SCIS 2014 The 31st Symposium on
Cryptography and Information Security
kagoshima, Japan, Jan. 21 - 24, 2014
The Institute of Electronics,
Information and Communication Engineers

Lattice-based Threshold Signature with Message Block Sharing

Rakyong Choi * Kwangjo Kim *

Abstract— In this paper, we introduce an interesting tool to construct the k-out-of-N threshold signature schemes, which we call as message block sharing. To achieve our goal, we first separate an original message with large size into multiple message blocks with random size. Then, we give a partial number of message blocks to N signers with combinatorial approach. We propose an example scheme with our tool by implementing this to the lattice-based group signature scheme by Gordon *et al.* in ASIACRYPT 2010.

Keywords: Lattice-based Cryptography, Threshold Signature, Message Block Sharing

1 Introduction

After Ajtai's seminal work [1] on lattices, lattices have come up as a fascinating tool in cryptology. The advantage of lattice-based cryptography can be certified from the fact that their security is based on the worst-case hardness assumptions instead of average-case hardness assumptions and the lattice-based cryptography remains secure against quantum computers. From these advantages, the range of applications of lattices varies in many areas in cryptology recently.

Separately, a digital signature scheme is a protocol that one party makes a signature for a message with the private signing key and the other party can verify the signature if he has the public verification key. But if we want to decrease the strength of the signer, the signature should be signed by several members in a group instead of one party. One example is a k-out-of-N threshold signature scheme, which is a protocol that approves any subset of k members among N members to produce a valid signature, but it is impossible to generate a valid signature in case fewer than k members are involved in the protocol. So, any conspiracy of less than k corrupted members cannot produce a valid signature.

In the current technology with quantum computer and big data, one interesting issue is how to apply the cryptographic primitive to be robust to quantum computer attack and the other is how we distribute the power of the signer so that we can control the message with huge size more carefully. Indeed, our lattice-based threshold signature scheme satisfies both conditions.

1.1 Related Approach in Threshold Lattice-based Cryptography

Only a few works have considered lattice-based signature schemes in the threshold setting. Feng *et al.* [10] give a threshold signature scheme whose signing algorithm proceeds sequentially through each member of the group. This scheme is highly interactive and the scheme is based on NTRUSign, which has been broken [17]. Cayrel *et al.* [8] give a lattice-based threshold *ring* signature scheme, in which at least t members are needed to create an *anonymous* signature. In that system, each member has its own public key, and verification time grows linearly with the number of members. Also, Bendlin *et al.* [3] give a threshold variant of Gentry *et al.*'s signature scheme [12] by giving the way to share a lattice trapdoor.

1.2 Related Approach to Our Construction

Unlike the threshold setting, there have been a lot of works on lattice-based signature schemes [7, 8, 11, 12, 14, 15, 16, 19, 20]. But we briefly introduce two previous approach [11, 12]. First one is Gentry *et al.*'s paper which shows how to construct trapdoor functions by sampling lattice points from a specific discrete Gaussian probability distribution and using that distribution, they construct simple and efficient "hash-then-sign" signature scheme [12].

The second paper by Gordon *et al.* is the main reference for this paper and this paper introduces the first construction of a lattice-based group signature scheme with a new algorithm for sampling a basis for an orthogonal lattice and the trapdoor [11].

The most common and the simplest tool to make a threshold signature scheme from the original signature is using the well-known secret sharing technique by Shamir [21]. But in this paper, we do not use Shamir's secret sharing scheme so that we do not need to make any polynomials for secret sharing. Instead, we suggest the simpler concept called "message block sharing" by partitioning the original message into d blocks with some basic combinatorial idea. Then, we apply this idea into Gordon *et al.*'s lattice-based group signature scheme to construct our threshold signature scheme.

* Computer Science Department, KAIST, 291 Gwahak-ro, Yuseong-gu, Daejeon, 305-701, Korea. {thepride, kkj}@kaist.ac.kr

1.3 Outline of the Paper

In Section 2, we introduce the notation and some requirements on lattices to get into this paper such as the lattice problem, trapdoor functions in Gentry *et al.*'s paper, and the group signature suggested by Gordon *et al.* with their Non-Interactive Witness-Indistinguishable (NIWI) proof system. We explain how to construct our main technique, message block sharing with the binomial coefficients in Section 3. In Section 4, with our message block sharing technique, we describe the construction of our lattice-based threshold signature scheme derived from the scheme in Gordon *et al.*'s paper. We discuss the security of our scheme in Section 4 as well. In Section 5, we give a conclusion.

2 Background on Lattices

In the paper, we use n for the security parameter and other parameters are taken as a function of n. When we say something is "statistically close," we mean "their difference is negligible in n."

2.1 Notation and Orthogonal Lattices

We denote vectors using small bold letters (*e.g.*, $\mathbf{x}$, $\mathbf{y}$) and denote matrices using big bold letters (*e.g.*, $\mathbf{A}$, $\mathbf{B}$). $\|\mathbf{x}\|$ represents the Euclidean norm of $\mathbf{x}$ and $\|\mathbf{B}\|$ represents the maximum of Euclidean norms of the columns of $\mathbf{B}$. For instance, when $\mathbf{B} = \{\mathbf{b}_1|\cdots|\mathbf{b}_m\}$, $\|\mathbf{B}\| = \max_i \|\mathbf{b}_i\|$. Then, we denote $\widetilde{\mathbf{B}} = (\widetilde{\mathbf{b}}_1|\cdots|\widetilde{\mathbf{b}}_m)$ for the Gram-Schmidt orthogonalization of columns of $\mathbf{B}$.

If a matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ is given, we define m-dimensional lattice $\mathcal{L}(\mathbf{B}^T)$ as $\mathcal{L}(\mathbf{B}^T) = \{y \in \mathbb{Z}^m | y \equiv \mathbf{B}^T\mathbf{s} \mod q$ for some $\mathbf{s} \in \mathbb{Z}^m\}$. We define the *orthogonal lattice* (or dual lattice) $\Lambda^\perp(\mathbf{B})$ as $\Lambda^\perp(\mathbf{B}) = \{\mathbf{w} \in \mathbb{Z}^m | \mathbf{B}^T\mathbf{w} \equiv 0 \mod q\}$.

2.2 Gaussian Error Distribution

The normal (Gaussian) distribution over $\mathbb{R}$ is a continuous normal distribution and we normally define this as the density function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp(-\frac{(x-\mu)^2}{2\sigma^2}).$$

But in this paper, we replace $\frac{1}{\sigma\sqrt{2\pi}}$ by s and assume that the mean μ is 0. Then, we have the density function

$$D_s(x) = \frac{1}{s} \cdot \exp(-\frac{\pi x^2}{s^2}).$$

Similarly, m-dimensional Gaussian distribution can be defined as

$$D_s(\mathbf{x}) = \frac{1}{s} \cdot \exp(-\frac{\pi\|\mathbf{x}\|^2}{s^2}).$$

When we say Gaussian distribution in the paper, we always mean the *truncated* Gaussian which is the Gaussian distribution whose support is restricted to $x \in \mathbb{R}$ with $|x| < s \cdot \omega(\sqrt{\log n})$.

For a lattice $\Lambda \subset \mathbb{Z}^m$, we can define *discrete Gaussian distribution*, which is the m-dimensional Gaussian distribution whose support is restricted to the lattice Λ and its density function is defined by

$$D_{\Lambda,s}(\mathbf{x}) = \frac{D_s(\mathbf{x})}{\sum_{y\in\Lambda} D_s(\mathbf{y})}.$$

Gentry *et al.* [12] showed that this distribution can be sampled efficiently for $s \geq \|\mathbf{B}\| \cdot \omega(\sqrt{\log n})$.

2.3 Learning With Errors Problem

The "Learning With Errors" (LWE) problem is firstly introduced by Regev in 2005 [18]. LWE problem generalizes the "learning parity with noise" problem and many lattice-based cryptographic protocols like encryption schemes and fully homomorphic encryption schemes and signature schemes assume that LWE problem is hard [6, 11, 13, 18].

We follow the problem described in Gordon *et al.*'s paper [11]. For a positive integer m, n with $m > n$, integer $q \geq 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution χ on the interval $[0, q)^m$, we define the decisional version of the LWE problem as the problem of distinguishing between uniformly chosen $(\mathbf{A}, \mathbf{y}) \leftarrow \mathbb{Z}_q^{n\times m} \times [0, q)^m$ and the sampling $(\mathbf{A}, \mathbf{A}^T\mathbf{s}+\mathbf{e} \mod q)$ where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{e} \leftarrow \chi$. We say that the $\mathrm{LWE}_{m,q,\chi}$ problem is *hard* when $(\mathbf{A}, \mathbf{y})$ and $(\mathbf{A}, \mathbf{A}^T\mathbf{s}+\mathbf{e} \mod q)$ are indistinguishable.

For discrete Gaussian distribution $D_{\mathbb{Z}^m,s} \mod q$, we denote the $\widehat{\mathrm{LWE}}_{m,q,s}$ problem for an abbreviation of the $\mathrm{LWE}_{m,q,D_{\mathbb{Z}^m,s}}$ problem. We remark that hardness of the LWE problem with error distribution Ψ_α^m implies hardness of the LWE problem with error distribution $D_{\mathbb{Z}^m,\alpha\cdot q\cdot\sqrt{2}}$.

Lemma 2.1. [11] *For any α, hardness of the $LWE_{m,q,\alpha}$ problem implies hardness of the $\widehat{LWE}_{m,q,\alpha q\sqrt{2}}$*

2.4 How to Sample an Orthogonal Lattices with Trapdoor

Alwen and Peikert [2] show the trapdoor sampling algorithm which generates an almost uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with a "trapdoor" matrix $\mathbf{T} \in \mathbb{Z}^{m\times m}$ satisfying the following lemma:

Lemma 2.2. [2] There is a probabilistic poly-time algorithm $\mathsf{TrapSamp}(1^n, 1^m, q)$ *with $q \geq 2$ and $m \geq n + 8n\log q$, which outputs matrices $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and $\boldsymbol{T} \in \mathbb{Z}^{m\times m}$ such that*:

- *The distribution on $\mathbf{A}$ as output by $\mathsf{TrapSamp}$ is stastistically cloase to uniform over $\mathbb{Z}_q^{n\times m}$,*
- *the columns of $\mathbf{T}$ form a basis of the lattice $\Lambda^\perp(\mathbf{A})$, implying in particular $\mathbf{A}\cdot\mathbf{T} = 0 \pmod q$,*
- *$\|\mathbf{T}\| = O(n\log q)$ and $\|\tilde{\mathbf{T}}\| \leq C \cdot \sqrt{n\log q}$, for some absolute constant $C < 40$.*

For orthogonal lattices, we modify the documents described in Lemma 2.2 and give another lemma for sampling orthogonal basis and the trapdoor.

Lemma 2.3.[11] *There is a probabilistic poly-time algorithm* $\mathsf{OrthoSamp}(1^n, 1^m, q, \mathbf{B} \in \mathbb{Z}_q^{n\times m})$ with $q \geq 2$ and $m \geq 8n\log q$, which outputs matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{T} \in \mathbb{Z}^{m\times m}$ such that:

- $\mathbf{A}\mathbf{B}^T = 0 \pmod q$. *Moreover, the distribution on* $\mathbf{A}$ *is statistically close to uniform over* $\mathbb{Z}_q^{n\times m}$*, subject to this condition,*
- *the columns of* $\mathbf{T}$ *form a basis of the lattice* $\Lambda^{\perp}(\mathbf{A})$*, implying in particular* $\mathbf{A}\cdot\mathbf{T} = 0 \pmod q$*,*
- *Furthermore, each column* t_i *of* $\mathbf{T}$ *is distributed (independently) according to* $D_{\Lambda^{\perp}(\boldsymbol{A}),s}$*, where* $s = C\cdot\sqrt{n\log q}\cdot\omega(\sqrt{\log m})$ *and* C *is the constant from Lemma 2.2.*

We also need the trapdoor inversion algorithm from Gentry *et al.*'s paper, $\mathsf{GPVInvert}(\mathbf{A}, \mathbf{T}, \mathrm{s}, \mathbf{u})$ outputs $\mathbf{e} \leftarrow D_{\Lambda^{\perp}(\mathbf{A})+\mathbf{t},s}$ after computing an arbitrary $\mathbf{t} \in \mathbb{Z}^m$ with $\mathbf{A}\mathbf{t} = \mathbf{u} \pmod q$.

2.5 NIWI Proof System and the Lattice-based Group Signature Scheme

Let $\mathbf{B}_1, \cdots, \mathbf{B}_n \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{z}_1, \cdots, \mathbf{z}_n \in \mathbb{Z}_q^m$. Then we specify how to construct NIWI proof system that Gordon *et al.* used in their paper for the gap language $L_{s,\gamma} = (L_{YES}, L_{NO})$ defined by:

$$L_{YES} = \left\{ \begin{pmatrix} \mathbf{B}_1, \cdots, \mathbf{B}_n \\ \mathbf{z}_1, \cdots, \mathbf{z}_n \end{pmatrix} \middle| \exists\ \mathbf{s} \in \mathbb{Z}_q^n \text{ and } i \in [N] : \|\mathbf{z}_i - \mathbf{B}_i^T\mathbf{s}\| \leq s\sqrt{m} \right\}$$

$$L_{NO} = \left\{ \begin{pmatrix} \mathbf{B}_1, \cdots, \mathbf{B}_n \\ \mathbf{z}_1, \cdots, \mathbf{z}_n \end{pmatrix} \middle| \forall\ \mathbf{s} \in \mathbb{Z}_q^n \text{ and } i \in [N] : \|\mathbf{z}_i - \mathbf{B}_i^T\mathbf{s}\| > \gamma\cdot s\sqrt{m} \right\}$$

Here, L_{YES} represents the collection of N points at least one of which is close to the corresponding lattice, and L_{NO} represents the collection of N points all of which are pretty far from the corresponding lattice.

Lemma 2.4.[11] *Let* $\gamma \geq O\sqrt{\dfrac{m}{\log m}}$*. Then, there is a NIWI proof system for the language* $L_{s,\gamma}$ *in the random oracle model, where the length of the proof is* $O(mnN\log q)$ *bits.*

From Lemma 2.1 through Lemma 2.4, we can construct the lattice-based group signature scheme by Gordon *et al.*'s. Formally, group signature scheme $\mathcal{GS}$= (G.KeyGen, G.Sign, G.Verify, G.Open) is a collection of four poly-time algorithms [4, 11] defined by:

- **Group key-generation algorithm** G.KeyGen $(1^n, 1^N)$ with a security parameter n and the group size N is a randomized algorithm that outputs (PK, TK, **gsk**), where PK is the public key, TK is the tracing key for the opener, and **gsk** is a vector of N signing keys for each member i.
- **Group signature signing algorithm** G.Sign $(\mathsf{gsk}[i], M)$ with signing key $\mathsf{gsk}[i]$ and the message M is a randomized algorithm that outputs a signature σ.
- **Group signature verification algorithm** G.Verify(PK, M, σ) is a deterministic algorithm which outputs either 1 or 0. (accept or reject)
- **Opening algorithm** G.Open(TK, M, σ) is a deterministic algorithm that outputs an identity $i \in [N]$.

The group signature should satisfy *anonymity* and *traceability* for the security. Anonymity means that we cannot determine which group member signed the particular message without the tracing key, even adversary has all the signing keys. Traceability means that adversary cannot forge a valid signature which is untraceable by the opener.

We give a description of Gordon *et al.*'s scheme briefly. We let n be the security parameter, $q = \mathsf{poly}(n), m \geq 8n\log q$ and $s \geq C\sqrt{n\log q}\cdot\omega(\sqrt{\log m})$ be other parameters of the system. We let $H : \{0,1\}^* \to \mathbb{Z}_q^n$ be a hash function for a random oracle.

GKV.KeyGen$(1^n, 1^N)$: Extract $(\mathbf{B}_1, \mathbf{S}_1), \cdots, (\mathbf{B}_N, \mathbf{S}_N) \leftarrow \mathsf{TrapSamp}(1^n, 1^m, q)$ and for each $i \in [N]$, compute $(\mathbf{A}_i, \mathbf{T}_i) \leftarrow \mathsf{OrthoSamp}(1^n, 1^m, q, \mathbf{B}_i)$. It outputs $\mathsf{PK} = \left((\mathbf{A}_i, \mathbf{T}_i)_{i=1}^N\right)$ as the public key, $\mathsf{TK} = (\mathbf{S}_i)_{i=1}^N$ as the tracing key, $\mathbf{gsk} = (\mathbf{T}_i)_{i=1}^N$ as a signing key for each member.

GKV.Sign $(\mathsf{gsk}[j], M)$: Choose random $r \leftarrow \{0,1\}^n$, set $\bar{M} = M\|r$, and compute $\mathbf{h}_i = H(\bar{M}\|i)$ for each $i \in [N]$. Then,

- Compute $\mathbf{e}_j \leftarrow \mathsf{GPVInvert}(\mathbf{A}_j, \mathbf{T}_j, s, \mathbf{h}_j)$.
- For $i \neq j$, choose $\mathbf{e}_j \leftarrow \mathbb{Z}_q^m$ uniformly random, with $\mathbf{A}_i\mathbf{e}_i = \mathbf{h}_i \pmod q$

For all i, sample $\mathbf{s}_i \leftarrow \mathbb{Z}_q^n$ and compute $\mathbf{z}_i = \mathbf{B}_i^T\mathbf{s}_i + \mathbf{e}_i \pmod q$ in $\mathbb{Z}_q^n$. Then, using the witness $(\mathbf{s}_i, i)$, we construct the NIWI proof π for the gap language $L_{s,\gamma}$ as discussed above.
Output $(r, \mathbf{z}_1, \cdots, \mathbf{z}_N, \pi)$.

GKV.Verify(PK, M, σ): Resolve the signature σ into $(r, \mathbf{z}_1, \cdots, \mathbf{z}_N, \pi)$ and set $\bar{M} = M\|r$. Output 1 iff the proof π is correct, and $\mathbf{A}_i\mathbf{z}_i = H(\bar{M}\|i) \pmod q$ for all i.

GKV.Open(TK, M, σ): Resolve the signature σ into $(r, \mathbf{z}_1, \cdots, \mathbf{z}_N, \pi)$ and using the $\mathsf{TK} = (\mathbf{S}_i)_{i=1}^N$, output the smallest index i whose distance between $\mathcal{L}(\mathbf{B}_i^T)$ and $\mathbf{z}_i$ is at most $s\sqrt{m}$ or output $\perp$ in case to indicate the failure.

We can easily check the correctness of the scheme. This scheme assumes hardness of $\mathrm{LWE}_{m,q,\alpha}$ problem.

3 Message Block Sharing

Unlike Shamir's secret sharing technique, we focus more on basic structure of general 'hash-and-sign' signature scheme. The idea is that we first divide an original message into message blocks $M_1, \cdots, M_d$ with random size so that no block remains meaningful. Then, we distribute uniform number of message blocks into each member of the group. But to satisfy the condition of k-out-of-N threshold signature scheme, this distribution cannot be randomly shared. Moreover, alliance of less then k members do not have every message block and any alliance of k members must have the entire message like the condition of threshold signature scheme.

From here, we denote μ_i as the member of the group and Γ_i as the set of message blocks that a member μ_i gets. We briefly review the definition of the binomial coefficient and introduce how to construct our main tool called "message block sharing" technique.

3.1 Definition of Binomial Coefficient

We denote the binomial coefficient as $\binom{a}{b} = \frac{n!}{k!(n-k)!}$ and $\binom{a}{b}$ is interpreted as a coefficient of the monomial x^b of binomial formula $(x+y)^a$.

But in combinatorics, this number represents the number of ways that b instances are chosen from a instances. (*i.e.*, This number represents the number of a-element subsets of an b-element set.) This interpretation leads to our message block sharing technique.

3.2 2-out-of-3 Message Block Sharing

In this subsection, we give a toy example to get an inspiration to make k-out-of-N message block sharing. In other words, there are N members in the group and if k of them share their information, the whole message reveals.

For two-out-of-three message block sharing, we first divide the original message into several message blocks $M_1, \cdots, M_s$ and give t message blocks to each member so that two members can reveal the message. Indeed, $d = 3, t = 2$ is enough to achieve our goal. If we let $\Gamma_1 = \{M_2, M_3\}, \Gamma_2 = \{M_1, M_3\}$ and $\Gamma_3 = \{M_1, M_2\}$, one member do not have complete message blocks and any two members can get the message.

3.3 k-out-of-N Message Block Sharing

From the previous example, we notice that no message block is shared by all members. If we switch this into the fact slightly, we get that for each message block M_j, we can accept the situation that one member does not have that block M_j. We generalize this fact into k-out-of-N message block sharing model. Then, for each message block, it is sufficient although $k-1$ members do not have that block. Moreover, to exclude the possibility that the coalition of $k-1$ members make full message, $k-1$ members should not contain a block M_j for each block message M_j.

The last condition for message block sharing is that each member has the same number of messages and this can be satisfied with the way we find all possible $k-1$-element subsets of N-element set and for the rest of the set $\{\alpha_1, \cdots, \alpha_{l-k+1}\}$, each message block goes into members $\mu_{\alpha_1}, \cdots, \mu_{\alpha_{l-k+1}}$.

Clearly, if the number of message block s is equal to $\binom{N}{k-1}$, we get the desired message block sharing model. The conspiracy of at most $k-1$ members do not reveal the full message since there always exists exactly one message block missed from our distribution of message blocks and k members must obtain the message since at least one member has the message block M_j for each message block M_j. Furthermore, each member has exactly $t = \binom{N-1}{k-1}$ message blocks.

Namely, to construct k-out-of-N message block sharing, we need the message to be sufficiently large so that the message can be divided into $d = \binom{N}{k-1}$ message blocks. Thus, in the following section, we consider that security parameter n is determined by the size of the original message.

Summing up the above, we have the parameter extracting algorithm defined as:

- **Threshold parameter extracting algorithm** $\mathsf{TParamExt}(\mathcal{M}_M)$ with $\mathcal{M}$ which is a bit length of the message M is a deterministic poly-time algorithm that outputs (n, N, k), where n is the security parameter, N is the group size, and k is the number of members to generate the valid signature.

Using this algorithm, we can construct k-out-of-N threshold signature scheme in Section 4.

4 Our Threshold Signature Scheme Based on Lattices

In this section, we define the k-out-of-N threshold signature scheme more precisely. Then, we modify the lattice-based group signature scheme in Gordon *et al.*'s paper using message block sharing technique. We describe our threshold signature scheme.

4.1 Definitions and Security Model

We adopt the definition and security model from the work of Shoup [23] to define our k-out-of-N threshold signature schemes.

The k-out-of-N threshold signature $\mathcal{TS} = (\mathsf{T.Param}, \mathsf{T.KeyGen}, \mathsf{T.Sign}, \mathsf{T.Verify})$ is a collection of four poly-time algorithms defined by:

- **Threshold parameter preparing algorithm** $\mathsf{T.Param}(M)$ with the message M is a determinis-

tic algorithm which outputs (n, N, k), where n is the security parameter, N is the group size, and k is the number of required signers for verification.

- **Threshold key-generation algorithm** $\mathsf{T.KeyGen}$ $(1^n, 1^N, 1^k)$, with a security parameter n, the group size N, and the number of required signers for verification, is a randomized algorithm that outputs $(\mathsf{PK}, \mathbf{gsk}, (\Gamma_i)_{i=1}^N)$, where PK is the public key, $\mathbf{gsk}$ is a vector of N signing keys for each member i, and Γ_i is a message block set that the member i can sign on.
- **Threshold signature signing algorithm** $\mathsf{T.Sign}$ $(\mathsf{gsk}, M, (\Gamma_i)_{i=1}^N)$ with signing key gsk and the message M is a randomized algorithm that outputs a signature σ.
- **Threshold signature verification algorithm** $\mathsf{T.Verify}(\mathsf{PK}, M, (\Gamma_i)_{i=1}^N, \sigma)$ is a deterministic algorithm which outputs either 1 or 0 (depends on whether accepted or rejected).

For the security model, we assume there are τ corrupted players with the condition such that $k \geq \tau + 1$ and $N - \tau \geq k$ and security requirements of the threshold signature scheme is as follows:

Correctness. If there is no corrupted players, then combining signature of any k signers outputs a valid signature on the message.

Unforgeability. We say that the adversary forges a signature when he outputs a valid signature for a message in which was not submitted as a signing query to at least $k - \tau$ uncorrupted players. We say that the threshold signature scheme is unforgeable when the probability that the adversary forges a signature is negligible.

4.2 Our Construction

Similar to Gordon *et al.*'s scheme, we let n be the security parameter, $q = \mathsf{poly}(n), m \geq 8n \log q$ and $s \geq C\sqrt{n \log q} \cdot \omega(\sqrt{\log m})$ be other parameters of the system. We let $H : \{0,1\}^* \to \mathbb{Z}_q^n$ be a hash function for a random oracle. but n is decided after computing bit length of the message M and we do not need the opening algorithm in our scheme.

$\mathsf{T.Param}(M)$: Compute the bit length $\mathcal{M}_M$ of the message M and outputs $(n, N, k) \leftarrow \mathsf{TParamExt}(\mathcal{M}_M)$.

$\mathsf{T.KeyGen}(1^n, 1^N)$: Extract $(\mathbf{B}_1, \mathbf{S}_1), \cdots, (\mathbf{B}_N, \mathbf{S}_N) \leftarrow \mathsf{TrapSamp}(1^n, 1^m, q)$ and for each $i \in [N]$, compute $(\mathbf{A}_i, \mathbf{T}_i) \leftarrow \mathsf{OrthoSamp}(1^n, 1^m, q, \mathbf{B}_i)$. It outputs $\mathsf{PK} = \big((\mathbf{A}_i, \mathbf{T}_i)_{i=1}^N\big)$ as the public key, $\mathbf{gsk} = (\mathbf{T}_i)_{i=1}^N$ as a signing key for each member. Finally, compute and distribute the corresponding message block set $\Gamma_i = \{(M_{i_1}, \cdots, M_{i_t})$ for each member i where $t = \binom{N-1}{k-1}$ and the total number of message blocks is $d = \binom{N}{k-1}$.

$\mathsf{T.Sign}$ $(\mathsf{gsk}, M, (\Gamma_i)_{i=1}^N)$: For each member i, set $M[i] = M_{i_1} \| \cdots \| M_{i_t}$ and compute $\mathbf{h}_i = H(M[i])$ for each $i \in [N]$. Then,

- Compute $\mathbf{e}_j \leftarrow \mathsf{GPVInvert}(\mathbf{A}_j, \mathbf{T}_j, s, \mathbf{h}_j)$.

For all i, sample $\mathbf{s}_i \leftarrow \mathbb{Z}_q^n$ and compute $\mathbf{z}_i = \mathbf{B}_i^T \mathbf{s}_i + \mathbf{e}_i \pmod q$ in $\mathbb{Z}_q^n$. Then, using the witness $(\mathbf{s}_i, i)$, we construct the NIWI proof π for the gap language $L_{s,\gamma}$ as discussed above.
Output $(\mathbf{z}_1, \cdots, \mathbf{z}_N, \pi)$.

$\mathsf{T.Verify}(\mathsf{PK}, M, (\Gamma_i)_{i=1}^N, \sigma)$: First check whether $\Gamma_i \subset \{M_1, \cdots, M_d\}$ for all i. Resolve the signature σ into $(\mathbf{z}_1, \cdots, \mathbf{z}_N, \pi)$ and set$M[i] = M_{i_1} \| \cdots \| M_{i_t}$ and compute $\mathbf{h}_i = H(M[i])$ for each $i \in [N]$. Output 1 iff the proof π is correct, $\Gamma_i \subset \{M_1, \cdots, M_d\}$ for all i, and $\mathbf{A}_i \mathbf{z}_i = H(M[i]) \pmod q$ for all i.

Then, we show the security requirement of our proposed scheme as follows:

Theorem 4.1. *The proposed scheme produces the valid signature. (Correctness)*

Proof. $\mathbf{A}_i \mathbf{z}_i = \mathbf{A}_i(\mathbf{B}_i^T \mathbf{s}_i + \mathbf{e}_i) = \mathbf{A}_i \mathbf{e}_i = H(M[i]) \pmod q$ since $\mathbf{A}_i \mathbf{B}_i^T = 0 \pmod q$ for all i. This confirms for each i, $\mathbf{z}_i$ is a valid signature for $M[i]$.

Indeed, if we combine signatures of any k members, we can get the signature for the original message since $\sum_{\delta=1}^k \Gamma[i_\delta] =$ (the whole message blocks) from the construction of $\Gamma[i]$. Thus, any k members can generate the valid signature for the message using the message block sharing model and correctness is guaranteed in this model. □

Theorem 4.2. *The proposed scheme satisfies the unforgeability.*

Proof. We may assume that $k - 1$ members are corrupted. Then, this problem becomes the unforgeability of one signature $\mathbf{z}_i$. Indeed, this can be guaranteed from the hardness of LWE problem and collision-resistance of the hash function. Thus, our construction is also unforgeable. □

4.3 Comparison with Other Threshold Signature Schemes

We compare related signature scheme in threshold setting that we described in section 1 with our scheme. Feng *et al.*'s work [10] gives sequential signing process so that each member cannot generate their own signature simultaneously. This scheme is based on the standard NTRU lattice and the variation of Closest Vector Problem (CVP) which is a worst-case hardness problem to find the closest lattice vector $\mathbf{u}$ to the given vector w. But during the protocol, this scheme use NTRUSign

scheme which can be broken these days as discussed in Section 1. Cayrel *et al.*'s work [8] 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 small integer solution (SIS) problem which is weaker than LWE problem. Then, Bendlin *et al.* [3] provide a threshold signature scheme by using an algorithm to share a lattice trapdoor. This scheme is based on LWE problem but conceptually hard to understand.

In Table 1, we compare our scheme with other threshold schemes. Our scheme is based on LWE problem so that there is no known attack and the scheme is conceptually simpler since we do not change any trapdoor functions as Bendlin *et al.* did, but divide the message into parts. Moreover, our scheme can generate the signature concurrently without any sequence. But since our scheme generates the group size after we get the message M, this would be inefficient in some situation.

Table 1: Threshold Signatures on Lattices

Scheme	FGM[10]	CLRS[8]	BKP[3]	Ours
Problem	NTRU lattice	**Ideal lattice**	Lattice	Lattice
Security	CVP	SIS	**LWE**	**LWE**
Known attack	Yes	**No**	**No**	**No**
Key Idea	sequential signing	syndrome decoding	trapdoor share	message block

Moreover, if we compare other signature scheme in threshold setting based on other problem such as Discrete Logarithm Problem (DLP) [5, 9] and Integer Factorization Problem (IFP) [23], our scheme takes more operation than other schemes from the view of computation. But, unlike those schemes based on DLP and IFP which can be broken by quantum computer attack [22], our scheme is secure against quantum computer attack.

5 Conclusion

We construct a conceptually simpler lattice-based threshold signature scheme with a new idea using message block sharing tool. Compare to Gordon *et al.*'s scheme, each member has the different partial message and its corresponding hash value. Then, all members generate valid signature for partial message blocks as Gordon *et al.*'s scheme delegates one member to generate the valid signature on behalf of all group members and other member generates an invalid signature. This difference makes our scheme to be a threshold signature scheme instead of group signature scheme like Gordon *et al.*'s scheme.

Our scheme enables the signing process of the message with large bit length to be more secure but in some cases, the scheme might be inefficient as we set the group size after we get the message. As we can see in Table 1, we still need to improve our scheme like constructing its Ring-LWE variant to get more efficiency.

So, for future work, we consider how to construct the Ring-LWE version of our scheme and also, we plan to apply our tool to design a new threshold group signature scheme or a threshold ring signature scheme based on lattices.

Acknowledgement

This research was supported by the KUSTAR-KAIST Institute, Korea, under the R&D program supervised by the KAIST and funded by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

References

[1] Miklós Ajtai, "Generating hard instances of lattice problems," *Proceedings of the twenty-eighth annual ACM Symposium on Theory of Computing*, 99–108, 1996.

[2] Joël Alwen and Chris Peikert, "Generating shorter bases for hard random lattices," *26th International Symposium on Theoretical Aspects of Computer Science (STACS) 2009*, 75-86, 2009

[3] Rikke Bendlin, Sara Krehbiel, and Chris Peikert, "How to share a lattice trapdoor: threshold protocols for signatures and (H) IBE," *Applied Cryptography and Network Security*, 218-236, 2013.

[4] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi, "Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions," *Advances in Cryptology – EUROCRYPT 2003*, 614-629, 2003.

[5] Alexandra Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme," *Public key cryptography – PKC 2003*, 31-46, 2002.

[6] Zvika Brakerski and Vinod Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *Foundations of Computer Science (FOCS)*, 97-106, 2011.

[7] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert, "Bonsai Trees, or How to Delegate a Lattice Basis," *Advances in Cryptology – EUROCRYPT 2010*, 523-552, 2010

[8] Pierre-Louis Cayrel, Richard Lindner, Markus Rückert, and Rosemberg Silva, "A lattice-based threshold ring signature scheme," *Progress in Cryptology – LATINCRYPT 2010*, 255-272, 2010.

[9] Xiaofeng Chen, Fangguo Zhang, Divyan M. Konidala, and Kwangjo Kim, "New ID-based threshold signature scheme from bilinear pairings," *Progress in Cryptology – INDOCRYPT 2004*, 371-383, 2005.

[10] Tao Feng, Yongguo Gao, and Jianfeng Ma, "Changeable Threshold Signature Scheme Based on Lattice Theory," *E-Business and E-Government (ICEE), 2010 International Conference on*, 1311-1315, 2010.

[11] S. Dov Gordon, Jonathan Katz, and Vinod Vaikuntanathan, "A group signature scheme from lattice assumptions," *Advances in Cryptology – ASIACRYPT 2010*, 395-412, 2010.

[12] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," *Proceedings of the 40th annual ACM Symposium on Theory of Computing*, 197-206, 2008.

[13] Craig Gentry, Amit Sahai, and Brent Waters, "Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based," *Advances in Cryptology – CRYPTO 2013*, 75-92, 2013

[14] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman, "NSS: An NTRU lattice-based signature scheme," *Advances in Cryptology – EUROCRYPT 2001*, 211-228, 2001.

[15] Vadim Lyubashevsky and Daniele Micciancio, "Asymptotically efficient lattice-based digital signatures," *Theory of Cryptography*, 37-54, 2008.

[16] Vadim Lyubashevsky, "Lattice signatures without trapdoors," *Advances in Cryptology – EUROCRYPT 2012*, 738-755, 2012.

[17] Phong Q. Nguyen and Oded Regev, "Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures," *Advances in Cryptology – EUROCRYPT 2006*, 271-288, 2006.

[18] Oded Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Proceedings of the thirty-seventh annual ACM Symposium on Theory of Computing*, 84-93, 2005.

[19] Markus Rückert, "Lattice-based blind signatures," *Advances in Cryptology – ASIACRYPT 2010*, 413-430, 2010.

[20] Markus Rückert, "Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles," *Post-Quantum Cryptography*, 182-200, 2010.

[21] Adi Shamir, "How to share a secret," *Communications of the ACM 22.11*, 612-613, 1979.

[22] Peter W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM journal on computing 26.5*, 1484-1509, 1997

[23] Victor Shoup, "Practical threshold signatures," *Advances in Cryptology – EUROCRYPT 2000*, 207-220, 2000.