SCIS 2018 2018 Symposium on
Cryptography and Information Security
Niigata, Japan, Jan. 23 - 26, 2018
The Institute of Electronics,
Information and Communication Engineers

Revisiting CK17 Linearly Homomorphic Ring Signature based on SIS

Rakyong Choi * Kwangjo Kim *

Abstract: In SCIS 2017, Choi and Kim introduced the new linearly homomorphic ring signature scheme (CK17 scheme) based on the hardness of SIS problem, which overcomes the limitation of Boneh and Freeman's scheme to implement homomorphic signatures to the real world scenario under multiple signers setting for a message. They replace the original sampling algorithm SamplePre() by Gentry et al. with Wang and Sun's sampling algorithm GenSamplePre() to achieve the multiple-signer functionality but their work is lack of the rigorous security proof. Thus, this paper revisits the CK17 scheme and makes an advanced definition which is subring-identical linearly homomorphic signature, and suggests a security requirements on it. Then, we show the correctness and subring-identical linear homomorphism of the proposed scheme.

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

1 Introduction

1.1 Background and Motivation

Ring signature is a kind of group-oriented signatures which allow a member of a group to sign a message on behalf of the whole group. Ring signature provides the anonymity of the signer since the verifier cannot reveal who is the real signer in the group. In a ring signature scheme, a designated signer forms a ring of any set of possible signers including himself. The message signer can then generate a ring signature using his secret key and public keys of other members of the ring. Ring signature can be applied to many applications such as anonymous information source, cryptocurrency, *etc.*

Cloud computing system is one possible application area of a ring signature scheme. As the infrastructure of cloud computing systems increases, one of uprising security challenges is how the cloud server provides authenticity for the function of encrypted message via a signature scheme. Moreover, the cloud server should have the power to generate the proper signature for a computation of messages without permission of a single signer of each message.

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 [1, 2] and *fully homomorphic* when it supports constructing the proper signature for any function of messages [3, 4].

In 2017, Choi and Kim considered the convincing scenario that some information on cloud system is signed by a group instead of an individual. They define the linearly homomorphic ring signature and suggest a lattice-based linearly homomorphic ring signature scheme over binary fields by adopting Wang and Suns preimage sampling algorithm GenSamplePre(). But no security proof is given in their paper.

Thus, we revisits the CK17 scheme and suggests an advanced definition which is subring-identical linearly homomorphic signature with security requirements.

1.2 Related Work

In 2011, Boneh and Freeman [1] 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 [2] also suggested that some bounded homomorphic signature can be constructed using ideal lattices from Gentry's fully homomorphic encryption [5].

After Boneh and Freeman's work, lattices have become a main tool to make linearly and fully homomorphic signatures. Zhang *et al.* [6] introduced the notion 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.* [7] to generate multiple secret keys.

Jing [8] 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.* [7]. Both Zhang *et al.* [6] and Jing's [8] contributions are making multi-key linearly homomorphic signatures.

Choi and Kim [9, 10] suggested the concept of the linearly homomorphic multisignature and linearly ho-

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

momorphic ring signature. In these works, Choi and Kim 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.* [3] suggested the first fully homomorphic signature scheme with a homomorphic trapdoor function but there is only one secret key. Then, Fiore *et al.* [4] suggested a fully homomorphic signature scheme with multi-key (*i.e.,* multiple secret keys) setting.

1.3 Outline of the Paper

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

We give the security proof of our proposed scheme in Section 4 and make 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 Lattice-based Algorithms

Briefly, lattice is 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 Learning With Errors (LWE) 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 [11] 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.* [7] 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

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 $\rho_{\gamma,\mathbf{c}}(\mathbf{x}) = \exp\left(\frac{-\pi\|\mathbf{x}\text{-}\mathbf{c}\|^2}{\gamma^2}\right)$ 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.* [12] 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. *[12] 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. *[1] 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 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 [13] 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.* [7].

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 $\mathbf{A}_S = [\mathbf{A}_{S_1} \mid \mathbf{A}_{S_3}] \in \mathbb{Z}_q^{n\times(k_1+k_3)m}$ with its trapdoor $\mathbf{T}_S$. Then, one can sample a preimage from a vector $\mathbf{y}$ as below:

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

a. Sample $\mathbf{e}_{R_2} \in \mathbb{Z}_q^{k_2 m}$ and $\mathbf{e}_{R_4} \in \mathbb{Z}_q^{k_4 m}$.

b. Let $\mathbf{z} = \mathbf{y} - \mathbf{A}_{R_2}\mathbf{e}_{R_2} - \mathbf{A}_{R_4}\mathbf{e}_{R_4}$ and sample $\mathbf{e}_S = [\mathbf{e}_{R_1} \mid \mathbf{e}_{R_3}] \in \mathbb{Z}_q^{(k_1+k_3)m}$ from SamplePre $(\mathbf{A}_S, \mathbf{T}_S, \gamma, \mathbf{z})$.

c. Output $\mathbf{e} = [\mathbf{e}_{S_1} \mid \mathbf{e}_{S_2} \mid \mathbf{e}_{S_3} \mid \mathbf{e}_{S_4}]$.

3 Subring-Identical Linearly Homomorphic Ring Signature

3.1 Definition

In a ring signature, a signer chooses any subset of all possible signers including himself/herself to form a ring, without getting their permission [14]. 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 [13] as below:

Definition 2. (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, 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 **Verify**$(R, id, \mathbf{y}, \sigma)$ outputs 1 for all valid signatures $\sigma \leftarrow$ **R.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.

In this paper, we define a new concept called subring-identical linearly homomorphic ring signature by restricting the availability of linear homomorphism to identical subrings.

Definition 3. (subring-identical linearly homomorphic ring signature). A subring-identical linearly homomorphic ring signature $\mathcal{SILHR}$ is a tuple of PPT algorithms $\mathcal{SILHR} =$ (**SI.Setup**, **SI.Sign**, **SI.Combine**, **SI.Verify**) with the following functionality:

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

SI.Sign($pk, sk, id, R,$ v) :
Given a key pair (pk, sk) of a designated signer where $pk \in R$, a tag id, a public key R of the ring, and a vector $\mathbf{v}$, a signer chooses the subring $\mathcal{S}$ to make the signature. Then, this algorithm outputs a ring signature σ of the vector $\mathbf{v}$ under sk and a label lab_S for a subring $\mathcal{S} \subset \mathcal{R}$.

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

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

To check the correctness, 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 **Verify**$(R, id, \mathbf{y}, \sigma, lab_S)$ always outputs 1 for all valid signatures $\sigma \leftarrow$ **SI.Sign**$(pk_i, sk_i, id, R, \mathbf{y})$.

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

If the scheme holds the above property, we say that the scheme is *subring-identical linearly homomorphic.*

3.2 Security Requirements

The security requirements of our scheme adopts *unforgeability* and *weakly context hiding* property from linearly homomorphic signatures as well as *anonymity* from other ring signature. Here, based on the former research on ring signatures by Bender *et al.* [15], we define the security requirements of linearly homomorphic ring signature. For unforgeability, we define unforgeability against fixed-ring attack.

Definition 4. (unforgeability against fixed-ring attack). A linearly homomorphic ring signature is *unforgeable against fixed-ring attack* if the advantage of any PPT adversary $\mathcal{A}$, in the following security game is negligible in the security parameter n.

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

Queries :
Proceeding adaptively, $\mathcal{A}$ queries the signing query **SI.Sign**$(pk_s, sk_s, id_s, R, \mathbf{v}_s)$ to extract σ_s.

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

$\mathcal{A}$ wins the game if the signature σ^* is valid and **SI.Sign**$(\cdot, \cdot, id^*, R, \mathbf{y}^*)$ is never queried, *i.e.,* either (1) id^* is never queried or (2) $id^* = id_i$ for some signing query but $\mathbf{y}^*$ is not queried by the adversary.

Definition 5. (weakly context hiding). A linearly homomorphic ring 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 $(pk, sk) \leftarrow$ **SI.Setup**$(n,$ **params**$)$ and sends both public key pk and secret key sk to $\mathcal{A}$.

Challenge :
$\mathcal{A}$ outputs two 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 **SI.Combine**$(pk, id, \{(\alpha_i, \sigma_i)\}_{i=1}^{k}, lab_S)$ 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 f_j can be selected adaptively after choosing V_0 and V_1.

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

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

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. (*i.e.,* success probability of the adversary is close to $\frac{1}{2}$.)

Setup :
The challenger $\mathcal{C}$ generates key pairs $\{pk_i, sk_i\}_{i=1}^{r}$ $\leftarrow$ **SI.Setup** $(n,$ **params**$)$ 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 set of public keys of the ring $\mathcal{R}$, and $\mathbf{v}$ is a vector to be signed. Then, the challenger $\mathcal{C}$ chooses a tag $id_i \leftarrow \{0,1\}^n$ uniformly and gives id_i with a signature $\sigma_i \leftarrow$ **SI.Sign**$(pk_i, sk_i, id_i, R, \mathbf{v})$.

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

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

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

3.3 Concrete Design

To design lattice-based linearly homomorphic ring signature scheme, Choi and Kim [10] let each member of the ring take their own public key and secret key by trapdoor generation function $\mathsf{TrapGen}()$ during the setup phase. Then, they concatenated the public key of each member to make the common public key. In the signing phase, CK17 scheme modified the preimage sampling algorithm from well-known $\mathsf{SamplePre}()$ to $\mathsf{GenSamplePre}()$ to make the ring homomorphic signature.

We further modify algorithms **R.Sign**, **R.Combine**, and **R.Verify** of original paper by Choi and Kim [10] to adjust our new definition of subring-identical linearly homomorphic ring signature scheme.

SI.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 $H : \{0,1\}^* \rightarrow \mathbb{Z}_{2q}^{n\times m}$ be a hash function, viewed as a random oracle and choose the ring $\mathcal{R} = \{1, 2, \cdots, r\}$.
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 $\mathcal{R}$.

SI.Sign$(pk_i, sk_i, id, R,$ v$)$:
For a key pair $(pk_i, sk_i) = (\mathbf{A}_i, \mathbf{T}_i)$ of a designated signer i 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. Choose the subring $\mathcal{S} \subset \mathcal{R}$ that generates the signature of the given vector.
2. Set a matrix $\mathbf{A}_R = [\mathbf{A}_1 \mid \mathbf{A}_2 \mid \cdots \mid \mathbf{A}_r \mid H(id)] \in \mathbb{Z}_{2q}^{n\times(r+1)m}$.
3. Run $\mathsf{ExtBasis}(\mathbf{T}_i, \mathbf{S})$ to get the trapdoor $\mathbf{T}$ for the subring $\mathcal{S}$.
4. Output a signature $\sigma \leftarrow \mathsf{GenSamplePre}(\mathbf{A}_R, \mathbf{A}_S, \mathbf{T}_S, \gamma, q\cdot\mathbf{v})$ and a label lab_S for a subring $\mathcal{S} \subset \mathcal{R}$.

SI.Combine$(R, id, \{(\alpha_j, \sigma_j)\}_{j=1}^{l}, lab_S)$:
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 $\{\sigma_j\}_{j=1}^{r}$ with identical subring $\mathcal{S}$, output $\sigma = \sum_{j=1}^{r} \sigma_i \in \mathbb{Z}^{(r+1)m}$.

SI.Verify$(R, H, id,$ y$, \sigma, lab_S)$:
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 $\mathbf{y} \in \mathbb{F}_2^n$, a signature $\sigma \in \mathbb{Z}^{(r+1)m}$, and a label lab_S for a subring $\mathcal{S} \subset \mathcal{R}$, 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. Get a subring $\mathcal{S} \subset \mathcal{R}$ from the label lab_S and parse σ into $[\mathbf{e}_{S_1} \mid \mathbf{e}_{S_2} \mid \mathbf{e}_{S_3} \mid \mathbf{e}_{S_4}]$ where $S_1 \cup S_3 = S$ where S is the set of public keys of the signers in $\mathcal{S}$.
3. If $\|\mathbf{e}_S\| \leq L \cdot \gamma\sqrt{(r+1)m}$ and $\mathbf{A}_R \cdot \sigma = q \cdot \mathbf{y} \bmod 2q$, output 1 (accept). Otherwise, output 0 (reject).

4 Security Proof

4.1 Correctness and Linear Homomorphism

To verify the correctness of the proposed signature scheme, we must show that the correctness condition in **Definition 3** holds for any public key all key pairs $\{pk_i, sk_i\}_{i=1}^{r}$ where $pk_i \in R$ and $r \leq L$ is the number of the ring $\mathcal{R}$.

Theorem 1. *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, the proposed scheme $\mathcal{LHRS}$ always outputs a valid signature.*

Proof. Assume that a signature σ is extracted from **SI.Sign** algorithm with the designated signer $i \in \mathcal{R}$ with a key pair (pk_i, sk_i) where $pk_i \in R$.

In **SI.Sign**$(pk_i, sk_i, id, R, \mathbf{v})$ algorithm, $\mathsf{GenSamplePre}$ $(\mathbf{A}_R, \mathbf{A}_S, \mathbf{T}_S, \gamma, q\cdot\mathbf{v})$ algorithm outputs a sample $\mathbf{e}_{S_2} \in \mathbb{Z}_q^{k_2 m}$ and $\mathbf{e}_{S_4} \in \mathbb{Z}_q^{k_4 m}$ uniformly, then compute $q \cdot \mathbf{z} = q\cdot\mathbf{v} - \mathbf{A}_{R_2}\cdot\mathbf{e}_{R_2} - \mathbf{A}_{R_4}\cdot\mathbf{e}_{R_4}$ and sample $\mathbf{e}_S = [\mathbf{e}_{R_1} \mid \mathbf{e}_{R_3}] \in \mathbb{Z}_q^{(k_1+k_3)m}$ from $\mathsf{SamplePre}(\mathbf{A}_S, \mathbf{T}_S, \gamma, q\cdot\mathbf{z})$. The signature σ becomes the concatenation of all $\mathbf{e}_{S_i}$'s as $\sigma = [\mathbf{e}_{S_1} \mid \mathbf{e}_{S_2} \mid \mathbf{e}_{S_3} \mid \mathbf{e}_{S_4}]$.

Since $\mathbf{e}_S$ is extracted from $\mathsf{SamplePre}(\mathbf{A}_S, \mathbf{T}_S, \gamma, \mathbf{z})$ algorithm, $\mathbf{A}_S\cdot\mathbf{e}_S = q\cdot\mathbf{z} \bmod 2q$ and $\|\mathbf{e}_S\| \leq \gamma\sqrt{(r+1)m}$

from the definition of $\mathsf{GenSamplePre}(\mathbf{A}_R, \mathbf{A}_S, \mathbf{T}_S, \gamma, q \cdot \mathbf{v})$ algorithm.

Here, $q \cdot \mathbf{v} = q \cdot \mathbf{z} + \mathbf{A}_{R_2} \cdot \mathbf{e}_{R_2} + \mathbf{A}_{R_4} \cdot \mathbf{e}_{R_4} = \mathbf{A}_S \cdot \mathbf{e}_S + \mathbf{A}_{R_2} \cdot \mathbf{e}_{R_2} + \mathbf{A}_{R_4} \cdot \mathbf{e}_{R_4} = \mathbf{A}_R \cdot \sigma \bmod q$.

Thus, the proposed signature scheme is correct. □

Theorem 2. *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, the proposed scheme $\mathcal{SILHR}$ is subring-identical linearly homomorphic.*

Proof. Assume that we combine l messages $\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_l$ into $\mathbf{m}_{lin} = \sum_{j=1}^{l} \mathbf{m}_j$.

For all $\sigma_j \leftarrow$ **SI.Sign**$(pk_i, sk_i, id, R, \mathbf{v})$, we extract $\mathbf{e}_{S^j}$ where $\|\mathbf{e}_{S^j}\| \leq \gamma\sqrt{(r+1)m}$ and $\mathbf{A}_R \cdot \sigma_j = q \cdot \mathbf{m}_j \bmod 2q$.

For a signature $\sigma_{lin} = \sum_{j=1}^{l} \sigma_j \leftarrow$ **SI.Combine** $(R, id, \{(\alpha_j, \sigma_j)\}_{j=1}^{l}, labs_S)$, we extract $\mathbf{e}_{S_{lin}}$ and $\|\mathbf{e}_{S_{lin}}\| \leq \sum_{j=1}^{l} \|\mathbf{e}_{S^j}\| \leq l \cdot \gamma\sqrt{(r+1)m} \leq L \cdot \gamma\sqrt{(r+1)m}$ if $l \leq L$ with high probability by **Lemma 2** and $\mathbf{A}_R \cdot \sigma = q \cdot \mathbf{m}_{lin} \bmod 2q$.

i.e., **SI.Verify** $(R, H, id, \mathbf{m}_{lin}, \sigma_{lin}, labs_S) = 1$ and the proposed signature satisfies the linearly homomorphic property if $l \leq L$. Hence, the proposed signature scheme is subring-identical linearly homomorphic. □

4.2 Other Security Requirements

For the security requirements of our proposed scheme $\mathcal{SILHR}$, we give a proof sketch of unforgeability, weakly context hiding property, and anonymity in **Theorems 3, 4, and 5**, respectively.

Theorem 3. *For the proposed signature $\mathcal{SILHR}$, the proposed signature is unforgeable against fixed-ring attack in the random oracle model when $SIS_{q,(r+1)m,2\gamma}$ problem is infeasible.*

Proof. (sketch) Let $\mathcal{A}_1$ be an adversary that has the advantage ϵ_0 of the challenge-response game in **Definition 4**. We construct a polynomial time algorithm $\mathcal{B}_1$ to simulate the attacking environment for $\mathcal{A}_1$. Both $\mathcal{A}_1$ and $\mathcal{B}_1$ have the input q_A, which is the total number of queries issued by $\mathcal{A}_1$ and $\mathcal{B}_1$ interacts with $\mathcal{A}_1$ as below:

Setup: $\mathcal{B}_1$ guesses the size of the challenge ring $r \in [q_E]$ and obtains an instance $\mathbf{A}_{S_t} \in \mathbb{Z}_q^{n \times (r+1)m}$. Then, $\mathcal{B}_1$ parses it into $\mathbf{A}_j \in \mathbb{Z}_q^{n \times m}$ where $1 \leq j \leq r+1$. We assume that all ring members are in the set $[r+1]$ without loss of generality.
$\mathcal{B}_1$ runs $\mathsf{TrapGen}$ to generate a tuple $\{i, \mathbf{A}_i, \mathbf{T}_i$ and replace $\mathbf{A}_i$ into $\mathbf{A}_j$ if $i = j$. All $\mathbf{A}_i$'s after replacement will be sent to $\mathcal{A}_1$.

Query Phase: $\mathcal{B}_1$ answers hash queries and signing queries requested by $\mathcal{A}_1$.

Challenge $\mathcal{A}_1$ outputs a forgery $\{id^*, \mathbf{y}^*, \sigma^*, labs_{S^*}\}$. If $S^* \neq S_t$, $\mathcal{B}_1$ aborts. Otherwise, $\mathcal{B}_1$ checks whether (1) id^* is never queried or (2) $id^* = id_i$ for some signing query but $\mathbf{y}^*$ is not queried by the adversary.

In the above process, if $\mathcal{A}_1$ outputs a valid signature σ^* with a tuple $\{id^*, \mathbf{y}^*, labs_{S^*}\}$, $\mathcal{B}_1$ solves the SIS instance of $\mathrm{SIS}_{q,(r+1)m,2\gamma}$. □

Theorem 4. *For the proposed signature $\mathcal{SILHR}$, weakly context hiding property holds for our signature.*

Proof. (sketch) Let $\mathcal{A}_2$ be an adversary that has the advantage ϵ_0 of the challenge-response game in **Definition 5**. Assume $\mathcal{A}_2$ has the output $\{V_0, V_1, f_1, \cdots, f_s\}$ of this challenge-response game, where $\{\mathbf{v}_i^{(0)}\}_{i=1}^{k}$ and $\{\mathbf{v}_i^{(1)}\}_{i=1}^{k}$ are basis vectors of V_0 and V_1, respectively.

We know that

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

Let σ_i^0 and σ_i^1 be the challenger's signatures of $\mathbf{v}_i$ for V_0 and V_1, respectively.

$\sigma_{j,0}^*$ and $\sigma_{j,1}^*$ are signatures on $\mathbf{u}_j$ computed using **SI.Combine** algorithm.

We claim that the samples $\{\sigma_{j,0}^*\}_{j=1}^{s}$ and $\{\sigma_{j,1}^*\}_{j=1}^{s}$ are sampled from statistically close distributions so that the adversary cannot guess b with non-negligible probability if γ is sufficiently large. □

Lemma 3. *For the proposed signature $\mathcal{SILHR}$, let (i_0, i_1, R, v) be a tuple such that $v \in \{0,1\}^*$ is a message to be signed with the ring $\mathcal{R}$, i_0 and i_1 are indices with $\boldsymbol{A}_{i_0}, \boldsymbol{A}_{i_1} \in R$. If $SIS_{q,(r+1)m,2\gamma}$ is hard, $\sigma_{i_0} \leftarrow$ **SI.Sign**$(pk_{i_0}, sk_{i_0}, id, R, \boldsymbol{v})$ and $\sigma_{i_1} \leftarrow$ **SI.Sign** $(pk_{i_1}, sk_{i_1}, id, R, \boldsymbol{v})$ are computationally indistinguishable.*

Proof. (sketch) From **SI.Sign** and $\mathsf{GenSamplePre}$, samples from Gaussian distribution is computationally indistinguishable from random. Hence, the ring signatures σ_{i_0} and σ_{i_1} are computationally indistinguishable to random vectors in $\mathbb{Z}_q^{(r+1)m}$.

□

Theorem 5. *For the proposed signature $\mathcal{SILHR}$, the proposed signature provides anonymity.*

Proof. (sketch) Let $\mathcal{A}_3$ be an adversary that has the advantage ϵ_0 of the challenge-response game in **Definition 6**. We construct a polynomial time algorithm $\mathcal{B}_3$ to simulate the attacking environment for $\mathcal{A}_3$. Both $\mathcal{A}_3$ and $\mathcal{B}_1$ have the input q_A, which is the total number of queries issued by $\mathcal{A}_3$ and $\mathcal{B}_3$ interacts with $\mathcal{A}_3$ as below:

Setup: $\mathcal{B}_3$ runs $\mathsf{TrapGen}$ to generate a tuple $\{i, \mathbf{A}_i, \mathbf{T}_i$ and sends all $\mathbf{A}_i$ to $\mathcal{A}_3$.

Query Phase: $\mathcal{B}_3$ answers hash queries and signing queries requested by $\mathcal{A}_3$.

Challenge $\mathcal{A}_3$ provides a tuple $\{i_0, i_1 S^*, \mathbf{v}^*\}$ such that $\mathbf{v}^*$ is a message to be signed with the ring S^*, i_0 and i_1 are indices with $pk_{i_0}, pk_{i_1} \in S^*$. $\mathcal{B}_3$ chooses a bit $b^* \leftarrow \{0,1\}$ and computes the challenge signature σ_{b^*} with **SI.Sign** and provides σ_{b^*} to $\mathcal{A}_3$. Then, $\mathcal{A}_3$ outputs a guess $b' \in \{0,1\}$.

In the above process, $\mathcal{B}_3$ behaves like a real anonymity security experiment. Thus, if $\mathcal{A}_3$ guesses correctly with non-negligible probability, $\mathcal{A}$ can distinguish two signatures from different identities with non-negligible probability, which contradicts **Lemma 3**. □

5 Conclusion and Future Work

We have revisited the lattice-based linearly homomorphic signature scheme over binary fields by Choi and Kim in SCIS 2017 [10]. We define the new concept called subring-identical linearly homomorphic signature scheme and give a proof sketch of their requirements.

As future work, we plan to extend the linear homomorphism of our scheme to any subring and check how to embed this scheme to real-world cloud computing systems. Also, one of challenging problems along with this paper is to define and construct a homomorphic ring signature with resistance to chosen-subring attack and insider corruption [15].

Acknowledgement

This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00555, Towards Provable-secure Multi-party Authenticated Key Exchange Protocol based on Lattices in a Quantum World) and National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2015R1A2A2A01006812, Design and Security Analysis of Novel Lattice-based Fully Homomorphic Signatures Robust to Quantum Computing Attack).

References

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

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

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

[4] D. Fiore, A. Mitrokotsa, L. Nizzardo, and E. Pagnin, "Multi-key homomorphic authenticators," in *Advances in Cryptology–ASIACRYPT 2016*, pp. 499–530, Springer, 2016.

[5] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Annual ACM on Symposium on Theory of Computing*, pp. 169–178, ACM, 2009.

[6] 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.

[7] 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.

[8] 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`.

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

[10] R. Choi and K. Kim, "Design of new linearly homomorphic signatures on lattice," in *Symposium on Cryptography and Information Security*, 2017. 2B3-3).

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

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

[13] 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.

[14] 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.

[15] A. Bender, J. Katz, and R. Morselli, "Ring signatures: Stronger definitions, and constructions without random oracles," in *Conference on Theory of Cryptography*, pp. 60–79, Springer, 2006.