University of Technology Chemnitz-Zwickau

Department of Computer Science

Theoretical Computer Science
and Information Security

Technical Report

TR-95-16-F

Blind multisignature schemes and their relevance to electronic voting

Patrick Horster · Markus Michels · Holger Petersen

August 1995

Straße der Nationen 62
D-09111 Chemnitz, Germany

Phone: (+49) 371-531-1479
Fax: (+49) 371-531-1810

Blind multisignature schemes and their relevance for electronic voting[2]

Patrick Horster · Markus Michels[1] · Holger Petersen[1]
Theoretical Computer Science and Information Security
University of Technology Chemnitz-Zwickau,
Straße der Nationen 62, D-09111 Chemnitz, Germany
email: {pho,mmi,hpe}@informatik.tu-chemnitz.de

Abstract

In this paper we present the first blind multisignature scheme based on the discrete logarithm problem. Furthermore, we discuss several extensions of the scheme and introduce an extended concept of multisignature schemes.

Then we describe, how a blind multisignature scheme can be used as a building block in electronic voting schemes. In the new mixed-type voting scheme the administrator is splitted into several. If at least one is honest, then the fairness property can be satisfied without having an additional communication phase after the voting phase.

1 Introduction

The concept of blind signature schemes was introduced by Chaum in 1982 [1]. In a blind signature scheme an owner Alice wants to obtain a digital signature on a message chosen by herself from the notary Nancy, but Nancy shouldn't have any idea which message she signs. If she gets the message and the signature later, it must not be possible that Nancy can find a relationship between some blinded and unblinded parameters.

A multisignature scheme allows multiple signers to sign the same message. In a blind multisignature scheme, we have one owner Alice, who wants to obtain a digital signature from several signers (which are the notaries in the blind signature scheme), so that each signer doesn't know a relationship between the blinded and unblinded message and signature parameters. This means they can't recognize the signature later, even if they all collude. This kind of scheme has many applications, like shared anonymous access control or multiparty pseudonymous credentials.

A simple solution for a blind multisignature scheme is, that every signer signs the message using a conventional blind signature scheme. This has the drawback that the data expansion increases with the number of signers. In 1993 a multisignature scheme based on the discrete logarithm problem without data expansion was proposed by Harn [2, 3], which has been modified and generalized by Horster, Michels and Petersen [4]. Here we present the first blind multisignature scheme based on this scheme using the design criteria for blind signature schemes [5].

We give a brief review of the basic multisignature scheme in section two. In section three we present our approach for the first blind multisignature scheme and give a security analysis in section four. Section five describes some extensions. In section six we present an extended concept of multisignature schemes. Section seven describes, how a blind multisignature scheme can be used as building block in electronic voting schemes and in section eight we mention that blind multisignature scheme can be useful in electronic cash schemes as well.

2 The basic multisignature scheme

First we review the underlying signature scheme, which is a variant of the ElGamal signature scheme [6, 7]. Then we show how this can be extended to provide multisignatures [3].

The underlying signature scheme

A trusted authority chooses a large prime p, a generator $\alpha \in \mathbf{Z}_p^*$ and publishes them as system parameters. The signer *Alice* chooses a random number $x_A \in_R \mathbf{Z}_{p-1}$ 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

[1]Part of this work was done, while visiting the Crypto Group of Gemplus Card International, Sarcelles, France

[2]appeared in Proc. of 11th Annual Computer Security Applications Conference, New Orleans, IEEE Press, (1995), pp. 149 – 155.

$k \in_R \mathbf{Z}_p^*$, computes $r := \alpha^k \pmod p$ and solves the equation $s := x_A \cdot (m+r) + k \pmod{p-1}$. The triple $(m; r, s)$ is the signed message. It can be verified by checking the congruence $\alpha^s \equiv y_A^{m+r} \cdot r \pmod p$.

The multisignature scheme

In a multisignature scheme we have t different signers $u_1, \ldots, u_t$. Each of them has a secret key $x_i \in \mathbf{Z}_q^*$ and a related public key $y_i := \alpha^{x_i} \pmod p$. Each signer u_i chooses a random number $k_i \in_R \mathbf{Z}_q^*$ and computes $r_i := \alpha^{k_i} \pmod p$. Then he broadcasts r_i to all other signers and the clerk, such that every signer can compute $r := \prod_{i=1}^t r_i \pmod p$. Now each signer u_i computes his signature parameter s_i by $s_i := x_i \cdot (m+r) + k_i \pmod{p-1}$. He transmits s_i to a clerk who additionally knows m and r and whose task is to check each individual signature by verifying the congruence $y_i^{m+r} \cdot r_i \equiv \alpha^{s_i} \pmod p$. He also generates the multisignature of the message m by computing $s := \sum_{i=1}^t s_i \pmod{p-1}$. Now the triple (m, r, s) is the multisignature of the message m which can be verified by checking the congruence $y^{m+r} \cdot r \equiv \alpha^s \pmod p$, where $y := \prod_{i=1}^t y_i \pmod p$.

Security considerations

In [4], two attacks on this multisignature scheme have been described in detail, which we review briefly.

1. In the first attack, the attacker Carol chooses her public key y_C as $y_C := y_B \cdot y_D \pmod p$. But as the congruence
$$y_A \cdot y_B \cdot y_D \equiv y_A \cdot y_C \pmod p$$
and the verification equation of the multisignature scheme hold, if Alice, Bob and Donald have signed the message, no verifier can decide if the message was really signed by them or just by Alice and Carol. Note that this attack is weak because Carol doesn't know her own secret key and as a result she can't sign any message. The attack can be easily prevented by checking each signer's public key using a zero-knowledge proof for the discrete logarithm of it.

2. The second attack is more subtle. If several signers collude it is possible, that these signers choose their random numbers k_i such that the sum is equal to zero modulo $p-1$. As a result the verification equation also holds if their part of the multisignature is not considered. This attack can be prevented by adding the identity of the signers to the signed message.

3 The blind multisignature scheme

To construct a blind multisignature scheme, we use the design criteria for Meta-blind signature schemes described in [5]. The idea is, that every signer gives a blind signature on the message m and that all these signatures are combined afterwards.

A trusted authority chooses large primes p, q with $q|(p-1)$, a generator $\alpha \in \mathbf{Z}_p^*$ of a multiplicative subgroup of order q and publishes them as system parameters. The initialization is the same as in the basic multisignature scheme. To prevent the first attack described in the last section, each signer u_i proofs, that he has chosen his public key correctly. Then he chooses a random number $\tilde{k}_i \in_R \mathbf{Z}_q^*$ and computes $\tilde{r}_i := \alpha^{\tilde{k}_i} \pmod p$. He broadcasts $\tilde{r}_i$ to the owner of the signature *Alice*, who computes $\tilde{r} := \prod_{i=1}^t \tilde{r}_i \pmod p$. Alice unblinds the parameter $\tilde{r}$ and computes $r := \tilde{r}^a \cdot y^b \pmod p$, where $y := \prod_{i=1}^t y_i \pmod p$. Then she blinds the message m by $\tilde{m} := (m + r + b) \cdot a^{-1} - \tilde{r} \pmod q$. Alice transmits $\tilde{m}$ and $\tilde{r}$ to each signer u_i, who computes his signature parameter $\tilde{s}_i$ by $\tilde{s}_i := x_i \cdot (\tilde{m} + \tilde{r}) + \tilde{k}_i \pmod q$. He transmits $\tilde{s}_i$ to *Alice* who knows m and r, checks each individual signature by verifying the congruence
$$y_i^{\tilde{m}+\tilde{r}} \cdot \tilde{r}_i \equiv \alpha^{\tilde{s}_i} \pmod p.$$
She also generates the multisignature of the message m by computing $\tilde{s} := \sum_{i=1}^t \tilde{s}_i \pmod q$ and $s := \tilde{s} \cdot a \pmod q$. Now the tuple (r, s) is the blind multisignature of the message m which can be verified by checking the congruence
$$y^{m+r} \cdot r \equiv \alpha^s \pmod p,$$
where $y := \prod_{i=1}^t y_i \pmod p$. This equation is satisfied as the following equivalence holds:
$$\begin{aligned}
& y^{m+r} \cdot r \\
\equiv\ & y^{m+r} \cdot (\tilde{r}^a \cdot y^b) \\
\equiv\ & y^b \cdot y^{m+r} \cdot \left(\prod_{i=1}^t \tilde{r}_i^a\right) \\
\equiv\ & y^{m+r+b} \cdot \left(\prod_{i=1}^t \alpha^{\tilde{k}_i a}\right) \\
\equiv\ & y^{m+r+b} \cdot \left(\prod_{i=1}^t \alpha^{(\tilde{s}_i - x_i \cdot (\tilde{m}+\tilde{r}))a}\right) \\
\equiv\ & y^{m+r+b} \cdot \left(\prod_{i=1}^t \alpha^{\tilde{s}_i a}\right) \cdot \left(\prod_{i=1}^t \alpha^{-x_i(\tilde{m}+\tilde{r})a}\right) \\
\equiv\ & y^{m+r+b} \cdot \alpha^{a \sum_{i=1}^t \tilde{s}_i} \cdot \left(\prod_{i=1}^t y_i^{-(\tilde{m}+\tilde{r})a}\right) \\
\equiv\ & \alpha^{a\tilde{s}} \cdot y^{-(\tilde{m}+\tilde{r})a} \cdot y^{m+r+b} \\
\equiv\ & \alpha^s \cdot y^{-((m+r+b)a^{-1} - \tilde{r} + \tilde{r})a + m + r + b} \\
\equiv\ & \alpha^s \pmod p
\end{aligned}$$

The second attack mentioned in section two is possible, with the additionally condition that the colliding signers choose their public keys y_i such that the product is equal to one modulo p. This is necessary, as y is used for the computation of the parameter $r := \tilde{r}^a \cdot y^b$. The prevention of this attack by adding the identities into the unblinded message fails, if the owner *Alice* also colludes with this group. In this case they might cheat the other signers, who still believe, that the pretended persons are signing the document, while only the innocent are actually signing it.

To countermeasure this extended attack, the signers should use the modified signature equation

$$\tilde{s}_i := x_i \cdot (\tilde{m} + \tilde{r}) + \tilde{k}_i \cdot h(y_1, y_2, \ldots, y_t) \pmod q,$$

where $h : {\mathbf{Z}_p}^t \rightarrow \mathbf{Z}_q^*$ is a suitable one-way hash function. This results in the verification equation

$$y^{m+r} \cdot r^{h(y_1, y_2, \ldots, y_t)} \equiv \alpha^s \pmod p,$$

which can be checked by first computing the hash value $h(y_1, y_2, \ldots, y_t)$ of the public keys of all signers and verifying, if the product of these public keys is equal to y and the verification equation is satisfied.

4 Security analysis

In the blind multisignature scheme we have several possible attackers: *Outsiders*, who are not participating the issue of the blind signature, the *signers* who are registered users and can play an active part in the signing process and the owner *Alice*. Furthermore some of these different attackers might collude.

Roughly speaking, the possibility for outsiders to cheat relies on the security of the underlying signature scheme. The signers ability to cheat relies on the security of the underlying multisignature scheme. Their possibility to deanoymize the signing message during the signature process relies on the security of the underlying blind signature scheme. Alice can't generate a signature herself, as this relies on the security of the multisignature scheme. She also can't cheat some honest signers together with some dishonest one's as this is avoided by adding the hash value of the public keys of all signers to the signature equation.

1. **Outsider attack:** In the known-message scenario the outsider knows the public parameters, some blind multisignatures and the related individual signatures if he eavesdrops the communication channel. One possibility he has is a replay attack: He eavesdrops $\tilde{r}_{A,1}$ and $\tilde{s}_{A,1}$ of user Alice from the signature generation and tries to use this for generating a new signature. He puts $\tilde{r}_{A,1}$ into the broadcast channel as the new parameter $\tilde{r}_{A,2}$. As he doesn't know the related parameter $\tilde{k}_{A,2}$ he can't compute $\tilde{s}_{A,2}$. Thus he has to use the eavesdroped value $\tilde{s}_{A,1}$. But this is only correct if $\tilde{m}'_1 = \tilde{m}'_2$ and $r_1 = r_2$. As the hash function is collision-free the first condition is satisfied in those cases where the same group of signers signs the same message twice. Thus this attack is not successful. Other possibilities might be to forge the individual signature of Alice or the multisignature. Both cases can be seen as the attempt to forge the used variant of the Meta-ElGamal signature scheme. A detailed security analysis for the Meta-ElGamal scheme is described in [5]. Stronger attacks seems to be useless because m' changes if a different group of signers signs the same message or different messages are signed. It seems to be hard to combine them to get additional signatures.

2. **Signer attack:** Insiders can try to choose their parameters $\tilde{k}_i$ such that $\tilde{r}$ is chosen suitable, like in the second attack described in section two. If they fix $\tilde{r}$ in advance they must be able to compute discrete logarithms if they don't know at least one of the values $\tilde{r}_i$. This contradicts the discrete logarithm assumption.

   Concerning the anonymity, any signer can never obtain more information about the blinding of the message and the parameters, as in the case, where all signers collude. Thus we only have to consider the case, where the signers sum all their blind signature parameters $\tilde{s} := \sum \tilde{s}_i \pmod q$ and know the product of their blinded signature parameter $\tilde{r} := \prod \tilde{r}_i \pmod p$. Then they can try to find a relation between these values for a given blinded message $\tilde{m}$ and any triple (m, r, s) of an unblinded signed message.

   The anonymity of the blind multisignature in this case is guaranteed by the following theorem.

Theorem 1 *Given two large primes p and q, $q|(p-1)$, a generator α of a multiplicative subgroup of order q of $\mathbf{Z}_p$ and a public key $y \in \mathbf{Z}_p$ of order q. Assume, we have two (arbitrary) signed messages (m_i, r_i, s_i) with $\alpha^{s_i} \equiv y^{r_i + m_i} \cdot r_i \pmod p$, for $i \in \{1, 2\}$, and the two relations $a := s_1 \cdot s_2^{-1} \pmod q$ and $b := (m_2 + r_2) \cdot a - m_1 - r_1 \pmod q$. Then the equation (1) holds:*

$$r_1 \equiv r_2^a \cdot y^b \pmod p. \tag{1}$$

Proof:

Using the equation $\alpha^{s_2} \equiv y^{r_2 + m_2} \cdot r_2 \pmod p$ the proof is straightforward:

$$\begin{aligned} r_2^a \cdot y^b &\equiv r_2^a \cdot y^{(m_2+r_2)\cdot a - r_1 - m_1} \\ &\equiv r_2^a \cdot (\alpha^{s_2} \cdot r_2^{-1})^a \cdot y^{-(r_1 - m_1)} \\ &\equiv \alpha^{s_2 \cdot a} \cdot y^{-(r_1 - m_1)} \\ &\equiv \alpha^{s_2 s_1 s_2^{-1}} \cdot y^{-(r_1 - m_1)} \\ &\equiv \alpha^{s_1} \cdot y^{-(r_1 - m_1)} \equiv r_1 \pmod p \end{aligned}$$

Furthermore (a, b) is the unique solution for equation (1) for two given valid signed messages (m_1, r_1, s_1) and (m_2, r_2, s_2), because of cardinality reasons: There exist $(q-1)^2$ possibilities for choosing $a, b \in \mathbf{Z}_q^*$. Fixing (m_1, r_1, s_1) we have $(q-1)$ choices of $m_2 \in \mathbf{Z}_q^*$ and $(q-1)$ choices of r_2 in the subgroup generated by α ($r_2 \neq 1$). The computation of s_2 is uniquely fixed as

$s_2 := \log_\alpha(y^{r_2+m_2} \cdot r_2) \pmod p$. As a result, it's impossible, to find two different tuples (a_1, b_1) and (a_2, b_2), that satisfy equation (1) for these messages. Together with theorem 1, this proves untraceability.

3. **Owner attack:** Alice might reveal individual signatures but this doesn't endanger the security of the scheme. If she doesn't compute the blind multisignature correctly from the individual blind signatures the verification equation of the multisignature is not satisfied and thus this kind of attack can be discovered by the verifiers.

5 Extensions of the scheme

There are several possible extensions of the blind multisignature scheme:

1. **Different generators:** Each signer u_i can use his own generator $\alpha_i \in \mathbf{Z}_p^*$ and the related keys $x_i \in_R \mathbf{Z}_q^*$ and $y_i := \alpha_i^{x_i} \pmod p$. Then $\tilde{r}_i$ is computed as $\tilde{r}_i := \alpha_i^{\tilde{k}_i} \pmod p$ and the verification can be done more general by the equation $y^{m+r} \equiv r \cdot \prod_{i=1}^{t} \alpha_i^{s_i} \pmod p$. For $t = 2$ this can be regarded as a variant of Okamoto's first blind signature scheme in [8]. Note that the efficiency depends on the number of signers. The computational effort to compute the multi-exponentiation of the $\alpha_i^{s_i}$ is bounded by 4/3 times the effort for a single exponentiation, while the storage used for the precomputation in this case increases linear in the number of signers [9].

2. **Use of the Meta-Multisignature scheme:** Instead of using the Harn multisignature scheme, which uses the variant EG II.3 of the Meta-ElGamal signature scheme [7], we can also use the variants of the Meta-Multisignature scheme described in [4]. This gives us a multitude of blind multisignature schemes.

3. **Use of the Meta-Message recovery scheme:** Instead of using the Meta-Multisignature scheme with appendix, we can also use the Meta-Multisignature scheme with message recovery [4] based on the Meta-Message recovery signature scheme [5]. This results in the *first* blind multisignature schemes giving message recovery.

4. **Other designs for the computation of r:** Instead of computing $r := \tilde{r}^a \cdot y^b \pmod p$ we can also use the general computation formula $r := \tilde{r}^A \cdot y^B \cdot \alpha^C \pmod p$, where A, B and C are suitable general functions with arguments a and b. The function A shouldn't be equal to 0, to avoid the disappearance of the secret parameter $\tilde{k}$ in the signature equation. This leads to many different designs for blind multisignatures and generalizes the results in [5].

6 An extended concept of multisignature schemes

It is possible, that a verifier does not know the public keys of all signers authentically. In this case, he might be interested, if particular signers, whose authentic public keys he knows, participated in the multisignature or not. Using the original multisignature scheme, this verification can't be done, as the multisignature might be universally forged in this case. We briefly describe this attack here.

Assume, that the verifier does not know the public key y_1 authentically and that the attacker is aware of this. He can then choose an arbitrary message m and signature parameters r and s and solve the verification equation for the public key y_1:

$$y_1 := \prod_{i=2}^{t} y_i^{-1} \cdot \alpha^{s(m+r)^{-1}} \cdot r^{-(m+r)^{-1}} \pmod p. \quad (2)$$

This means, y_1 is a free parameter, thus the verifier couldn't mention this fraud.

To avoid this attack, we can add the hash value of all public keys y_i into the signature equation. Each signer u_i computes his signature parameter

$$s_i := x_i \cdot (m + r) + k_i \cdot h(y_1, \ldots, y_t) \pmod q.$$

Then the attack described above is no longer possible, as the (non authentic) public key y_1 appears in the exponent, so that equation (2) can no longer be evaluated.

7 Electronic voting schemes

A blind multisignature scheme can be used as a building block in cryptographic applications, e.g. in electronic voting schemes. We first outline a typical mix-type voting scheme, describe a multi-decryption encryption scheme and present a new voting scheme that uses a multi-decryption encryption scheme and a blind multisignature scheme as building blocks.

7.1 Voting schemes using an anonymous channel

Voting schemes using an anonymous channel can guarantee the anonymity of the voters, if this channel is secure. This channel can e.g. be implemented by a mix-net introduced by Chaum [10], therefore these schemes are sometimes called mix-type voting schemes. Here we outline the system due to Fujioka, Okamoto and Ohta [11] (others are similar): Every voter encrypts his vote by a bit commitment scheme. Then, in the voting slip issuing phase, the administrator signs the encrypted vote using a blind signature scheme. On the other hand, the administrator gets a voter's signature of the blinded encrypted vote. Therefore, the administrator does not know the encrypted vote and can, later, prove to anyone by showing the voter's signature that he issued a valid voting slip to this voter. In the voting phase, the voter sends the encrypted vote and the adminstrators's signature on that through an anonymous channel. In the counting phase, every voter sends the key to open the commitment, so that the administrator can decrypt the encrypted vote. If *all* communication is taped on a public board, the administrator can be prevented from cheating, if he does not dare to be caught, as was shown by Chen and Burmester [12]. The use of a bit commitment scheme is necessary to guarantee that the administrator doesn't reveal intermediate results which might influence the whole election (this is the fairness property). However, the price for that is an additional communicating phase for opening the commitment.

7.2 A multi-decryption cryptosystem

A *multi-decryption cryptosystem* is a public-key cryptosystem in which one sender sends an encrypted message to n receiver. All n receiver are needed to decrypt the encrypted message, less than n receiver can't decrypt it.

Now we describe an multi-decryption encryption scheme based on the ElGamal encryption scheme [6], as it can be used in our voting scheme later. This can be regarded as a special case of the ElGamal encryption scheme with threshold decryption due to Desmedt and Fraenkel [13]. We show how to extend it to guarantee that every receiver decrypts correctly.

A trusted authority chooses a large prime p, an integer $q|(p-1)$, a generator $\alpha \in \mathbf{Z}_p^*$ and publishes them as system parameters. All receivers i choose a random number $x_i \in_R \mathbf{Z}_q^*$ and computes $y_i := \alpha^{x_i} \pmod p$. Each receiver publishes y_i and keeps x_i secret.

To encrypt a message $m \in \mathbf{Z}_{p-1}$, Alice picks a uniformly chosen random number $k \in_R \mathbf{Z}_q^*$, computes $r := \alpha^{-k} \pmod p$, $K := (\prod_i y_i)^k \pmod p$ and $C := m \cdot K \pmod p$. Alice sends the tuple (C, r) to the receivers.

Every receiver computes $K_i' := r^{x_i} \pmod p$, all receivers can compute $K' := \prod_i K_i' \pmod p$ and $m := K' \cdot C \pmod p$. Every receiver i can show to anyone that he has decrypted correctly by proving in zero-knowledge that the discrete logarithm of K_i' to base r is the same as the discrete logarithm of y_i to base α using Chaum's proof [14], if the ciphertext is authentic. We assume that a receiver is guilty, if he refuses to give the proof. Then any receiver, who cheats during decryption, will be caught.

7.3 A new voting scheme

Using n administrators instead of one, where at least one administrator is honest, we can describe a simplified voting scheme. First we give a rough outline of the scheme:

The voter encrypts the vote using a multi-decryption encryption scheme. Then the n administrators issue a valid voting slip by signing the encrypted vote using a blind multisignature scheme. One the other hand, the administrators get the blinded encrypted message signed by the voter. During the voting phase, the voter sends the encrypted vote and the administrators' multisignature on it through an anonymous channel. After the voting phase the encrypted vote is decrypted and published. As at least one administrator is honest and refuses to decrypt during the voting phase, we don't need another communication phase here to guarantee fairness. Additionally, it is not necessary to tape *all* communication on a public board.

We explain the new scheme in more detail. We use the following notation: the integer t_j is related to the candidate j, v_i the vote of voter i, E the encryption function and D the decryption function of the multi-decryption encryption scheme, S_A is the administrators' signature and V_A their verification function, S_j is voter j's signature and V_j verification function, c_j is voter j's encrypted vote and $\tilde{c}_j$ is voter j's blinded encrypted vote. Now each phase is proceeded sequentially:

- *Initialization*: Candidate i is related to t_i, which satisfies a publically known redundancy scheme. System parameters and public parameters are authentically published, they are only used for one election.

- *Voting slip issuing phase*: Voter j chooses his vote $v_j := t_i||r$, where r is chosen randomly, if he likes to vote for candidate i. Then he encrypts this using a multi-decryption encryption scheme, that means he computes $c_j := E(v_i)$. By sending his identity, he now gets a blind multisignature on this ciphertext by the administrators, provided he asks for it the first time in the current election and signs the blinded ciphertext himself. To be more precise, the voter j knows $S_A(c_j)$ and the administrators knows $S_j(\tilde{c}_j)$.

- *Voting phase:* Every voter j sends the tuple $(c, S_A(c))$ through an anonymous channel to the administrators. The tuple is then published in a list on a public board.

- *Decryption phase:* The administrators decrypt together any ciphertext c to obtain $g := D_A(c)$. This value is added to each tuple in the list on the public board. Furthermore, each administrator adds additional parameter which can be used late to prove that he has decrypted correctly. If the multi-decryption encryption scheme described above is used, then the each administrator adds the parameter K_i' on the public board. In this case, each line in the list is given by $(c, S_A(c), g, K_1', \ldots, K_n')$ with $c = (C, r)$.

- *Claiming phase:* Each voter j checks if his vote was decrypted correctly. If not, he sends anonymously $(v_j, c_j, S_A(c_j))$ to a third person who checks if these values are related. Then the board will be corrected.

- *Counting phase:* The administrators can now compute and publish the result and every voter can do so as well.

7.4 Security considerations

We discuss the possibilities of cheating by voters and administrators, assuming that at least one administrator is honest.

The voter can't get more than one voting slip, as the administrator gets a signature of the voter during the voting slip issuing phase. Furthermore, as the honest administrator guarantees that the voter knows his voting slip after his signature on the blinded vote is received by the administrators, every claim of the voter that he did not get a voting slip can be regarded as an attempt to cheat. The voter can claim that a correctly decrypted vote is incorrect. But as the vote satisfies a suitable redundancy scheme, he is not able to compute another valid vote related to the same ciphertext. What he still can do is claiming that he didn't vote for a candidate (he just wants to support democracy) and has sent trash. Then the administrators must be able to prove to anyone that they decrypted correctly, which is possible with the multi-decryption encryption scheme described above.

The administrators can't distribute voting slips freely if one of them is honest. If a voter possesses a valid voting slip but doesn't vote, the honest administrator can avoid that dummy votes are added. He can further avoid that intermediate results are revealed. A cheating administrator can influence the decryption such that the published vote in the list of the public board is wrong. As the voter can prove to anyone anonymously that he knows the correct vote and the published vote is indeed wrong, this doesn't influence the election. Furthermore, as each administrator has published the additional parameters which are used in the proof that he has decrypted correctly, the cheating administrator will be caught, as he can't give this proof.

7.5 Discussion

The advantage of the scheme is that it gets rid off the assumption that *all* communication must be written on an public board (to be more precise, the encrypted vote and, later, the vote itself and the additional parameter must be written on the board) and gets rid off the additional communicating phase to open the commitment. Clearly, the price paid for this is that at least one administrator is assumed to be honest. We give two arguments that this assumption is reasonable:

First, by use in practise, the anonymous channel would be simulated by a mix-net where it is assumed that at least one mix-center is honest as well. Therefore, a trustworthy entity must be assumed anyway.

Second, in the initialization of the scheme, system parameters without trapdoors must be chosen by the administrators or other authorities. For example, Chen and Burmester [12] assumed the existence of a trusted center to generate a composite module which is needed to use the Fiat-Shamir scheme [15] in their system. Clearly, if this center is untrustworthy, the security is completely lost. While it is difficult in this case to distribute this center into several centers where only one is honest, it seems to be possible if the security of the used schemes are based on the discrete logarithm problem. One honest center can avoid that, say, a trapdoor-prime or a trapdoor-generator is chosen, and can guarantee that public keys of the administrators are authentic, if the extended concept of multisignature schemes described in section 6 is used for certification and at least one public key of the administrators is authentically known. Obviously, the centers' tasks can be done by the administrators, if at least one is honest. Then, the existence of a trusted center is not necessary.

As a result, it seems reasonable to assume that at least one administrator is honest.

A more serious problem of the schemes mentioned so far, is the possibility of verifiable buying of votes. The coercer, who taps the line between the administrators and the voter, might force the voter to use random numbers prepared by him in the voting slip issuing phase. Then he sees in the voting phase if these numbers will appear again or not and therefore check if the voter votes the "correct" candidate. This might be prevented physically if the voter can't determine the random numbers by himself. If he is supplied by random numbers by a (trusted) physical device, which also does the computation for him, then this attack will fail. Clearly, a more powerful coercer, who can physically see what the voter votes, can still be successful. This coercer model, however, seems to be less of practical than of theoretical interest as the effect of vote-buying is only non negligible in large scale elections, if the number of bought votes is high. As the powerful coercer can't see the vote of several voter simultaneously, he needs a large number of supporters for supervising the voters. This scenario seems not to be very realistic.

8 Electronic cash

Finally, an idea is mentioned, which allows to get rid off an undesirable property of a well known electronic cash scheme.

In electronic cash applications, the most well known cash system due to Brands [16] suffers from the property that the bank can compute the secret key of the user if he (erroneously) spent a coin of a small amount twice. Then a cheating bank (colluding with two shops) can compute a coin with a large amount and claims that this was the coin the user spent twice. As this bank knows the secret key of the user she can simulate the withdrawal and payment protocol. If the bank server is splitted into several bank servers and at least one is trusted by the user (e.g. the one which is controlled by the society of bank users), then this property can be avoided. In the withdrawal protocol a blind multisignature scheme can be used instead of a blind signature scheme. A more precise description is object to further work.

9 Conclusion

We have presented the first blind multisignature scheme, mentioned several ways to obtain various schemes and introduce an extended concept of multisignatures. As application, a new mix-type voting scheme was outlined, which is based on blind multisignatures. Finally, it has been mentioned that blind multisignatures can also be useful in electronic cash.

10 Acknowledgement

The authors would like to thank David Naccache for helpful comments during preparation of this paper.

References

[1] D. Chaum, "Blind signatures for untraceable payments", Advances in Cryptology: Proc. Crypto'82, New York: Plenum Press, (1983), pp. 199–203.

[2] L.Harn, "(t,n) threshold signature and digital multisignature", Workshop on Cryptography & Data Security, June 7–9., (1993), pp. 61–73.

[3] L.Harn, "New digital signature scheme based on discrete logarithm", Electronics Letters, Vol. 30, No. 5, (1994), pp. 396–398.

[4] P.Horster, M.Michels, H.Petersen, "Meta-Multisignature scheme based on the discrete logarithm problem", Proc. IFIP/SEC '95, Cape Town, South Africa, Chapman & Hall, (1995), pp. 128–142.

[5] P.Horster, M.Michels, H.Petersen, "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. 224–237.

[6] T.ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, Vol. IT-30, No. 4, July, (1985), pp. 469–472.

[7] P.Horster, M.Michels, H.Petersen, "Meta-ElGamal signature schemes", Proc. of the 2nd ACM Conference on Computer and Communications Security, Fairfax, Virginia, (1994), pp. 96–107.

[8] T.Okamoto, "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.

[9] S.-M.Yen, C.-S.Laih, A.K.Lenstra, "Multi-exponentiation", IEE Proc.-Comput. Digit. Tech., Vol. 141, No. 6, November, (1994), pp. 325–326.

[10] D.Chaum, "Untraceable electronic mail return addresses and digital pseudonyms", Communcations of the ACM, Vol. 24 (2), Feb., (1981), pp. 84–88.

[11] A.Fujioka, T.Okamoto, K.Ohta, "A practical secret voting scheme for large scale elections", Lecture Notes in Computer Science 718, Advances in Cryptology: Proc. Auscrypt'92, (1993), pp. 244–251.

[12] L.Chen, M.Burmester, "A practical voting scheme which allows voters to abstain", Proc. Chinacrypt'94, (1994), ISBN 7-03-004363-4 / TB 398, pp. 100–107.

[13] Y.Desmedt, Y.Fraenkel, "Threshold cryptosystems", Lecture Notes in Computer Science 435, Advances in Cryptology: Proc. Crypto'89, Berlin: Springer Verlag, (1990), pp. 307–315.

[14] D.Chaum, "Zero-knowledge undeniable signatures", Lecture Notes in Computer Science 473, Advances in Cryptology: Proc. Eurocrypt'90, Berlin: Springer Verlag, (1991), pp. 458–464.

[15] A.Fiat, A.Shamir, "How to prove yourself: Practical solutions to identification and signature problems", Lecture Notes in Computer Science 263, Advances in Cryptology: Proc. Crypto'86, Berlin: Springer Verlag, (1987), pp. 186–194.

[16] S.Brands, "Untraceable off-line cash in wallet with observers", Lecture Notes in Computer Science 773, Advances in Cryptology: Proc. Crypto'93, Berlin: Springer Verlag, (1994), pp. 302-318.