On the Security of the Li-Hwang-Lee-Tsai Threshold Group Signature Scheme

Guilin Wang
Laboratories for Information Technology
21 Heng Mui Keng Terrace, Singapore l19613
givang@lit.a-star.edu.sg

Abstract. A (t, n) threshold group signature scheme is a generalization of group signature, in which only t or more members from a given group with n members can represent the group to generate signatures anonymously and the identities of signers of a signature can be revealed in case of dispute later In this paper, we first present a definition of threshold group signatures, and propose several requirements to evaluate whether a threshold group signature scheme is secure and efficient. Then we investigate the security and efficiency of a threshold group signature scheme proposed by Li, Hwang, Lee and Tsai, and point out eight weaknesses in their scheme, The most serious weakness is that there is a framing attack on their scheme. In this framing attack, once the group private key is controlled (n-t+1) colluding group members can forge a valid threshold group signature on any given message, which looks as if it was signed by (t-1) honest group members and one cheating member. At the same time, all these (t- 1) honest members cannot detect this cheating behavior, because they can use the system to generate group signatures normally
Keywords: digital signatures, group signatures, threshold group sig-natures) threshold-multisignatures.

1. Introduction

As a relatively new concept, group signatures are introduced and realized by Chaum and van Heyst in [10]. In a group signature scheme, each member of a given group is able to sign messages anonymously on behalf of the group. However, in case of later dispute, a group signature can be opened by the group manager and then the actual identity of signer can be revealed. From verifiers' point of view, they only need to know a single group public key to verify group signatures. On the other hand, from the point of view of signing group, the group can conceal its internal organizational structures, but still can trace the signer's identity if necessary. In virtue of these advantages, it is believed that group signatures have many potentially practical applications, such as authenticating price

lists, press releases, digital contract, e-voting, e-bidding and e-cash etc.

Inspired by the pioneering work of Chaum and van Heyst, a number of improvements and new group signature schemes have been proposed [11,12,31,5-7,3,2,1] In [11], Chen and Pedersen constructed the first scheme which allows new members join the group dynamically. They also pointed out the idea of sharing group public key to realize a t out of n threshold scheme, but did not provide concrete schemes. Camenisch presented an efficient group signature scheme with ability to add (or remove) group members after the initialization and then extended his scheme to a generalized group signature such that authorized subset of group members can sign messages on the group's behalf collectively[5]. As example of his generalized group signature scheme, Camenisch presented the first threshold group signature scheme. But in [5], both lengths of the group public key and a group signature are proportional to the group size. In [6], Camenisch and Stadler proposed the first group signature scheme whose group public key and signatures have length independent of the group size. Thus their scheme can be used for large groups. Camenisch and Michels [8l aimed to design generic group signature schemes with *separability*, i.e, the group members can choose their keys independently of each other.

Ateniese et. al focused on some bothersome issues that stand in the way of real world applications and developments of group signatures, such as coalition attacks and member deletion [3, 2]. Based on their observation of an unsuitable number theoretic assumption in [6], Ateniese and Tsudik [3] presented some quasi-attacks on the basic scheme of [6] and then proposed some simple ways to prevent them. In [1], Ateniese et al. proposed a provably secure coalition-resistant group signature scheme. Kim, Lim and Lee [23] proposed a new group signature scheme with a member deletion procedure. Based on the notion of dynamic accumulators, Camenisch and Lysyanskaya proposed a new efficient method for the member deletion problem in group signature schemes. At the same time, they pointed out that the scheme proposed by Kim et al. in [23] is broken, i.e., deleted group members can still prove membership.

In [24], Langford pointed out attacks on the group public key generation protocols in several threshold cryptosystems [19, 17, 25, 30], i.e, a group member can control the group private key. Michels and Horster [27] discovered some attacks against several multiparty signature schemes [17, 18, 25, 20]. Their attacks are in common that the attacker is an *insider*, i,e., a dishonest group member, and the protocol will be disrupted. Joye et. al [22, 21] showed that several newly designed group signature schemes are *universally forgeable*, that is, anyone (not necessarily a group member) is able to generate a valid group signature on arbitrary message, which cannot be traced by the group manager.

By combining the idea of the (t, n) threshold signatures [13, 14, 17, 16] with the multisignatures [28, 4, 29, 15], Li, Hwang, Lee and Tsai [26] proposed a new type of signature called the (t, n) *threshold-multisignatures* with three properties:(1) Threshold characteristic only t or more members of a given group cangenerate valid group signatures;(2) *Anonymity*, the group members generate group signatures anonymously, and they use pseudonyms as their identities in the public directory;(3) Traceability the identities of signers can be revealed in exceptional cases, such as legal dispute. At the same, time, they presented two concrete such schemes[26], one needs a trusted share distribution center(SDC)[1] while the other does not. Furthermore, they extended their proposed schemes to realize the generalized-multisignatures such that the group signatures can only be generated by some specified subsets of group members rather than by any subset of t members.

We notice that in a multisignature scheme the identities of signers are often public and the public keys of signers are needed to verify a signature. At the same time, anonymity and traceability are two essential properties of a group signature scheme [10]. So, we believe that it is more accurate to call the(t, n) threshold - multisignature schemes in [25, 26] as (t, n) *threshold group signature schemes.*

In this paper, we first present a definition to (t, n) threshold group signature schemes because such definition is not given previously. Then, we list several requirements to evaluate whether a threshold group signature scheme is secure and efficient. After that, we investigate the security and efficiency of the second scheme proposed by Li, Hwang, Lee and Tsai in [26]. For convenience, we will refer hereafter this scheme as LHLT scheme. According to these evaluation criteria, we point out eight weaknesses in LHLT scheme. The most serious weakness is that there is a *framing attack*[2]. The reason is that we find Langford's attack can also be applied to LHLT scheme. Based on this weakness in the group public key generation protocol, we present the detailed procedure of this framing attack on LHLT scheme by demonstrating how ($n-t+1$) colluding group members can forge a valid threshold group signature on any given message, which looks as if it was signed by (t - 1) honest group members and one cheating member.

The rest of this paper is organized as follows Section 2 proposes a definition of (t,, n) threshold group signature schemes, and addresses the security and efficiency of these schemes. Section 3 reviews LHLT scheme briefly, and section 4 points out some weaknesses of it. After that, section 5 demonstrates how($n-t+1$) colluding group members can forge a valid threshold group signature on any given message to frame (t-1) honest group members.

---

1) A SDC can also be called as a group manager, authority or dealer

2) In a framing attack, one or several group members have to be responsible for a signature generated by several other group members and/ or non group members[11].

Section 6 gives an example to explain the disadvantage of this framing attack and remarks to compare our framing attack with Michels and Horster's attack [27].

2. Definition

Based on the formal definitions of group signatures given by [12, 5-8, 3, 2, 1] and our understanding to threshold group signatures, we present the following formal definition

Definition 1. A(t, n) threshold group signature scheme *is a digital signature scheme comprised of the following six procedures*.

- *SETUP: A protocol among group managers for setting system parameters and generating the initial group public key and group private key.*
- *JOIN : A protocol between group managers and a user that results in th user becoming a new group member.*
- *SIGN : A protocol among t or more group members for producing a group signature of a given message.*
- *VERIFY : An algorithm for establishing the validity of a group signature when the group public key and a signed message are given.*
- *OPEN : A protocol among group managers that reveals the actual identities of the t signers when a signed message and the group public key ar given*
- *QUIT : A Protocol between a group member and group managers for removing the group member form the system.*

A secure threshold group signature scheme must satisfy the following properties:

1. **Correctness**: All signatures on any message generated by any honest authorized subset of group members using SIGN will get accepted by VERIFY.
2. **Unforgeability**: Only group members are able to generate valid partial signatures for given messages.
3. **Threshold characteristic**: Only t or more group members are able to generate valid threshold group signatures for given messages.
4. **Anonymity**: Given a threshold group signature, identifying the real signers is computationally hard for everyone but group managers.
5. **Unlinkability** : Deciding whether two different signatures were generated by the same subset of group members is computationally hard.
6. **Exculpability**: Any subset of group members or group managers cannot sign a message on behalf of another subset[3], i.e, without the existence of framing attacks.

3) But this property does not preclude group managers from creating nonexistent members and then generating valid group signatures.

7. **Traceability**: In case of dispute, a group signature can be opened and the real identities of signers can be revealed; moreover, the subset of signers cannot prevent the opening of a valid group signature.
8. **Coalition-resistance**: A colluding subset of group members cannot generate a valid group signature such that it cannot be traced.

The efficiency of a threshold group signature scheme is typically based on the following parameters:

- Whether the size of the group public key is independent of the size of the group
- Whether the size of a group signature is independent of the size of the group.
- The computational complexity and communications cost of SIGN, VERIFY and OPEN.
- The efficiency of SETUP, JOIN and QUIT.

3. Review of LHLT Scheme

In LHLT (t, n) threshold group signature scheme [26], it is assumed that all communication channels among group members are secure and reliable. The whole scheme consists of four stages: system initialization, group public key and secret shares generation, partial signature generation and verification, and group signature generation and verification.

Stage 1. system Initialization

Some or all members collectively agreed on the public system parameters {p, q, α, H}, where:

- p:a prime modulus such that $2^{511} < p < 2^{512}$;
- q: a prime such that $q \mid (p-1)$ and $2^{159} < q < 2^{160}$;
- α: a random generator of order q in GF(p);
- H: a collision free one-way hash function.

Stage 2. Group Public Key and Secret Shares Generation

Each of member i in group A={1, 2,…, n} randomly selects a polynomial $f_i(x)$, whose degree is no more than (t-1), and a number $x_i \in_R$ {1, 2,…, q-1}, denoted his pseudonym, then he computes y_i as follows:

$$y_i = \alpha^{f_i(0)} \bmod p.$$

(x_i, y_i) are the public key of member i, $i \in A$, and the polynomial $f_i(x)$ (especially $f_i(0)$) is kept secretly. When all members have released (x_i, y_i) through a broadcast channel, the group public key y can be determined as:

$$y = \prod_{i \in A} y_i \bmod p \ \left(= \alpha^{\sum_{i \in A} f_i(0) \bmod q} \bmod p \right). \qquad (1)$$

Then, member i generates following values to member $j (j \in A, j \neq i)$ as:

$$\begin{aligned} u_{ij} &= g_{ij} + f_i(x_j) \bmod q, \quad \text{where } g_{ij} \in_R \{1, 2, \cdots, q-1\}; \\ y_{ij} &= \alpha^{u_{ij}} \bmod p \ \left(= \alpha^{g_{ij} + f_i(x_j) \bmod q} \bmod p \right); \\ z_{ij} &= \alpha^{g_{ij}} \bmod p. \end{aligned} \qquad (2)$$

u_{ij} is sent privately to member j as his secret shares, but y_{ij} and z_{ij} are published as public information[4].

Stage 3. Partial Signature Generation and Verification

When t members of group A want to generate a signature for message m, each member i, $i \in B$ (B $\subset$A and $|B| = t$) sects a random number $k_i \in_R [1, q-1]$, computes and broadcasts a public value r_i as:

$$r_i = \alpha^{k_i} \bmod p.$$

Once all $r_j (j \in B)$ are available, each member i computes values R and E, and then his partial signature s_i as follows:

$$\begin{aligned} R &= \textstyle\prod_{j \in B} r_j \bmod p \ \left(= \alpha^{\sum_{j \in B} k_j \bmod q} \bmod p \right), \\ E &= H(m, R), \\ s_i &= f_i(0) + \textstyle\sum_{j \in A \setminus B} (u_{ji} \cdot C_{Bi}) + k_i \cdot E \bmod q. \end{aligned} \qquad (3)$$

Where, C_{Bi} is the Lagrange interpolating coefficient given by

$$C_{Bi} = \prod_{j \in B \setminus \{i\}} \frac{x_j}{x_j - x_i} \bmod q. \qquad (4)$$

---

4) Member j can use y_{ij} and z_{ij} to check whether he received correct secret shares from member i For detase consult[26].

Then, each member i(i ∈ B) sends his partial signature (m, i, r_i, s_i) to the designated combiner DC (any member in group A or the verifier of a signature can play this role). After computed the values R and E displayed by equation (3), DC uses public information (x_i, , y_i) and $y_{ji}(j \in A \ RSLANT B)$ to verify the validity of (m , $i, r_i s_i$):

$$\alpha^{s_i} \stackrel{?}{\equiv} y_i \cdot \Big(\prod_{j \in A \setminus B} y_{ji} \Big)^{C_{Bi}} \cdot r_i^E \bmod p, \quad \forall i \in B. \qquad (5)$$

Stage 4. Group Signature Generation and Verification

If all partial signatures(m , $i, r_i s_i$), i∈B, are valid, then DC produces the group signatue (m, B, R, S) by the following two equations:

$$R = \prod_{i \in B} r_i \bmod p, \qquad S = \sum_{i \in B} s_i \bmod q. \qquad (6)$$

When a verifier want to verify the validity of a group signature(m, B, R,S), he first computes values E and T as follows:

$$E = H(m, R); \\ T = \prod_{i \in B} \Big(\big(\prod_{j \in A \setminus B} z_{ji} \big)^{C_{Bi}} \Big) \bmod p. \qquad (7)$$

Then, the verifier uses the group public key y to check whether the following equality holds

$$\alpha^S \stackrel{?}{\equiv} y \cdot T \cdot R^E \bmod p. \qquad (8)$$

If yes, he accepts (m, B, R, S) as a valid group signature.

Li et al did not provide the proof to the correctness of this scheme, so we give the following theorem to guarantee the correctness of LHLT scheme[5)]

5) Theorem 1,2 and 4 in[26] do not express the correctness of the three schemes but repeat the definitions of valid group signatures.

Theorem 1. *If all members $i \in B$ and DC are honest, then group signature(m,B,R,S) generated by them is valid, i. e., it satisfies the equation(8).*

Proof. First, from the definitions of S and s_i we have

$$S = \sum_{i \in B} s_i = \sum_{i \in B} f_i(0) + \sum_{i \in B} \sum_{j \in A \setminus B} (C_{Bi} \cdot u_{ji}) + \sum_{i \in B} (k_i \cdot E) \bmod q.$$

If we replace u_{ji} in the above equation by $f_j(x_i)$ and g_{ji} according to the first equation of (2) and sum them separately, then we can get the following equation:

$$S = \sum_{i \in B} f_i(0) + \sum_{j \in A \setminus B} \sum_{i \in B} (C_{Bi} \cdot f_j(x_i)) + \sum_{i \in B} (C_{Bi} \cdot \sum_{j \in A \setminus B} g_{ji}) + E \cdot \sum_{i \in B} k_i \bmod q.$$

Furthermore, we replace the items in the second expression of the above equation by the following Lagrange interpolating equation

$$f_j(0) = \sum_{i \in B} (C_{Bi} \cdot f_j(x_i)) \bmod q. \qquad (9)$$

Then, we get

$$S = \sum_{i \in A} f_i(0) + \sum_{i \in B} (C_{Bi} \cdot \sum_{j \in A \setminus B} g_{ji}) + E \cdot \sum_{i \in B} k_i \bmod q.$$

Finally, if we do the exponential operations on base α to the both sides of the above equation, then we will know equation (8) holds.

4. Weaknesses in LHLT Scheme

In [26], Li et al. indeed presented elaborate security analysis for their schemes. However, from the above description of LHLT scheme, it is not difficult to see that this threshold signature scheme has the following eight weaknesses: first four of them are about the efficiency, and others about the security.

(1) *Public key length.* In fact, the public key of LHLT scheme not only consists of y_i, but also includes (i, x_i, y_i) and (y_{ij}, z_{ij}), $\forall i, j \in [1, n]$. Because the DC needs y_i, and y_{ij} to check the validity of each partial signature s_i according to equation (5), verifiers need z_{ij} to calculate the value T in equation (7), and both of them need: x_i to calculate C_{Bi} (recall equation (4)).

So, the public key length is dependent of the size n of the group.

(2) **The size of signatures**. In a signature pair(m, B, R, S), B is dependent of the size of threshold t, If n and t are big integers, then the size of signatures becomes big, too.

(3) **Member deletion**. This is an open problem[3] on the design a practical group signature scheme. LHLT scheme did not provide any solution to it, i.e., this scheme lacks QUIT procedure.

(4) *Member addition*. LHLT scheme mentioned that a new member $n+1$ can be dynamically added without affecting the shares of old members. But in fact, in addition to publish his public key pair(x_{n+1}, y_{n+1}), many things have to be done before the new member $n+1$ becomes a group member. For example, new member $n+1$ has to distribute $u_{n+1,j}$ to old members and publish $y_{n+1,j}$ and $z_{n+1,j}$ as Public information; old member j has to Send $u_{j.n+1}$ to new member $n+1$ and publish $y_{j.n+1}$ and $z_{j.n+1}$ as public information. Moreover, in some cases, this procedure will reveal the real identity of the new member $n+1$, because it is possible that all the real identities of members in the group are known publicly (but the corresponding map between identities and public key pairs is a secret). An example of these cases is the directorate of a corporation, where the public key pair (x_j, y_j) of each old member is not changed. But by comparing the identities and public key pairs of the old group and new group, everyone can recognize the real identity of the new member and his public key pair. So there is no anonymity for the new member. In this scenario, maybe the only choice is to reset the system by updating the group public key, and all parameters and secret shares of all members.

(5) *Anonymity*. From the subset B of a valid signature pair (m, B, R, S), each verifier can learn the pseudonyms of all signers, so LHLT scheme can only provide weak anonymity.

(6) **Unlinkablility**. Using information revealed by B, verifiers can link all signatures signed by the same subset or the same member. Therefore, LHLT scheme does not possess unlinkability.

(7) *Traceability*. LHLT scheme does not provide any method to bind the real identity of a member with his pseudonym, so the tracing procedure is not described in details. However, in distribution environments, how to record members' real identities and maintain the relationship between real identities and pseudonyms is really not easy.

(8) *Exculpability*. In [26], Li et al. claimed that the signing set of a group signature cannot be impersonated by any other set of group members, i.e., without the existence of framing attacks. But, in fact there is a framing attack on LHLT scheme. So this threshold group signature scheme does not have exculpability. Details of the framing attack are given in next section.

5. A Framing Attack on LHLT Scheme

In this section, we present the details about how (n −t+ 1) colluding members can forge a valid group signature on any message. This forged signature looks as if it is signed by other (t −1) honest members and one of these corrupted members. At the same time, some of the group members, including all honest members, can generate group signature properly. So, honest members feel the system works normally and cannot detect the existence of any deceit. But in the case of disputes, such forged signatures are opened, and then these honest members have to take responsibility for them.

For convenience, we assume that the first (t−1) members, i.e., member 1, … , t−1, are honest: each of them honestly selects parameters, distributes secret shares, receives and checks his secret shares sent by other members to meet there requirements of section 2, and does not reveal any u_{ji} sent by other members and g_{ij} selected by himself to anybody. But all other members collude with member n: they also select parameters and distribute secret shares to meet the requirements described in section 2;however, some of them reveal the values g_{ij} selected by them selves to member n, others of them intentionally ignore the fact that member n does not send values u_{ni} to them. The whole procedure includes three steps: member n controlling the group private key, member n distributing secret shares, and forging valid group signatures.

5.1 Member n Controlling the Group Private Key

In LHLT scheme, it is not required whether all public keys y_i should be published simultaneously when generating the group public key y according to equation(1). So member n can publish his public key y_n last after he has learned all other $y_i, i \in \{1.\cdots,n-1\}$, in spite that he has prepared his public key y_n as follows,

$$y_n = \alpha^{f_n(0)} \bmod p.$$

Now, member n computes and broadcasts the following value $\overline{y_n}$ as his public key by using all published values $y_i, i \in \{1.\cdots, n-1\}$,

$$\bar{y}_n = y_n \cdot \prod_{i=1}^{n-1} y_i^{-1} \bmod p.$$

Hence, all members in group A will take y_n as the group public key y, but member n knows the group private key $f_n(0)$ corresponding to y, because the following equation holds:

$$y = \bar{y}_n \cdot \prod_{i=1}^{n-1} y_i = y_n = \alpha^{f_n(0)} \bmod p.$$

Of course, member n does not know his private key $\overline{f_n}(0)$ corresponding to $\overline{y_n}$ unless he can solve the following discrete logarithm problem:

$$\bar{y}_n = \alpha^{\bar{f}_n(0)} \bmod p.$$

Once member n controlled the group private key, he can collude with other(n-t+ 1)) members to forge a valid group signature.

5.2 Member n Distributing Secret Shares

By imagining knowledge of a polynomial $\overline{f_n}(X) \in Z_q[X]$ with degree less than t and such that the free term of $\overline{f_n}(X)$ is $\overline{f_n}(0)$, member n can successfully share his private key $\overline{f_n}(0)$ with other members, although he does not know the value of it. Here is the basic idea: Member n selects random numbers as secret shares for the first t-1 (honest) members, but computes other shares for the rest members (his accomplices). The concrete method is described as follows.

1. Member n selects $2(t-1)$ random numbers a_{nj}, $b_{nj} \in_R [1, q-1] (1 \le j \le t-1)$ as the corresponding $\overline{g_{nj}}$ and $\overline{f_n}(X_j)$, respectively, and computes:

$$\begin{aligned} u_{nj} &= \bar{g}_{nj} + \bar{f}_n(x_j) \bmod q \ (= a_{nj} + b_{nj} \bmod q), \\ y_{nj} &= \alpha^{u_{nj}} \bmod p \ \left(= \alpha^{\bar{g}_{nj} + \bar{f}_n(x_j) \bmod q} \bmod p \right), \\ z_{nj} &= \alpha^{\bar{g}_{nj}} \bmod p \ \left(= \alpha^{a_{nj}} \bmod p \right). \end{aligned} \qquad (10)$$

Then, for every $j \in \{1,\cdots,t-1\}$, member n sends u_{nj} to member j secretly, and publishes y_n, and z_{nj} as public information.

2. Because t values of the function $\overline{f_n}(X)$, i,e., $\overline{f_n}(0)$, $\overline{f_n}(X_1)$, $\cdots$ $\overline{f_n}(X_{j-1})$, has been fixed (although member n does not know the exact value of $\overline{f_n}(0)$), the

function $\overline{f_n}(x)$ is determined. For every $l \in [t, n-1]$, if let $B_l = \{1, 2, \cdots, t-1\} \cup \{l\}$, then the following equation holds

$$\bar{y}_n = \alpha^{\bar{f}_n(0)} = \alpha^{C_{B_l l} \cdot \bar{f}_n(x_l)} \cdot \prod_{j=1}^{t-1} \alpha^{C_{B_l j} \cdot \bar{f}_n(x_j)} \bmod p, \quad \forall l \in [t, n-1].$$

From this equation, member n can compute the value of $\alpha^{f_n(xl)}$ as follows:

$$\alpha^{\bar{f}_n(x_l)} = \left(\bar{y}_n \cdot \prod_{j=1}^{t-1} \alpha^{-C_{B_l j} \cdot \bar{f}_n(x_j)}\right)^{C_{B_l l}^{-1} \bmod q} \bmod p, \quad \forall l \in [t, n-1]. \quad (11)$$

3. For the next k ($1 \le k \le n-t$) members (i.e., number $t.\cdots$, $t+k-1$) after the firt ($t-1$) members, member n selects k random numbers $u_{nt} \in {}_R[1, q-1]$, and computes

$$y_{nl} = \alpha^{u_{nl}} \bmod p,$$
$$z_{nl} = y_{nl} \cdot \alpha^{-\bar{f}_n(x_l)} \bmod p.$$

Where, $\alpha^{-\overline{f_n}}(x_l)$ is the inverse of $\alpha^{\overline{f_n(x_l)}}$ determined by equation (11). But in this case, member n does not know the value of $\overline{g_{nl}}$, for each $l \in [t, t+k-1]$.

4. For the last ($n-t-k$) members (i.e., member $t+k, \cdots, n-1$), member n selects $\overline{g_{nl}} / \in {}_R[1, q-1]$, and computes z_{nt} and y_{nl} as follows:

$$z_{nl} = \alpha^{\bar{g}_{nl}} \bmod p,$$
$$y_{nl} = \alpha^{\bar{g}_{nl}} \cdot \alpha^{\bar{f}_n(x_l)} \bmod p \ (= z_{nl} \cdot \alpha^{\bar{f}_n(x_l)} \bmod p).$$

Where, $\alpha^{\overline{f_n}(xl)}$ is determined by equation (11). In this case, member n does not know the value of u_{nl}, for each $l \in [t+k, n-1]$.

5. Up to now, the knowledge of member n is showed in table 1.

Index l	Member n knows	Member n does not know
$l \in [1, t-1]$	$\bar{g}_{nl}, \bar{f}_n(x_l), u_{nl}, y_{nl}, z_{nl}$	$\bar{f}_n(0)$
$l \in [t, t+k-1]$	$u_{nl}, y_{nl}, z_{nl}, \alpha^{\bar{f}_n(x_l)}$	$\bar{f}_n(0), \bar{f}_n(x_l), \bar{g}_{nl}$
$l \in [t+k, n-1]$	$\bar{g}_{nl}, y_{nl}, z_{nl}, \alpha^{\bar{f}_n(x_l)}$	$\bar{f}_n(0), \bar{f}_n(x_l), u_{nl}$

Tab. 1 The Knowledge of Member *n*.

Let set C={1, 2, ···, t-1, t,···, t + k-1}, table 1 shows that each member $i \in C$ knows $u_{\ni}$, so any t members in C can generate valid group signatures normally by using equations (3) and (6). But member n does not know $\overline{f_n}(0)$, and member l ($l \in [t + k, n-1]$does not know $u_{\ni}$, so they cannot take part in the normal generation of threshold group signatures.

Moreover, the situation is worse than this, because there is a framing attack on LHLT scheme.

5.3 Forging Valid Group Signatures

After member n distributed secret shares, he can collude his (n- t) conspirators(i.e., all members j, $j \in [t, n-1)$ to forge a valid group signature for any message m. But (t - 1) honest members and one cheating member have to take responsibility for this forged signature because it includes their pseudonyms and all pseudonyms can be opened if necessary. Now we describe the whole procedure as follows.

1. Member n first selects t random numbers $k_i \in_R [1, q-1]$($i \in B_i$ = {1,2,···, t-1, l} and $l \in [t + k, n]$), then computes values R and E as follows

$$R = \alpha^{\sum_{i \in B_l} k_i \bmod q} \bmod p \left(= \prod_{i \in B_l} r_i \bmod p \right),$$
$$E = H(m, R).$$

2. If $l \in$ [t-k,n -1], each conspirator j (j $\in$ AB_1{n}) sends his secrets g_{ji}(for all i $\in$ B $_i$) to member n. According to table 1, member n knows all $\overline{g_{\ni}}$= (i=1, ···,t-1) and $\overline{g_{\ni}}$ because $l \in$ [t +k, n-1], so he can compute a signature S $_l$ as follows:

$$S_l = f_n(0) + \sum_{i \in B_l} C_{B_l i} \cdot \bar{g}_{ni} + \sum_{j \in A \setminus B_l \setminus \{n\}} \sum_{i \in B_l} C_{B_l i} \cdot g_{ji} + \sum_{i \in B_l} k_i \cdot E \bmod q. \quad (12)$$

3. If l= n, each conspirator j ($j \in$ A$RSLANT$ B $_n$) sends g_{ji}(for all $i \in$ B $_n$) to member n, so member n can compute a signature S_n as follows:

$$S_n = f_n(0) + \sum_{j \in A \setminus B_n} \sum_{i \in B_n} C_{B_n i} \cdot g_{ji} + \sum_{i \in B_n} k_i \cdot E \bmod q. \quad (13)$$

4. Thus, all (n-t +1) corrupted members, including member n, forged a group signature (m, B_l, R, S_l) for message m such that a verifier believes that it is signed collectively by member 1, …, t - 1, and l.

The following theorem guarantees the validity of the forged group signature(m, B_l, R, S_l) obtained from the above procedure.

Theorem 2. *The above forgery attacks are successful, i.e:*
(1) If $l \in [t-k, n-1]$, *then the forged signature*(m, B_l, R, S_l) *computed from equation(12) is a valid threshold group signature for message m;*
(2) If $l = n$, *then the forged signature* (m, B_l, R, S_l)*computed from equation(13) is a valid threshold group signature for message m;*
Proof. (1) In the case of $l \in [t+k, n-1]$, if t members in subset B_l select the same t numbers k_i as in the first step of the above procedure, then their valid signature for message m is given by the following S:

$$S = \sum_{i \in B_l} s_i = \sum_{i \in B_l} f_i(0) + \sum_{i \in B_l} \left(C_{B_l i} \cdot u_{ni} + \sum_{j \in A \setminus B_l \setminus \{n\}} C_{B_l i} \cdot u_{ji} \right) + \sum_{i \in B_l} k_i \cdot E \bmod q.$$

By replacing the u_{ni} and all u_{ji} by the right sides of the first equation in (10)and (2)J exploting the Lagrange interpolating equation (9), and using the fact that $f_n(0) = \overline{f}_n(0) + f_{n-1}(0) + \cdots + f_1(0)$, the above equation can be rewritten as

$$S = f_n(0) + \sum_{i \in B_l} C_{B_l i} \cdot \bar{g}_{ni} + \sum_{j \in A \setminus B_l \setminus \{n\}} \sum_{i \in B_l} C_{B_l i} \cdot g_{ji} + \sum_{i \in B_l} k_i \cdot E \bmod q.$$

By comparing the right sides of the above equation and (12), it is showed that $S = S_l \cdot S_o$, according to Theorem 1, the forged tuple (m, B_l, R, S_l) computed from equation (12) is a valid threshold group siganture for message m.
(2) When $l = n$, the validity of signature (m, B_n, R, S_n) can be proved similarly.

6 An Example and Remarks

In this section, we first give a simple example to explain the disadvantage of the above framing attack. Then, we compare our framing attack with Michels and Horster's attack [271·At last, several simple methods to avoid these attacks are

given.

As an example, we assume that ten members in the directorate of a corporation use a (7,10) threshold group signature scheme to vote on a proposal m by setting t =7 and n =10. As a regulation of this corporation directorate, proposal m is passed if and only if a valid threshold group signature for m is produced, i.e., at least seven members agree on this proposal and then produce valid partial signatures for it. But in fact, the first six members of this directorate disagree on m, while other four members agree on it.

If a secure threshold group signature scheme is used, it is impossible to generate a valid group signature in this scenario. But now, we assume that the LHLT threshold group signature scheme is used and member l0 has controlled the group private key. Therefore, the last four members can forge a valid group signature for m in the pseudonyms of {1, 2, 3, 4, 5, 6, 9} or {1 2, 3, 4, 5, 6, 10} (let k = 2). The result is that the proposal m is passed by the directorate of this corporation, although most members disagree on it. All the honest members do not detect the existence of deceit, because any 7 members of set {1, 2, 3, 4, 5, 6, 7, 8} can produce group signatures normally. But member 9 and l0 cannot generate valid partial signatures. In [27], Michels and Horster also pointed out a framing attack on two schemes in [25][6]. Their attack can also be applied to LHLT scheme since Li et. al [26], did not take any countenneasure to prevent this attack. In Michels and Horster's at-tack (see §4.3 of [27]), it is assumed that member 1 colludes with member t,…n, and the DC (or called as the clerk) to cheat member 2,…, t - 1. The result is that when members in B = {1,2,…,t- 1,t) generate a signature (m, B, R, S)on a message m, member 1(and other creating members) can generate a valid threshold group signature (m, $\hat{B}$, $\hat{R}$, $\hat{S}$) on the same massage m under the name of $\hat{B}$= {1 ,2, …, t-1, t+ 1}. Our attack is stronger than Michels and Horster's in the following senses:

- In Michels and Horster's attack, dishonest members can only forge valid signatures on those messages that honest member 2,…, t - 1 agree to sign. In our attack, however, dishonest members can forge valid signatures on any messages selected by themselves.
- In order to generate signature pair (m, $\hat{B}$, $\hat{R}$, $\hat{S}$), member 1 has to disrupt the signing protocol for one time. So, this abnormal action can be detected by honest

---

6) Note that there are two typos on page 343[27]: the symbol $\overline{r}_1$ appeared in line 8 and 10 should be replaced by a new symbol, for example $\overline{r}_1$ Since $\overline{r}_1$ has been defined as $\overline{r}_1 := r_1^{b_1^{-1} \cdot \overline{b}_1} \bmod p$, a new symbol $\overline{r}_1 = \overline{R} \cdot R^{-1} \cdot r_1 \bmod p$ should be used such that when member 1 reveals $\overline{r}_1$ to all co-signers in B, all signers in B compute $\overline{R}$ instead of R, where $\overline{R} \equiv \overline{r}_1 \cdot \Pi_{i=2}^{t} r_i \bmod p$.

members. But, in our attack, dishonest members don't need to interact with any honest member. Therefore, honest members can only find something wrong when they get a valid group signature signed under their names but they did not sign it at all.

- When a threshold group signature is opened, the true signers are identified. Then, they will deserve awards or punishments corresponding to whether their decision on signing message m is good or bad. In Michels and Horster's attack, only one member is exchanged in B and $\hat{B}$, i.e., member t and $t+1$.Moreover, both of them are dishonest members.

So, their attack means that one dishonest member $t+1$ substitutes another dishonest member t to take awards or bear punishments. But, in our attack all honest members are involved in the dispute.

- To overcome their attack, Michels and Horster proposed an improvement to schemes in [25]: compute $E = E = H(m, R, B)$ instead of $E = H(m, R)$ and use a simultaneous channel for the distribution of r_i or require that all signers prove knowledge of the discrete logarithm of r_i without revealing its value. Even though LHLT scheme is modified according to this improvement, our attack will work as well. The reason is that our attack roots in the public key generation protocol instead of the distribution of values of all Υ_i.

To prevent the above framing attack, the synchronous submissions of each member's public key !h have to be reached in the public key generation protocol. To achieve this goal, we can require that all members have to commit their public keys y_i before any of these values are revealed or each member submits the signed x_i by using his private key $f_i(0)$ when he submits his public key y_i. At the same time, to avoid Michels and Horster's attack their improvement for distribution of r_i should also be adopted.

However, there is no straightforward way to improve LHLT scheme to get rid of other weaknesses described in §4. In fact, to our best knowledge, no existing threshold group signature schemes satisfy all security and efficiency requirements proposed in this paper.

Aknowledgements

The author would like to thank Dr. Jianying Zhou, Dr. Feng Bao, and Dr. Yongdong Wu as well as the anonymous referees for their helpful comments.

References

1. G. Ateniese, J. Camenisch , M. Joye, and G. Tsudik, A practical and provably secure coalition-resistant group signature scheme In: *Crypto'2000, LNCS 1880*. pp.255-270, Springer-Verlag, 2000.

2. G. Ateniese, M. Joye, and G. Tsudik. On the difficulty of coalition-resistant in group signature schemes. In Second *Workshop on Security in Communication Networks (SCN'99)*, September 1999.

3. G. Ateniese, and G. Tsudik. Some open issues and new directions in group signature schemes. In. *Financial Cryptography (FC'99), LNCS 1648*, pp.196-211. Springer-verlag, 1999.

4. C. Boyd. Digital multisignatures. In: *Cryptography Coding*, pp. 241-246. Oxford University Press, 1989.

5. J. Camenisch. Efficient and generalized group signatures. In: *Eurocrypt'97, LNCS 1233*, pp. 465-479. Springer-Vrlag, 1997.

6. J. Camenisch, and M. Stadler, Efficient group signature schemes for large groups. In: *Crypto'97, LNCS 1294*, pp. 410-424. Springer-Verlag, 1997,

7, J. Camenisch. Group signature schemes and payment systems based on the discrete logarithm problem. *Vol. 2 of ETH-Series in information Security on Cryptography,* ISBN 3-89649-286-1, Hartung-Gorre Verlag, Konstanz, 1998.

8. J. Camenisch, and M. Michels. Separability and efficiency for generic group signature schemes. In: *Crypto '99, LNCS 1666*, pp. 413-430. Springer-Verlag, 1999·

9. J, Camenisch, and A, Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In: *Crypto '2002, LNCS 2442*, pp. 61-76. Springer-Verlag, 2002.

10. D. Chaum, E. van Heyst. Group signatures. In: *Eurocrypt'91, LNCS 547*, pp. 257-265 springer-Verlag, 1991.

11. L. Chen, and T.P. Pedersen. New group signature schemes. In: *Eurocrypt'94, LNCS 950*, pp. 171-181. Springer-Verlag. 1995.

12. L. Chen, and T.P. Pedersen. On the efficiency of group signatures providing information-theoretic anonymity, In: *Eurocrpt'95, LNCS 921*, pp. 39-49. Springer-Verlag, 1995.

13. Y. Desmedt. Society and group oriented cryptography: a new concept. In *Crypto'87, LNCS 293* pp.120-127. Springer-Verlag, 1988.

14. Y. Desmedt, and Y. Frankel. Threshold cryptosystems. In *Crypto '89, LNCS 435*, pp. 307-315. Springer-Verlag, 1990.

15. A. Fujioka, T. Okamoto, and K. Ohta. A practical digital multisignature scheme based on discrete logarithms. In: *Auscrypt '92 LNCS 718*, pp. 244-251. Springer-Verlag, 1992.

16. R.Gennaro, S. Jarecki, H.Krawczyk, and T. Rabin. Robust threshold DSS signatures. In: *Eurocrypt'96, LNCS 1070*, pp. 354-371. Springer-Veriag. 1996.

17. L. Harn. Group-oriented (t, n) threshold digital signature scheme and multisignature. *IEE Proceedings - Computers and Digital Techniques*, 1994, 141(5): 307-313.
18. L Harn. New digital signature scheme based on discrete logarithm. *Electronic Letters*, 1994, 30(5): 396-398.
19, L. Harn, and S. Yang, Group-oriented undeniable signature schemes without the assistance of a mutually trusted party. In *Auscrypt'92 LNCS 718*, pp.133-142 Springer-Verlag, 1993.
20. P. Horster, M. Michels and H. Petersen. Meta-multisignature schemes based on the discrete logarithm problem. In *Proc. of IFIP/SEC'95*, pp. 128-141. Chapman& Hall, 1995.
21. M, Joye, S. Kim and N-Y. Lee. Cryptanalysis of two group signature schemes. *Information Security (ISW'99), LNCS 1729*, pp. 271-275. Springer-Verlag, 1999.
22. M. Joye, N-Y. Lee, and T. Hwang. On the security of the Lee-Chang group signature scheme and its derivatives. In: *Information Security* (ISW'99), LNCS 1729, pp. 47-51. Springer-Verlag, 1999.
23. H-J. Kim, J.I. Lim and D.H. Lee. Efficient and secure member deletion in group signature schemes. In: *Information Security and Cryptology (ICISC 2000), LNCS 2015*, pp. 150-161, Springer-Verlag, 2001.
24. S.K. Langford. Weaknesses in some threshold cryptosystems. *In Crypto '96, LNCS 1109*, pp.74-82. Springer-Verlag, 1996.
25. C-M. Li, T. Hwang and N-Y. Lee. Threshold-multisignature schemes where suspected forger implies traceability of adversarial shareholders. In: *Eurocrypt '94, LNCS 950*, pp.194-204. Springer-Verlag, 1995.
26. C-M. Li, T. Hwang, N-Y. Lee, and J-J. Tsai, (t, n) threshold-multisignature schemes and generalized-multisignature scheme where suspected forgery implies traceability of adversarial shareholders. *Cryptologia*, July 2000, 24(3): 250-268.
27. M Michels, and P. Horster. On the risk of discruption in several multiparty signature schemes. In *Asiacrypt'96, LNCS 1163*, pp.334-345. Springer-Verlag, 1995.
28. T Okamoto. A digital multisignature scheme using bijective public-key cryptosystem. *ACM Transactions on Computer Systems*, 1988, 6(8): 432-441.
29. T. Ohata. and T. Okamoto. A digital multisignature scheme based on the Fiat-Shamir scheme. In *Asiacrypt'91, LNCS 739*, pp. 75-79, Springer-Verlag. 1991.
30. C. Park, and K. Kurosawa. New Elgamal type threshold digital signature scheme. *IEICE Trans. Fundamentals*, January 1996, E79-A(1): 86-93.
31. H, Petersen. How to convert any digital signature scheme into a group signature scheme, In: *Proc. of Security Protocols Workshop '97, LNCS 1361*, pp. 67-78. Springer-Verlag, 1997.
32. A. Shamir. How to share a secret, Communications of the ACM, l979,22(11). 612-613.