SCIS 2017 2017 Symposium on
Cryptography and Information Security
Naha, Japan, Jan. 24 - 27, 2017
The Institute of Electronics,
Information and Communication Engineers

Design of New Linearly Homomorphic Signatures on Lattice

Rakyong Choi * Kwangjo Kim *

Abstract: This paper introduces two designs to enhance the Boneh and Freemans linearly homomorphic signature over binary fields, to overcome the limitations to implement homomorphic signatures to the real world scenario due to the heavy calculation and under multiple signers setting for a message.

Based on our concurrent work on classification on lattice-based trapdoor functions in SCIS 2017, we modify some algorithms from the original signature. We design the linearly homomorphic ring signature by adopting Wang and Sun's sampling algorithm GenSamplePre() instead of the original sampling algorithm SamplePre() by Gentry *et al.* Also, we adopt the mixing and vanishing technique of trapdoors by Boyen to design more efficient linearly homomorphic signature scheme with short signatures.

Keywords: ring signature, homomorphic signature, lattices, trapdoor function, sampling algorithm

1 Introduction

1.1 Background and Motivation

As the infrastructure of cloud systems increases, one of uprising security challenges is how the cloud server computes a function of encrypted messages without decryption. Despite of numerous studies on fully homomorphic encryption on lattices [1–3] to enable the server to calculate any function of encrypted message without decryption, there is another security issue in cloud system as how the cloud server gives authenticity for the function of encrypted message.

For authenticity of cloud systems, a signature is a well-known cryptographic primitive. But, the cloud server should have the power to generate the proper signature for a computation of messages without permission from the signer of each message as well. If the signature satisfies this condition, we say that the signature has the *homomorphic property.* Especially, a signature is called *linearly homomorphic* when it supports constructing the proper signature for the linear combination of messages [4, 5] and *fully homomorphic* when it supports constructing the proper signature for any function of messages [6, 7].

But, there are limitations to implement homomorphic signatures to the real world scenario due to the heavy calculation and under multiple signers setting for a message.

1.2 Our Contribution

In the work of Choi and Kim in SCIS 2017 [8], they summarize the characteristics of lattice-based trapdoor functions and their preimage sampling algorithms using a trapdoor. Based on their paper, we modify the algorithms from lattice-based linearly homomorphic signature over binary fields by Boneh and Freeman [4].

We first consider the linearly homomorphic ring signature over binary fields by adopting Wang and Sun's preimage sampling algorithm GenSamplePre() [9].

We let each member in the ring take their own public key and secret key by trapdoor generation function TrapGen() during the setup phase in the first design. Then, we concatenate the public key of each member to make the common public key. Then, in the signing phase, we modify the preimage sampling algorithm from well-known SamplePre() to GenSamplePre().

This design can be used in the real world scenario since some information on cloud system is signed by an organization instead of an individual and there should be at least two people to authenticate the message where each person has his/her secret key. In this situation, we need multiple signers with different secret keys for a single message and the corresponding signature is valid only if all signers are trustworthy.

Also, we adopt the mixing and vanishing technique of trapdoors introduced by Boyen [10] to design the linearly homomorphic signature scheme over binary fields with short signatures so that we have more practical linearly homomorphic scheme.

1.3 Related Work

In 2011, Boneh and Freeman [4] published their seminal work on linearly homomorphic signature over binary fields based on lattices with new lattice-based hard problems called k-SIS problem. Boneh and Freeman [5] also suggested that some bounded homomorphic signature can be constructed using ideal lattices from Gentry's fully homomorphic encryption [1].

After Boneh and Freeman's work, lattices have become a main tool to make linearly and fully homomorphic signatures. Zhang *et al.* [11] introduced the notion

* School of Computing, KAIST. 291, Daehak-ro, Yuseong-gu, Daejeon, South Korea 34141. {thepride, kkj}@kaist.ac.kr

of a homomorphic aggregate signature which doesn't need to have the same secret key to combine multiple messages. Then, they suggested a linearly homomorphic aggregate signature using the random basis generation algorithm RandBasis() by Cash *et al.* [12] to generate multiple secret keys.

Jing [13] separately suggested an efficient homomorphic aggregate signature with linear homomorphism as they concatenate a public key of each signer and use the extending trapdoor basis algorithm ExtBasis() by Cash *et al.* [12]. Both Zhang *et al.* [11] and Jing's [13] contributions are making multi-key linearly homomorphic signatures.

Choi and Kim [14] used the same technique suggested by Jing but pre-shared the message to multiple signers of a message to get the linearly homomorphic multisignature. This work suggested the first construction of multi-key multi-party linearly homomorphic signatures to the best of our knowledge.

Besides the linearly homomorphic signatures, Gorbunov *et al.* [6] suggested the first fully homomorphic signature scheme with a homomorphic trapdoor function but there is only one secret key. Recently, Fiore *et al.* [7] suggested a fully homomorphic signature scheme with multi-key setting, *i.e.*, there are multiple secret keys.

1.4 Outline of the Paper

Section 2 gives a notation and a background on a lattice and lattice-based cryptography from the definition of lattices and hard problems on lattices to lattice-based algorithms for trapdoor generation and sampling. Then, formal definition and security requirement of linearly homomorphic signature with detailed construction is given in Section 3.

We give the design of new linearly homomorphic signatures in Section 4 and we give a concluding remark with future work in Section 5.

2 Preliminaries

2.1 Notation

We denote vectors as small bold letters (*e.g.*, $\mathbf{x}$, $\mathbf{y}$) and matrices as big bold letters (*e.g.*, $\mathbf{A}$, $\mathbf{B}$).

Let $\mathbb{R}$ and $\mathbb{Z}$ express the set of real numbers and the set of integers, respectively and small alphabet letters express real numbers (*e.g.*, a, b, c).

For any integer $q \geq 2$, $\mathbb{Z}_q$ denotes the ring of integers modulo q and $\mathbb{Z}_q^{n\times m}$ denotes the set of $n \times m$ matrices with entries in $\mathbb{Z}_q$. When $\mathbf{A} \in \mathbb{Z}_q^{n\times m_1}$, $\mathbf{B} \in \mathbb{Z}_q^{n\times m_2}$, we write the concatenation of $\mathbf{A}$ and $\mathbf{B}$ as $[\mathbf{A} \mid \mathbf{B}] \in \mathbb{Z}_q^{n\times(m_1+m_2)}$.

Let $f(a, b)$ be a function f on a and b. We say a function $f : \mathbb{Z} \to \mathbb{R}^+$ is *negligible* when $f = O(n^{-c})$ for all $c > 0$ and denoted by $\mathrm{negl}(n)$. A function $g(m) = \lceil m\rceil$ is the ceiling function from $\mathbb{R}$ to $\mathbb{Z}$ such that $g(m)$ is the smallest integer which is greater than or equal to m.

$\|\mathbf{x}\|$ 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 $\mathbf{B} = \{\mathbf{b}_1|\mathbf{b}_2|\cdots|\mathbf{b}_m\}$, $\|\mathbf{B}\| = \max_i \|\mathbf{b}_i\|$. Then, we denote $\widetilde{\mathbf{B}} = (\widetilde{\mathbf{b}}_1|\widetilde{\mathbf{b}}_2|\cdots|\widetilde{\mathbf{b}}_m)$ for the Gram-Schmidt orthogonalization of columns of $\mathbf{B}$ and denote $\|\widetilde{\mathbf{B}}\| = \max_i \|\widetilde{\mathbf{b}}_i\|$ for *Gram-Schmidt norm* of $\mathbf{B}$.

2.2 Lattices and Algorithm for Trapdoor Basis Delegation

Briefly, lattices are a fascinating tool in modern cryptography and a lattice Λ can be defined as a discrete subgroup of $\mathbb{R}^m$ with its basis $\mathcal{S}$. A basis $\mathcal{S}$ of Λ is a set of linearly independent vectors $\mathcal{S} = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m\}$ which spans the lattice Λ and $\mathbf{S} = (\mathbf{b}_1|\mathbf{b}_2|\cdots|\mathbf{b}_m)$ is a basis matrix of lattice Λ.

Integer lattices are defined as a subgroup of $\mathbb{Z}^m$ instead of $\mathbb{R}^m$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we can denote lattices as a set $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m | \mathbf{A}\cdot\mathbf{e} = \mathbf{u} \bmod q\}$ and as a set $\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m | \mathbf{A}\cdot\mathbf{e} = \mathbf{0} \bmod q\}$ when $\mathbf{u} = \mathbf{0}$.

Lattice-based cryptography has a lot of advantages that their security is based on the average-case hardness problems like Small Integer Solution (SIS) problem and LWE(Learning With Errors) problem, which remain secure against quantum computing attacks and can be reduced to the worst-case hardness problem in lattices like Shortest Vector Problem (SVP) and Closest Vector Problem (CVP). Among them, SIS problem is defined as below.

Definition 1. (SIS problem) Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with $m \geq n\log q$ and its corresponding lattice $\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m | \mathbf{A}\cdot\mathbf{e} = \mathbf{0} \bmod q,\}$, it is hard to find a small vector $\mathbf{e} \in \Lambda_q^{\perp}(\mathbf{A})$, such that $\|\mathbf{e}\| \leq \beta$ for some $\beta \geq \sqrt{n\log q}$ and $\mathbf{A}\cdot\mathbf{e} = 0 \pmod q$, whose coefficients are either $-1, 0,$ or 1.

If we have the short "trapdoor" basis, all hard problems in lattice become solvable efficiently. Alwen and Peikert [15] introduced the trapdoor generation algorithm $\mathsf{TrapGen}(n, m, q)$ which generates a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with its "trapdoor" matrix $\mathbf{T} \in \mathbb{Z}^{m\times m}$ satisfying the following functionality:

TrapGen(n, m, q) :
: For the security parameter n, $m = \lceil 6n\log q\rceil$ and an integer q, this algorithm outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and its trapdoor $\mathbf{T}$ such that $\mathbf{T}$ is a basis of $\Lambda_q^{\perp}(\mathbf{A})$ with low Gram-Schmidt norm $\|\widetilde{\mathbf{T}}\| \leq 30\sqrt{n\log q}$.

Without loss of generality, we assume that a matrix $\mathbf{A}$ extracted from $\mathsf{TrapGen}(n, m, q)$ has a full rank. In our construction, a matrix $\mathbf{A}$ and its trapdoor $\mathbf{T}$ are used as a public key and a secret key, respectively.

Cash *et al.* [12] introduced the technique to randomly generate the basis from the matrix and to extend the basis to higher dimension in the concept of bonsai trees using the following algorithms.

RandBasis(T, s) :
For the trapdoor matrix $\mathbf{T}$ of $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a parameter $s \geq \|\mathbf{T}\| \cdot \omega(\sqrt{\log n})$, this algorithm outputs a basis $\mathbf{T}'$ for $\Lambda_q^{\perp}(\mathbf{A})$ with $\|\mathbf{T}'\| \leq s\cdot\sqrt{m}$.

ExtBasis(T, B) :
For the trapdoor matrix $\mathbf{T}$ of $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and the matrix $\mathbf{B} = \mathbf{A}\|\mathbf{A}' \in \mathbb{Z}_q^{n\times(m+m')}$, this algorithm outputs a basis $\mathbf{S}$ for $\Lambda_q^{\perp}(\mathbf{B})$ with $\|\widetilde{\mathbf{S}}\| = \|\widetilde{\mathbf{T}}\|$ in polynomial time, *i.e.*, Gram-Schmidt norm of $\mathbf{S}$ is equal to that of $\mathbf{T}$.

The extending trapdoor basis algorithm $\mathsf{ExtBasis}(\mathbf{T}, \mathbf{B})$ can be implemented to get a short basis of the higher-dimensional lattice from the lower-dimensional lattice.

2.3 Discrete Gaussian Distribution and Sampling Algorithm

For any subset $L \subset \mathbb{Z}^m$, a Gaussian function on $\mathbb{R}^m$ with center $\mathbf{c}$ and parameter γ can be defined as $\rho_{\gamma,\mathbf{c}}(\mathbf{x}) = \exp\left(\dfrac{-\pi\|\mathbf{x}\text{-}\mathbf{c}\|^2}{\gamma^2}\right)$ for any vector $\mathbf{c} \in \mathbb{R}^m$ and any positive parameter $\gamma > 0$ and a density function of discrete Gaussian distribution on a subset L, center $\mathbf{c}$, and parameter γ can be defined as

$$\mathcal{D}_{L,\gamma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\gamma,\mathbf{c}}(\mathbf{x})}{\sum_{\mathbf{y}\in L}\rho_{\gamma,\mathbf{c}}(\mathbf{y})}.$$

For the simplicity, we denote $\rho_{\gamma}(\mathbf{x})$ and $\mathcal{D}_{L,\gamma}(\mathbf{x})$ when center $\mathbf{c} = \mathbf{0}$.

Gentry *et al.* [16] proved that this distribution can be sampled efficiently for $\gamma \geq \|\widetilde{\mathbf{T}}\| \cdot \omega(\sqrt{\log n})$ where $\mathbf{T}$ is a trapdoor matrix of an n-dimensional lattice Λ as follows:

SamplePre(A, T, γ, u) :
For the matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, its trapdoor matrix $\mathbf{T} \in \mathbb{Z}_q^{m\times m}$, a real number $\gamma > 0$, and a vector $\mathbf{u} \in \mathbb{Z}^n$, this algorithm outputs a sample σ from a distribution that is statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\gamma}$.

The smoothing parameter $\eta_{\epsilon}(\Lambda)$ of Λ enables every coset of Λ to get roughly equal mass in the following **Lemmas 1** and **2**.

Lemma 1. *[16] Let q be a prime and n, m be integers with $m > 2n\log q$. Let f be some $\omega(\sqrt{\log m})$ function. Then, there is a negligible function $\epsilon(m)$ such that for all but at most q^{-n} fraction of matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, we have $\eta_{\epsilon(m)}(\Lambda_q^{\perp}(\boldsymbol{A})) < f(m)$.*

Lemma 2. *[4] Let $\Lambda \subset \mathbb{R}^n$ be a lattice. Suppose $\rho \geq \eta_{\epsilon}(\Lambda)$ for some negligible ϵ. Then, we have*

$$\Pr\left[0 \leq \|\mathbf{v}\| \leq 2\rho\sqrt{\frac{n}{2\pi}} : \mathbf{v} \leftarrow \mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\gamma}\right] \geq 1-\mathrm{negl}(n).$$

Lemma 1 declares that a sample vector from $\mathsf{SamplePre}$ $(\mathbf{A},\ \mathbf{T}, \gamma, \mathbf{u})$ with proper parameters can be extracted uniformly and **Lemma 2** determines the upper bound on the length $\|\mathbf{v}\|$ of a sample vector $\mathbf{v}$ from the Gaussian distribution $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\gamma}$.

Wang and Sun [9] suggested a new preimage sampling algorithm $\mathsf{GenSamplePre}(\mathbf{A}_R, \mathbf{A}_S, \mathbf{T}_S, \mathbf{v}, \gamma)$ to construct a ring trapdoor function and a ring signature on lattice. They use the idea of the lattice basis delegation technique by Cash *et al.* [12].

Let k, k_1, k_2, k_3, k_4 be positive integers as $k = k_1 + k_2 + k_3 + k_4$. We write $\mathbf{A}_S = [\mathbf{A}_{S_1} \mid \mathbf{A}_{S_2} \mid \mathbf{A}_{S_3} \mid \mathbf{A}_{S_4}] \in \mathbb{Z}_q^{n\times km}$ where $\mathbf{A}_{S_i} \in \mathbb{Z}_q^{n\times k_i m}$ for each i and $\mathbf{A}_R = [\mathbf{A}_{S_1} \mid \mathbf{A}_{S_3}] \in \mathbb{Z}_q^{n\times(k_1+k_3)m}$ with its trapdoor $\mathbf{T}_R$. Then, one can sample a preimage from a vector $\mathbf{y}$ as below:

GenSamplePre($\mathbf{A}_S, \mathbf{A}_R, \mathbf{T}_R, \gamma, \mathbf{y}$) :

a. Sample $\mathbf{e}_{S_2} \in \mathbb{Z}_q^{n\times k_2 m}$ and $\mathbf{e}_{S_4} \in \mathbb{Z}_q^{n\times k_4 m}$.
b. Let $\mathbf{z} = y - \mathbf{A}_{S_2}\mathbf{e}_{s_2} - \mathbf{A}_{S_4}\mathbf{e}_{s_4}$ and sample $\mathbf{e}_R = [\mathbf{e}_{S_1} \mid \mathbf{e}_{S_3}] \in \mathbb{Z}_q^{n\times(k_1+k_3)m}$ from $\mathsf{SamplePre}(\mathbf{A}_R, \mathbf{T}_R, \gamma, \mathbf{z})$.
c. Output $\mathbf{e} = [\mathbf{e}_{S_1} \mid \mathbf{e}_{S_2} \mid \mathbf{e}_{S_3} \mid \mathbf{e}_{S_4}]$.

2.4 Lattice Mixing and Vanishing Technique

Boyen [10] proposed the general framework to encode all bits at once by lattice trapdoor mixing and vanishing techniques.

He introduced a new trapdoor generation algorithm $\mathsf{TwoSideGen}(1^{\lambda})$ by slightly modifying Cash *et al.*'s extending trapdoor basis algorithm $\mathsf{ExtBasis}(\mathbf{T}, \mathbf{B})$, as below:

TwoSideGen(1^{λ}) :
For a security parameter λ, this algorithm outputs two random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{R} \in \mathbb{Z}_q^{m\times m}$ where $\mathbf{A}$ is uniform and $\mathbf{R}$ is from some distribution $\mathcal{R}$. Then, for some $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{F} = [\mathbf{A} \mid \mathbf{AR} + \mathbf{B}] \in \mathbb{Z}_q^{n\times 2m}$ and q defines the public parameters of a two-sided function.

$(\mathbf{F}, q)$ is a trapdoor function that samples the preimage with a trapdoor for either $\mathbf{A}$ or $\mathbf{B}$.

The characteristic of using a two-sided function is that we use the *firm* preimage trapdoor $\mathbf{T_A}$ that can always sample the preimage in the real scheme, whereas we use the *fickle* preimage trapdoor $\mathbf{T_B}$ for a matrix $\mathbf{B}$ which sometimes *vanishes* depending on a given message. With $\mathsf{TwoSideGen}(1^{\lambda})$ algorithm, Boyen constructed a signature $\mathcal{BS}$ to get shorter signatures as below:

B.KeyGen(1^{λ}) :
Given a security parameter λ and corresponding public parameters $n = n(\lambda), m = m(\lambda), q = q(\lambda)$,

1. Use $\mathsf{TrapGen}(n, m, q)$ to extract a matrix $\mathbf{A}_0$ and its trapdoor $\mathbf{T}_{\mathbf{A}_0}$.
2. Choose $t{+}1$ random matrix $\mathbf{R}_0, \mathbf{R}_1, \cdots, \mathbf{R}_t \in \mathbb{Z}^{m\times m}$ from discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}^m,\gamma}(\mathbf{x})$.

3. Choose t uniformly random integers $h_1, h_2, \cdots, h_t \in \mathbb{Z}_q$ and fix $h_0 = 1 \in \mathbb{Z}_q$.
4. Output a tuple $(\mathbf{A}_0, \{\mathbf{C}_i = \mathbf{A}_i\mathbf{R}_i + h_i\mathbf{B}_0\}_{i=0}^t)$ as a public key pk and $\mathbf{T}_{\mathbf{A}_0}$ as a secret key sk.

B.Sign$(sk, \mathbf{v})$:
Given a secret key sk and a message $\mathbf{v} \in \{0\} \times \{0,1\}^t$,

1. Define $\mathbf{C}_\mathbf{v} = \sum_{i=0}^t (-1)^{\mathbf{v}[i]}\mathbf{C}_i$ where $\mathbf{v}[i]$ is the i-th value of the message $\mathbf{v}$ and let a message-dependent matrix $\mathbf{F}_\mathbf{v} = [\mathbf{A}_0 \mid \mathbf{C}_\mathbf{v}] \in \mathbb{Z}_q^{n\times 2m}$.
2. Get the extending trapdoor basis $\mathbf{T}_\mathbf{F}$ of $\mathbf{F}_\mathbf{v}$ using $\mathsf{ExtBasis}(\mathbf{T}_{\mathbf{A}_0}, \mathbf{F}_\mathbf{v})$.
3. Sample a non-zero random vector $\mathbf{d} \in \Lambda^\perp(\mathbf{F}_\mathbf{v}) \subset \mathbb{Z}^{2m}$ using $\mathsf{SamplePre}(\mathbf{F}_\mathbf{v}, \mathbf{T}_\mathbf{F}, \mathbf{v}, \gamma)$ and output a signature $\sigma_\mathbf{v} = \mathbf{d}$.

B.Verify$(pk, \mathbf{v}, \sigma)$:
Given a public key pk, a message $\mathbf{v}$, and a signature $\sigma_\mathbf{v}$,

1. Check that $\mathbf{v}$ is in $\{0\} \times \{0,1\}^t$ and $\sigma_\mathbf{v}$ is a small non-zero vector, *i.e.*, $0 < \|\sigma_\mathbf{v}\| \le \gamma\sqrt{2m}$.
2. Check that $\sigma_\mathbf{v}$ satisfies that
$$\left[\mathbf{A}_0 \mid \sum_{i=0}^t (-1)^{\mathbf{v}[i]}\mathbf{C}_i\right]\mathbf{d} = \mathbf{0} \bmod q$$
3. If both are correct, accept the signature. Otherwise, reject.

3 Linearly Homomorphic Signature

We restate the formal definition and security requirements of linearly homomorphic signature over binary fields from Boneh and Freeman's work [4]. Then, we illustrate detailed construction.

3.1 Definition and Security Requirements

Boneh and Freeman [4] defined the linearly homomorphic signature over binary fields $\mathcal{LHS}$ as below:

Definition 2. (linearly homomorphic signature). A linearly homomorphic signature $\mathcal{LHS}$ over $\mathbb{F}_2$ is a tuple of PPT algorithms $\mathcal{LHS}$ = (**Setup**, **Sign**, **Combine**, **Verify**) with the following functionality:

Setup(n, **params**) :
Given the security parameter n and other public parameters **params**, this algorithm outputs a public key pk and a secret key sk.

Sign(sk, id, $\mathbf{v}$) :
Given a secret key sk, a tag id and a vector $\mathbf{v}$, this algorithm outputs a signature σ.

Combine$(pk, id, \{(\alpha_i, \sigma_i)\}_{i=1}^l)$:
Given a public key pk, a tag id and pairs $\{(\alpha_i, \sigma_i)\}_{i=1}^l$ where $\alpha_i \in \mathbb{F}_2 = \{0,1\}$ and σ_i is the signature of a vector $\mathbf{v}_i$ for each i, this algorithm outputs a signature σ for a vector $\Sigma_{i=1}^l \alpha_i \mathbf{v}_i$.

Verify(pk, id, $\mathbf{y}$, σ) :
Given a public key pk, a tag id, a vector $\mathbf{y}$ and a signature σ, this algorithm outputs either 0 (reject) or 1 (accept).

To check the correctness, for each (pk, sk), we must have

a. For all tags id and all vectors $\mathbf{y}$, the verification algorithm **Verify**(pk, id, $\mathbf{y}$, σ) outputs 1 for all valid signatures $\sigma \leftarrow$ **Sign**$(sk, id, \mathbf{y})$.

b. Whenever we operate a linear combination of some vectors $\{\mathbf{v}_i\}_{i=1}^l$, we can output the valid signature for that linear combination.

The security requirements of linearly homomorphic signature are *unforgeability* and *weakly context hiding* property as below:

Definition 3. (unforgeability). A linearly homomorphic signature is *unforgeable* if the advantage of any PPT adversary $\mathcal{A}$, in the following security game is negligible in the security parameter n.

Setup :
The challenger $\mathcal{C}$ sets $(pk, sk) \leftarrow$ **Setup**(n, **params**), then sends the public key pk to $\mathcal{A}$.

Queries :
Proceeding adaptively, $\mathcal{A}$ specifies a sequence of k-dimensional subspaces V_i with basis vectors $\{\mathbf{v}_j^{(i)}\}_{j=1}^k$. For each i, $\mathcal{C}$ chooses a tag $id_i \leftarrow \{0,1\}^n$ uniformly and gives id_i with j signatures $\sigma_{ij} \leftarrow$ **Sign**$(sk, id_i, \mathbf{v}_j^{(i)})$ for $j = 1, 2, \cdots, k$.

Output :
$\mathcal{A}$ outputs a tag $id^* \in \{0,1\}^n$, a non-zero vector $\mathbf{y}^*$, and a signature σ^*.

$\mathcal{A}$ wins the game if the signature σ is valid and either (1) $id^* \neq id_i$ for all i, or (2) $id^* = id_i$ for some i but $\mathbf{y}^* \notin V_i$.

Definition 4. (weakly context hiding). A linearly homomorphic signature is *weakly context hiding* if the advantage of any PPT adversary $\mathcal{A}$, in the following security game is negligible in the security parameter n.

Setup :
The challenger $\mathcal{C}$ sets $(pk, sk) \leftarrow$ **Setup**(n, **params**) and sends both public key pk and secret key sk to $\mathcal{A}$.

Challenge :

$\mathcal{A}$ outputs two k-dimensional vector spaces V_0, V_1 with basis vectors $\{\mathbf{v}_i^{(0)}\}_{i=1}^k$ and $\{\mathbf{v}_i^{(1)}\}_{i=1}^k$, respectively and linear functions on both $\{\mathbf{v}_i^{(0)}\})_{i=1}^k$ and $\{\mathbf{v}_i^{(1)}\}_{i=1}^k$ which satisfies

$$f_j\left(\mathbf{v}_1^{(0)}, \mathbf{v}_2^{(0)}, \cdots, \mathbf{v}_k^{(0)}\right) = f_j\left(\mathbf{v}_1^{(1)}, \mathbf{v}_2^{(1)}, \cdots, \mathbf{v}_k^{(1)}\right)$$

for all $j = 1, 2, \cdots, s$.

$\mathcal{C}$ chooses $b \in \{0,1\}$ and a tag $id \in \{0,1\}^n$ and signs the vector space V_b with a tag id.

Then, $\mathcal{C}$ uses **Combine**$(pk, id, \{(\alpha_i, \sigma_i)\}_{i=1}^k)$ algorithm to derive signatures σ_j of the function $f_j\left(\mathbf{v}_1^{(b)}, \mathbf{v}_2^{(b)}, \cdots, \mathbf{v}_k^{(b)}\right)$ for all $j = 1, 2, \cdots, s$.

$\mathcal{A}$ gets signatures σ_j. The function can be out adaptively after choosing V_0 and V_1.

Output :

$\mathcal{A}$ outputs a bit b'.

$\mathcal{A}$ wins the game if $b = b'$.

3.2 Construction by Boneh and Freeman

We let the public parameters **params**$=(N, k, L, m, q, \gamma)$ where $N = n$ is the dimension of vectors to be signed, k is the dimension of the subspace to be signed ($k < n$), L is the maximum number of signatures in linear combinations, $m(n, L) > n$ is an integer, $q(n, L)$ is an odd prime, and $\gamma(n, L)$ is a real number.

With those parameters, Boneh and Freeman [4] presented the first linearly homomorphic signature over binary fields with a tuple of PPT algorithms $\mathcal{LHS}$ = (**Setup**, **Sign**, **Combine**, **Verify**) with the following functionality:

Setup$(n$, **params**$)$:

Given a security parameter n and public parameters **params**$=(N, k, L, m, q, \gamma)$,

1. $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TrapGen}(n, m, 2q)$ where a matrix $\mathbf{A} \in \mathbb{Z}_{2q}^{n\times m}$ and its trapdoor basis $\mathbf{T}$ of $\Lambda_{2q}^{\perp}(\mathbf{A})$ satisfies that $\|\widetilde{\mathbf{T}}\| \leq 30\sqrt{n \log 2q}$.
2. Let $H : \{0,1\}^* \to \mathbb{Z}_{2q}^{n\times m}$ be a hash function, viewed as a random oracle.
3. Output the public key $pk \leftarrow (\mathbf{A}, H)$ and the secret key $sk \leftarrow (\mathbf{A}, H, \mathbf{T})$.

Sign$(sk, id$, $\mathbf{v})$:

Given a secret key $sk \leftarrow (\mathbf{A}, H, \mathbf{T})$, a tag $id \in \{0,1\}^n$ and a vector $\mathbf{v} \in \mathbb{F}_2^n$,

1. Set $\mathbf{B} \leftarrow \mathbf{A}\|H(id) \in \mathbb{Z}_{2q}^{n\times 2m}$.
2. Let $\mathbf{S} \leftarrow \mathsf{ExtBasis}(\mathbf{T}, \mathbf{B})$ be a basis for $\Lambda_{2q}^{\perp}(B)$ with $\|\widetilde{\mathbf{S}}\| = \|\widetilde{\mathbf{T}}\|$.
3. Output $\sigma \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathbf{S}, \gamma, q\cdot\mathbf{v})$.

Combine$(pk, id, \{(\alpha_i, \sigma_i)\}_{i=1}^l)$:

Given a public key $pk = (\mathbf{A}, H)$, a tag $id \in \{0,1\}^n$ and pairs $\{(\alpha_i, \sigma_i)\}_{i=1}^l$ where $\alpha_i \in \mathbb{F}_2 = \{0,1\}$ and σ_i is a signature of the i-th vector $\mathbf{v}_i$, output $\sigma \leftarrow \Sigma_{i=1}^l \alpha_i\sigma_i \in \mathbb{Z}^{2m}$.

Verify$(pk, id$, $\mathbf{y}$, $\sigma)$:

Given a public key $pk = (\mathbf{A}, H)$, a tag $id \in 0, 1^n$, a vector $\mathbf{y} \in \mathbb{F}_2^n$ and a signature $\sigma \in \mathbb{Z}^{2m}$,

1. Set $\mathbf{B} \leftarrow [\mathbf{A} \mid H(id)] \in \mathbb{Z}_{2q}^{n\times 2m}$.
2. If $\|\sigma\| \leq L\cdot\gamma\sqrt{2m}$ and $\mathbf{B}\cdot\sigma = q\cdot\mathbf{y} \bmod 2q$, output 1 (accept). Otherwise, output 0 (reject).

Lemma 3. *Let $\mathcal{LHS}$ be the linearly homomorphic signature over $\mathbb{F}_2$ as above. Suppose q be a prime, n, m be integers with $m > 2n\log q$, and $\gamma > 30\sqrt{n\log 2q}\cdot\omega(\sqrt{\log n})$. Then $\|\sigma\| \leq L\cdot\gamma\sqrt{2m}$ and $\boldsymbol{B}\cdot\sigma = q\cdot\boldsymbol{y} \bmod 2q$ for all valid signatures $\sigma \leftarrow$ **Combine**$(pk, id, \{(\alpha_i, \sigma_i)\}_{i=1}^l)$*

Moreover, **Lemmas 4** and **5** from Boneh and Freeman's work show that this signature is unforgeable in the random oracle model and it holds the weakly context hiding property [4].

Lemma 4. *Let $\mathcal{LHS}$ be the linearly homomorphic signature over $\mathbb{F}_2$ as above. Suppose that $m = \lceil 6n\log 2q\rceil$ and $\gamma = 30\sqrt{n\log 2q}\log n$. Let $\beta = L\cdot\gamma\sqrt{2m}$. Then $\mathcal{LHS}$ is unforgeable in the random oracle model assuming that k-$SIS_{q,2m,\beta,\gamma}$ problem is infeasible.*

Lemma 5. *Let $\mathcal{LHS}$ be the linearly homomorphic signature over $\mathbb{F}_2$ as above. Suppose that $k < \dfrac{\log n}{2\log\log n}$, $m = \lceil 6n\log 2q\rceil$ and $\gamma = 30\sqrt{n\log 2q}\log n$. Then $\mathcal{LHS}$ is weakly context hiding.*

4 Design of New Linearly Homomorphic Signatures

In this section, we define the linearly homomorphic ring signature and its security requirements and design the linearly homomorphic ring signature using a new preimage sampling algorithm $\mathsf{GenSamplePre}(\mathbf{A}_R, \mathbf{A}_S, \mathbf{T}_S, \mathbf{v}, \gamma)$ by Wang and Sun [9]. We also construct the signature scheme with short signature using lattice mixing and vanishing technique by Boyen [10].

We set the public parameters **params**$=(N, k, L, m, q, \gamma)$ same as the signature by Boneh and Freeman in Section 3.2.

4.1 Linearly Homomorphic Ring Signature

In a ring signature, a signer chooses any subset of all possible signers including himself/herself to form a ring, without getting their permission [17]. Thus, ring signature provides the anonymity of the signer since the signature of the message only convinces that one member in the ring signed the message without revealing a signer's identity. We define the linearly homomorphic ring signature using a new preimage sampling algorithm $\mathsf{GenSamplePre}(\mathbf{A}_R, \mathbf{A}_S, \mathbf{T}_S, \mathbf{v}, \gamma)$ by Wang and Sun [9] as below:

Definition 5. (linearly homomorphic ring signature). A linearly homomorphic ring signature $\mathcal{LHRS}$ is a tuple of PPT algorithms $\mathcal{LHRS}$ = (**R.Setup**, **R.Sign**, **R.Combine**, **R.Verify**) with the following functionality:

R.Setup(n, params) :
Given the security parameter n and public parameters **params**, this algorithm outputs a public key pk and a secret key sk.

R.Sign(pk, sk, id, R, v) :
Given a key pair (pk, sk) of a signer where $pk \in R$, a tag id, a public key R of the ring, and a vector $\mathbf{v}$, this algorithm outputs a signature σ of the vector $\mathbf{v}$ under sk.

R.Combine($R, id, \{(\alpha_i, \sigma_i)\}_{i=1}^{l}$) :
Given a public key R of the ring, a tag id, and pairs $\{(\alpha_i, \sigma_i)\}_{i=1}^{l}$ where $\alpha_i \in \mathbb{F}_2 = \{0, 1\}$ and σ_i is the signature of a vector $\mathbf{v}_i$ for each i, this algorithm outputs a signature σ for a vector $\Sigma_{i=1}^{l} \alpha_i \mathbf{v}_i$.

R.Verify(R, id, y, σ) :
Given a public key R of the ring, a tag id, a vector $\mathbf{y}$, and a signature σ, this algorithm outputs either 0 (reject) or 1 (accept).

To check the correctness, for each (pk, sk), we must have

a. For all key pairs (pk_i, sk_i) where $pk_i \in R$, tags id, and all vectors $\mathbf{y}$, the verification algorithm $\mathbf{Verify}(R, id, \mathbf{y}, \sigma)$ outputs 1 for all valid signatures $\sigma \leftarrow \mathbf{Sign}(pk_i, sk_i, id, R, \mathbf{y})$.

b. Whenever we operate a linear combination of some vectors $\{\mathbf{v}_i\}_{i=1}^{l}$, we can output the valid signature for that linear combination.

The security requirements of linearly homomorphic signature are *unforgeability* and *weakly context hiding* property for linearly homomorphic ring signatures as well as *anonymity* like other ring signature. Here, we only give the formal definition of the anonymity since the others are analogous to the one for linearly homomorphic signatures.

Definition 6. (anonymity). A linearly homomorphic ring signature is *anonymous* if the advantage of any PPT adversary $\mathcal{A}$, in the following security game is negligible in the security parameter n.

Setup :
The challenger $\mathcal{C}$ obtains $(pk_i, sk_i) \leftarrow$ **R.Setup** $(n, \mathbf{params})$ r times where r is the size of the ring $\mathcal{R}$, then sends public keys $\{pk_i\}_{i=1}^{r}$ to $\mathcal{A}$.

Queries :
$\mathcal{A}$ specifies the pair $(i, R, \mathbf{v})$ where i is a signer index, R is a public key of the ring $\mathcal{R}$, and $\mathbf{v}$ is a vector to be signed. Then, the challenger $\mathcal{C}$ chooses a tag $id_j \leftarrow \{0,1\}^n$ uniformly and gives id_j with a signature $\sigma_i \leftarrow \mathbf{R.Sign}(pk, sk,\ id_i, R, \mathbf{v})$.

Challenge :
$\mathcal{A}$ requests a challenge by sending $(i_0, i_1, R^*, \mathbf{v}^*)$ to $\mathcal{C}$, where i_0 and i_1 are signer indices, R^* is a public key of the ring $\mathcal{R}^*$ which contains pk_{i_0} and pk_{i_1}, and $\mathbf{v}^*$ is a vector to be signed. a non-zero vector $\mathbf{y}^*$, and a signature σ^*.
Then, $\mathcal{C}$ chooses a bit $b \leftarrow \{0, 1\}$ and a tag $id^* \leftarrow \{0,1\}^n$ and sends a challenge signature $\sigma_b \leftarrow \mathbf{R.Sign}(pk_{i_b}, sk_{i_b}, id^*, R^*, \mathbf{v}^*)$ to the adversary $\mathcal{A}$.

Output :
$\mathcal{A}$ outputs a bit b'.

$\mathcal{A}$ wins the game if $b = b'$.

With these definition and security requirements, we design the linearly homomorphic ring signature on lattice as below:

R.Setup(n, g, params) :
Given a security parameter n, a number of all possible signers g, and public parameters **params** $=(N, k, L,\ m, q, \gamma)$, do the following:

1. Run $\mathsf{TrapGen}(n, m, 2q)$ to generate a matrix $\{\mathbf{A}_i\}_{i=1}^{g} \in \mathbb{Z}_{2q}^{n\times m}$ and its corresponding trapdoor basis $\{\mathbf{T}_i\}_{i=1}^{g}$ of $\Lambda_{2q}^{\perp}(\mathbf{A}_i)$ such that $\|\widetilde{\mathbf{T}}_i\| \leq 30\sqrt{n \log 2q}$.
2. Let $H : \{0,1\}^* \to \mathbb{Z}_{2q}^{n\times m}$ be a hash function, viewed as a random oracle and choose the ring $\mathcal{R}$.
3. Output the public key $pk_i = (\mathbf{A}_i, H)$ and the secret key $sk_i = \mathbf{T}_i$ for each signer i of the ring and R is a subset of public keys of all possible signers including pk_i to form a ring $\mathcal{R}$.

R.Sign(pk_i, sk_i, id, R, v) :
For a key pair $(pk_i, sk_i) = (\mathbf{A}_i, \mathbf{T}_i)$ of a signer where $pk_i \in R$ when the size of the ring is r, a tag $id \in \{0,1\}^n$, and a vector $\mathbf{v}_i \in \mathbb{F}_2^n$, do the following:

1. Set a matrix $\mathbf{A}_R = [\mathbf{A}_1 \mid \mathbf{A}_2 \mid \cdots \mid \mathbf{A}_r \mid H(id)] \in \mathbb{Z}_{2q}^{n\times(r+1)m}$.
2. Output a signature $\sigma \leftarrow \mathsf{GenSamplePre}(\mathbf{A}_R, \mathbf{A}_S, \mathbf{T}_S, q \cdot \mathbf{v}, \gamma)$.

R.Combine($R, id, \{(\alpha_j, \sigma_j)\}_{j=1}^{l}$) :
Given a public key R of the ring of size r, a hash function H, a tag $id \in \{0,1\}^n$, and set of signatures $\{\sigma_j\}_{j=1}^{r}$, output $\sigma = \sum_{j=1}^{r} \sigma_i \in \mathbb{Z}^{(r+1)m}$.

R.Verify$(R, H, id,\ \mathbf{y},\ \sigma)$:
Given a public key R of the ring with the size r, a hash function H, a tag $id \in \{0,1\}^n$, a vector $\mathbf{y} \in \mathbb{F}_2^n$, and a signature $\sigma \in \mathbb{Z}^{(r+1)m}$, do the following:

1. Set a matrix $\mathbf{A}_R = [\mathbf{A}_1 \mid \mathbf{A}_2 \mid \cdots \mid \mathbf{A}_r \mid H(id)] \in \mathbb{Z}_{2q}^{n\times(r+1)m}$.
2. If $\|\sigma\| \le L \cdot \gamma\sqrt{(r+1)m}$ and $\mathbf{A}_R \cdot \sigma = q \cdot \mathbf{y} \bmod 2q$, output 1 (accept). Otherwise, output 0 (reject).

We believe that the above construction holds the linearly homomorphic property for signatures from the same ring $\mathcal{R}$.

4.2 Linearly Homomorphic Signature with Short Signatures

We design a new linearly homomorphic signature with short signatures as a tuple of PPT algorithms $\mathcal{SLH} = (\mathbf{S.Setup}, \mathbf{S.Sign}, \mathbf{S.Combine}, \mathbf{S.Verify})$ by adopting lattice mixing and vanishing technique by Boyen [10] as below:

S.Setup$(n,\ \mathbf{params})$:
Given a security parameter n and public parameters **params**$=(N, k, L,\ m, q, \gamma)$, do the following:

1. Use $\mathsf{TrapGen}(n, m, 2q)$ to extract a matrix $\mathbf{A}_0$ and its trapdoor $\mathbf{T}_{\mathbf{A}_0}$ such that $\|\widetilde{\mathbf{T}}_{\mathbf{A}_0}\| \le 30\sqrt{n \log 2q}$.
2. Choose $n+1$ random matrix $\mathbf{R}_0, \mathbf{R}_1, \cdots, \mathbf{R}_n \in \mathbb{Z}^{m\times m}$ from discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}^m,\gamma}(\mathbf{x})$.
3. Choose n uniformly random integers $h_1, h_2, \cdots, h_n \in \mathbb{Z}_{2q}$ and fix $h_0 = 1 \in \mathbb{Z}_{2q}$.
4. Let $H : \{0,1\}^* \to \mathbb{Z}_{2q}^{n\times m}$ be a hash function, viewed as a random oracle.
5. Calculate $\mathbf{C}_i = \mathbf{A}_i\mathbf{R}_i + h_i\mathbf{B}_0$ for each i and output a public key $pk = (\mathbf{A}_0, \{\mathbf{C}_i\}_{i=0}^n, H)$ and a secret key $sk = \mathbf{T}_{\mathbf{A}_0}$.

S.Sign$(sk, id,\ \mathbf{v})$:
Given a secret key $sk = \mathbf{T}_{\mathbf{A}_0}$, a tag $id \in \{0,1\}^n$ and a vector $\mathbf{v} \in \{0,1\}^n$, do the following:

1. Define $\mathbf{C}_\mathbf{v} = \mathbf{C}_0 + \sum_{i=1}^n (-1)^{\mathbf{v}[i]}\mathbf{C}_i$ where $\mathbf{v}[i]$ is the i-th value of the message $\mathbf{v}$ and set $\mathbf{F}_\mathbf{v} = [\mathbf{A}_0 \mid \mathbf{C}_\mathbf{v} \mid H(id)] \in \mathbb{Z}_q^{n\times 3m}$ as a message-dependent matrix.
2. Get the extending trapdoor basis $\mathbf{T}_\mathbf{F}$ of $\mathbf{F}_\mathbf{v}$ using $\mathsf{ExtBasis}(\mathbf{T}_{\mathbf{A}_0}, \mathbf{F}_\mathbf{v})$.
3. Sample a non-zero random vector $\mathbf{d} \in \Lambda^\perp(\mathbf{F}_\mathbf{v}) \subset \mathbb{Z}^{3m}$ using $\mathsf{SamplePre}(\mathbf{F}_\mathbf{v}, \mathbf{T}_\mathbf{F}, q\cdot\mathbf{v}, \gamma)$ and output a signature $\sigma_\mathbf{v} = \mathbf{d}$ of the vector $\mathbf{v}$.

S.Combine$(pk, id, \{(\alpha_j, \sigma_j)\}_{j=1}^l)$:
Given a public key $pk = (\mathbf{A}_0, \{\mathbf{C}_i\}_{i=0}^n, H)$, a tag $id \in \{0,1\}^n$, and pairs $\{(\alpha_i, \sigma_i)\}_{i=1}^l$ where $\alpha_i \in \{0,1\}$ and σ_j is a signature of the j-th vector $\mathbf{v}_j$, output $\sigma = \sum_{j=1}^l \alpha_j\sigma_j \in \mathbb{Z}^{3m}$.

S.Verify$(pk, id,\ \mathbf{y},\ \sigma)$:
Given a public key $pk = (\mathbf{A}, H)$, a tag $id \in \{0,1\}^n$, a vector $\mathbf{y} \in \{0,1\}^t$, and a signature $\sigma \in \mathbb{Z}^{3m}$, do the following:

1. Check that $\mathbf{v}$ is in $\{0,1\}^n$ and $\sigma_\mathbf{v}$ is a small non-zero vector, *i.e.*, $0 < \|\sigma_\mathbf{v}\| \le L \cdot \gamma\sqrt{3m}$.
2. Set $\mathbf{C}_\mathbf{y} = \mathbf{C}_0 + \sum_{i=1}^n (-1)^{\mathbf{v}[i]}\mathbf{C}_i$ and check that $\sigma_\mathbf{v}$ satisfies that
$$[\mathbf{A}_0 \mid \mathbf{C}_\mathbf{y} \mid H(id)]\,\mathbf{d} = q \cdot \mathbf{y} \bmod 2q.$$
3. If both are correct, accept the signature. Otherwise, reject.

We expect that the above construction extracts the shorter signatures than the signature from Boneh and Freeman's signature scheme.

5 Concluding Remark

We have two different methods to enhance lattice-based linearly homomorphic signature scheme over binary fields by modifying linearly homomorphic signature by Boneh and Freeman [4]. We design the linearly homomorphic ring signature and linearly homomorphic signature with short signatures.

But, we haven't proved the validity of the suggested schemes or their security analysis yet. Thus, first of all, we have to give the concrete proof of designed signatures so that these signatures can be safely applicable to the real world scenario.

Aside from this, it will be also challenging to design new fully homomorphic signatures by adding new functionalities to the existing homomorphic trapdoor functions like lattice mixing and vanishing technique used in this paper or by changing the hard problems on lattice from SIS problem to Ring-SIS, LWE or LWR problems.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. NRF-2015R1A2A2A01006812).

References

[1] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Forty-First Annual ACM on Symposium on Theory of Computing*, pp. 169–178, ACM, 2009.

[2] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) lwe," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, 2014.

[3] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology–CRYPTO 2013*, pp. 75–92, Springer, 2013.

[4] D. Boneh and D. M. Freeman, "Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures," in *Public Key Cryptography–PKC 2011*, pp. 1–16, Springer, 2011.

[5] D. Boneh and D. M. Freeman, "Homomorphic signatures for polynomial functions," in *Advances in Cryptology–EUROCRYPT 2011*, pp. 149–168, Springer, 2011.

[6] S. Gorbunov, V. Vaikuntanathan, and D. Wichs, "Leveled fully homomorphic signatures from standard lattices," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pp. 469–477, ACM, 2015.

[7] D. Fiore, A. Mitrokotsa, L. Nizzardo, and E. Pagnin, "Multi-key homomorphic authenticators," in *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Part II*, pp. 499–530, Springer, 2016.

[8] R. Choi and K. Kim, "A classification of lattice-based trapdoor functions," in *Proceedings of the Forty-Fourth Symposium on Cryptography and Information Security*, 2017. (to appear).

[9] J. Wang and B. Sun, "Ring signature schemes from lattice basis delegation," in *International Conference on Information and Communications Security*, pp. 15–28, Springer, 2011.

[10] X. Boyen, "Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more," in *Public Key Cryptography–PKC 2010*, pp. 499–517, Springer, 2010.

[11] P. Zhang, J. Yu, and T. Wang, "A homomorphic aggregate signature scheme based on lattice," *Chinese Journal of Electronics*, vol. 21, no. 4, pp. 701–704, 2012.

[12] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," *Journal of Cryptology*, vol. 25, no. 4, pp. 601–639, 2012.

[13] Z. Jing, "An efficient homomorphic aggregate signature scheme based on lattice," *Mathematical Problems in Engineering*, 2014. Avaliable online at `http://dx.doi.org/10.1155/2014/536527`.

[14] R. Choi and K. Kim, "Lattice-based multi-signature with linear homomorphism," in *Proceedings of the Forty-Third Symposium on Cryptography and Information Security*, 1D1-3, 2016.

[15] J. Alwen and C. Peikert, "Generating shorter bases for hard random lattices," *Theory of Computing Systems*, vol. 48, no. 3, pp. 535–553, 2011.

[16] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the Fortieth Annual ACM on Symposium on Theory of Computing*, pp. 197–206, ACM, 2008.

[17] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 552–565, Springer, 2001.