University of Technology Chemnitz-Zwickau

Department of Computer Science

Theoretical Computer Science
and Information Security

Technical Report

TR-95-19-D

On the risk of disruption in several multiparty signature schemes

Markus Michels

Version: February 2, 1996

Straße der Nationen 62
D-09111 Chemnitz, Germany

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

On the risk of disruption in several multiparty signature schemes

Markus Michels
Theoretical Computer Science and Information Security
University of Technology Chemnitz-Zwickau,
Straße der Nationen 62, D-09111 Chemnitz, Germany
E-mail: mmi@informatik.tu-chemnitz.de
(Revised Preliminary Draft - Please do not distribute)

February 2, 1996

Abstract

Multiparty cryptography is an important topic in contemporary cryptography. In this paper we clarify the difference between several multiparty signature concepts and point out some weaknesses in some group signature schemes and multisignature schemes. More precisely, we point out that a multisignature scheme is vulnerable to universal forgery by an insider attacker under reasonable assumptions. This attack can be applied to some generalizations as well. Then we present a universal forgery attack on two group signature schemes with anonymous (t, n) shared signature generation. Furthermore, we show that in two group signature schemes with non-anonymous (t, n) shared signature generation presented at EUROCRYPT'94, it can't be guaranteed that a signer can decide with whom he is going to sign a message. All attacks have in common that the protocol is disrupted. Thus the attacks are not undetectable, what might limit their scope. However, as they can only be detected afterwards and knowledge leaked by protocol disruptions must be useless, such attacks are not acceptable in general and must be avoided. Finally, we suggest some heuristic fixes. As a result, the design of multiparty signature schemes is not an easy task and requires a great amount of care.

Key words: Data Security, Public-Key Cryptography, Group-oriented Cryptography, Cryptanalysis, Digital Signature

1. Introduction

In the well known conventional digital signature concept by Diffie and Hellman [6] one signer is sufficient to sign a message known to him and one verifier is sufficient to check the validity of any (given) signature. Therefore signatures are transferable. Some concepts has been suggested to modify the property of transferibility, e.g. undeniable signature schemes and designated confirmer signatures [3, 2], another the property that the signer knows the message he is going to sign, e.g. blind signature schemes [1]. We can call these signature concepts *property concepts*, as some properties of the conventional model are modified.

In other concepts, we may call them *multiparty concepts*, it is required that only several signers should be able to sign, that only several verifiers should be able to verify a signature or that only members of a group should be able to sign. All these signature concepts can be regarded as *multiparty signature schemes*. They can be distinguished in the classes *non-group signature schemes* and *group signature schemes*. The difference between these classes is that in the first one only individual (public) keys exist while in the later one a group (public) key exists.

One example of the first class are *signature schemes with multiple signature generation* (mostly called *multisignature schemes*), in which some signers want sign the same message together. Another example is the concept of *signature schemes with multiple signature verification*, in which only some predetermined verifiers together can verify a signature.

The class of group signature schemes can be further distinguished in group signature schemes with

- *non-anonymous* (t, n) *shared signature generation*, where only at least t out of n signers together are able to generate a signature and the identities of these t signers must be known to the verifier(s).

- *anonymous* (t, n) *shared signature generation*, where only at least t out of n signers together are able to generate a signature and it's unknown whose t signers have signed. Therefore the signers remain anonymously.

- *abolishable anonymous* (t, n) *shared signature generation*, where only at least t out of n signers together are able to generate a signature and the signers remain anonymously. However, under certain circumstances it's possible to abolish the anonymity of the signers.

- *non-anonymous* (t, n) *shared signature verification*, where a special group of t out of n verifiers together are able to verify a signature and only those t verifiers of that group can verify.

- *anonymous* (t, n) *shared signature verification*, where only at least t out of n verifiers together are able to verify a signature and any t verifiers of that group can verify.

Clearly, it's further possible to combine one the first three concepts of the list with the two last concepts. Additionally the resulting multiparty concept can be combined with any property concept.

Most of these concepts have been proposed in the literature before. Multisignature schemes where first introduced by Itakura and Nakamura [10], group signature schemes with abolishable anonymous (t, n) shared signature generation (and $t = 1$) by Chaum and van Heijst [4], group signature schemes with anonymous (t, n) shared signature generation by Desmedt and Fraenkel [5], group signature schemes with non-anonymous (t, n) shared signature generation by Li, Hwang and Lee [11] and group signature schemes with anonymous (t, n) shared signature verification by de Soete, Vedder and Quisquater [13]. However, other names for some of the concepts are used, which might be somewhat misleading: Chaum and van Heijst called their concept just *group signature schemes* and Li, Hwang and Lee *threshold multisignature schemes*. We are of the opinion that the notation given above is more suitable to stress the differences between the several concepts.

In this paper, we will point out that several multiparty signature schemes are vulnerable to some active attacks, which have in common that the attacker is an insider and the protocol will be disrupted. Therefore the attacks are not undetectable, what might limit their scope. On the other hand, protocol disruptions shouldn't be a threat for a scheme and the attack can only be detected *afterwards*. Further, it should be emphasized that during the signature generation the attacker may choose his value after receiving some values from his co-signers and victims in the broadcast phase, that is we allow *rushing*. We are of the opinion that this assumption is reasonable. Furthermore, this assumption is *not* excluded in the description of the attacked schemes in [7, 8, 9, 11].

In the sections 2,3 and 4 it will be shown that

- a multisignature scheme and some generalizations can be forged universally [7, 8, 9].
- two group signature schemes with anonymous (t, n) shared signature generation can be forged universally [8]
- and in two proposals for a group signature scheme with non-anonymous (t, n) shared signature generation given in [11] dishonest signers can force their victims to sign a document together although the victims are not aware of all of their co-signers.

In section 5 it is pointed out how to repair the schemes using heuristic methods.

2. Multisignature schemes

The new attack can be applied to Harn's multisignature scheme, which is based on a normal signature scheme [7].

2.1 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 $x \in \mathbf{Z}_{p-1}$ and computes $y := \alpha^x \pmod{p}$. She publishes y and keeps x 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 $k \in \mathbf{Z}_{p-1}$, computes $r := \alpha^k \pmod{p}$ and solves the equation

$$s := x \cdot (m + r) - k \pmod{p - 1}.$$

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

$$r \cdot \alpha^s \equiv y^{m+r} \pmod{p}.$$

Clearly, to avoid existentiell forgery m must satisfy a redundancy scheme or its hashvalue $h(m)$ is signed. *We assume that this countermeasure will be used in the following although it will not be mentioned explicitely.*

2.2 Multisignature schemes

In a multisignature scheme we have t different signers $u_1, \ldots, u_t$ and 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 $k_i \in \mathbf{Z}_q^*$ and computes $r_i := \alpha^{k_i} \pmod{p}$. Then he broadcasts r_i to all other signers, 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 $y_i^{m+r} \equiv r_i \cdot \alpha^{s_i} \pmod{p}$. 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

$$y^{m+r} \equiv r \cdot \alpha^s \pmod{p},$$

where $y := \prod_{i=1}^{t} y_i \pmod{p}$.

2.3 Known attacks

Horster, Michels and Petersen pointed out that the scheme is vulnerable to two different attacks [9]. The first is rather simple: An insider attacker chooses his public key as product of the public keys of his victims. Therefore, if the victims (and some other signers) generate a multisignature on a message, then he can claim that he (and the other signers) have generated it. This attack can be easily prevented by checking, if each signer has chosen his public key correctly, e.g. using a zero-knowledge proof. The second is more subtle and the result is similar to the result of the cryptanalysis in the section 4. 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 into the signed message.

2.4 New attack

First the possible attacker models are described. In this multisignature environment we can distinguish between an *insider*-, an *outsider*- and a *clerk* attack. The outsider does not play an active role in a protocol, while the insider does. The *clerk* attack can be *active* (he does not follow the protocol) or *passive* (he just reveals some parameters). Furthermore, the strength of the insider attack depends on the *number* of the attacker. *In the following, our attack assumes that one insider attacker and the passive clerk collude*, while the overwhelming majority of signers, more precisely all except the insider attacker, can assumed to be honest. Furthermore, we assume that in the broadcast phase during the signature generation the attacker can choose him parameters *after* he received the parameters of his co-signers and victims, that is we allow *rushing*.

The *aim* of the attack is universal forgery, that is the attacker obtains a multisignature on any message $\tilde{m}$ chosen by him.

Assume that an insider attacker u_1, say Charly, wants his victims $u_2, \ldots, u_t$ to sign a message $\tilde{m}$ with him. They reject, but agree to sign the innocent message m with him.

Therefore, co-signer and victim u_i (for all $i \in \{2, \ldots, t\}$) chooses a random number $k_i \in \mathbf{Z}_{p-1}^*$, computes $r_i := \alpha^{k_i} \pmod p$ and sends r_i all other co-signers. Charly waits until he received $r_2, \ldots, r_t$. Then he picks a random $k_1 \in \mathbf{Z}_{p-1}^*$, computes $r_1 := \alpha^{k_1} \pmod p$, $r := \prod_{i=1}^t r_i \pmod p$, $\tilde{r} = \tilde{m} + r - m \pmod{p-1}$ and $\tilde{r}_1 := \tilde{r} \cdot \prod_{i=2}^t r_i^{-1} \pmod p$. He sends $\tilde{r}_1$ instead of r_1 to all co-signers. Now each co-signer u_i (for all $i \in \{2, \ldots, t\}$) computes $\tilde{r} := \tilde{r}_1 \cdot \prod_{i=2}^t r_i \pmod p$ (although she thinks that r is computed), calculates

$$s_i := x_i \cdot (m + \tilde{r}) + k_i \pmod{p-1}$$

and sends s_i to the clerk. As Charly is not aware of $\tilde{k}_1$, he can't compute a valid individual signature to message m.

He disrupts the protocol by telling all co-signers that he losts the parameter (e.g. by a crash). Then, the protocol will be repeated and this time Charly behaves honestly. As the new signature is not needed in the further attack, it doesn't matter, if the co-signers refuse to repeat the protocol. Therefore, this attack doesn't neccessarily assume *rejoining* (that is a faulty processor (here Charly) can continue later), if the old signatures co-signed by Charly remain valid. Clearly, it would be unreasonable to assume that all old signatures of a signer are invalid, only because he errorously broadcasts a wrong value.

The clerk, colluding with Charly, reveals $s_2, \ldots, s_t$ and Charly computes

$$s_1 := x_1 \cdot (m + \tilde{r}) + k_1 \pmod{p-1}$$

using the known parameter k_1. Charly computes $s := \sum_{i=1}^t s_i \pmod{p-1}$. Then $(\tilde{m}, r, s)$ is a valid signature, as

$$\alpha^s \equiv \alpha^{\sum_{i=1}^t s_i} \equiv \alpha^{\sum_{i=1}^t x_i \cdot (m+\tilde{r}) + k_i} \equiv \alpha^{\sum_{i=1}^t x_i \cdot (\tilde{m}+r) + k_i} \equiv y^{(\tilde{m}+r)} \cdot r \pmod p.$$

Again, the main assumption of this attack is that Charly can choose his $\tilde{r}_1$ according to his wishes and the parts $r_2, \ldots, r_t$ of the victims are already known to him, that is rushing is allowed.

2.5 Extensions

The attack can also be applied to the multisignature given by Harn in [8] and some generalizations of Harn's multisignature scheme given by Horster, Michels and Petersen in [9].

3. Group signature schemes with anonymous (t,n) shared signature generation

The attack also works in the group signature scheme with anonymous (t, n) shared signature generation suggested by Harn [8]. In fact, the attack can be regarded as an improvement of the clerk's attack discussed in section 3.4.iv) in [8]. We first review the underlying signature scheme and then the first group signature scheme with anonymous (t, n) shared signature generation.

3.1 Underlying signature scheme

A trusted authority chooses two large primes p, q with $q|(p-1)$, a generator $\alpha \in \mathbf{Z}_p^*$ of order q and publishes them as system parameters. The signer *Alice* chooses a random number $x \in \mathbf{Z}_q$ and computes $y := \alpha^x \pmod p$. She publishes y and keeps x 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 $k \in \mathbf{Z}_q$, computes $r := \alpha^k \pmod p$ and solves the equation

$$s := x \cdot m - k \cdot r \pmod q.$$

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

$$r^r \cdot \alpha^s \equiv y^m \pmod p.$$

3.2 First group signature scheme with anonymous (t,n) shared signature generation

In the first group signature scheme with anonymous (t,n) shared signature generation a trusted authority is assumed. The authority picks two large primes p, q with $q|(p-1)$, a generator $\alpha \in \mathbf{Z}_{p-1}$ of order q and a polynomial $f(x) := \sum_{i=1}^{t} a_i \cdot x^t$ with $a_1, \ldots, a_t \in \mathbf{Z}_q^*$. Then the authority determines $x := f(0)$ as group secret key and computes $y := \alpha^x \pmod p$ as the group public key. The secret share of each user u_i is $x_i := f(ID_i) \pmod q$ and the related public key is $y_i := \alpha^{x_i} \pmod p$.

If t different signers, say $u_1, \ldots, u_t$ without loss of generality, wants to sign a message m, each signer u_i chooses a random number $k_i \in \mathbf{Z}_q^*$ and computes $r_i := \alpha^{k_i} \pmod p$. Then he broadcasts r_i to all other signers, 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

$$s_i := x_i \cdot \prod_{j=1, i \neq j}^{t} \frac{-ID_j}{ID_i - ID_j} \cdot m - k_i \cdot r \pmod q.$$

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

$$y_i^{m \cdot \prod_{j=1, i \neq j}^{t} \frac{-ID_j}{ID_i - ID_j}} \equiv r_i^r \cdot \alpha^{s_i} \pmod p.$$

He also generates the signature of the message m by computing $s := \sum_{i=1}^{t} s_i \pmod q$. Now the triple (m, r, s) is the signature of the message m which can be verified by checking the congruence

$$y^m \equiv r^r \cdot \alpha^s \pmod p.$$

3.3 Cryptanalysis of the first scheme

It's possible for one insider attacker to forge this scheme universally in a similar manner and with the same assumed attacker model as described in section 2. Assume that an insider attacker u_1, say Charly, wants his victims $u_2, \ldots, u_t$ to sign a message $\tilde{m}$ with him. They reject, but agree to sign the innocent message m with him. It should be stressed that only one signer must be dishonest.

Therefore, victim u_i with $2 < i < t$ chooses a random number $k_i \in \mathbf{Z}_q^*$, computes $r_i := \alpha^{k_i} \pmod p$ and sends r_1 to the other signer. Insider attacker Charly waits until he received all $r_2, \ldots, r_t$. Charly picks a random $k_1 \in \mathbf{Z}_{p-1}^*$, computes $r_1 := \alpha^{k_1} \pmod p$, $r := \prod_{i=1}^t r_i \pmod p$, $d := \tilde{m} \cdot m^{-1} \pmod q$, $\tilde{r} = r^d \pmod p$ and $\tilde{r}_1 := \tilde{r} \cdot \prod_{j=2}^t r_j^{-1} \pmod p$. He sends $\tilde{r}_1$ instead of r_1 to the other signer. Now victim u_i computes $\tilde{r} := \tilde{r}_1 \cdot \prod_{j=2}^t r_j \pmod p$ (although he thinks that r is computed), calculates

$$s_i := x_i \cdot \prod_{j=1, i \neq j}^{t} \frac{-ID_j}{ID_i - ID_j} \cdot m - k_i \cdot \tilde{r} \pmod q$$

and sends s_i to the clerk. As Charly is not aware of $\tilde{k}_1$, he can't compute a valid s_1 to message m. He disrupts the protocol by finding a suitable excuse. Then, the protocol might be repeated and this time Charly behaves honestly. It's worth noting that the resulting signature of this repetition is not needed in the following. Therefore, if the victims refuse to repeat the protocol, this doesn't matter. As a result, it's not neccessary to assume rejoining. Anyway, the re-start of the protocol can be done by another dishonest insider-signer as well. But then we have to assume that two signers are dishonest.

The clerk, colluding with Charly, reveals $s_2, \ldots, s_t$ to Charly. Charly computes

$$s_1 := x_1 \cdot \prod_{j=1, j \neq 1}^{t} \frac{-ID_j}{ID_1 - ID_j} \cdot m - k_1 \cdot \tilde{r} \pmod q$$

as the other signer, but using k_1. Now Charly can compute $s := d \cdot \sum_{i=1}^t s_i \pmod q$. Then the triple

$$(\tilde{m}, \tilde{r}, s)$$

is a valid signature for the message $\tilde{m}$, as

$$\tilde{r} \equiv r^d \equiv \alpha^{d \cdot \sum_{i=1}^t k_i} \pmod p \tag{1}$$

and

$$\alpha^s \equiv \alpha^{d \cdot \sum_{i=1}^t s_i} \equiv \alpha^{d \cdot (\sum_{i=1}^t x_i \cdot \prod_{j=1, j\neq i}^t \frac{-ID_j}{ID_i - ID_j} \cdot m - k_i \cdot \tilde{r})}$$
$$\equiv \alpha^{\sum_{i=1}^t x_i \cdot \prod_{j=1, j\neq i}^t \frac{-ID_j}{ID_i - ID_j} \cdot m \cdot d - k_i \cdot \tilde{r} \cdot d} \equiv \alpha^{\sum_{i=1}^t x_i \cdot \prod_{j=1, j\neq i}^t \frac{-ID_j}{ID_i - ID_j} \cdot \tilde{m} - k_i \cdot \tilde{r} \cdot d} \overset{(1)}{\equiv} y^{\tilde{m}} \cdot \tilde{r}^{-\tilde{r}} \pmod p.$$

Again, the main assumption of this attack is that Charly can choose his r_1 according to his wishes and the parts $r_2, \ldots, r_t$ of the victims are already known to him, that is we allow rushing.

3.4 Second group signature scheme with anonymous (t,n) shared signature generation

In the second group signature scheme with anonymous (t,n) shared signature generation no trusted authority is assumed. Every user is a authority, but all users choose two large primes p, q with $q|(p-1)$, a generator $\alpha \in \mathbf{Z}_{p-1}$ of order q together. Each user u_i chooses a polynomial $f_i(x) := \sum_{j=1}^t a_{i,j} \cdot x^t$ with $a_{i,1}, \ldots, a_{i,t} \in \mathbf{Z}_q^*$. Then $x_i := f_i(0)$ is the user u_i's (main) secret key and the related (main) public key is $y_i := \alpha^{x_i} \pmod p$. Furthermore each user u_i gets $x_{i,j} := f_j(ID_i)$ from all other users u_j as additional secret keys with the related public keys are $y_{i,j} := \alpha^{x_{i,j}} \pmod p$.

If t different signers, say $u_1, \ldots, u_t$ without loss of generality, wants to sign a message m, each signer u_i chooses a random number $k_i \in \mathbf{Z}_q^*$ and computes $r_i := \alpha^{k_i} \pmod p$. Then he broadcasts r_i to all other signers, 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

$$s_i := \left(x_i + \sum_{j=t+1}^{n} x_{i,j} \cdot \prod_{e=1, i \neq e}^{t} \frac{-ID_e}{ID_i - ID_e} \right) \cdot m - k_i \cdot r \pmod q.$$

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

$$\left(y_i \cdot (\prod_{j=t+1}^{n} y_{i,j})^{\prod_{e=1,i\neq j}^{t} \frac{-ID_e}{ID_i - ID_e}} \right)^m \equiv r_i^r \cdot \alpha^{s_i} \pmod p.$$

He also generates the signature of the message m by computing $s := \sum_{i=1}^t s_i \pmod q$. Now the triple (m, r, s) is the signature of the message m which can be verified by checking the congruence

$$y^m \equiv r^r \cdot \alpha^s \pmod p.$$

3.5 Cryptanalysis of the second scheme

Now the attack can be applied in the same manner as described above.

4. Group signature schemes with non-anonymous (t,n) shared signature generation

In this section we point out that in a slightly modified way it's possible to show that in two group signature schemes with non-anonymous (t,n) shared signature generation by Li, Hwang and Lee [11], the signer can't be sure who his co-signers are. There are situations where this property might be important in a *non-anonymous* setting, e.g. a signer Alice agrees to sign a special message together Bob and Charly but not with Charly and Donald (with $t = 3$). Note that in contrary to the schemes of the last section, the verifier needs to know whose signers have generated the signature. Therefore, in this concept there might be difference if, say Alice, Bob and Charly or Alice, Charly and Donald sign the same message m. It should be explicitly stressed that the schemes given by Li, Hwang and Lee can still be used in those situations in which this property is not important.

4.1 First signature schemes with non-anonymous (t,n) shared signature generation

Now we review the first scheme by Li, Hwang and Lee [11]. In this scheme a trusted dealer is assumed. The dealer picks two large primes p, q with $q|(p-1)$, a generator $\alpha \in \mathbf{Z}_{p-1}$ of order q and a polynomial $f(x) := \sum_{i=1}^t a_i \cdot x^t$ with $a_1, \ldots, a_t \in \mathbf{Z}_q^*$. Then the dealer determines $x := f(0)$ as group secret key and computes $y := \alpha^x \pmod p$ as the public key of the group. The secret share of each user i is $u_i := g_i + f(ID_i) \pmod q$ and the public keys are $y_i := \alpha^{u_i} \pmod p$ and $z_i := \alpha^{g_i} \pmod p$.

If a group B with $|B| = t$ of signers would like to generate a signature of a message m, then each signer $i \in B$ picks k_i and computes $r_i := \alpha^{k_i} \pmod p$. This r_i is sent to the other signers using a broadcast channel. If all r_i are available, then each signer computes

$$R := \prod_{i \in B} r_i \pmod p \text{ and } E = H(m, R)$$

using a suitable hash function H. Then each signer i computes

$$s_i := u_i \cdot \left(\prod_{j \in B, i \neq j} \frac{0 - ID_j}{ID_i - ID_j} \right) + k_i \cdot E \pmod q.$$

Each signer i sends the values m and s_i to a clerk who can check the validity of the individual signature by

$$\alpha^{s_i} \equiv y_i^{\left(\prod_{j\in B, i\neq j} \frac{0-ID_j}{ID_i-ID_j}\right)} \cdot r_i^E \pmod p.$$

The clerk combines the individual signature by

$$S := \sum_{i\in B} s_i \pmod q.$$

Then (B, m, R, S) is the group signature of the signers in B to message m. This signature can be checked by computing

$$T := \prod_{i\in B} z_i^{\left(\prod_{j\in B, i\neq j} \frac{0-ID_j}{ID_i-ID_j}\right)} \pmod p,$$

$$E := H(m, R)$$

and checking, whether the equation

$$\alpha^S \equiv y \cdot T \cdot R^E \pmod p$$

holds.

4.2 Security analysis

Here we point out that the above mentioned property can't be gained. We assume that signer u_1, say Charly, is an insider attacker. He is a member of the set of signers $B = \{1, \ldots, t\}$ and $\tilde{B} = \{1, \cdots, t-1, t+1\}$. All signers $u_2, \ldots, u_{t-1}$ agree to sign the message m with all co-signers in B, but not with co-signer u_{t+1} instead of u_t, that is all co-signers in $\tilde{B}$. We assume that co-signer u_{t+1} is aware of the attack and colludes.

Now the protocols with the signer in B will be proceeded according to the protocol: Fix

$$b_i := \prod_{j\in B, j\neq i} \frac{0-ID_j}{ID_i-ID_j} \pmod q \text{ and } \tilde{b}_i := \prod_{j\in \tilde{B}, j\neq i} \frac{0-ID_j}{ID_i-ID_j} \pmod q$$

Each signer u_i with $i \in B - \{1\}$ picks k_i and computes $r_i := \alpha^{k_i} \pmod p$ and broadcasts r_i to the other co-signers in B. Charly waits until he knows $r_2, \ldots, r_t$ and gets $r_{t+1} := \alpha^{k_{t+1}} \pmod p$ from co-signer u_{t+1}. Then he picks a random k_1 and computes $r_1 := \alpha^{k_1} \pmod p$,

$$R := \prod_{i=1}^{t} r_i \pmod p,$$

$$\tilde{r}_i := r_i^{b_i^{-1}\cdot \tilde{b}_i} \pmod p$$

and

$$\tilde{R} := r_{t+1} \cdot \prod_{i=1}^{t-1} \tilde{r}_i \pmod p.$$

He reveals $\tilde{r}_1 := \tilde{R} \cdot R^{-1} \cdot r_1 \pmod p$ to all co-signers in B. Therefore, all signers in B computes $\tilde{R}$ instead of R, as

$$\tilde{R} \equiv \tilde{r}_1 \cdot \prod_{i=2}^{t} r_i \pmod p,$$

although the honest signers $u_2, \ldots, u_t$ are not aware of it. Signer u_i with $i \in B$ computes $E := H(m, \tilde{R})$ and

$$s_i := u_i \cdot b_i + k_i \cdot E \pmod q.$$

and sends s_i to the clerk. Then the protocol for signing the message m by the group B will be disrupted by Charly, finding a suitable excuse. The protocol for group B will be repeated and this time Charly behaves honestly. If the co-signers in B refuse that, this doesn't matter.

The clerk, colluding with Charly, reveals the parameter $s_2, \ldots, s_{t-1}$ to Charly. Furthermore, signer u_{t+1} computes

$$s_{t+1} := u_i \cdot \tilde{b}_{t+1} + k_{t+1} \cdot E \pmod q.$$

and sends s_{t+1} to Charly. Charly computes

$$s_1 := u_1 \cdot b_1 + k_1 \cdot E \pmod q$$

and $S := s_{t+1} + \sum_{i \in \tilde{B} \cap B} \tilde{b}_i \cdot b_i^{-1} \cdot s_i \pmod q$. Then $(\tilde{B}, m, S, \tilde{R})$ is a valid signature of group $\tilde{B}$ for message m, as

$$E = H(m, \tilde{R}), \tilde{R} \equiv r_{t+1} \cdot \prod_{i \in B \cap \tilde{B}} \tilde{r}_i \equiv \alpha^{k_{t+1} + \sum_{i \in B \cap \tilde{B}} k_i \cdot (b_i^{-1} \cdot \tilde{b}_i)} \pmod p$$

and

$$\begin{aligned} \alpha^S &\equiv \alpha^{s_{t+1} + \sum_{i \in \tilde{B} \cap B} \tilde{b}_i \cdot b_i^{-1} \cdot s_i} \equiv \alpha^{s_{t+1} + \sum_{i \in \tilde{B} \cap B} u_i \cdot b_i \cdot \tilde{b}_i \cdot b_i^{-1} + \tilde{b}_i \cdot b_i^{-1} \cdot k_i \cdot E} \\ &\equiv \alpha^{(\sum_{i \in \tilde{B}} u_i \cdot \tilde{b}_i) + k_{t+1} + \sum_{i \in \tilde{B} \cap B} \tilde{b}_i \cdot b_i^{-1} \cdot k_i \cdot E} \equiv y \cdot T \cdot \tilde{R}^E \pmod p. \end{aligned}$$

If Charly can control the broadcast channel, that is he can avoid that the message r_{t+1} sent by u_{t+1} will be received by the co-signers $u_2, \ldots, u_{t+1}$ and he can avoid that message r_t sent by u_t will be received by u_{t+1}, then it's possible to assume that signer u_{t+1} is not aware of the attack.

4.3 Second scheme signature schemes with non-anonymous (t,n) shared signature generation

Li, Hwang and Lee suggested another group signature scheme with non-anonymous (t,n) shared signature generation where no trusted authority is needed [11]. Here every user i acts like a dealer. Each user selects a $t-1$-th degree polynomial, $f_i(x)$, where $f_i(0)$ is his secret and $y_i := \alpha^{f_i(0)}$ is his public key. The group key is computed by $y := \prod_{i \in A} y_i \pmod p$. Each signer $i \in A$ sends the following parameters to each signer $j \in A$:

$$u_{i,j} := g_{i,j} + f_i(ID_j) \pmod q, y_{i,j} := \alpha^{u_{i,j}} \pmod p, z_{i,j} := \alpha^{g_{i,j}} \pmod p,$$

where the $g_{i,j}$ are chosen randomly.

If a group B with $|B| = t$ of signers would like to generate a signature of a message m, then each signer $i \in B$ picks k_i and computes $r_i := \alpha^{k_i} \pmod p$. This r_i is sent to the other signers using a broadcast channel. If all r_i are available, then each signer computes

$$R := \prod_{i \in B} r_i \pmod p \text{ and } E = H(m, R)$$

using a suitable hash function H. Then each signer i computes

$$b_{i,B} := \prod_{e \in B, i \neq e} \frac{0 - ID_e}{ID_i - ID_e} \pmod q$$

$$l_{i,B} := \sum_{j \in A, j \notin B} u_{j,i} \cdot b_{i,B} \pmod q,$$

$$s_i := f_i(0) + l_{i,B} + k_i \cdot E \pmod q.$$

Each signer i sends the values m and s_i to a clerk who can check the validity of the individual signature by

$$\alpha^{s_i} \equiv y_i \cdot \left(\prod_{j \in A, j \notin B} y_{j,i} \right)^{b_i} \cdot r_i^E \pmod p$$

The clerk combines the individual signatures by

$$S := \sum_{i \in B} s_i \pmod{q}.$$

Then (m, B, R, S) is the group signature of the signers in B on message m. This signature can be checked by computing

$$b_{i,B} := \prod_{e \in B, i \neq e} \frac{0 - ID_e}{ID_i - ID_e} \pmod{q}, Z_i := \prod_{j \in A, j \notin B} z_{j,i} \pmod{p},$$

$$T := \prod_{i \in B} Z_i^{b_{i,B}} \pmod{p}, E := H(m, R)$$

and checking, whether the equation

$$\alpha^S \equiv y \cdot T \cdot R^E \pmod{p}$$

holds.

4.4 Security analysis

It's possible to point out that the property is not gained in this scheme in a similar way as before. With $B = \{1, \ldots, t\}$, $\tilde{B} = \{1, \ldots, t-1, t+1\}$ and $A = 1, \ldots, n$ this time signer u_1, say Charly, and $u_t, \ldots, u_n$ must collude. Therefore, in general a stronger attacker model must be assumed here.

We give an brief outline how the attack can be applied here just focusing on victim u_2, say Alice. All variables in

$$l_{2,B} := b_{2,B} \cdot \sum_{j \in A, j \notin B} u_{j,2} \pmod{q}$$

and

$$l_{2,\tilde{B}} := b_{2,\tilde{B}} \cdot \sum_{j \in A, j \notin \tilde{B}} u_{j,2} \pmod{q}$$

are known to the colluders, as $u_{j,2}$ is distributed by signer u_j with $j \in A - (B \cap \tilde{B}) = \{t, \ldots, n\}$ and these signers $u_t, \ldots, u_n$ belong to the colluders. The signature equation of Alice is

$$s_2 := f_2(0) + l_{2,B} + k_2 \cdot E \pmod{q}$$

if Alice signs with group B. By defining $\tilde{s}_2 := s_2 - l_{2,B} + l_{2,\tilde{B}} \pmod{q}$, the signature equation

$$\tilde{s}_2 := f_2(0) + l_{2,\tilde{B}} + k_2 \cdot E \pmod{q}$$

holds, which is Alices individual signature equation for message m if Alice signs with group $\tilde{B}$. Note that Alice has to use $E := h(m, \tilde{R})$ what can be easily prepared by the insider attacker Charly analogous to the attack described above.

5. Heuristic countermeasures

It's possible to countermeasure all described attack in a heuristic way. However, the security of the repaired schemes can't be guaranteed.

The attack in the multisignature schemes and the group signature schemes with (t, n) shared signature generation can be countermeasured if a simultaneous broadcast channel is used (and therefore the assumption is not satisfied) or every co-signer i proves the knowledge of the discrete logarithm of r_i without revealing it (e.g. by a zero-knowledge proof or a signature on a fixed message with key pair (k_i, r_i)). The disadvantage of both countermeasures is the additional amount of computations. A more elegant countermeasure is to prevent the attacker to compute $\tilde{r}$. For example, if in the signature equation of the signature scheme in section 2.1 the expression

$m+r$ is substituted by $h(r, m)$ with a suitable hash function h, then $\tilde{r}$ can't be computed anymore and the attack fails. Then the multisignature scheme based on Schnorr's signature scheme [12] is obtained, as the underlying conventional signature scheme is equivalent to Schnorr's scheme now [12].

In both schemes by Li, Hwang and Lee it seems to be sufficient to compute $E := H(m, R, B)$ instead of $E := H(m, R)$, to use a simultaneous broadcast channel for the distribution of the r_i or to require that all signers prove knowledge of the discrete logarithm of r_i without revealing it, e.g. by a zero-knowledge proof [12].

6. Conclusion

We pointed out some serious flaws in several multiparty signature schemes under reasonable assumptions. Although the attack is not undetectable as the insider attacker has to cause a disruption during a signature generation, the success of such attacks is not acceptable, as the attacks can only be detected afterwards and the knowledge obtained by protocol disruptions must be useless. We further suggested heuristic countermeasures to overcome the weaknesses. As a result, the design of multiparty signature schemes is not easy and requires a great amount of care.

References

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

[2] D.Chaum, "Designated confirmer signatures", LNCS 950, Proc. Eurocrypt'94, Springer Verlag, (1995), pp. 86–91.

[3] D.Chaum, H.van Antwerpen, "Undeniable Signatures", LNCS 435, Proc. Crypto '89, Springer Verlag, (1990), S. 212–216.

[4] D.Chaum, E.van Heijst, "Group signatures", LNCS 547, Proc. Eurocrypt '91, Springer Verlag, (1992), pp. 257–265.

[5] Y.Desmedt, Y.Frankel, "Shared generation of authenticators and signatures", LNCS 576, Proc. Crypto '91, Springer Verlag, (1992), pp. 457–469.

[6] W.Diffie, M.Hellmann, "New directions in cryptography", IEEE Transactions on Information Theory, Vol. IT-22, No. 6, (1976), pp. 644–654.

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

[8] L.Harn, "Group-oriented (t,n) threshold digital signature scheme and digital multisignature", IEE Proc.-Comput. Digit. Tech., Vol. 141, No. 5, September, (1994), pp. 307–313.

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

[10] K.Itakura, K.Nakamura, "A public key cryptosystem suitable for digital multisignatures", NEC Research and Development, Vol. 71, (1983).

[11] C.-M.Li, T.Hwang, N.-Y.Lee, "Threshold multisignature schemes where suspected forgery implies traceability of adversarial shareholders", LNCS 950, Proc. Eurocrypt'94, Springer Verlag, (1995), pp. 194–204.

[12] C.P.Schnorr, "Efficient signature generation by smart cards", Journal of Cryptology, Vol. 4, (1991), pp. 161–174.

[13] M.de Soete, J.-J.Quisquater, K.Vedder, "A signature with shared verification scheme", LNCS 435, Proc. Crypto '89, Springer Verlag, (1990), pp. 253–262.