Cryptanalysis of Threshold-Multisignature Schemes

Lifeng Guo

Institute of Systems Science, Academy of Mathematics and System Sciences,
Chinese Academy of Sciences, Beijing 100080, P.R. China
E-mail address: lfguo@mail.amss.ac.cn

Abstract. In [1], Li et al. proposed a new type of signature scheme, called the (t, n) threshold-mutisignature scheme. The first one needs a mutually trusted share distribution center (SDC) while the second one does not. In this paper, we present a security analysis on their second schemes. We point out that their second threshold-multisignature scheme is vulnerable to universal forgery by an insider attacker under reasonable assumptions. In our attack, $(n - t + 1)$ colluding members can control the group secret key. Therefore, they can generate valid threshold-multisignautre for any message without the help of other members. Furthermore, honest members cannot detect this security flaw in the system, since any t members can generate threshold-multisignatures according to the prescribed protocols.

keywords: threshold-multisignature; secret sharing

1 Introduction

In the well known conventional digital signature concept by Diffie and Hellman [2] one signer is sufficient to sign a message m know to him and one verifier is sufficient to check the validity of any given signature. In other concepts, we may call them multiparty signature concepts, it is required that only several signers should be able to sign or that only several verifiers should be able to verify a signature. A lot of schemes for slightly different concepts have been suggested, for an overview see [3].

In a multisignature scheme [4] some signers can generate a signature on a message together, one verifier is sufficient to verifier a given signature and the verifier needs the identity of the signers for verification. In particular, the signers are not anonymous.

The concepts of group oriented cryptography and threshold cryptosystems were developed by Frankel and Desmedt [5,6,7]. In a threshold cryptosystem, the private key is not held by a individual. Instead, the key is shared among a group such that a certain minimum number of them can work together to use the key without compromising its value. Any subset of the group with fewer than the threshold number of members will have no information about the key. This distribution of the key provides protection against dishonest group members and accidental disclosure of the key.

Finally, in a threshold multisignature scheme [1], t out of a group of n signers are able to generate a signature on a together, one verifier is sufficient to verify a given signature and the verifier needs the identity of the signers for verification. In particular, the signers are not anonymous.

In [1], Li et al. proposed two (t, n) threshold-multisignature : One of their schemes needs the assistance of a mutually trusted SDC, while the other does not. In this paper, we demonstrate an attack to show that in their second threshold-multisignature scheme, $(n-t+1)$ colluding members can control the group secret key. Therefore, they can generate valid threshold signature for any message without the help of other members. Furthermore, honest members cannot detect any security flaw in the system, since under the assumption that the clerk is also corrupt, any t members can generate threshold-multisignatures according to the prescribed protocols.

The rest of this paper is organized as follows. Section 2 introduces Li et al.' second threshold-multisignature schemes are based. Section 3 presents our security analysis on their schemes. The conclusion is drawn in section 4.

2. Review of Li et al.'s Second Threshold-Multisignature Schemes

In [1], Li et al. proposed a new type of signature scheme, called the (t, n) threshold-mutisignature scheme. The first one needs a mutually trusted SDC while the second one does not. In this section we only review their second scheme which does not need a trusted center.

Their second scheme consists of three parts: group public key and secret shares generation phase, partial signature generation and verification phase, and group signature generation and verification phase.

Part 1: Group Public Key and Secret Shares Generation Phase

Let $A(|A| = n)$ be the set of all shareholders, B be any subset in A of size t ($|B| = t$). The public parameters, (H, p, q, α), should be agreed by all shareholders in advance.

- a collision free one-way hash function H.
- p= a large prime modulus, where $2^{511} < p < 2^{512}$.
- q= a prime divisor of p-1, where $2^{159} < p < 2^{160}$.
- a positive integer $\alpha = h^{p-1/q} \mod p$, where $1 \leq h \leq p-1$, and g is a generator with order q in $GF(p)$.

Each shareholders $i, i \in A$, randomly selects a $(t-1)$-th degree polynomial, $f_i(x)$, and an integer x_i associated with each shareholder i, where $x_i \in [1, q-1]$. Then he computes a corresponding public key y_i, as

$$y_i = \alpha^{f_i(0) \mod q} \mod p. \tag{1}$$

$\{x_i, y_i\}$ are the public keys of the shareholders $i, i \in A$, and the polynomial $f_i(x)$ is his secret parameter. The group public key y can be determined by all shareholders as

$$y = \prod_{i \in A} y_i \mod p. (= \alpha^{\sum_{i \in A} f_i(0) \mod q} \mod p). \tag{2}$$

Since there is no trusted SDC, each shareholder i must act as a trusted SDC to generate and distribute following values to the shareholder j, $j \in A$, $j \neq i$, as :

$$u_{ij} = g_{ij} + f_i(x_j) \mod q, \tag{3}$$

$$y_{ij} = \alpha^{u_{ij}} \mod p, (= \alpha^{g_{ij} + f_i(x_j) \mod q} \mod p,) \tag{4}$$

$$z_{ij} = \alpha^{g_{ij}} \mod p. \tag{5}$$

Where x_j is the public key of shareholder j, and g_{ij} is random integer with $0 < g_{ij} < q$. The value of u_{ij} is the secret share generated by shareholder i for shareholder j, and both y_{ij} and z_{ij} are shareholder j's public values.

Part 2: Partial Signature Generation and Verification Phase

Each shareholder $i, i \in B$, randomly selects an integer $k_i, k_i \in [1, q-1]$, and compute a public value r_i, as

$$r_i = \alpha^{k_i} \mod p. \tag{6}$$

Then each shareholder $i, i \in B$, makes r_i publicly available through a broadcast channel. Once all $r_i, i \in B$ are available, each shareholder i in B computes the product R and a hash value E as

$$R = \prod_{i \in B} r_i \mod p, (= \alpha^{\sum_{i \in B} k_i \mod q} \mod p) \tag{7}$$

$$E = H(m, R) \mod q \tag{8}$$

Then shareholder i uses his secret keys $f_i(0)$,k_i and u_{ij} $j \in A$. j is not belong to B. To calculate the partial signature s_i as

$$s_i = f_i(0) + \sum_{j \in A, j \notin B} (u_{ji} \cdot \prod_{e \in B, e \neq i} \frac{0 - x_e}{x_i - x_e}) + k_i \cdot E \mod q. \tag{9}$$

Each shareholders i in B sends the values $\{m, s_i\}$ to the designated combiner, DC. The DC firstly computes the values of R and E from the broadcast channel, and then he uses shareholder i's public key x_i, y_i and y_{ji}, for $j \in A, j \in B$ to verify the validity of the partial signature as

$$\alpha^{s_i} \equiv (y_i \cdot (\prod_{j \in A, j \notin B} y_{ji})^{\prod_{e \in B, e \neq i} \frac{0-xe}{x_i - xe}) \mod q}) \cdot r_i^E \mod p. \tag{10}$$

If the above equation holds, then the partial signature $\{m, r_i, s_i\}$ is valid.

Part 3: Group Signature Generation and Verification Phase

Once all these t partial signature are verified by the DC, the DC can generate the group signature for the message m as $\{m, B, R, S\}$, where

$$S = \sum_{i \in B} s_i \mod q \tag{11}$$

To verify the validity of the group signature $\{m, B, R, S\}$, the verifier has to compute the verification value T and the hash value E as

$$T = \prod_{i \in B} ((\prod_{j \in A, j \notin B} z_{ji})^{\prod_{e \in B, e \neq i} \frac{0-xe}{x_i - xe}}) \mod p \tag{12}$$

$$E = H(m, R) \mod q \tag{13}$$

Then, the verifier uses the group public key y to check

$$\alpha^S \equiv y \cdot T \cdot R^E \mod p \tag{14}$$

If the above equation holds, the group signature $\{m, B, R, S\}$ is valid.

3. An Attack on Li Chuan Ming et al.'s second threshold-multisingature scheme

First the possible attacker models are described. In the multisignature environment we can distinguish between an insider-,an outsider- and designated combiner (DC) attack . The outsider does not play an active role in a protocol, while the insider does. The DC attack can be active (he does not follow the protocol) or passive (he just reveals some parameters). Furthermore, one strength of the insider attack depends on the number of the attack.

In this section, we show an attack on Li Chuan Ming et al.'s second threshold-multisingature schemes in which n group members generate the group public key in a distributed way. We present the details about how $(n-t+1)$ colluding members can cheat other $(t-1)$ honest members by controlling the group secret key. In our attack, we make the following two assumptions:

- **Assumption 1.** Except $(t-1)$ honest members in the system, all other $(n-t+1)$ members are dishonest and collude together.

-**Assumption 2.** The designated combiner DC is also corrupted by dishonest members. Here, the DC is passive. He just reveals some parameters.

For simplicity, but without loss of generality, we assume that the first $(t-1)$ members , i.e., $U_1, U_2, \cdots, U_n$ are honest members and all other $(n-t+1)$ members are dishonest. We further assume that as the head of dishonest members, U_n controls the group secret key, while other colluding members tolerate his faults in the group public key generation. When needed, these conspirators send their secret keys to U_n.

The details of our attack are given as follows. The whole procedure consists of three steps: member U_n controlling the group private key, member U_n distributing secret shares, and dishonest members generating valid individual signatures.

Step 1. Member U_n controlling the group private key

In the group public key generation of Li et al.'s second scheme, it is not required that all public keys y_i's should be published simultaneously. Thus, member U_n can be the last one to publish his public key y_n. By choosing a random number x as the group secret key, he first sets the group public key by $y = \alpha^x \mod p$. Then, when all other $y_i's$ $(i \in \{1, 2, \cdots, n-1\})$ are published, he computes and publishes his public key y_n as follows:

$$y_n = y \cdot \prod_{i=1}^{n-1} y_i^{-1} \mod p \tag{15}$$

Therefore, all members in group A will take y as the group public key, since the following equation holds:

$$y = y_n \cdot \prod_{i=1}^{n-1} y_i = \alpha^x \mod p \tag{16}$$

Hence, member U_n has controlled the group private key x corresponding to y. Of course, member U_n does not know his private key s_n corresponding to y_n since he cannot find discrete logarithm of y_n to the base α.

Step 2. Member U_n distributing secret shares

The difficulty is how member U_n can distribute his secret key s_n to other members even though he does not know the value of s_n. For this sake, U_n does as follows.

1. Firstly, U_n assumes that he has chosen a $(t-1)$-degree polynomial $f_n(X)$ such that $f_n(0) = s_n$, where s_n is the unknown but fixed number satisfying $y_n = \alpha^{f_n(0)} \mod p$. At the same time he selects $(t-1)$ random numbers $b_i \in [0, q-1]$, and sets $f_n(x_i) = b_i$, for each $i \in \{1, 2, \cdots, t-1\}$.

2. Secondly, he chooses random numbers $g_{ni} \in Z_q$ and he sends $u_{ni} = g_{ni} + b_i$, $y_{ni} = \alpha^{g_{ni}+b_i} \mod p$, $z_{ni} = \alpha^{g_{ni}} \mod p$ privately to each member U_i $i \in \{1, 2, \cdots, t-1\}$. However, U_n cannot send u_{nj}, y_{nj}, z_{nj} to each of his conspirators since he does not know the value of $f_n(x_j)$ for each $j \in \{t, t+1, \cdots, n-1\}$. But, as U_n's conspirators, each member U_j tolerates this fault.

3. Thirdly, U_n has to publish the related public key u_{nj}, y_{nj}, z_{nj}, for all $j \in \{1, 2, \cdots, n-1\}$. Of course, for $i \in \{1, 2, \cdots, t-1\}$, U_n computes $u_{ni} = g_{ni} + b_i$, $y_{ni} = \alpha^{g_{ni}+b_i} \mod p$, $z_{ni} = \alpha^{g_{ni}} \mod p$. The remainder is to compute $u_{nj} = g_{nj} + f_n(x_j)$, $y_{nj} = \alpha^{g_{nj}+f_n(x_j)} \mod p$, $z_{nj} = \alpha^{g_{nj}} \mod p$. This key step is that he compute $\alpha^{f_n(x_j)}$. We now explain that U_n can also carry out $\alpha^{f_n(x_j)}$ for each $j \in \{t, t+1, \cdots, n\}$ even though he does not know the value of $f_n(x_j)$. According to the Lagrange interpolating formula, the following equation holds.

$$f_n(0) = \sum_{i=1}^{t-1} f_n(x_i) \cdot \prod_{k \in B_j, k \neq i} \frac{-x_k}{x_i - x_k} + f_n(x_j) \cdot \prod_{k \in B_j, k \neq j} \frac{-x_k}{x_j - x_k} \mod p \tag{17}$$

Here we write $C_{B_ji} = \prod_{k\in B_j, k\neq i} \frac{-x_k}{x_i - x_k}$, $C_{B_jj} = \prod_{k\in B_j, k\neq j} \frac{-x_k}{x_j - x_k}$, where $B_j = \{1, 2, \cdots, t-1, j\}$, $j \in \{t, t+1, \cdots, n-1\}$.

$$f_n(0) = \sum_{i=1}^{t-1} f_n(x_i) \cdot C_{B_ji} + f_n(x_j) \cdot C_{B_jj} \mod p \tag{18}$$

We have

$$\alpha^{f_n(0)} = \prod_{i=1}^{t-1} \alpha^{f_n(x_i)\cdot C_{B_ji}} \alpha^{f_n(x_j)\cdot C_{B_jj}} \mod p, \quad f_n(x_i) = b_i \tag{19}$$

$$\alpha^{f_n(x_j)} = (\alpha^{f_n(0)} \cdot \prod_{i=1}^{t-1} \alpha^{-c_{B_ji}\cdot b_i})^{C_{B_jj}^{-1}} \quad \forall j \in \{t, t+1, \cdots, n-1\} \tag{20}$$

After all $\alpha^{f_n(x_j)}$ $(j \in \{1, 2, \cdots, n-1\})$ are computed, U_n publishes $u_{ni} = g_{ni} + f_n(x_i)$, $y_{ni} = \alpha^{g_{ni}+f_n(x_i)} \mod p$, $z_{ni} = \alpha^{g_{ni}} \mod p$ to the honest members, where $i \in \{1, 2, \cdots, t-1\}$. Though $\alpha^{f_n(x_j)}$ $(j \in \{t, t+1, \cdots, n-1\})$ are computed, U_n has no way to know $f_n(x_j)$ unless he solve discrete logarithm. But members U_j $(j \in \{t, t+1, \cdots, n-1\})$ are his conspirators , they can tolerate that. U_n can randomly send u_{nj} to their conspirators, but correctly sends $y_{nj} = \alpha^{g_{ni}+f_n(x_j)} \mod p$, $z_{nj} = \alpha^{g_{nj}} \mod p$. Any member can verify that all y_{nj}, z_{nj}'s are consistent since the following equation:

$$\alpha^{f_n(0)} = \prod_{j\in B} (y_{nj} \cdot z_{nj}^{-1})^{C_{B_j}}, \quad \forall \ B \subseteq \{1, 2, \cdots, n-1\} \quad and \quad |\ B\ | = t \tag{21}$$

When U_n and all other members published all $u_{ij}, y_{ij}, z_{ij}, i, j \in \{1, 2, \cdots, n-1\}$ and $i \neq j$, the system is set up. After that, U_n can use the known group secret key x to forge valid threshold-multisignature on any message m. That is, he first chooses a random $k \in [0, q-1]$ and computes $R = \alpha^k \mod p$ and $E = H(m, R)$. Then, he gets S from equations:

$$\alpha^S \equiv y \cdot T \cdot R^E \mod p \tag{22}$$

and

$$T = \prod_{i\in B} ((\prod_{j\in A, j\notin B} z_{ji})^{\prod_{e\in B, e\neq i} \frac{0-xe}{x_i - xe}}) \mod p. \tag{23}$$

Where $z_{ji} = \alpha^{g_{ji}} \mod p$.

We have

$$S \equiv x + \sum_{i\in B} \sum_{j\in A, j\notin B} g_{ji} \prod_{e\in B, e\neq i} \frac{0 - x_e}{x_i - x_e} + kE. \tag{24}$$

In this equation, U_n knows x, k. Moreover, $j \notin B$, these numbers g_{ji}, are his conspirators' random numbers, so he can obtain g_{ji} from his conspirators. It is easy to know that such forged pair (R, S) is a valid threshold-multisignature on message m, since it satisfies Equation (14). Furthermore, as we have mentioned, under the help of the corrupted clerk, dishonest members can also generate valid individual signature. So they can cheat honest members that the system is normal and secure.

Step 3. Dishonest members generating their individual signatures

Now, we assume t members of a subset B want to sign a message m. According to the individual signature generation (10), U_n cannot generate valid individual signature since he does not know the value $f_n(0)$. In addition, if $n \notin B$, any malicious member $U_j (t \leq j \leq n-1)$ cannot generate valid individual signature since he does not know the value of $f_n(x_j)$. However, note that if under our assumption 2, we will see that the corrupted DC can help dishonest members to generate valid individual signatures in two cases: (1) $n \in B$ and (2) $B = \{1, 2, \cdots, t-1, j\}$ where $j \in \{t, t+1, \cdots, n-1\}$. In the following,

we only describe how dishonest members can generate their valid individual signatures in case 1. As we mentioned above, in this case $n \in B$, any other (honest or dishonest) member can generate his individual signature normally. Therefore, we now focus on how member U_n can generate his individual signature.

In Li et al.'s scheme, it is also not required that all r_i's should be published simultaneously, thereby member U_n can be the last one to publish r_n. That is, U_n first selects a random number $k \in [0, q-1]$, and sets

$$R = g^k \mod p. \tag{25}$$

When all other r_i's have been broadcast, he computes and broadcasts the following r_n:

$$r_n = R \cdot \prod_{i \in B/\{n\}} r_i^{-1} \mod p. \tag{26}$$

Consequently, each member U_j $(j \in B/\{n\})$ computes R by $R = r_n \cdot \prod_{i \in B/\{n\}} r_i^{-1} \mod p$. Then, by using Equation (9), each U_j generates and sends his individual signature (r_i, s_i) to the DC. The corrupted clerk reveals the values of all (r_i, s_i)'s to U_n. To get his individual signature on the message m, U_n first solves S from the following equation

$$S \equiv x + \sum_{i \in B} \sum_{j \in A, j \notin B} g_{ji} \prod_{e \in B, e \neq i} \frac{0 - x_e}{x_i - x_e} + kE. \tag{27}$$

Next, he computes his individual signature (r_n, s_n) as follows.

$$r_n = R \cdot \prod_{i \in B/\{n\}} r_i^{-1} \mod p, \quad \text{and} \quad s_n = S - \sum_{i \in B/\{n\}} s_i \mod q. \tag{28}$$

Finally, U_n sends (r_n, s_n) to the DC so that the clerk can generate the threshold-multisignature (R, S) for the message m. If necessary, theDC publishes all individual signatures (r_i, s_i) $(i \in B)$ as the evidences that all members in B indeed generated valid individual signatures for the message m. After all (r_i, s_i)'s have been broadcast, each member in B can verify the validity of each pair (r_i, s_i) by using Equation (10). Up to this point, U_n generated his individual signature pair (r_n, s_n).

The following theorem proves that the above (R, S) is valid threshold-multisignature for the message m, and that (r_n, s_n) is U_n's valid individual signature for the message m.

Theorem 1. The above procedure that U_n generates his individual signature is successful. That is,

(1) The pair (R, S) generated by Equation (23),(25) is a valid threshold-multisignautre for the message m, and

(2) The pair (r_n, s_n) generated by Equation (26) is U_n's valid individual signature for the same message m.

Proof: It is obvious that the pair (R, S) generated by Equation (25) satisfies Equation (14). We now prove (2): we need to show that the pair (r_n, s_n) generated by Equation (26) satisfies Equation (10). This is justified by the following equalities.

$$\begin{aligned}
\alpha^{s_n} &= \alpha^{S - \sum_{i \in B/\{n\}} s_i} \mod p \\
&= y \cdot T \cdot R^E \Big(\prod_{i \in B/\{n\}} \alpha^{-s_i} \Big) \mod p \\
&= y \cdot T \cdot R^E \prod_{i \in B/\{n\}} \Big(y_i \Big(\prod_{j \in A, j \notin B} y_{ji}^{\prod_{e \in B, e \neq i} \frac{0 - x_e}{x_i - x_e} \bmod q} r_i^E \Big) \Big)^{-1} \mod p \\
&= \Big(R \cdot \prod_{i \in B/\{n\}} r_i^{-1} \Big)^E \cdot y \cdot \prod_{i \in B/\{n\}} y_i^{-1} \cdot T \cdot \Big(\prod_{i \in B/\{n\}} \prod_{j \in A, j \notin B} y_{ji}^{-1} \Big)^{\prod_{e \in B, e \neq i} \frac{0 - x_e}{x_i - x_e} \bmod q} \mod p
\end{aligned}$$

$$
\begin{aligned}
&= r_n^E \cdot y \cdot \prod_{i\in B/\{n\}} y_i^{-1} \cdot (\prod_{i\in B}(\prod_{j\in A, j\notin B} z_{ji})^{\prod_{e\in B,e\neq i}\frac{0-xe}{x_i-xe}}) \cdot (\prod_{i\in B/\{n\}}(\prod_{j\in A,j\notin B} y_{ji}^{-1})^{\prod_{e\in B,e\neq i}\frac{0-xe}{x_i-xe} mod q}) \ \ mod\ \ p \\
&= r_n^E \cdot y \cdot \prod_{i\in B/\{n\}} y_i^{-1} \cdot (\prod_{j\in A,j\notin B} y_{jn}^{\prod_{e\in B,e\neq i}\frac{0-xe}{xn-xe}}) \cdot (\prod_{i\in B}(\prod_{j\in A,j\notin B} z_{ji}\cdot y_{ji}^{-1})^{\prod_{e\in B,e\neq i}\frac{0-xe}{x_i-xe} mod q}) \ \ mod\ \ p \\
&= r_n^E \cdot y \cdot \prod_{i\in B/\{n\}} y_i^{-1} \cdot (\prod_{j\in A,j\notin B} y_{jn}^{\prod_{e\in B,e\neq n}\frac{0-xe}{xn-xe}}) \cdot (\prod_{i\in B}(\prod_{j\in A,j\notin B} \alpha^{-f_j(x_i)})^{\prod_{e\in B,e\neq i}\frac{0-xe}{x_i-xe} mod q}) \ \ mod\ \ p \\
&= r_n^E \cdot y \cdot \prod_{i\in B/\{n\}} y_i^{-1} \cdot (\prod_{j\in A,j\notin B} y_{jn}^{\prod_{e\in B,e\neq i}\frac{0-xe}{xn-xe}}) \cdot \prod_{j\in A,j\notin B} \alpha^{\sum_{i\in B}-f_j(x_i)\prod_{e\in B,e\neq i}\frac{0-xe}{x_i-xe}} \ \ mod\ \ p \\
&= r_n^E \cdot y \cdot \prod_{i\in B/\{n\}} y_i^{-1} \cdot (\prod_{j\in A,j\notin B} y_{jn}^{\prod_{e\in B,e\neq i}\frac{0-xe}{xn-xe}}) \cdot \prod_{j\in A,j\notin B} \alpha^{-f_j(0)} \ \ mod\ \ p \\
&= r_n^E \cdot y \cdot \prod_{i\in B/\{n\}} y_i^{-1} \cdot (\prod_{j\in A,j\notin B} y_{jn}^{\prod_{e\in B,e\neq i}\frac{0-xe}{xn-xe}}) \cdot \prod_{j\in A,j\notin B} y_j^{-1} \ \ mod\ \ p \\
&= r_n^E \cdot y_n \cdot (\prod_{j\in A,j\notin B} y_{jn})^{\prod_{e\in B,e\neq i}\frac{0-xe}{xn-xe} \ \ mod\ \ q} \ \ mod\ \ p
\end{aligned}
$$

4 Conclusion

In this paper, we presented a security analysis to Li et al.'s second threshold-multisignature scheme without a mutually thrusted SDC, we demonstrated an attack that allows $(n-t+1)$ colluding members to control the group secret key and then generate valid threshold-multisignature for any message. However, honest members cannot detect this security flaw in the system since t members can generate threshold-multisignatures according to the specified protocols. Consequently, colluding dishonest members can cheat honest members successfully.

References

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

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

[3] Y.Desmedt, Threshold cryptosystems, ETT, 5 (4), August, (1994), pp. 449-457.

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

[5] Y.Desmedt, Society and group oriented cryptography, Advances in Cryptology-Crypto'87 proceedings, Springer-Verlag, 1988, pp. 120-127.

[6] Y.Desmedt and Y.Frankel, Shared generation of authenticators and signatures,Advances in Cryptology-Crypto'91 proceedings, Springer-Verlag, 1992, pp. 457-469. Advances in

[7] Y.Frankel and Y.Desmedt, Parallel reliable threshold multisignatures, Tech. Report TR-92-04-02, Dep. of EE & CS, Univ.of Wisconsin-Milwaukee, April 1992.