer given some auxiliary information about each group member. This actually uses a general principle, which also applies to some of the schemes in [CvH91].

Furthermore, the auxiliary information used to identify the signer in the two schemes presented in this paper can very easily be shared verifiably among the group members such that for some $k \geq 1$, any set of k group members can identify the signer (solving the above mentioned open problem).

1.3 Contents

The next section presents the method for identifying the signer. Section 3 then sketches a protocol due to Berry Schoenmakers (see [S93]) for proving knowledge of at least one out of many discrete logarithms. This idea underlies both

group signature schemes presented in this paper. Section 4 describes the scheme giving unconditional privacy, and Section 5 the scheme giving computational anonymity. Section 6 sketches a solution to the open problem mentioned above.

2 Identifying the Signer

This section describes a general way of identifying the signer of group signatures. Let the group members be $P_1, P_2, ..., P_n$ for some $n \in \mathbb{N}$, and let P_i have a secret key, s_i. The public key of the group is denoted by K, and a signature on the message, m, with respect to this public key is denoted by $\sigma_K(m)$. Suppose that a given scheme satisfies all requirements to group signatures except that the signer cannot be identified. In order to add this, the following property is needed:

> Given $s_1, \ldots, s_n$ and $(m, \sigma_K(m))$ it is possible to determine which of the secret keys was used to construct the signature $\sigma_K(m)$.

Then a group signature scheme can be constructed as follows. The group selects two public keys and each member gets two secret keys in the given scheme. Let the secret key of P_i be (s_i, t_i) and the public key of the group be (K_1, K_2). Using s_i and t_i, P_i makes signatures with respect to K_1 and K_2, respectively. Thus each member signs a message by signing it twice.

A pair (σ_1, σ_2) is a valid signature on m with respect to (K_1, K_2) if

$$\sigma_1 = \sigma_{K_1}(m) \quad \text{and} \quad \sigma_2 = \sigma_{K_2}(m).$$

The authority is given $(t_1, \ldots, t_n)$ as auxiliary information (and the identity of the member having t_i as secret key). By the above assumption this information enables the authority to identify the signer from σ_2, but of course not to sign (unless it can forge signatures with respect to K_1).

This way of identifying signer will be called double-signing. Using double-signing the signer in three of the schemes in [CvH91] can be identified much easier than using the interactive protocols proposed there (at the cost of twice as long signatures).

3 Knowledge of One out of Many Witnesses

The group signatures in this paper are based on a protocol for proving knowledge of one out of many witnesses presented in [S93].

Let G_q denote a group of prime order q and let g be a generator of G_q. The common input to the prover and verifier is $(g, h_1, \ldots, h_n)$ for some $n \in \mathbb{N}$, where each $h_i \in G_q$. Let $h = g^{x_i}$. Given one of these x_i's as secret input, the prover shows that he knows w such that for some $i \in \{1, 2, \ldots, n\}$: $h_i = g^w$. Schoenmakers protocol (based on [S91]) for doing this is sketched below for the case $w = x_1$.

1. P chooses $s_i, d_j \in \mathbb{Z}_q^*$ at random for $i = 1, 2, \ldots, n$ and $j = 2, 3, \ldots, n$. He then computes $a_1 = g^{s_1}$, $a_i = g^{s_i} h_i^{-d_i}$ for $i = 2, 3, \ldots, n$ and sends $(a_1, a_2, \ldots, a_n)$ to V.
2. V chooses a challenge $c \in \mathbb{Z}_q^*$ at random and sends it to P.
3. P first computes $d_1 = c - \sum_2^n d_i$ and then

$$r_i = \begin{cases} s_1 + x_1 d_1 & \text{for } i = 1, \\ s_i & \text{for } 2 \leq i \leq n \end{cases}$$

and sends $(d_1, \ldots, d_n, r_1, \ldots, r_n)$ to the verifier.
4. V verifies that

$$\sum_{i=1}^{n} d_i = c$$

and that

$$g^{r_i} = a_i h_i^{d_i} \qquad \text{for } i = 1, 2, \ldots, n.$$

Theorem 1 [S93]. *The above protocol is a witness indistinguishable (see [FS90]) proof of knowledge of w satisfying*

$$h_i = g^w \qquad \textit{for some } i \in \{1, 2, \ldots, n\}.$$

The proof is omitted here, but the intuition is that each challenge c defines a set of q^{n-1} possible choices for $d_1, \ldots, d_n$ and the prover cannot guess which. Witness indistinguishability follows from the fact that the verifier cannot tell which values of d_i the prover has selected before getting c.

Remark. An extension of this protocol allows the prover to show that he knows at least k out of n secret keys (see [CDS93]).

4 Unconditional Anonymity

This section presents a group signature scheme giving unconditional anonymity. We only consider the case with two persons (P_1 and P_2) in the group. The general case is obtained by a straightforward extension.

First a scheme allowing only one message to be signed is presented, and then it is shown how to extend it to $l \in \mathbb{N}$ signatures.

4.1 Signing One Message

Let two generators g_1 and g_2 of G_q be given (the actual selection of these generators is not important as long as no group member can express one as the power of the other). The secret key of P_i is $(x_{i1}, x_{i2}) \in \mathbb{Z}_q^2$ for $i = 1, 2$ and the public key is $h_i = g_1^{x_{i1}} g_2^{x_{i2}}$. Assuming $h_1 \neq h_2$ two such persons can from a group with public key

$$(h_1, h_2).$$

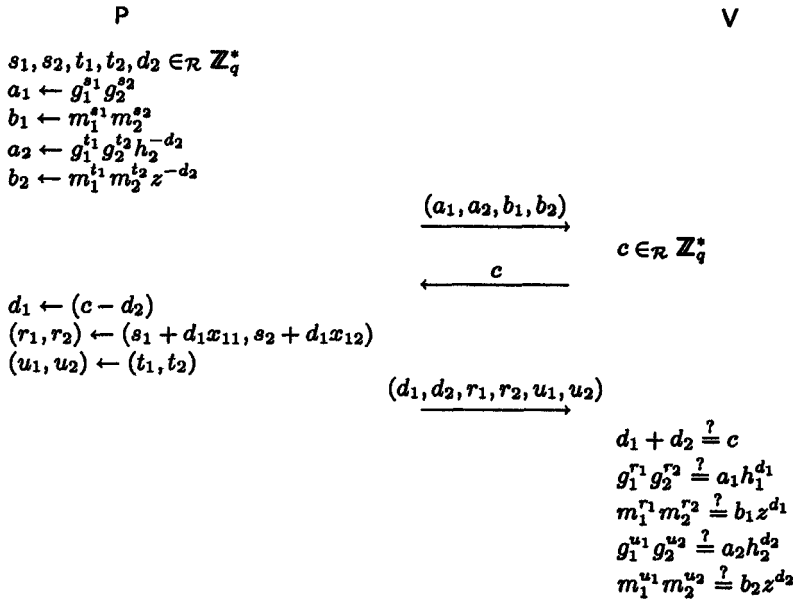

Fig. 1. Proving that z is constructed correctly with respect to h_1

Note, that the relation between P_i and h_i is revealed by the group's public key. If new members join the group their public keys are simply added.

P_i's signature on a message $m = (m_1, m_2)$ is $z = m_1^{x_{i1}} m_2^{x_{i2}}$ plus a proof that this is correct with respect to either h_1 or h_2. Next this proof will be explained and then its application to the group signature is described. The proof uses the idea of [S93] and is shown in Figure 1.

Using the arguments in [S93] it can be shown that the protocol in Figure 1 constitutes a proof that the prover knows a pair (s, t) such that

$$z = m_1^s m_2^t \quad \wedge \quad \left(h_1 = g_1^s g_2^t \quad \vee \quad h_2 = g_1^s g_2^t\right).$$

If $\log_{m_1} m_2 \neq \log_{g_1} g_2$, there are two possible witnesses to this claim: a pair (s, t) such that $h_1 = g_1^s g_2^t$ and a pair satisfying $h_2 = g_1^s g_2^t$.

Lemma 2. *If $\log_{m_1} m_2 \neq \log_{g_1} g_2$, the proof is witness indistinguishable (see [FS90]).*

Proof sketch. Given z there is exactly one pair (α_1, α_2) such that

$$h_1 = g_1^{\alpha_1} g_2^{\alpha_2} \qquad \text{and} \qquad z = m_1^{\alpha_1} m_2^{\alpha_2}$$

and exactly one pair (β_1, β_2) such that

$$z = m_1^{\beta_1} m_2^{\beta_2} \qquad \text{and} \qquad h_2 = g_1^{\beta_1} g_2^{\beta_2}.$$

We have to show that a prover, knowing (α_1, α_2) would construct messages with the same distribution as a prover knowing (β_1, β_2) (the protocol for a prover knowing the witness to h_2 is symmetric).

First, notice that the distribution of (a_1, a_2, b_1, b_2) is independent of whether the prover knows (α_1, α_2) or (β_1, β_2). Thus these numbers contain no information about the witness and no information about d_1 and d_2.

Next, given (a_1, a_2, b_1, b_2) there is exactly one possible tuple (v_1, v_2, w_1, w_2) such that

$$a_1 = g_1^{v_1} g_2^{v_2}, \quad b_1 = m_1^{v_1} m_2^{v_2}, \quad a_2 = g_1^{w_1} g_2^{w_2} \quad \text{and} \quad b_2 = m_1^{w_1} m_2^{w_2}.$$

Furthermore, the numbers (r_1, r_2, u_1, u_2) are uniquely determined by:

$$r_1 = v_1 + d_1\alpha_1, \quad r_2 = v_2 + d_1\alpha_2, \quad u_1 = w_1 + d_2\beta_1 \quad \text{and} \quad u_2 = w_2 + d_2\beta_2.$$

Thus the messages sent by the prover reveal no information about which of the two witnesses he knows. □

Given three hash-functions, $\mathcal{H}$, $\mathcal{H}_1$ and $\mathcal{H}_2$, P_i now signs a message m as follows:[2]

1. Compute from m a pair $(m_1, m_2) \in G_q^2$ as $m_j = \mathcal{H}_j(m)$ for $j = 1, 2$.
2. P_i computes z and executes the proof, computing c as $\mathcal{H}(a_1, b_1, a_2, b_2, m_1, m_2)$ (thus $\mathcal{H}$ must be "pseudo-random" as required by Fiat and Shamir in [FS87]).
3. The signature on m is $(z, d_1, d_2, r_1, r_2, u_1, u_2)$. It is verified by computing (a_1, b_1, a_2, b_2) and then verifying that c is the correct hash value and that $c = d_1 + d_2$.

Using the notation from the proof of Lemma 2 $(\alpha_1, \alpha_2) = (x_{11}, x_{12})$ and with very high probability $(\beta_1, \beta_2) \neq (x_{21}, x_{22})$ if P_1 is the signer. Thus, a person knowing the secret keys of P_1 and P_2 can tell whether P_1 or P_2 is the signer. Therefore, double-signing (see Section 2) can be used to turn this scheme into a group signature in which the authority can easily determine the signer.

Proposition 3. *The above group signature scheme satisfies the following three properties*

1. *Before a member signs a message he is unconditionally protected against framing,*[3] *but afterwards the other group members together can frame him (given sufficient computing power).*
2. *If P_i signs two different messages, then an unlimited powerful receiver can easily tell that both signatures correspond to h_i.*
3. *If both P_1 and P_2 sign a single message, then an unlimited powerful receiver can see that the two signatures were made by different members, but he cannot tell which member made which signature (in the case of n group members it can be shown that all $n!$ permutations are equally likely).*

[2] By choosing two random, but fixed strings ρ_1 and ρ_2, $\mathcal{H}_j$ can for example be implemented as $\mathcal{H}(\rho_j, m)$ for $j = 1, 2$. Thus only one hash function is really needed.

[3] A member is said to be *framed* if other members and non members together make a signature for which he will later be held responsible.

Proof sketch. Very briefly, all three claims follow from the fact that before a person has signed a message, all q possible secret keys are equally likely, whereas a signature together with the public key uniquely determines the secret key. □

Thus in the application to submission of tenders even an all powerful entity cannot decide (from the signatures) which candidate submits which tender.

Remark. A member able to compute discrete logarithms can make a group signature for which no other member will be held responsible.

4.2 Signing Many Messages

There are many ways to extend the above scheme to allow each member to sign $l \in \mathbb{N}$ messages. The following sketches one possibility for groups of two persons. Let $l+1$ generators $g_1, \ldots, g_{l+1}$ of G_q be given. The secret key of P_i is $(x_{i1}, \ldots, x_{i,l+1}) \in \mathbb{Z}_q^{l+1}$ for $i = 1, 2$. The public key of the group is

$$(g_1, \ldots, g_{l+1}, h_1, h_2)$$

where $h_i = g_1^{x_{i1}} \cdots g_{l+1}^{x_{i,l+1}}$ for $i = 1, 2$ (assume $h_1 \neq h_2$).

P_i's signature on a message $m = (m_1, \ldots, m_{l+1})$ is $z = m_1^{x_{i1}} \cdots m_{l+1}^{x_{i,l+1}}$ plus a proof that this is correct with respect to either h_1 or h_2. A witness indistinguishable proof of this can be constructed by modifying the protocol in Figure 1. The digital signature is then obtained as before.

Each members public key gives together with t signatures $t+1$ equations for determining that members secret key. The matrix corresponding to these equations is the same for all members of the group. If it has full rank the underlying proof system is witness indistinguishable and the signatures reveal no information about who signed which message. Assuming that all messages are chosen at random (this is reasonable if the hash functions used are good), the probability that this matrix has rank $t+1$ is:

$$1 - q^{-(l+1)} \sum_{j=1}^{t} \frac{q^j}{q^{l+1}} \approx 1 - \frac{q^t}{q^{l+1}}.$$

Thus in this scenario the scheme gives unconditional anonymity.

Remark. An unlimited powerful attacker can obtain messages for which the matrix mentioned above does not have full rank, and in that case the attacker might be able to rule out possible combinations of signers. Note, however, that this attack requires both the computation of discrete logarithms and the ability to control the hash function.

In the proposed application to submit tenders, it is not possible to perform such a chosen message attack as the signer selects the message.

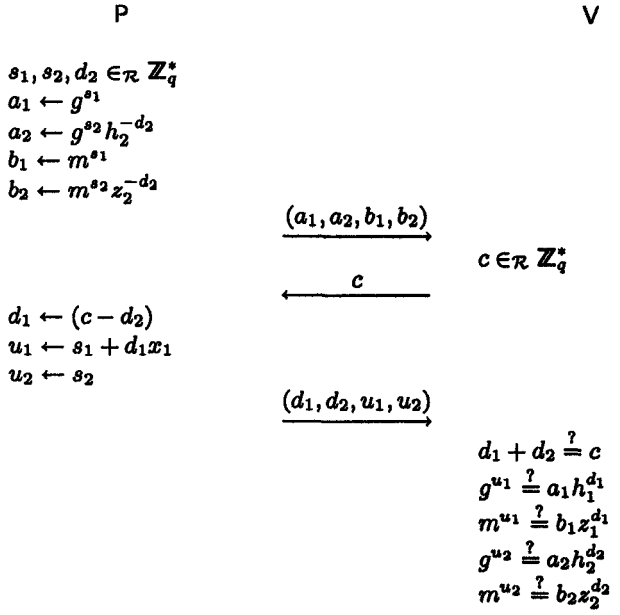

Fig. 2. Proving (*)

5 Computational Anonymity

Again the scheme will be described for groups consisting of two persons, P_1 and P_2. The public key of the group is (g, h_1, h_2) and the secret key of P_i is $x_i = \log_g h_i$ for $i = 1, 2$. When signing a message $m \in G_q$, P_i computes $z_i = m^{x_i}$, chooses $z_{3-i} \in G_q$ at random and proves that he knows w such that

$$(h_1 = g^w \quad \wedge \quad z_1 = m^w) \quad \vee \quad (h_2 = g^w \quad \vee \quad z_2 = m^w) \qquad (*)$$

The common input is $(g, h_1, h_2, m, z_1, z_2)$ and the secret input of the prover is x_1 or x_2 (in Figure 2 the prover knows x_1 — the case of x_2 is symmetric). By a straightforward modification of [S93], it can be shown that the protocol is a proof of knowledge of a witness to (*).

This protocol can be turned into a signature scheme as in Section 4 (and [FS87]). Next it will be argued that the receiver of such signatures cannot tell whether the signature was made using x_1 or x_2. The protocol is not witness indistinguishable in the sense of [FS90], where it is required that even a distinguisher who knows the possible witnesses cannot tell which witness the prover knows. That clearly does not hold for this protocol. Therefore the following contains a less formal argument for the anonymity of the group members. First, it is shown (based on a discrete logarithm assumption stated below) that if no group member has previously made any signatures it is infeasible to tell who made a given signature. Then it is argued that knowledge of other signatures

plus the identity of the corresponding signer does not help the receiver deciding which member made a given signature. The security of the scheme depends on the following two assumptions.

Assumption 4. Let D be a probabilistic polynomial time machine which takes (g, h, m, z) as input. Let $Pr_{=}$ denote the probability that D outputs 1, when m is chosen uniformly at random and $\log_g h = \log_m z$. Let $Pr_{\neq}$ denote the probability that D outputs 1 when m and z are chosen uniformly at random. Then for all D: $|Pr_{=} - Pr_{\neq}|$ is negligible as a function of the order of the group (for all but a negligible fraction of the pairs (g, h)).

For some of the arguments a stronger version of Assumption 4 is needed. Consider an oracle algorithm, A, working as follow on input (p, q, g, h_1, h_2):

1. Repeat the following a polynomial number of times:
   (a) Choose a message, m at random and select $j \in \{1, 2\}$.
   (b) Get m^{x_j} from the oracle.
2. Get a pair (m_0, z_0), where m_0 is chosen at random and z_0 is either $m_0^{x_1}$ or $m_0^{x_2}$.
3. Output j.

Assumption 5. For every polynomially bounded A as above, the probability that A outputs j such that $z_0 = m_0^{x_j}$ is "polynomially close" to $\frac{1}{2}$.

Now consider three possible provers:

P_0: The input satisfies $z_1 = m^{x_1}$ and $z_2 = m^{x_2}$;
P_0 just chooses d_1 at random.
P_1: The input satisfies $z_1 = m^{x_1}$ and z_2 is chosen uniformly at random;
P_2: z_1 is chosen uniformly at random and $z_2 = m^{x_2}$.

Lemma 6. *Under Assumption 4 the following holds. No polynomially bounded verifier can distinguish between P_0 and P_2. Similarly, P_0 and P_1 cannot be distinguished.*

Proof sketch. Let a verifier $\tilde{V}$ be given.
Given g, h_1, m, z_1. We want to decide whether z_1 is chosen at random or $z_1 = m^{x_1}$.

1. Choose x_2 at random and compute $h_2 = g^{x_2}$ and $z_2 = m^{x_2}$.
2. Execute the protocol $(P_2, \tilde{V})$.
3. If $\tilde{V}$ outputs P_0 then output 0. Otherwise output 1.

It is easy to see that $\tilde{V}$'s view in case $z_1 = m^{x_1}$ is that generated by P_0 and if z_1 is chosen at random it is the same as that generated by P_2. □

This lemma shows that given a signature from either P_1 or P_2 it is not feasible to tell which secret was actually used. However, in general the distinguisher may

have received many signatures before trying to recognize which secret key was used in a given signature.

Assume that the hash function used in the signature scheme is such that making a signature is equivalent to executing the basic proof system with an honest verifier (i.e., choosing the challenge using $\mathcal{H}$ corresponds to choosing the challenge at random).

If a distinguisher can identify the signer of a signature given some previous signatures and the identity of the corresponding signers, then the distinguisher could also identify the signer after executing the protocol in Figure 2 acting as the honest verifier. However, given the identity of the prover, the transcript of the honest verifier can be generated with the correct distribution by the distinguisher itself. Thus the only help, which the distinguisher obtains from these previous signatures is $m_i^{x_j}$ for the various messages $m_1, m_2, \ldots$ and $j = 1$ or $j = 2$. Assumption 5 says that this information is of no help.

6 Shared Identification of the Signer

When using double-signing each member has two secret keys of which one (called t_i in Section 2) is sent to the authority. If this key is shared among the n group members in a k out of n threshold scheme any k members can identify the signer.

For the two schemes presented here the distribution as well as the identification can be done quite efficiently. Due to space limitations only the main idea will be sketched here for the computationally secure scheme (the same techniques can be used for the scheme in Section 4 using the non-interactive, verifiable secret sharing scheme from [P92]).

First, each P_i shares his secret key t_i verifiably among all n group members. This can be done quite efficiently using the fact that g^{t_i} is part of the group's public key. Each member P_j gets a share t_{ij} of this key, for which $g^{t_{ij}}$ is publicly known (see [P91]).

Later, when k members want to decide whether a given signature z on a message m was made by P_i, each computes $m^{t_{ij}}$. Each of these partial results can then easily be combined into m^{t_i}. Cheating parties can be discovered by requiring a proof that this result is correct (e.g., using the efficient protocol in [Cha91]). They conclude that P_i was indeed the signer if and only if this equals z.

7 Conclusion

The first group signature scheme presented here protects the individual group member unconditionally. The second scheme is more efficient but the analysis is harder. Both schemes allow new group members to join the group dynamically, and they are very well suited to distributing the authority's information among the group members. This solves a problem posed by Chaum and van Heijst.

Furthermore, both schemes use a general method for identification of the signer, which can also be used in other schemes.

Acknowledgments

We thank Ronald Cramer, Ivan Damgård and Berry Schoenmakers for discussions about the results of this paper.

References

[Cha91] D. Chaum. Zero-Knowledge Undeniable Signatures. In *Advances in Cryptology - proceedings of EUROCRYPT 90*, Lecture Notes in Computer Science, pages 458-464. Springer-Verlag, 1991.

[CvH91] D. Chaum, E. van Heijst. Group Signatures. In *Advances in Cryptology - proceedings of EUROCRYPT 91*, Lecture Notes in Computer Science, pages 257-265. Springer-Verlag, 1991.

[CDS93] R. Cramer, I. Damgård and B. Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. *To be presented at CRYPTO'94.*

[D93] Y. Desmedt. Threshold Cryptosystems. In *Advances in Cryptology - proceedings of AUSCRYPT 92*, Lecture Notes in Computer Science, pages 3–14, 1993.

[H92] E. van Heijst. *Special Signature Schemes.* PhD thesis, CWI, 1992.

[FS90] U. Feige and A. Shamir. Witness Indistinguishable and Witness Hiding Protocols. In *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing*, pages 416 – 426, 1990.

[FS87] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - proceedings of EUROCRYPT 86*, Lecture Notes in Computer Science, pages 186 – 194. Springer-Verlag, 1987.

[O88] T. Okamoto. A Digital Multisignature Scheme Using Bijective Public-Key Cryptosystems. *ACM Trans. on Comp. Sys.*, 6(8):432 – 441, 1988.

[OO93] K. Ohta and T. Okamoto. A Digital Multisignature Scheme Based on the Fiat-Shamir Scheme. In *Advances in Cryptology - proceedings of ASIACRYPT 91*, Lecture Notes in Computer Science, pages 139 – 148. Springer-Verlag, 1993.

[P91] T. P. Pedersen. A Threshold Cryptosystem without a Trusted Party. In *Advances in Cryptology - proceedings of EUROCRYPT 91*, Lecture Notes in Computer Science, pages 522 – 526. Springer-Verlag, 1991.

[P92] T. P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *Advances in Cryptology - proceedings of CRYPTO 91*, Lecture Notes in Computer Science, pages 129–140. Springer-Verlag, 1992.

[S93] B. Schoenmakers. Efficient Proofs of Or. Manuscript, 1993.

[S91] C. P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[WP90] M. Waidner and B. Pfitzmann. The Dining Cryptographer in the Disco: Unconditional Sender and Recipient Untraceability with Computationally Secure Serviceability. In *Advances in Cryptology - proceedings of EUROCRYPT 89*, Lecture Notes in Computer Science, page 690. Springer-Verlag, 1990.