Security Analysis Concerning the Random Numbers of Threshold Ring Signatures

Kaibin Huang
Department of Computer Science
National Chengchi University
Taipei 11605, Taiwan
foxleooo@gmail.com

Raylin Tso
Department of Computer Science
National Chengchi University
Taipei 11605, Taiwan
raylin@cs.nccu.edu.tw

***Abstract*—Since the first ring signature was introduced by Revist et al. in 2001, several schemes have been proposed extending from the original "1-out-of-n" ring signature. For instance, "t-out-of-n" threshold ring signatures are proposed and often researched. We noticed that the careless random number generation in threshold ring signature schemes might influence the security of threshold ring signatures. In this work, we demonstrate the vulnerability of the careless random number generation; moreover, we provide a solution to the problem as well as perform a security analysis of the modification.**

Keywords- hash function; random number generation; ring signature; security analysis; threshold ring signature

I. INTRODUCTION

Group-oriented signatures [1][2][3][4] were introduced to provide a signer anonymous signature scheme in some useful applications. In group-oriented signature schemes, a member of the group may arbitrarily choose other members to generate a group-oriented signature (including group signatures [3][4] and ring signatures [1][2][7][10]) without their assistance. Anyone can verify that the signature comes from someone in the group (or ring), but has no idea of the real signer among the group's (or ring's) members. In some cases as in the case of leaking a secret from a specific organization, group-oriented signatures are quite useful.

There are two major kinds of group-oriented signatures: group signatures [3][4] and ring signatures [1][2][7][10]. Both of them preserve signer anonymity so that the signer is protected anonymously in the group (or ring). The major difference between the two kinds of group-oriented signatures is that there is a trusted group manager in a group signature, who is able to convert a group signature into a traditional single-signer signature when necessary. Otherwise, ring signatures are distributed and not convertible. In this paper, the discussion is focused on ring signatures.

Extended from the "1-out-of-n" ring signature, a "t-out-of-n" threshold ring signature makes it possible to invite some ring members as signers to co-generate a ring signature that remains signer ambiguous. Some people in the group contribute to the signature, but signers are anonymous, and hence, no one can discover which members signed the document. Here is a threshold ring signature scenario:

There is a congress voting by n people in progress. Assume that t of them agree and that the others disagree.

- Can the t signatures be combined into a single signature, instead of t signatures?
- How can they show the voting result with signatures, but eliminate the risk of backlash?

The threshold ring signature is definitely suitable for achieving the two requirements above. On one hand, a "t-out-of-n" threshold ring signature generated by t members in the ring; on the other hand, the signer ambiguity of the threshold ring signature uncertainly as to which t members were involved in its generation. Several research studies have been performed on threshold ring signatures [5][6][9][11]. In this paper, we discuss the security analysis of threshold ring signatures.

A. Motivation

We noticed that some "t-out-of-n" threshold ring signature schemes ignore the security process of the generation of random numbers. We termed this problem as "the careless random number generation problem." In general, the generation of random number is controlled by a unique signer and kept secret in most signing algorithms. The means of generating the random numbers is rarely discussed and purposely not made a point of focus. In other words, in many schemes, it does not matter how the random number is generated or who is responsible for the generation of the random numbers. However, in the threshold ring signature (multi-signer signing algorithm), there is a problem because one or more signers generate the random numbers. What is the difference between them? Could the effort required to ensure the security of the signature be a problem?

B. Our contribution

We analyze the security and vulnerability of the careless random number generation and point out that the problem may damage the security of the threshold ring signature. Then, we propose a modification to the careless random

978-0-7695-4654-4/12 $26.00 © 2012 IEEE
DOI 10.1109/ITNG.2012.95

number generation problem. Finally, a security analysis of the modification is provided.

C. Paper organization

The rest of paper is organized as follows: preliminaries are briefly mentioned in section 2. In section 3, Liu et al.'s threshold ring signature scheme [5]is revisited in detail so that we may take it as an example for analyzing security, especially with respect to random number generation. The security and vulnerability issues resulting from the careless random number generation, which are analyzed in section 4. Countermeasures against the careless random number generation problem are given in section 5. Finally, the conclusion is provided in section 6.

II. PRELIMINARIES

In this section, we list some preliminary works. First, hash function plays an important role in this paper so that it will be listed. Then, the security of Liu et al.'s scheme (in DL case) is based on DLP, so DL problem is revisited. Finally the syntax and security requirement of threshold ring signature are introduced at the end of this section.

A. Cryptographic hash function

According to [12], a cryptographic hash function is a deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string, the (cryptographic) hash value, such that an accidental or intentional change to the data will change the hash value. The data to be encoded is often called the "message," and the hash value is sometimes called the message digest or simply digest.

The ideal cryptographic hash function has four main or significant properties:

- It is easy (but not necessarily quick) to compute the hash value for any given message.
- It is infeasible to generate a message that has a given hash.
- It is infeasible to modify a message without changing the hash.
- It is infeasible to find two different messages with the same hash.

B. Discrete Logarithm Problem

p and q are two large prime numbers, $q \mid (p-1)$, $g \in Z_p^*$, $order(g) = q$, $< G >$ is a subgroup of Z_p^*, h is randomly selected in $< G >$. Discrete logarithm problem: given $g, h \in < G >$, find $x \in Z_q$ such that $g^x \equiv h \bmod p$. Discrete logarithm problem is regarded as difficult in general.

C. Model of Threshold Ring Signatures

According to [5], (t, n) -threshold ring signature scheme consists of the algorithms $(G, S_{t,n}, V_{t,n})$.

- *Key Generation Algorithm*

$(\hat{s}, P) \leftarrow G(1^k)$ is a probabilistic polynomial time algorithm (PPT) which takes as input a security parameter k, and produces a private key $\hat{s}$ and a public key P of a user.

- *Signing Algorithm*

$\sigma \leftarrow S_{t,n}(1^k, \hat{S}, \mathrm{L}, \mathrm{m})$ is a PPT which accepts as inputs a security parameter k, a set of private keys $\hat{S}$, a set of public keys L including the ones corresponding to the private keys in $\hat{S}$ and a message m, produces a signature σ. We require that $|\hat{S}| = t$ and $|L| = n$ where $0 < t \le n$.

- *Verification algorithm*

$1/0 \leftarrow V_{t,n}(1^k, L, m, \sigma)$ is a polynomial-time algorithm which accepts as inputs a security parameter k, a set L consists of n public keys, a message m and a signature σ, returns 1 or 0 for **accept** or **reject** the signature respectively. We require that $V_{t,n}(1^k, L, m, S_{t,n}(1^k, \hat{S}, L, m)) = 1$ for any message m and any set L of n public keys in which the public keys corresponding to all the private keys of $\hat{S}$ are included.

For simplicity, we usually omit the input of security parameter when using $S_{t,n}$ and $V_{t,n}$ in the rest of the paper.

L may include public keys based on different security parameters. The security of the signature scheme defined above is set to the smallest k among them. G may also be extended to take the description of key types.

D. Security Requirement of Threshold Ring Signatures

The security of a (t,n) -threshold ring signature scheme [5]consists of two requirements, namely *Signer Ambiguity* and *Unforgeability*. They are defined as follows.

- *Signer Ambiguity*

Let $L=\{P_1,...,P_n\}$ where each key is generated as $(\hat{s}_i,P_i)\neg\ G(1^{k_i})$ for some $k_i\in N$. Let $k=\min(k_1,...,k_n)$. A (t,n) -threshold ring signature scheme is unconditionally signer ambiguous if, for any L , any message m , and any signature $\sigma\leftarrow S_{t,n}(1^k,\hat{S},L,m)$ where $\hat{S}\subseteq\{\hat{s}_1,...,\hat{s}_n\}$ and $|\hat{S}|=t$, any unbound adversary E accepts as inputs L , and σ , outputs π such that $\hat{s}_\pi\in\hat{S}$ with probability t/n .

Intuitively, signer ambiguity means that it is infeasible to identify that which t signers out of n possible signers actually work jointly to generate a (t,n) -threshold ring signature.

- *Unforgeability*

Let $L=\{P_1,...,P_n\}$ be the set of n public keys in which each key is generated as $(\hat{s}_i,P_i)\neg\ G(1^{k_i})$ where $k_i\in N$. Let $k=\min(k_1,...,k_n)$. Let $SO(L',i_1,...i_{t'},m)$ be a signing oracle that takes any set L' of public keys, where $L'\subseteq L$ and $n'=|L'|$, any t' signers indexed by $i_1,...i_{t'}$ where $1\le i_j\le n$, $1\le j\le t'$ and $t'\le n'$, and any message m , produces a (t',n') -threshold ring signature $\sigma\leftarrow S_{t',n'}(\{\hat{s}_{i_1},...,\hat{s}_{i_{t'}},L',m\})$, such that $V_{t',n'}(L',m,\sigma)=1$. Let $\hat{S}_{t-1}$ be any set of $t-1$ private keys corresponding to the public keys in L . A (t,n) -threshold ring signature scheme is unforgeable if, for any PPT , A with signing oracle SO , for any L , and for all sufficiently large k , $\Pr[A^{SO}(1^k,L,\hat{S}_{t-1})\rightarrow(m,\sigma):1\leftarrow V_{t,n}(L,m,\sigma)]\le\in(k)$ where $\in$ is a negligible function. Restriction is that $(L,i_1,...,i_t,m,\sigma)$ should not be found in the set of all oracle queries and replies between A and SO for any $1\le i_j\le n$, $1\le j\le t'$. The probability is taken over all the possible inputs of A , oracle queries and coin flips of A . A real-valued function $\in$ is negligible if for every $c>0$, there exists a $k_c>0$ such that $\in(k)<k^{-c}$ for all $k>k_c$. We say that a (t,n) -threshold ring signature scheme is secure if it satisfies the above requirements.

That is to say, to generate a (t,n) -threshold ring signature, there must be at least t ring members participating in the scheme; otherwise according to the unforgeability, any k ring members $(1\quad k<t)$ are not able to sign a (t,n) -threshold ring signature.

III. LIU ET AL.'S (t,n) -THRESHOLD RING SIGNATURE

In this section, we review Liu et al.'s [5]threshold ring signature scheme (in DL-Problem case) and give a discussion.

A. Model of Liu et al.'s scheme

For $i=1,...,n$, user i owns public key (p_i,q_i,g_i,y_i) and private key x_i , where p_i and q_i are prime, $q_i\,|\,(p_i-1)$, $g_i\in z^*_{p_i}$ of order q_i and $y_i=g_i^{x_i}\bmod p_i$. We assume that the discrete logarithm problem modulo p_i is hard. Let L be the set of all public keys of the n users.

Let ρ be twice the bit length of the largest q_i and N_i , for $1\le i\le n$. Let $G:\{0,1\}^*\rightarrow\{0,1\}^\rho$ be some cryptographic hash function. Without loss of generality, suppose that user j , for $1\le j\le t$, are participating signers and user i , for $t+1\le i\le n$, are non-participating signers. To generate a (t,n) -threshold ring signature on a message $m\in\{0,1\}^*$, the t participating signers carry out the following steps.

- *The Signing Algorithm*

1. For $i=t+1,...,n$, pick $c_i\in_R\{0,1\}^\rho$ and $s_i\in_R Z_{q_i}$. Compute $z_i=g_i^{s_i}y_i^{c_i}\bmod p_i$.
2. For $j=1,...,t$, pick $r_j\in_R Z_X$ and compute $z_j=g_j^{r_j}\bmod p_j$.
3. Compute $c_0=G(L,t,m,z_1,...,z_n)$ and construct a polynomial f over $GF(2^\rho)$ such that $\deg(f)=n-t$, $f(0)=c_0$ and $f(i)=c_i$, for $t+1\le i\le n$.

4. For $j = 1,\ldots,t$, compute $c_j = f(j)$ and $s_j = r_j - c_j x_j \bmod q_j$.
5. Output the signature for m and L as $\sigma = (s_1,\ldots,s_n,f)$.

- *The Verification Algorithm*

A verifier checks a signature $\sigma = (s_1,\ldots,s_n,f)$ with a message m and a set of public keys L as follows.

1. Check if $\deg(f) = n - t$. If yes, proceed. Otherwise, reject.
2. For $i = 1,\ldots,n$, compute $c_i = f(i)$ and $z_i' = g_i^{s_i} y_i^{c_i} \bmod p_i$.
3. Check whether $f(0) \overset{?}{=} G(L,t,m,z_1',\ldots,z_n')$. If yes, accept. Otherwise, reject.

B. Discussion

Liu et al.'s threshold ring signature scheme [5]is an extension of Abe et al.'s [1] scheme. Follow [1], the Liu et al. scheme also takes advantage of the secret sharing idea [8] proposed by Shamir in 1979. Most threshold ring signatures make use a of secret sharing idea [8] to achieve "t-out-of-n" signatures. In those threshold ring signature schemes that use the secret sharing idea, signers use their secret keys to co-generate the ring signature; meanwhile, signers know nothing about non-signers secret keys, and they do not need them. In general, signers generate some random numbers to achieve the variables corresponding to non-signers. The security of the Liu et al. scheme is rigorously proved in [5], but random number generation is not strictly defined; it is not defined by a function or by one or more participating signers. Specific signers are not required to execute the signing steps (especially steps 1, 3, and 5). In the next section, we describe the conditions that exist when there is a dishonest signer in the ring.

IV. SECURITY IN CARELESS RANDOM NUMBER GENERATION

Signer ambiguity and unforgeability are rigorously required security properties in ring signature schemes. In this section, we focus on signer ambiguity. We use "A Separable Threshold Ring Signature Scheme" [5](in the case of DLP) proposed by Liu et al. as an example. The security of signer ambiguity is strictly proved in their paper. The probability of guessing the real signer is t/n, however, we noticed that there is a security flaw that arises from a neglected part of the scheme, that is, the generation of random numbers. Random numbers in the scheme include c_i and s_i $(t+1 \le i \le n)$, which are just selected randomly in the specific fields ($\{0,1\}^{\rho}$ and Z_q) respectively. However, there is no rule on how to generate the random numbers. It will be vulnerable to damage from signer ambiguity.

Before introducing the vulnerability, without loss of generality, let us assume that there are n members in the ring and t members participate in the (t,n)-threshold ring signature. For $k = \{1,\ldots,n\}$, user U_k represents the set of all members in the ring signature; U_i denotes the set of non-signers $(t+1 \le i \le n)$, and U_j represents the set of signers $(1 \le j \le t)$.

The key generation phase is the same as the Liu et al. scheme. Then, we define a signer U_{evil}, $U_{evil} \in U_j$, who is a dishonest signer in the ring signature. The propose of U_{evil} is to expose the identity of non-signers when necessary, in other words, to break the signer ambiguity of the scheme. U_{evil} creates cryptographic hash functions $H_k : \{0,1\}^* \to Z_{q_k}$ for all ring members, U_k. Here, we demonstrate how U_{evil} influences the security of the threshold ring signature, as follows:

- In step 1 of the signing algorithm phase in [5], U_{evil} picks $s_i' \in_R Z_{q_i}$ and generates $s_i = H_i(s_i')$ $(i = t+1,\ldots,n)$, instead of randomly selecting s_i.
- U_{evil} preserves $(s_{t+1}',\ldots s_n')$ secretly.

As shown in the two tricks above, U_{evil} follows the Liu et al. scheme to generate a (t,n)-threshold ring signature, $\sigma = (s_1,\ldots,s_n,f)$, that passes the verification algorithm with other signers. Random numbers, s_i, are generated through hash functions $s_i = H_i(s_i')$ $(t+1 \le i \le n)$ instead of being randomly selected. Nevertheless, s_i is regarded as a random number by everyone except U_{evil}.

In steps 1 and 4 of the signing algorithm, non-signers U_i and signers U_j own $s_i = H_i(s_i')$ and $s_j = r_j - c_j x_j \bmod q_j$, respectively. By the one-way property in preliminaries section, it is infeasible to calculate s_i' from s_i and H_i; further, it is infeasible to calculate a number " s_j' " from s_j and H_j so that $s_j = H_j(s_j')$. Once U_{evil} reveals some s_i' and H_i that satisfy $s_i = H_i(s_i')$, s_i is regarded as being generated from a hash function instead of by signers $s_j = r_j - c_j x_j \bmod q_j$; in other words, by doing this, anyone can be convinced that U_i is a non-signer.

In summary, the careless random number generation gives U_{evil} the chance to revoke non-signers via revealing some critical message in the signing algorithm. Random number generation is a neglected part of most threshold ring signature schemes that use the secret sharing idea and is rarely discussed. However, as we demonstrated, the negligence may be a fatal vulnerability that influences signer ambiguity. We propose improvements in the following section.

V. A SOLUTION TO THE PRPBLEM

As discussed in section 4, neglecting random number generation may cause fatal damage to signer ambiguity. Obviously, if there are two or more signers who participate in random number generation, the problem will be solved trivially. Without loss of generality and fairness, we may regulate the way in which random numbers are generated. Take the Liu et al. scheme [5] for example. There are three different kinds of random numbers, c_i, s_i and r_j, where c_i and s_i belong to U_i, U_i denotes non-signers; r_j belongs to U_j, and U_j represents signers. Random numbers corresponding to signers should be kept secret (ex: U_j keeps r_j). On the other hand, random numbers about non-signers, such as c_i and s_i in [5], should be co-generated by all signers. For instance, $c_i = \sum_{j=1}^{t} c_{ran_j}$, $s_i = \sum_{j=1}^{t} s_{ran_j}$, each c_i and s_i is composed from the sums of $c_{ran_j} \in_R \{0,1\}^{\rho}$ and $s_{ran_j} \ {}_R Z_{q_i}$, respectively, where c_{ran_j} and s_{ran_j} represent random numbers selected by signers U_j $(1 \le j \le t)$. We adjust the signing algorithm of the Liu et al. scheme [5] so that each step of the signing algorithm is assigned to be executed by some signers, U_j, in the ring signature.

A. *Adjusted Signing Algorithm*

1. For $i = t+1, \ldots, n$, signers U_j pick $c_{ran_j} \in_R \{0,1\}^{\rho}$ and $s_{ran_j} \ {}_R Z_{q_i}$. U_j compute $c_i = \sum_{j=1}^{t} c_{ran_j}$, $s_i = \sum_{j=1}^{t} s_{ran_j}$,and $z_i = g_i^{s_i} y_i^{c_i} \bmod p_i$.
2. For $j = 1, \ldots, t$, U_j pick $r_j \in_R Z_{q_j}$ and compute $z_j = g_j^{r_j} \bmod p_j$.
3. U_j compute $c_0 = G(L, t, m, z_1, \ldots, z_n)$ and construct a polynomial f over $GF(2^{\rho})$ such that $\deg(f) = n - t$, $f(0) = c_0$ and $f(i) = c_i$, for $t+1 \le i \le n$.
4. For $j = 1, \ldots, t$, U_j compute $c_j = f(j)$ and $s_j = r_j - c_j x_j \bmod q_j$.
5. U_j output the signature for m and L as $\sigma = (s_1, \ldots, s_n, f)$.

B. *Security Analysis*

By the one-way property in preliminaries section, it is infeasible to calculate a corresponding "s_i'" from s_i and H_i that satisfies $s_i = H_i(s_i')$; similarly, it is infeasible to compute "c_i'" from c_i and H_i where $c_i = H_i(c_i')$. Without the threat of revealing non-signers, signer ambiguity is guaranteed. On the other hand, each random number, such as c_i and s_i, comes from the sum of t random numbers in fields $\{0,1\}^{\rho}$ and Z_{q_i}, respectively. The original security of the ring signature is not influenced when some random numbers are composed from the sums of several random numbers in the same field.

VI. Conclusion

In this paper, we find out a fatal vulnerability in threshold ring signatures, which comes from random number generation. Take Liu et al.'s [5] for example, we demonstrate how an evil ring signature signer breaks signer anonymous through revealing non-signers because of the carless random number generation. Finally, we propose a solution to avoid this vulnerability and provide security analysis about the proposed solution.

REFERENCE

[1] M. Abe, M. Ohkubo and K. Suzuki, 1-out-of-n signatures from a variety of keys, *Advances in cryptology – ASIACRYPT'02*, Lecture Notes in Computer Science 2501, pp.415–432, 2002.

[2] A. K. Awasthi and S. Lal, ID-based ring signature and proxy ring signature schemes from bilinear pairings, *Cryptology ePrint Archive*, Report 2004/184, 2004.

[3] D. Chaum and E. van Heyst. Group signatures. In Advances in *Cryptology — EROCRYPT* '91, vol. 547 of LNCS, pp. 257–265, Springer-Verlag, 1991.

[4] D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *Proceedings of Asiacrypt 2010*, volume 6477 of LNCS, pages 395–412. Springer-Verlag, 2010.

[5] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. A Separable Threshold Ring Signature Scheme. In Jong In Lim and Dong Hoon Lee, *Information Security and Cryptology - ICISC* 2003, 6th *International Conference Seoul*, Korea, November 27-28, 2003, Revised Papers, volume 2971 of Lecture Notes in Computer Science, pages 352–369. Springer, 2003

[6] Patrick P. Tsang, Victor K. Wei, Man Ho Au, Tony K. Chan, Joseph K. Liu, and Duncan S. Wong. Separable linkable threshold ring signatures. In *Indocrypt 2004*, volume 3348 of LNCS, pages 384–398. Springer-Verlag, 2004.

[7] R. Rivest, A. Shamir and Y. Tauman, How to leak a secret, *Advances in cryptology – ASIACRYPT'01*, Lecture Notes in Computer Science 2248, pp.552–565, 2001.

[8] A. Shamir. How to Share a Secret. *Communication of ACM (CACM)*, Vol. 22, pages 612–613, 1979.

[9] Zhong-hua SHEN, Xiu-yuan, Threshold signature scheme with threshold verification based on multivariate linear polynomial, in Journal of Jiaotong University. (Science.), pp. 551–556, 2011

[10] Raylin Tso, Convertible ring signatures with gradual revelation of non-signers, Computer and Communication Networks, Article first published online: 18 MAY 2011, available at http://onlinelibrary.wiley.com/doi/10.1002/sec.334/full

[11] Raylin Tso, Tadakiko Ito, Takeshi Okamoto, Eiji Okamoto, Design and analysis of "flexible k-out-of-n signatures", in Proceedings of ATC'10, Springer, Lecture Notes in Computer Science, Vol. 6407, pp. 255–267, 2010.

[12] Cryptographic hash function, from Wikipedia, available at: http://en.wikipedia.org/wiki/Cryptographic_hash_function