Lattice-based Threshold Ring Signature with Message Block Sharing

Jiangshan CHEN[1,2*], Yupu HU[1], Wen GAO[3], Hongmei Liang[2]
[1] State Key Laboratory of Integrated Service Networks, Xidian University
Xi'an 710071-China.
[e-mail: yphu@mail.xidian.edu.cn]
[2] School of Mathematics and Statistics, Minnan Normal University
Zhangzhou Fujian 363000-China.
[e-mail: jschen@mnnu.edu.cn]
[3]College of Electrical & Information Engineering, Shaanxi University of Science & Technology
Xi'an 710021-China.
[e-mail: gaowen@sust.edu.cn]
*Corresponding author: Jiangshan Chen

Received November 21, 2017; revised March 9, 2018; revised June 29, 2018; accepted August 23, 2018; published February 28, 2019

Abstract

Threshold ring signature scheme enables any *t* entities from *N* ring members to spontaneously generate a publicly verifiable *t*-out-of-*N* signature anonymously. The verifier is convinced that the signature is indeed generated by at least *t* users from the claimed group, but he cannot tell them apart. Threshold ring signatures are significant for ad-hoc groups such as mobile ad-hoc networks. Based on the lattice-based ring signature proposed by Melchor *et al.* at AFRICRYPT'13, this work presents a lattice-based threshold ring signature scheme, employing the technique of message block sharing proposed by Choi and Kim. Besides, in order to avoid the system parameter setup problems, we proposed a message processing technique called "pad-then-permute", to pre-process the message before blocking the message, thus making the threshold ring signature scheme more flexible. Our threshold ring signature scheme has several advantages: inherits the quantum immunity from the lattice structure; has considerably short signature and almost no signature size increase with the threshold value; provable to be correct, efficient, indistinguishable source hiding, and unforgeable.

Keywords: Threshold ring signature, lattice, anonymous, unforgeable

This work was supported by the National Key R&D Program of China under Grants No. 2017YFB0802000, National Natural Science Foundations of China(61472309, 61672412), National Cryptography Development Fund (MMJJ20170104).

1. Introduction

In 1997, Ajtai[1] showed a vital conclusion that, the worst-case hardness of the standard lattice problem GapSVP can be reduced to the average-case hardness of SIS problem. After that, lattice-based cryptography become an appealing alternative to number theory based cryptology, due to its quantum resistant assumptions. According to the reference [26], lattice-based cryptography is becoming practical.

In the real world, ring signatures are widely used, such as block-chain, e-commerce, and so on. In particular, when people not only need to protect the identity of the signer, but also need to disperse the power of the signer, the threshold ring signature can meet their needs, such as, electronic voting. The concept of ring signature was introduced by Rivest et al.[2] based on RSA. In certain circumstances, the designer would prefer to decrease the power of the signer, such as major decisions in party and country, or military and national defense. This matter can be settled by requiring the signature generated by several members in a group instead of one party. Bresson et al.[3] first defined the threshold ring signature. A t-out-of-N threshold ring signature allows any at least t users to generate a signature in the name of N members, without leaking any information about the set of signers that generate the signature (the t signers can be combined freely from N members). For any case that fewer than t members involved during the signing process, it is impossible to generate a valid signature. Any conspiracy of less than *t* corrupted members cannot produce a valid signature. Several threshold ring signature schemes ([4-12,24,25]) were put forward. The application of threshold technology is widely in many subject areas, including in the cutting-edge direction such as big data ([27],[28]).

Feng et al.'s[11] threshold signature is highly interactive and the scheme is based on NTRUSign, which has been broken by Nguyen and Regev[13]. Cayrel et al.'s[10] work gives a ring threshold signature by modifying the threshold signature scheme based on the syndrome decoding problem with identification scheme, but their scheme is based on small integer solution (SIS) problem which is weaker than LWE problem. Bendlin et al.[12] propose a threshold signature scheme by using an algorithm to share a lattice trapdoor. Their scheme is based on LWE problem but conceptually hard to understand. Bettaieb and Schrek[14] propose an improved lattice-based threshold ring signature scheme based on Cayrel et al.'s work, they generalize the same identification scheme CLRS to obtain a more efficient one, and the main improvement is a significant reduction of the size of the signature, but the signature size still increases obviously with the threshold value.

Based on the lattice-based ring signature proposed by Melchor et al[15] at AFRICRYPT'13, we present a lattice-based threshold ring signature scheme in this work, by employing the message block sharing technique (Choi and Kim[16]). Besides, in order to avoid the system parameter setup problem, a new technique which we call "Pad-then-Permute", is adopted to pre-process the message before blocking it, thus making the threshold ring signature scheme more flexible in practice. Our proposal has considerably short signature, and there is little or no increase in signature size with the threshold value. Besides, we proved that the proposed scheme is correct, efficient, indistinguishable source hiding, and unforgeable.

The rest of the paper is organized as follows. The preliminaries are provided in Section 2, and the key techniques are introduced in Section 3. We describe the syntax of threshold ring signature in Section 4 and our construction in Section 5. The security analysis is shown in Section 6, and the performance analysis is given in Section 7. This paper concludes in Section 8.

2. Preliminaries

2.1 Notations

$\mathbb{R}$ denotes the set of real numbers, and $\mathbb{Z}$ denotes the set of integers. For positive integer *d*, [*d*] denotes the set $\{1,\cdots,d\}$. For a given set *S*, $x \leftarrow S$ represents that *x* is a uniformly random sample chosen from *S*. For positive integer *m* and *n* such that $m \geq n$, C_m^n describes the combination number which equals to $m!/(n!(m-n)!)$. $\mathbb{Z}_p$ denotes the quotient ring $\mathbb{Z}/p\mathbb{Z}$. In this work, we build our construction upon the ring $\mathcal{D} = \mathbb{Z}_p[x]/\langle x^n+1\rangle$, where x^n+1 is irreducible, *n* is a power of two, and *p* is a prime such that $p = 3 \bmod 8$. Elements in $\mathcal{D}$ are represented by polynomials of degree $n-1$ with coefficients in $\{-(p-1)/2,\cdots,(p-1)/2\}$. Generally, polynomials are denoted by Roman letters $(a,b,\cdots)$, and vector of polynomials by roman letters with hats as $(\hat{a},\hat{b},\cdots)$. For a positive integer *m* and polynomials $a_1,\cdots,a_m \in \mathcal{D}$, the vector of polynomials $(a_1,\cdots,a_m)$ is denoted by $\hat{a}$. For any polynomial *a*, its infinity norm l_∞ is defined as $\| a \|_\infty = \max_i | a^{(i)} |$, where $a^{(i)}$ are coefficients of *a*. Similarly, the infinity norm of vector of polynomials $\hat{a} = (a_1,\cdots,a_m)$ is defined as $\| \hat{a} \|_\infty = \max_i \| a_i \|_\infty$ where $i \in [m]$.

Let *n* be the system security parameter, other parameters are implicitly determined by *n*. We use the notations *O*, ω to show the growth of functions. *poly*(*n*) denotes functions such that $f(n) = O(n^c)$ for some *c*. We say that *f*(*n*) is negligible (denoted by *negl*(*n*)) if, $f(n) < n^{-c}$ holds for all positive *c* and a sufficiently large *n*. A probability is *overwhelming* if it is $1 - negl(n)$.

2.2 Foundation of Lattice

We recall the definitions of lattice and the shortest independent vector problem(SIVP).

Definition 1. Let $B = \{\mathbf{b}_1,\cdots,\mathbf{b}_m\}$ be a set of *m* linearly independent vectors over $\mathbb{R}^n$. The lattice generated by *B* is defined by

$$\mathcal{L}(B) = \left\{ \sum_{i=1}^{m} x_i \mathbf{b}_i \mid x_i \in \mathbb{Z} \right\}, \text{ for } i \in \{1,\cdots,m\}. \quad (1)$$

Generally, $\lambda_1(\mathcal{L}(B))$ denotes the shortest vector of on lattice $\mathcal{L}(B)$.

Definition 2. Let $\mathcal{L}(B)$ be a lattice with rank *m*, for $i \in \{1,\cdots,m\}$, the *i*-th successive minima of $\mathcal{L}(B)$ is defined by

$$\lambda_i(\mathcal{L}) = \inf \left\{ r \in R \mid \dim\left(span\left(\mathcal{L} \cap \bar{B}(0,r) \right) \right) \geq i \right\}, \quad (2)$$

where $\bar{B}(0,r) = \left\{ x \in \mathbb{R}^n \mid \| x \| \leq r \right\}$ is the *m*-dimensional sphere with the origin as its center and radius *r*, inf represents the lower bound, and dim defines the dimension of the space spanned by the lattice points in $\bar{B}(0,r)$.

Definition 3 (SIVP_γ **problem).** Given an *n*-dimensional lattice $\mathcal{L}$, the SIVP_γ problem find *n* linearly independent lattice vectors of length at most $\gamma \cdot \lambda_n(\mathcal{L})$.

2.3 Collision-Resistant Hash Functions

The collision-resistant hash function family $\mathcal{H}$ in definition 4 was introduced by Lyubashevsky and Micciancio[17].

Definition 4. For integer m and $D_\times \subseteq \mathcal{D}$, $\mathcal{H}(\mathcal{D}, D_\times, m) = \{h_{\hat{a}} : \hat{a} \in \mathcal{D}^m\}$ is the function family that for any $\hat{z} \in D_\times^m$, the equation $h_{\hat{a}}(\hat{z}) = \hat{a} \cdot \hat{z} = \sum a_i z_i$ holds where $\hat{a} = (a_1, \cdots, a_m)$ and $\hat{z} = (z_1, \cdots, z_m)$. Here, the inner products of $a_i z_i$ are operated in $\mathcal{D}$.

Note that, for any $\hat{y}$, $\hat{z} \in \mathcal{D}^m$ and $c \in \mathcal{D}$, hash functions in $\mathcal{H}(\mathcal{D}, D_\times, m)$ meet two conditions:

$$h(\hat{y} + \hat{z}) = h(\hat{y}) + h(\hat{z}) \tag{3}$$

$$h(\hat{y}c) = h(\hat{y})c \tag{4}$$

Besides, the function family is collision resistant when its domain is the set $D_\times^m \subset \mathcal{D}^m$. Given an element $h \in \mathcal{H}(\mathcal{D}, D_\times, m)$, the collision problem $Col(h, D_\times)$ (where $D_\times \subset \mathcal{D}$) asks to find two distinct elements $\hat{z}_1, \hat{z}_2 \in D_\times$ that $h(\hat{z}_1) = h(\hat{z}_2)$). Lyubashevsky and Micciancio showed that, when $D_\times$ was limited to a set of small norm polynomials, $Col(h, D_\times)$ was as hard as SVP_γ in the worst case over lattices that correspond to ideals in $\mathcal{D}$.

2.4 Statistical Distance

The statistical distance shows the difference between two probability distributions.

Definition 5 (Statistical Distance). Let *X* and *X'* be two random variables over a finite set *S*. The statistical distance between *X* and *X'* is defined by

$$\Delta(X, X') = \frac{1}{2} \sum_{x \in S} |\Pr[X = x] - \Pr[X' = x]|. \tag{5}$$

The following proposition suggests that the statistical distance will not increase by a randomized algorithm.

Proposition 1 (Micciancio and Goldwasser[18]**)** Let *X* and *X'* be two random variables come from a common set *S*. For any function *f* with domain *S*, the statistical distance between *f*(*X*) and *f*(*X'*) is at most

$$\Delta(f(X), f(X')) \le \Delta(X, X'). \tag{6}$$

That is to say, if the statistical distance between two random variables (X_λ) and (X_λ') is negligible, an attacker can only achieve a negligible advantage in distinguishing the distributions of (X_λ) and (X_λ') with a sample. In proposition 1, there is no assumption on the computational complexity *f*, so it holds no matter the attacker is computationally bounded or unbounded.

However, the statistical distance may grow when considering multiple variables. We can get the conclusion from definition 4 that, if *X*, *Y* come from a distribution ϕ and *X'*, *Y'* come from a distribution ϕ', the following inequality holds:

$$2\Delta(X, X') \ge \Delta((X, Y), (X', Y')) \ge \Delta(X, X'). \tag{7}$$

An attacker with many samples of the same distribution may be able to distinguish better than with one. Therefore, if the statistical distance of two random variables has a upper-bound of $\varepsilon(k)$, given *s* samples of the same distribution and, the attacker's advantage over a wild guess is bounded by $s \cdot \varepsilon(k)$.

3. Key Techniques

3.1 Message Block Sharing

The concept of message block sharing was introduced by Choi and Kim[16]. The main idea can be described as: first divide the original message into *d* message blocks $M_1,\cdots,M_d$ with random sizes, and then distribute the message blocks to each member in a uniform number *t*.

To achieve a *t*-out-of-*N* threshold signature scheme, the distribution of the message blocks cannot be shared randomly. The desired goal is that the alliance of less than *t* members in a group will not have all message blocks and any alliance of *t* members will get the whole message like threshold signature scheme.

Let u_i and Γ_i be the member of the group and the set of message blocks u_i received during the distribution procedure, respectively. Next, we first describe two-out-of-three message block sharing as a toy example, and then deduce *t*-out-of-*N* message block sharing technique from it.

Two-out-of-three Message Block Sharing. The original message can be divided into three message blocks M_1,M_2,M_3 and each member gets two message blocks, so that any two members can recover the whole message. If $\Gamma_1=\{M_1,M_2\}$, $\Gamma_2=\{M_2,M_3\}$, and $\Gamma_3=\{M_1,M_3\}$, one member does not have the whole message blocks and any two members can recover the original message. **Fig. 1** shows the process of two-out-of-three message block sharing.

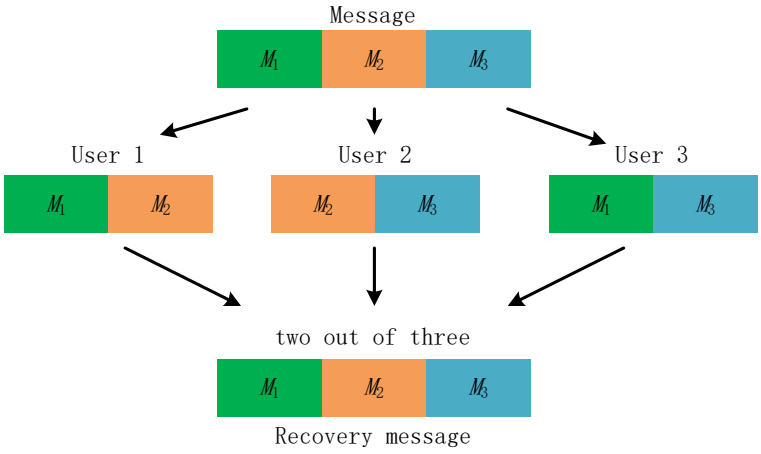

Fig. 1. Two-out-of-three message block sharing

***t*-out-of-*N* Message Block Sharing.** In the above instance, none of the message blocks is shared by all members, and none of the members has all message blocks. Moreover, for each message block M_j, one member does not have that block M_j. Now, we extend these facts to the *t*-out-of-*N* message block sharing case.

For each message block M_j, it is sufficient that $t-1$ members do not have that block. Besides, for each block M_j, to avoid the coalition of $t-1$ members recovering the full message, $t-1$ members should not contain that block. Another requirement for the message block sharing is, each member should have the same number of message blocks. This matter can be solved by finding all possible $t-1$-element subsets of *N*-element set, and for the rest of

the set $\{\alpha_1,\cdots,\alpha_{N-t+1}\}$, each message block is distributed to members $u_{\alpha_1},\cdots,u_{\alpha_{N-t+1}}$.

In fact, when the number of message blocks d is set to be C_N^{t-1}, we can get the desired message block sharing model. On the one hand, the alliance of at most $t-1$ members will not get the whole message as there will always exist exactly one message block missed during the distribution of message blocks. On the other hand, t members will always get the whole message since at least one member has each message block M_j. Moreover, each member has exactly $k=C_{N-1}^{t-1}$ message blocks. Therefore, to construct t-out-of-N message block sharing, the message is required to be large enough to be divided into d message blocks.

3.2 Message Preprocessing Technique: Pad-then-Permute

Choi and Kim employed a parameter extracting algorithm called *Threshold parameter extracting algorithm* which took the message and its bit length as inputs, and outputted (λ,N,t), where λ was the security parameter, N was the group size, and t was the threshold value of the number of members to generate the valid signature. This approach leads to a serious disadvantage when the signature scheme is implemented. The threshold parameter extracting algorithm requires a message at least d bits, so the threshold signature cannot work if the message was short. Besides, the bit length of the message to be signed determines the value of the threshold t and N. It is infeasible in practice because the threshold value is always set with the system parameter before generating a threshold signature for a message. To avoid these disadvantages, we employ a message preprocessing technique called Pad-then-Permute to make the threshold signature more practical.

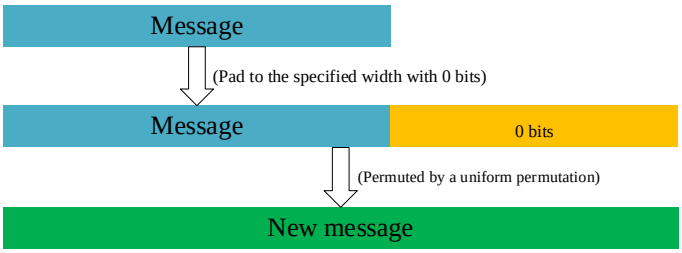

Fig. 2. Pad-then-permute technique

A brief introduction of this technique is shown in **Fig. 2**. In a t-out-of-N threshold signature scheme, the Pad-then-Permute algorithm first pads then permutes, and finally divides the message M into $d=C_N^{t-1}$ message blocks $\{M_1,\cdots,M_d\}$. Specific for, this technique first computes $w=\lceil g/d\rceil$ and $r=H_0(M)$, where g is the bit length of M, H_0 is a public hash function. Second, it pads M to $w\cdot d$ bits with zero bits and the result is denoted as M'. Then, the pad-then-permute algorithm randomly chooses a uniform permutation π from the $w\cdot d$ bits permutation set. Finally, this technique divides the message $\pi(M')$ into d message blocks $\{M_1,\cdots,M_d\}$. In a word, taking the system parameter and the message M as inputs, this technique outputs the relevant parameters as well as the message block sets that to be distributed to the members.

4. Syntax of Threshold Ring Signature

Hereinafter, we follow the syntax and security requirements for threshold ring signature in the work of Bettaieb and Schrek[14]. Assume that t users try to co-produce a signature in the name of N users anonymously, and we use $U=\{1,\cdots,N\}$ and S to denote the set of users and the set of t signers with the conditions of $S\subset U$, respectively. Each user $i\in U$ has a public/secret key pair (PK_i,SK_i). Generally, a threshold ring signature scheme includes three algorithms:

Setup: this algorithm takes the system security parameter as inputs, and outputs the public parameters and the threshold parameters (n,N,t).

T.RingKeyGen : this algorithm generates the key pair (PK_i,SK_i) for each user $i\in U$.

T.RingSign(M,U,S) : this algorithm is an interactive protocol between t users that takes a set of public keys of users in U, a set of t secret keys of users in S, and a message M as inputs, and outputs a t-out-of-N threshold ring signature σ on M.

T.RingVerify(M,σ,t,U) : this algorithm is a deterministic algorithm which outputs **accept** or **reject**.

Security requirements for a threshold signature scheme include indistinguishability source hiding and unforgeability.

Indistinguishability source hiding. A t-out-of-N threshold ring signature is indistinguishable source hiding if, for a probabilistic polynomial time adversary $\mathcal{A}$, the success probability of $\mathcal{A}$ is negligible in the following game:

a. For $i\in U$, the challenger generates (PK_i,SK_i), and sends the public keys $PK=\{PK_1,\cdots,PK_N\}$ to $\mathcal{A}$. Besides, a signing oracle **OT.Sign**$(\cdot)$ is allowed to be accessed by $\mathcal{A}$, which returns $\sigma=$ **T.Sign**(M,PK,S), where S denotes the set of signers.

b. $\mathcal{A}$ outputs a message M, two distinct sets $\{PK_{i_{1,0}},\cdots,PK_{i_{t,0}}\}$ and $\{PK_{i_{1,1}},\cdots,PK_{i_{t,1}}\}$ with $PK_{i_{l,j}}\in PK$ for $l\in\{0,1\}$, $j\in\{1,\cdots,t\}$. $\mathcal{A}$ can access to $\{SK_1,\cdots,SK_N\}\setminus\{SK_{i_{1,0}},\cdots,SK_{i_{t,0}}\}$. Finally, the challenger chooses a random bit b and returns $\sigma\leftarrow$ **T.Sign**$(\mu,P,\{SK_{i_{1,b}},\cdots,SK_{i_{t,b}}\})$ to $\mathcal{A}$.

c. $\mathcal{A}$ outputs a bit b', and succeeds if $b=b'$.

Unforgeability. A threshold ring signature scheme is unforgeable under a chosen message attack if, for any probabilistic polynomial time forger $\mathcal{F}$, the success probability of $\mathcal{F}$ is negligible in the following game:

a. The challenger $\mathcal{C}$ generates key pairs $\{PK_i,SK_i\}_{i=1}^N$, and sends the set of public keys $PK=\{PK_i\}_{i=1}^N$ to $\mathcal{F}$.

b. $\mathcal{F}$ can access to **OT.Sign**$(\cdot)$.

c. $\mathcal{F}$ can make corruption queries that return secret keys SK_i on input i.

d. $\mathcal{F}$ outputs a t-out-of-N threshold ring signature σ^* for a new message M^*.

$\mathcal{F}$ succeeds if all of the following conditions are satisfied:

a. **T.Verify**$(M^*, \sigma^*, t, PK) = 1$.

b. $\mathcal{F}$ never asked M^* in a signing query.

c.The number of corruption queries is strictly less than $t-1$.

5. Our Construction

5.1 Verbal Description

We show how to construct a threshold ring signature scheme by the message block sharing technique. Our key generation algorithm comes from Lyubashevsky's work[19]. The signer has a secret signing key $\hat{s}$ and a public verification key (h, C) such that $h(\hat{s}) = C$.

Before generating a signature, the system first generates public parameters for the threshold ring signature scheme, and then preprocesses the message *M*. Assume that the message *M* is divided into *d* blocks $M = \{M_1, \cdots, M_d\}$ for $d = C_N^{t-1}$, and $\Gamma_i = \{M_{i_1}, \cdots, M_{i_k}\}$ is distributed to *N* users where $k = C_{N-1}^{t-1}$. In order to coproduce a signature on message *M*, *t* users (include a leader *L* among them) form a ring. Here, we denote the set of their identity indexes by *S*. They proceed as follows:

a. Each signer $j \in S$ chooses a random vector of polynomials $\hat{y}_j$ in secret, computes and sends $h_i(\hat{y}_i)$ to *L*.

b. Upon receiving *t* values of $h_j(\hat{y}_j)$ for $j \in S$, leader *L* chooses random vectors of polynomials $\hat{y}_i$ for each $i \in U \setminus S$. Finally, *L* computes and publishes $\Psi = \sum_{i \in U} h_i(\hat{y}_i)$ to *t* signers.

c. For *t* signers, each of them combines his message blocks to get $M[j] = M_{j_1} \| \cdots \| M_{j_k}$, and then computes $e_j = H(\Psi, M[j], r)$. Finally, they output and send $(\hat{z}_j, e_j)$ to *L*, where $\hat{z}_j = \hat{s}_j e_j + \hat{y}_j$.

d. *L* sets $\hat{z}_j = \hat{y}_j$ for the rest *N*-*t* users, and outputs $\sigma = \{(\hat{z}_1, \cdots, \hat{z}_N), (e_j)_{j \in S}\}$ as the signature on *M*.

For each Γ_j, the verifier recombines $M[j] = M_{j_1} \| \cdots \| M_{j_k}$, and checks if $e_j = H\left(\sum_{i \in U} h_i(\hat{z}_i) - C \cdot \sum e_j, M[j], r\right)$. This is true for a correct signature because the linearity of hash function h_j has the property of $h_j(\hat{z}_j) - Ce_j = h_j(\hat{s}_j e_j + \hat{y}_j) - Ce_j = h_j(\hat{y}_j)$.

As the growth of ring size will make forgery attacks easier, there is a constant *c* that acceptable the ring size are bounded by λ^c, where λ is security parameter. Since the sizes of the signature and the verification key grow with the ring size, the implementer can replace *c* by 1 or 2 which will satisfy any reasonable applications. To resist the attacks on malicious chosen parameters, our threshold ring signature algorithm takes an initial step in which the inputs are required to pass simple tests (it is similar to the ring signature scheme proposed by Mechor *et al.*[15]).

5.2 Formal Description

Algorithm 1 Setup.

Description: Given the security parameters λ, N, and t, this algorithm determines the common public parameters.

Input: λ, N, t,

1. Compute $d = C_N^{t-1}$.
2. Set n as a power of two larger than λ.
3. Set $m = 3\log n$, and p as a prime larger than n^4 such that p=3mod 8.

--Note: these parameters define the sets $\mathcal{D}$, D_h, D_z, D_y, $D_{s,c}$, and the family $\mathcal{H}$.

$$\mathcal{D} = \mathbb{Z}_p[x] / \left\langle x^n + 1 \right\rangle,$$

$$D_h = \{g \in \mathcal{D} : \| g \|_\infty \le mn^{1.5} \log n + \sqrt{n} \log n\},$$

$$D_y = \{g \in \mathcal{D} : \| g \|_\infty \le mn^{1.5} \log n\},$$

$$D_z = \{g \in \mathcal{D} : \| g \|_\infty \le mn^{1.5} \log n - \sqrt{n} \log n\},$$

$$D_{s,c} = \{g \in \mathcal{D} : \| g \|_\infty \le 1\}.$$

4. Set $C \leftarrow \mathcal{D}$, $C \neq 0$.
5. $H_0 : \{0,1\}^* \to D_{s,c}$ is a public collision-resistant hash function that will be employed to verify the integrity of the message.
6. $H' : \{0,1\}^* \to D_{s,c}$, $H : \{0,1\}^* \to D_{s,c}$ are two collision-resistant hash functions, which simulate as random oracles in the proof of security.

Output: $P_1 = \{\lambda, t, N, d, n, m, p, C, H_0 H, H'\}$.

Algorithm 2 T.RingKeyGen.

Description: Generate a keypair for user *i*.

Input: P_1.

1. Set $\hat{s}_i = (s_{i,1}, \cdots, s_{i,m}) \leftarrow \mathcal{D}_{s,c}^m$.
2. If none of $s_{i,j}$ is invertible, go to step 1.
3. Let $j_0 \in \{1, \cdots, m\}$ such that s_{i,j_0} is invertible.
4. $(a_{i,1}, \cdots, a_{i,j-1}, a_{i,j+1}, \cdots, a_{i,m}) \leftarrow \mathcal{D}^{m-1}$
5. Let $a_{i,j_0} = s_{i,j_0}^{-1}(C - \sum_{j \neq j_0} a_{i,j} s_{i,j})$ and denote $\hat{a}_i = (a_{i,1}, \cdots, a_{i,m})$.

Output: $(PK_i, SK_i) = (h_i, \hat{s}_i)$, h is the hash function in $\mathcal{H}$ defined by $\hat{a}_i$.

Algorithm 3 Message Processing.

Description: Given a message M, this algorithm first pads, then permutes, and finally divides M into d message blocks $\{M_1, \cdots, M_d\}$.

Input: P_1, M.

1. Compute $w = \left\lceil \frac{g}{d} \right\rceil$ and $r = H_0(M)$, where g is the bit length of M, and H_0 is a public hash function.

2. Pad M to $w \cdot d$ bits with zero bits, denote the resulting bit string as M'.
3. Randomly choose a uniform permutation π from the $w \cdot d$ bits permutation set.
4. Compute and divide $\pi(M')$ into d message blocks $\{M_1, \cdots, M_d\}$.

Output: $P_2 = \{\pi, r\}$, and distribute the corresponding message block set $\Gamma_i = \{M_{i_1}, \cdots, M_{i_k}\}$ to user i for $k = C_{N-1}^{t-1}$.

Note: The users must keep their message blocks in secret.

Algorithm 4 T.RingSign.

Description: Given message block sets $\Gamma_j = \{M_{j_1}, \cdots, M_{j_t}\}$, t signers whose identity indexes $j \in S$ coproduce a signature on M under the name of U such that $S \subset U$.

Input: P_1, P_2, SK_i, M

1. Verify that the public parameters indeed meet the constraints of steps 1-3 in algorithm **T.RingKeyGen**, and SK_i is in $\mathcal{D}_z^m$, if not, output **failed**.
2. Each signer $j \in S$ randomly chooses $\hat{y}_j \leftarrow \mathcal{D}_y^m$, then computes and sends $h_j(\hat{y}_j)$ to L (one of the singers).
3. Upon receiving the values $h_j(\hat{y}_j)$ from signers $j \in S$, L first chooses random vectors of polynomials $\hat{y}_i \leftarrow \mathcal{D}_y^m$ for each $i \in U \setminus S$, and then computes $\Psi = \sum_{i \in U} h_i(\hat{y}_i)$ and $y' = H'(h_1(\hat{y}_1), \cdots, h_N(\hat{y}_N), \Psi)$. Afterwards, L shares $h_1(\hat{y}_1), \cdots, h_N(\hat{y}_N)$, Ψ, and y' with the rest $t-1$ signers.
4. Upon receiving Ψ, each signer j ensures his $h_j(\hat{y}_j)$ is in $h_1(\hat{y}_1), \cdots, h_N(\hat{y}_N)$, and checks the correctness of the value of Ψ and y'. Then, he sets $M[j] = M_{j_1} \| \cdots \| M_{j_k}$, and computes $e_j = H(\Psi, M[j], y', r)$ (e_j is therefore in $\mathcal{D}_{s,c}$), $\hat{z}_j = \hat{s}_j e_j + \hat{y}_j$.
5. If $\hat{z}_j \notin D_z^m$, go to Step 2.
6. Finally, the signer $j \in S$ sends $\hat{z}_j$ and (e_j, Γ_j) to L.
7. L recombines the original message $\pi(M')$ by Γ_j and recovers M, then he checks if $r = H_0(M)$, if not, output **failed**.
8. L sets $\tilde{e} = \{e_j\}_{j \in S}$ and $\hat{z}_i = \hat{y}_i$ for $i \in U \setminus S$, and combines $\tilde{z} = (\hat{z}_1, \cdots, \hat{z}_N)$ in order.

Output: $\sigma = \{\tilde{z}, \tilde{e}, y'\}$.

Algorithm 5 T.RingVerify.

Description: Given public parameters, public key set, message block sets $\{\Gamma_j\}_{j \in S}$, and a signature on a message M, output 1 to **accept** or 0 to **reject** the signature.

Input: $P_1, P_2, \{\Gamma_j\}_{j \in S}, \sigma = \{\tilde{z}, \tilde{e}, y'\}$.

The verifier first parses the signature $\sigma = \{\tilde{z}, \tilde{e}, y'\}$ into $\tilde{z} = (\hat{z}_1, \cdots, \hat{z}_N)$, $\tilde{e} = \{e_j\}_{j \in S}$ and y', and then checks as follows:

1. Recombine the original message $\pi(M')$ by Γ_j and recover M, check if $r = H_0(M)$.
2. $\hat{z}_i \in \mathcal{D}_z^m$ for all i that $PK_i \in U$.

3. For each $j \in S$, the verifier sets $M[j] = M_{j_1} \| \cdots \| M_{j_k}$, and checks if $e_j = H(\sum_{i \in U} h_i(\hat{z}_i) - C \cdot \sum_{j \in S} e_j, M[j], y', r)$.

Output: If all the conditions are satisfied, the verifier output 1, otherwise 0.

5.3 Correctness and Efficiency

Assume that $\sigma = \{\tilde{z}, \tilde{e}, y'\}$ is a signature for *M*, the message block sharing technique ensures that the message must come from at least *t* signers from the ring, otherwise the message cannot be integrally composed, so a legally generated signature must satisfy the first condition. The second test is always passed as step 2 and step 5 guarantee the signature only contain elements in D_z^m. In terms of the third test,

$$\begin{aligned}
\sum_{i \in U} h_i(\hat{z}_i) - C \cdot \sum_{j \in S} e_j &= \sum_{i \in U} h_i(\hat{z}_i) - C \cdot \sum_{j \in S} e_j \\
&= \sum_{i \in U \setminus S} h_i(\hat{z}_i) + \sum_{j \in S} h_j(\hat{z}_j) - C \cdot \sum_{j \in S} e_j \\
&= \sum_{i \in U \setminus S} h_i(\hat{y}_i) + \sum_{j \in S} h_j(\hat{s}_j e_j + \hat{y}_j) - C \cdot \sum_{j \in S} e_j \\
&\quad \textit{by replacing } \hat{z}_i \textit{ by } \hat{y}_i \textit{ and } \hat{s}_j e_j + \hat{y}_j \textit{ for} \\
&\quad i \in U \setminus S \textit{ and } j \in S, \textit{respectively.} \\
&= \sum_{i \in U \setminus S} h_i(\hat{y}_i) + \sum_{j \in S} (C \cdot e_j + h_j(\hat{y}_j)) - C \cdot \sum_{j \in S} e_j \\
&\quad \textit{using the homomorphic properties of} \\
&\quad h_i \in \mathcal{H}, \textit{and } h_i(\hat{s}_i) = C. \\
&= \sum_{i \in U \setminus S} h_i(\hat{y}_i) + \sum_{j \in S} h_j(\hat{y}_j) \\
&= \sum_{i \in U} h_i(\hat{y}_i).
\end{aligned}$$

So the third test of **T.RingVerify** always holds for a valid signature.

Therefore, a correctly created signature always passes the verification. The following proposition shows the computational costs of algorithms in the proposed scheme. Before it, we first give a lemma that will be used in the proof of Proposition 1.

Lemma 1. Let $D_{s,c}^{\times}$ denote the set of non-invertible polynomials of $D_{s,c}$. We have $\Pr_{f \leftarrow D_{s,c}^{\times}}[f \in D_{s,c}^{\times}] \leq \frac{2}{3^{n/2}}$.

Theorem 1. The expected running times of **Setput**, **T.RingKeyGen**, **T.RingSign**, and **T.RingVerify** are polynomial in the security parameter.

Proof. The proof line comes from proposition 2 in the work of Mechor *et al*[15]. Firstly, we know that all the computations in these algorithms can be executed in polynomial time in the security parameter. Secondly, **Setup** and **T.RingVerify** can be operated in polynomial time, and we only need to consider the iterations of **T.RingKeyGen** and **T.RingSign**.

Lemma 3 tells us that, each of the polynomials chosen in step 1 of algorithm **T.RingKeyGen** is invertible with probability exponentially close to one and thus the expected iteration is approximately one. So, the expected running time of **T.RingKeyGen** is polynomial.

Step 1 in algorithm **T.RingSign** has a polynomial amount of iterations, and the loop between steps 2 and 5 which will continue as long as $\hat{z}_i = \hat{s}_i \cdot e_i + \hat{y}_i \notin D_z^m$. Corollary 6.2 in [19]

tells us that, for any $\hat{s} \in D_s^m$, $\Pr_{c \leftarrow D_{s,c}, \hat{y} \leftarrow D_y^m}[\hat{s}c + \hat{y} \in D_z^m] = e^{-1} - o(1)$. In our scheme, e and $\hat{y}_j$ are drawn uniformly from $D_{s,c}$ and D_y^m, by using the above result, the expected iteration number of **T.RingSign** is less than 3, so **T.RingSign** also runs in expected polynomial time.

Above all, our scheme is not very complex, so it requires low computational costs. The computation of algorithm 1 mainly includes some parameters sampling and hash functions choosing. There only needs one vector sampling and one inner product of vector in algorithm 2. In algorithm 3, the computation includes bits-pad and an uniform permutation. The computation of algorithm 4 mainly contains two hash operation, several vector sampling and operations. The computation of algorithm 5 also includes several hash and vector operations in it.

6. Security Analysis

Theorem 2. Under the assumption that Mechor *et al.*'s ring signature scheme is anonymous, the proposed threshold ring signature is statistically indistinguishable source hiding.

Proof. In the indistinguishable source hiding game, the adversary can access a signature which depends on a random bit b as well as on the system public parameters P_1, two distinct secret key sets $S\hat{K}_0 = \{SK_{i_{1,0}}, \cdots, SK_{i_{t,0}}\}$, $S\hat{K}_1 = \{SK_{i_{1,1}}, \cdots, SK_{i_{t,1}}\}$, and a message M. These parameters are known to the adversary except $\{SK_{i_{1,0}}, \cdots, SK_{i_{t,0}}\}$ and the random bit b. Let $X_{b,P,S\hat{K}_b,M}$ be the random variable that represents the signature obtained by the adversary for a given set of parameters. Similar to the anonymity proof of Mechor *et al.*'s ring signature scheme, we can regard $S\hat{K}_b$ as the sk_{i_b} in theorem 2 (Mechor *et al.*[15]).

Next, we show the difference between our threshold ring signature from an existing ring signature (Mechor *et al.*[15]). In Mechor *et al.*'s ring signature scheme, $\hat{z}$ in our signature has one component generated by $\hat{z}_i \leftarrow \hat{s}_i \cdot e_i + \hat{y}_i$ with $\hat{s}_i$ the secret key, $\hat{y}_i$ randomly chosen from D_y^m, and e_i computed by the hash function H. Other components are randomly chosen from D_z^m. While in our signature, $\tilde{z}$ in our signature has t components generated by $\hat{z}_i \leftarrow \hat{s}_i \cdot e_i + \hat{y}_i$ with $\hat{s}_i$ the secret key, $\hat{y}_i$ randomly chosen from D_y^m, and e_i computed by the hash function H. Other N-t components are randomly chosen from D_z^m.

Theorem 6.5 in Lyubashevsky's work[19] stated that, for any h chosen from the hash family $\mathcal{H}$, message M, and any two private keys $\hat{s}, \hat{s}' \in D_s^m$ such that $h(\hat{s}) = h(\hat{s}')$, $\Delta((\hat{z}, e), (\hat{z}', e')) = n^{-\omega(1)}$ holds for random variables $\hat{z}$ ($\hat{z}'$) and e (e'), where e (e') is the output of algorithm $Sign(M, h, \hat{s})$ ($Sign(M, h, \hat{s}')$), respectively. Let $X_{b,P,sk_{i_b},\mu,R}$ be the random variable describing the output of **Ring - sign**(P, sk_{i_b}, μ, R) with $P_1 = \{\lambda, t, N, d, n, m, p, C, H_0 H, H'\}$, $P_2 = \{\pi, r\}$, sk_{i_b}, μ, and R as inputs to the algorithm. From Mechor *et al.*'s (2013) work, if the domains of these variables are both different from $\{failed\}$ we have $\Delta(X_{0,P,sk_{i_0},\mu,R}, X_{1,P,sk_{i_1},\mu,R}) = n^{-\omega(1)}$ for $b \in \{0,1\}$. Iterating this result for t-1 times in our scheme, we can get the conclusion that $\Delta(X_{0,P,S\hat{K}_{i_0},M}, X_{1,P,S\hat{K}_{i_1},M}) = n^{-\omega(1)}$.

Theorem 3. Suppose there exists a polynomial time forger $\mathcal{F}$ who makes at most $t-1$ corruption queries and can output a valid forgery of the proposed threshold ring signature scheme with probability of ε. By employing the power of $\mathcal{F}$, we can construct an algorithm $\mathcal{B}$ that outputs a forgery of the underlying scheme with probability at least $(N-t+1)t\varepsilon / N^2$.

Proof. Assume that there exists a forger $\mathcal{F}$ that can output a forgery of the proposed threshold ring signature scheme with non-negligible advantage ε. By employing $\mathcal{F}$, we construct a polynomial time algorithm $\mathcal{B}$ that outputs a forgery for Lyubashevsky's scheme with non-negligible advantage.

$\mathcal{B}$ receives as input a public key PK^*. $\mathcal{B}$ employs $\mathcal{F}$ with input public keys $PK=\{PK_1,\cdots,PK_N\}$, which is generated as follows. $\mathcal{B}$ chooses an index $i^* \leftarrow \{1,\cdots,N\}$ and sets $PK_{i^*}=PK^*$. Other public keys are generated directly by algorithm **T.RingKeyGen** with their corresponding private key $SK_{i\neq i^*}$. Note that, parameter S in the underling scheme (Lyubashevsky[19]) is denoted by C to avoid confusion with the set of signer. $\mathcal{B}$ simulates oracle queries of $\mathcal{F}$ as follows:

1. When $\mathcal{F}$ makes a signature queries on message M, $\mathcal{B}$ generates a response by running the **T.RingSign** algorithm with arbitrary t signers whose identity index $i \neq i^*$.
2. To reply consistently, $\mathcal{B}$ maintains a list L' in a form of $\left(h_1(y_1),\cdots,h_N(y_N),\sum_{i\in U}h_i(y_i);y'\right)$. When $\mathcal{F}$ makes queries with input as $\left(h_1(y_1),\cdots,h_N(y_N)\right)$ to random oracle H', $\mathcal{B}$ first looks it up in L'. If it exists, returns y'; if not, generates $y' \leftarrow D_{s,c}$ at random, and stores $\left(h_1(y_1),\cdots,h_N(y_N),\sum_{i\in U}h_i(y_i);y'\right)$ in L'.
3. Similarly, $\mathcal{B}$ need to maintain another list L in a form of $\left(\sum_{i\in U}h_i(y_i),M[j],y',r;e_j\right)$ to reply consistently When $\mathcal{F}$ makes queries with input as $\left(\sum_{i\in U}h_i(y_i),M[j],y',r\right)$ to random oracle H, $\mathcal{B}$ first searches $\left(\sum_{i\in U}h_i(y_i),M[j],y',r;e_j\right)$ in L, if it exists, returns e_j; if not, it searches y' in L'. If it exists, $\mathcal{B}$ makes queries with input $\left(h_{i*}(y_{i*}),M[j]\|r\right)$ to the random oracle of the underlying signature, adds the answer in L and honestly returns it to $\mathcal{F}$; otherwise, $\mathcal{B}$ chooses $e_j \leftarrow D_{s,c}$ at random, and stores it in L.
4. $\mathcal{B}$ faithfully answers any corruption query that is submitted by $\mathcal{F}$ for a user $i \neq i^*$. If $\mathcal{F}$ makes a corruption query for i^*, $\mathcal{B}$ simply aborts. Note that, $\mathcal{F}$ can totally make $t-1$ corruption queries.

At a given point, $\mathcal{F}$ finishes running and outputs a valid forgery $\sigma=\{\tilde{z},\tilde{e},y'\}$ on message M, where $\tilde{z}=(\hat{z}_1,\cdots,\hat{z}_N)$, $\tilde{e}=\left\{e_j\right\}_{j\in S}$. Note that, $\mathcal{F}$ never make signing queries on M. Next, we will show that, from this forgery on the proposed threshold signature, $\mathcal{B}$ can output a valid forgery of the underlying signature. Observe that $\mathcal{F}$ must make a random oracle query to get

y', so $\left(h_1(y_1),\cdots,h_N(y_N),\sum_{i\in U}h_i(y_i);y'\right)$ must be stored in L'. $\mathcal{B}$ checks if $h_{i^*}(\hat{z}_{i^*})=h_{i^*}(\hat{y}_{i^*})$, if so, $\mathcal{B}$ outputs "*failed*"(this occurs in a probability of $1-\frac{t}{N}$); if not, it reveals that i^* is one of t signers in the forgery and one of $\{e_j\}_{j\in S}$ is used in generating $\hat{z}_{i^*}$. So, $\mathcal{B}$ finds it out by checking that $h_{i^*}(\hat{z}_{i^*})=h_{i^*}(\hat{y}_{i^*})+C\cdot e_j$ and denotes it as e_{i^*}. Finally, $\mathcal{B}$ searches $\left(\sum_{i\in U}h_i(y_i),M[i^*],y',r;e_{i^*}\right)$ and outputs $\left(\hat{z}_{i^*},e_{i^*}\right)$ as a forgery on message $\left(M[i^*]\|r\right)$. Seeing that the abort probability of $\mathcal{B}$ is $\frac{t-1}{N}$, and the probability of failure is $1-\frac{t}{N}$. Therefore, if $\mathcal{F}$ can output a valid forgery of the proposed threshold ring signature scheme with probability of ε, by employing the power of $\mathcal{F}$, we can construct an algorithm $\mathcal{B}$ to output a forgery of the underlying scheme with probability of $\left(1-\frac{t-1}{N}\right)\cdot\frac{t}{N}\cdot\varepsilon=(N-t+1)t\varepsilon/N^2$. This completes the proof.

7. Analysis

In **Table 1**, we analyze the security of our scheme and other lattice-based threshold signature schemes. Feng *et al.*'s[11] scheme requires a sequential signing procedure, and thus each member cannot generate their own signature simultaneously. Moreover, their scheme is based on the standard NTRU lattice and the variation of CVP, thus can be broken. Cayrel *et al.*'s[10] proposal gives a ring threshold signature by modifying the threshold signature scheme based on the syndrome decoding problem with identification scheme, but this scheme is based on SIS problem which is weaker than LWE problem. Bendlin *et al.*'s[12] work is a threshold signature scheme based on LWE problem but conceptually hard to understand. The scheme proposed by Choi and Kim[16] is based on LWE problem and there is no known attack and the scheme is conceptually simpler. However, their scheme generates the group size after we get the message *M*, this would be inefficient in some situations. Our scheme is based on ideal lattice, there is no known attack so far. Like Choi and Kim's work, we do not adopt any trapdoor functions as Bendlin *et al.* did, but divide the message into blocks. Besides, our scheme generates signature concurrently without any sequence.

Table 1. Threshold Signatures on Lattices

Scheme	Hard Problem	Security	Known attack	Key Idea
(Feng *et al.*, 2010)	NTRU lattice	CVP	Yes	Sequential signing
(Cayrel *et al.*, 2010)	Ideal lattice	SIS	No	Syndrome decoding
(Bendlin *et al.*, 2013)	Lattice	LWE	No	Trapdoor share
(Choi and Kim, 2014)	Lattice	LWE	No	Message block
This work	Ideal lattice	SVP	No	Message block

Table 2 shows the signature sizes of our scheme in different values of *t* and *N*. Compared with previous works, our threshold scheme has considerably short signature sizes for a same *N*, and the signature size does not depend so much on the parameter *t* for the given parameters. Here, we use parameters used in subsection 5.1 (Cayrel *et al.*[10]) and subsection 6.3 (Bettaieb and Schrek[14]) to compare the performance with these schemes. The parameters are set n=64, m=2048, q=257 and the length of the commitment of COM is 224 bits for bit-security equals to

111. Note that, in our construction, the signature consists of two parts, each component in the first part is in D_z^m and each component of the second part is in $D_{s,c}$. (Parameter setups are in Algorithm 1, $D_z = \{g \in \mathcal{D} : \| g \|_\infty \leq mn^{1.5} \log n - \sqrt{n} \log n\}$, $D_{s,c} = \{g \in \mathcal{D} : \| g \|_\infty \leq 1\}$).

Table 2. Comparison of Lattice-based Threshold Ring Signature Schemes in Mbytes

N	*t*	(Cayrel *et al*., 2010)100	(Bettaieb and Schrek, 2013)	This work
100	2	24.43	0.52	0.0351
100	10	24.43	2.56	0.0352
100	50	48.85	12.80	0.0355
200	2	48.85	0.54	0.0702
200	10	48.85	2.68	0.0703
200	50	48.85	13.39	0.0706
1000	2	244.24	0.73	0.3509
1000	10	244.24	3.63	0.3510
1000	50	244.24	18.11	0.3513

When we compare other signature schemes in threshold setting based on other problems such as Discrete Logarithm Problem (DLP) (Boldyreva[20]; Chen *et al*.[21]) and Integer Factorization Problem (IFP) (Shoup[22]), our scheme takes more operations than other schemes from the view of computations. Superior to schemes based on DLP and IFP, which can be broken by quantum computer attack (Shor[23]), our scheme can resist the quantum attacks so far.

8. Conclusion

Based upon the lattice-based ring signature proposed by Melchor *et al*[15], this work constructs a lattice-based threshold ring signature scheme by the technique of message block sharing. To solve the system parameter setup problem and make the threshold ring signature scheme more flexible in practice, we preprocess the message by a new technique called "pad-then-permute" before blocking the message. The proposed scheme has considerably short signature size, with hardly any increase with its threshold value. Moreover, our scheme inherits the quantum immunity from lattice structure, and is proved to be correct, efficient, indistinguishable source hiding, and unforgeable. We will research threshold ring signature with message block sharing based on other advaced cryptographic systems as the future work.

References

[1] M. Ajtai, "Generating Hard Instances of Lattice Problems," in *Proc. of The twenty-eight annual ACM Symposium on Theory of Computing*, pp. 99-108, May 22 - 24, 1996. Article (CrossRef Link)

[2] R.L. Rivest, A. Shamir, Y. Tauman, "How to Leak a Secret," *ASIACRYPT 2001*, pp. 552-565, Dec 9 - 13, 2001. Article (CrossRef Link).

[3] E. Bresson, J. Stern, M. Szydlo, "Threshold Ring Signatures and Applications to Ad-Hoc Groups," *CRYPTO 2002*, pp. 465-480, Aug 18 - 22, 2002. Article (CrossRef Link)

[4] J.K. Liu, V.K. Wei, D.S. Wong, "A Separable Threshold Ring Signature Scheme," *Information Security and Cryptology - ICISC 2003*, pp.12-26, Nov 27-28, 2003. Article (CrossRef Link)

[5] S.S.M. Chow, L.C.K. Hui, S.M. Yiu, "Identity Based Threshold Ring Signature," *Information Security and Cryptology – ICISC 2004*, pp.218-232, Dec 2-3, 2004. Article (CrossRef Link)

[6] J.K. Liu, D.S. Wong, "On the Security Models of (Threshold) Ring Signature Schemes," *Information Security and Cryptology – ICISC 2004*, pp.204-217, Dec 2-3, 2004. Article (CrossRef Link)
[7] S. Chang, D.S. Wong, Y. Mu, Z. Zhang, "Certificateless Threshold Ring Signature," *Information Sciences*, vol. 179, no. 20, pp. 3685–3696, September, 2009. Article (CrossRef Link)
[8] C.A. Melchor, P.L. Cayrel, P. Gaborit, F. Laguillaumie, "A New Efficient Threshold Ring Signature Scheme Based on Coding Theory," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4833-4842, July, 2011. Article (CrossRef Link)
[9] P.P. Tsang, M.H. Au, J.K. Liu, W. Susilo, D.S. Wong, "A Suite of Non-pairing ID-Based Threshold Ring Signature Schemes with Different Levels of Anonymity (Extended Abstract)," *Provable Security-ProvSec 2010*, pp.166-183, Oct 13-15, 2010. Article (CrossRef Link)
[10] P.L. Cayrel, R. Lindner, M. Rückert, R. Silva, "A Lattice-Based Threshold Ring Signature Scheme," *Progress in Cryptology – LATINCRYPT 2010*, pp.255-272, Aug 8-11, 2010. Article (CrossRef Link)
[11] T. Feng, Y. Gao, J. Ma, "Changeable Threshold Signature Scheme Based on Lattice Theory," in *Proc. of International Conference on E-Business and E-Government(ICEE)*, pp.1311-1315, May 7-9, 2010. Article (CrossRef Link)
[12] R. Bendlin, S. Krehbiel, C. Peikert, "How to share a lattice trapdoor: threshold protocols for signatures and (H)IBE," *Applied Cryptography and Network Security-ACNS 2013*, pp.218-236, June 25-28, 2013. Article (CrossRef Link)
[13] P.Q. Nguyen, O. Regev, "Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures," *Journal of Cryptology*, vol. 22, no. 2, pp. 139–160, April, 2009. Article (CrossRef Link)
[14] S. Bettaieb, J. Schrek, "Improved Lattice-based Threshold Ring Signature Scheme," *PQCrypto 2013*, pp.34-51, June 4-7, 2013. Article (CrossRef Link)
[15] C.A. Mechor, S. Bettaie, X. Boyen, L. Fousse, P. Gaborit, "Adapting Lyubashevsky's Signature Schemes to the Ring Signature Setting," *AFRICACRYPY'13*, pp.1-25, June 22-24, 2013. Article (CrossRef Link)
[16] R. Choi, K. Kim, "Lattice-based Threshold Signature with Message Block Sharing," in *Proc. of The 31st Symposium on Cryptography and Information Security*, pp.1-7, Jan 21-24, 2014.
[17] V. Lyubashevsky, D. Micciancio, "Generalized Compact Knapsacks are Collision Resistant," *ICALP 2006*, pp.144-155, July 10-14, 2006. Article (CrossRef Link)
[18] D. Micciancio, S. Goldwasser, "Complexity of Lattice Problems: A Cryptographic Perspective," *The Kluwer International Series in Engineering and Computer Science*, vol. 671. Kluwer Academic Publishers, Boston, 220 pages, March, 2002.
[19] V. Lyubashevsky, Towards Practical Lattice-Based Cryptography, PhD thesis, University of California, San Diego, 2008.
[20] A. Boldyreva, Threshold Signatures, "Multisignatures and Blind Signatures Based on the Gap Diffie-Hellman-Group Signature Scheme," *Public key cryptography-PKC 2003*, pp.31-46, Jan 6-8, 2003. Article (CrossRef Link)
[21] X. Chen, F. Zhang, D.M. Konidala, K. Kim, "New ID-based Threshold Signature Scheme from Bilinear Pairings," *Progress in Cryptology-INDOCRYPT 2004*, pp.371-383, Dec 20-22, 2004. Article (CrossRef Link)
[22] V. Shoup, "Practical threshold signatures," *Advances in Cryptology-EUROCRYPT 2000*, pp. 207-220, May 14-18, 2000. Article (CrossRef Link)
[23] P.W. Shor, "Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM journal on computing*, vol. 26, no. 5, pp. 1484-1509, July, 2006. Article (CrossRef Link)
[24] A. PetzoldtEmail, S. Bulygin, J. Buchmann, "A multivariate based threshold ring signature scheme," *Applicable Algebra in Engineering, Communication and Computing*, vol. 24, no. 3-4, pp. 255-275, August 2013. Article (CrossRef Link)
[25] G. Zhou, P. Zeng, X. Yuan, S. Chen, K. R. Choo, "An Efficient Code-Based Threshold Ring Signature Scheme with a Leader-Participant Model," *Security and Communication Networks*, vol.

2017, Article ID 1915239, 7 pages, August, 2017. Article (CrossRef Link)
[26] J. Howe, T. Pöppelmann, M. O'neill, E. O'sullivan, T. Güneysu, "Practical Lattice-Based Digital Signature Schemes," *ACM Transactions on Embedded Computing Systems*, vol. 14, no. 3, May 2015. Article (CrossRef Link)
[27] J. Wang, S. Liu, H. Song, "Fractal Research on the Edge Blur Threshold Recognition in Big Data Classification," *Mobile Networks and Applications*, vol. 23, no. 2, pp. 251–260, April, 2018. Article (CrossRef Link).
[28] E. Baccarelli, N. Cordeschi, A. Mei, M. Panella, M. Shojafar, J. Stefa, "Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study," *IEEE Network*, vol. 30, no. 2, pp. 54-61, March-April, 2016. Article (CrossRef Link).

Jiangshan CHEN is a PhD student in Xidian University. He received his BS in college of science from Xidian University, China in 2003 and MS in college of mathematics and statistics from Minnan Normal University, China in 2012. His research interests include public key cryptography and network security.
(Email:JSChen@mnnu.edu.cn)

Yupu HU is a professor and PhD supervisor of the School of Tele-communications Engineering, Xidian University, China. He received his PhD in cryptography from Xidian University, China in 1999, and received his MS and BS in mathematics from Xidian University, China in 1987 and 1982, respectively. His main research interests include public key cryptography based on lattices and the analysis and application of fully homomorphic encryption schemes.
(Email:yphu@mail.xidian.edu.cn)

Wen GAO is a lecturer in college of electrical and information engineering of Shaanxi University of Science and Technology, China. She received her PhD in cryptography from Xidian University, China in 2017, and received her BS in Electronic Information Engineering from Henan University of Technology, China in 2011. Her research interests include lattice-based cryptography.
(Email: gaowen@sust.edu.cn)

Hongmei LIANG is a experimenter in college of mathematics and statistics of Minnan Normal University, China. She received his BS in college of science from Xidian University, China in 2003 and MS in college of mathematics and statistics from Minnan Normal University, China in 2008. Her research interests include public key cryptography and network security.
(Email:116745346@qq.com)