PAPER Special Section on Cryptography and Information Security

Fully Homomorphic Encryption Scheme Based on Decomposition Ring

Seiko ARITA[†a)] ***and*** **Sari HANDA**[†b)]**,** ***Members***

SUMMARY In this paper, we propose the decomposition ring homomorphic encryption scheme, that is a homomorphic encryption scheme built on the decomposition ring, which is a subring of cyclotomic ring. By using the decomposition ring the structure of plaintext slot becomes $\mathbb{Z}_{p^l}$, instead of GF(p^d) in conventional schemes on the cyclotomic ring. For homomorphic multiplication of integers, one can use the full of $\mathbb{Z}_{p^l}$ slots using the proposed scheme, although in conventional schemes one can use only one-dimensional subspace GF(p) in each GF(p^d) slot. This allows us to realize fast and compact homomorphic encryption for integer plaintexts. In fact, our benchmark results indicate that our decomposition ring homomorphic encryption schemes are several times faster than HElib for integer plaintexts due to its higher parallel computation.

key words: *fully homomorphic encryption, ring-LWE, cyclotomic ring, decomposition ring, plaintext slots*

1. Introduction

Background.

Homomorphic encryption (HE) scheme enables us computation on encrypted data. One can add or multiply (or more generally "evaluate") given ciphertexts and generate a new ciphertext that encrypts the sum or product (or "evaluation") of underlying data of the input ciphertexts. Such computation (called *homomorphic* addition or multiplication or evaluation) can be done without using the secret key and one will never know anything about the processed or generated data.

Since the breakthrough construction given by Gentry [10], many efforts have been dedicated to make such homomorphic encryption scheme more secure and more efficient. Especially, HE schemes based on the Ring-LWE problem [5], [9], [20], [21] have obtained theoretically-sound proof of security and well-established implementations such as HElib [14] and SEAL v2.0 [19]. Nowadays many researchers apply HE schemes to privacy-preserving tasks for mining of outsourced data such as genomic data, medical data, financial data and so on [7], [13], [16]–[18].

Our perspective: GF(p^d) versus $\mathbb{Z}_{p^l}$ slots.

In order to obtain higher throughput, batching technique is widely adopted in many HE schemes, that allows us to encrypt multiple messages in a single ciphertext and enables a parallel processing in SIMD manner. The space where each of values of parallel processing is set is called "plaintext slot".

The HE schemes based on the Ring-LWE problem (*Ring-HE schemes* in short), depend on arithmetic of cyclotomic integers [20]. Cyclotomic integers a are algebraic integers generated by some primitive m-th root of unity ζ and have the form like $a = a_0 + a_1\zeta + \cdots + a_{n-1}\zeta^{n-1}$ where a_i are ordinary integers in $\mathbb{Z}$ and $n = \phi(\mathsf{m})$. In the Ring-HE schemes based on cyclotomic ring, its structure of the plaintext slot is known to be Galois field GF(p^d) of some degree d. For small primes p, this degree $d\,(> \log_p(\mathsf{m}))$ will be large.

Such plaintext structure is good for applications that use data represented by elements of Galois field GF(p^d), such as error correcting codes or AES ciphers. However, many applications will use integers modulo a power of prime p^l (i.e., elements in $\mathbb{Z}_{p^l}$) rather than elements of Galois field GF(p^d).

We focus on the fact that restricting the cyclotomic ring to its subring called "Decomposition Ring", the slot structure shrinks from GF(p^d) to $\mathbb{Z}_p$. Then, by using Hensel lifting, we can enlarge the modulus from $\mathbb{Z}_p$ to $\mathbb{Z}_{p^l}$. We believe in that such plaintext structure will be more natural, easy to handle, and significantly efficient for many applications.

Method.

To realize plaintext structure composed of slots of mod-p^l integers, we use decomposition ring R_Z with respect to the prime p, instead of cyclotomic ring R.

Let ζ be a primitive m-th root of unity. The m-th cyclotomic ring $R = \{a_0 + a_1\zeta + \cdots + a_{n-1}\zeta^{n-1} \mid a_i \in \mathbb{Z}\}$ is a ring of all cyclotomic integers generated by ζ, where $n = \phi(\mathsf{m})$ is the value of Euler function at m. Plaintext space of Ring-HE schemes will be the space of mod-p cyclotomic integers, i.e., $R_p = R/pR$ for some small prime p. It is known that in the cyclotomic ring R, the prime number p is not prime any more (in general) and it factors into a product of g prime ideals $\mathfrak{P}_i$ (with some divisor g of n): $pR = \mathfrak{P}_0\mathfrak{P}_1\cdots\mathfrak{P}_{g-1}$. The residual fields $R/\mathfrak{P}_i$ of each factor $\mathfrak{P}_i$ are nothing but the space of plaintext slots of Ring-HE schemes, which are isomorphic to GF(p^d) with $d = n/g$. Thus, the plaintext space is

$$R_p \simeq R/\mathfrak{P}_0 \oplus \cdots \oplus R/\mathfrak{P}_{g-1} \simeq \mathrm{GF}(p^d) \oplus \cdots \oplus \mathrm{GF}(p^d).$$

Here note that we can use only 1-dimensional subspace GF(p) = $\mathbb{Z}_p$ in each d-dimensional slot GF(p^d) for homomorphic multiplication of mod-p integers.

Manuscript received March 18, 2019.
Manuscript revised July 11, 2019.

[†]The authors are with the Institute of Information Security, Yokohama-shi, 221-0835 Japan.
a) E-mail: arita@iisec.ac.jp
b) E-mail: dgs158101@iisec.ac.jp

DOI: 10.1587/transfun.2019CIP0027

The decomposition ring R_Z with respect to prime p is the minimum subring of R in which the prime p has the same form of prime ideal factorization as in R , that is,

$$pR_Z = \mathfrak{p}_0\mathfrak{p}_1\cdots\mathfrak{p}_{g-1} \tag{1}$$

with the same number g of factors. By the minimality of R_Z, the residual fields $R_Z/\mathfrak{p}_i$ of each factor $\mathfrak{p}_i$ must be one-dimensional, that is, isomorphic to $\mathbb{Z}_p$. So the plaintext space on R_Z will be

$$(R_Z)_p \simeq R_Z/\mathfrak{p}_0 \oplus \cdots \oplus R_Z/\mathfrak{p}_{g-1} \simeq \mathbb{Z}_p \oplus \cdots \oplus \mathbb{Z}_p.$$

Applying Hensel lift $l-1$ times, we get $(R_Z)_{p^l} \simeq \mathbb{Z}_{p^l} \oplus \cdots \oplus \mathbb{Z}_{p^l}$ for p^l. Thus, the decomposition ring R_Z realizes plaintext slots of integers modulo p^l, as desired. Note that now we can use *all of the dimensions* of R_Z as its plaintext slots for homomorphic multiplication of mod-p^l integers. This high parallelism of slot structure will bring us significantly more efficient SIMD operations for mod-p^l integer plaintexts.

Two bases.

The cyclotomic ring R has attractive features that enable efficient implementation of addition/multiplication of and noise handling on their elements. Can we do similar things even if we use the decomposition ring R_Z instead of cyclotomic ring R?

The cyclotomic ring R's nice properties are consolidated to the existence of two types of bases [21]:

- The power(ful) basis: Composed of short and nearly orthogonal vectors to each other. Used when rounding rational cyclotomic numbers to their nearest cyclotomic integers.
- The CRT basis: Related to the FFT transformation and multiplication. Vectors of coefficients of given two cyclotomic integers w.r.t. the CRT basis can be multiplied component-wise, resulting a new vector corresponding to the multiplied cyclotomic integer.

We investigate structure of the decomposition ring R_Z, following the way in cyclotomic cases given by Lyubashevsky, Peikert, and Regev [21]. Then, we will give two types of bases of R_Z, called η-basis and ξ-basis, which can substitute well for the power(ful) and CRT bases in cyclotomic cases, respectively.

Construction.

Based on the above investigation, we construct two types of homomorphic encryption schemes over the decomposition ring: DR-FV and DR-BGV. The DR-FV and DR-BGV schemes realize the FV [9] and the BGV scheme [5] over the decomposition ring, respectively. We show several bounds on the noise growth occurring among homomorphic computations and prove that both of DR-FV and DR-BGV schemes are fully homomorphic on modulus of magnitude $q = O(\lambda^{\log \lambda})$.

Security.

For security we will need hardness of a variant of the decisional Ring-LWE problem over the decomposition ring. Recall the search version of Ring-LWE problem is already proved to have a quantum polynomial time reduction from the approximate shortest vector problem of ideal lattices in *any number field* by Lyubashevsky, Peikert, and Regev [20]. They proved equivalence between the search and decisional versions of the Ring-LWE problems only for cyclotomic rings. However, it is not difficult to see that the equivalence holds also over the decomposition rings, since those are subrings of cyclotomic rings and inherit good properties about prime ideal decomposition from them.

Efficiency.

Here, we compare efficiency of DR-FV (or DR-BGV) with the conventional HE scheme on the cyclotomic ring (CR-HE for short). In CR-HE, the ring dimension is $n = \phi(\mathsf{m})$, the number of slots is $g = n/d$ and the dimension of each slot is d. So, one can encrypt g integer plaintexts into a single ciphertext of n $(= gd)$ dimension. On the contrary, in DR-FV (or DR-BGV), the ring dimension and the number of slots are both g and the dimension of each slot is 1. One can encrypt g integer plaintexts into a single ciphertext of the same dimension g. Thus, on the same level of security (i.e. same dimension), DR-FV (or DR-BGV) can handle d times as many plaintexts as CR-HE in a single ciphertext. This means that DR-FV (or DR-BGV) achieves more faster and compact HE than conventional CR-HE for integer plaintexts. More concrete benchmark results are given in Sect. 5.

Related works.

In 2009, Gentry [10] established the fully homomorphic encryption scheme for the first time. After this breakthrough, representative two schemes, BGV scheme [5] and FV [9] scheme, are proposed depending on the techniques such as key switching [6] and modulus switching [5]. Since the computational cost of homomorphic operations are very expensive, parallel computing is needed for higher throughput. The SIMD technique, proposed by [22], enables parallel homomorphic computation using the CRT over polynomials, and has been adopted in many HE schemes. We focus on the wasteful slot structure of such HE schemes based on cyclotomic ring, and improve the slot structure using the decomposition ring. Kim and Song [15] also focus on the similar issue and construct HE based on another subring of the cyclotomic ring, called "conjugate-invariant ring", aiming for efficient homomorphic fixed-point number computation. Terada, Nakano, Okumura and Miyaji [23] conducted some experiments regarding lattice attack against Ring-LWE problem over the decomposition ring. They concluded that the Ring-LWE problem on the decomposition ring is expected to be as secure as the ordinal Ring-LWE problem on the cyclotomic ring. This paper is a full version of [1].

Organization.

In Sect. 2 we prepare notions and tools needed for our work, especially about cyclotomic rings. Section 3 investigates structure and properties of the decomposition ring, and gives its η-basis and ξ-basis as well as quasi-linear time conversion between them. In Sect. 4 we state a variant of the Ring-LWE problem over the decomposition ring and construct our two homomorphic encryption schemes over the ring. Finally, Sect. 5 shows our benchmark results, comparing efficiency of our two homomorphic encryption schemes and HElib. Proofs of lemmas or theorems are collected in the appendices.

2. Preliminaries

Notation.

For a positive integer m, $\mathbb{Z}_m$ denotes the ring of congruent integers mod m, and $\mathbb{Z}_m^*$ denotes its multiplicative subgroup. For an integer a (that is prime to m), $\mathrm{ord}_m^{\times}(a)$ denotes the order of $a \in \mathbb{Z}_m^*$. Basically vectors are supposed to represent column vectors. The symbol $\mathbf{1}$ denotes a column vector with all entries equal to 1. I_n denotes the $n \times n$ identity matrix. The symbol $\mathrm{Diag}(\alpha_1, \cdots, \alpha_n)$ means a diagonal matrix with diagonals $\alpha_1, \ldots, \alpha_n$. For vectors $\boldsymbol{x}, \boldsymbol{y}$ ($\in \mathbb{C}^n$), $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^n x_i \overline{y}_i$ denotes the inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$. $\|\boldsymbol{x}\|_2 = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$ denotes the l_2-norm and $\|\boldsymbol{x}\|_\infty = \max\{|x_i|\}_{i=1}^n$ denotes the infinity norm of $\boldsymbol{x}$. For vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, $\boldsymbol{a} \odot \boldsymbol{b} = (a_i b_i)_i$ denotes the component-wise product of $\boldsymbol{a}$ and $\boldsymbol{b}$. For a square matrix M over $\mathbb{R}$, $s_1(M)$ denotes the largest singular value of M. For a matrix A over $\mathbb{C}$, $A^* = \overline{A}^T$ denotes the transpose of complex conjugate of A.

2.1 Homomorphic Encryption Scheme

A homomorphic encryption scheme is a quadruple HE=(Gen, Encrypt, Decrypt, Evaluate) of probabilistic polynomial time algorithms. Gen generates a public key pk, a secret key sk and an evaluation key evk: $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk}) \leftarrow \mathsf{Gen}(1^\lambda)$. Encrypt encrypts a plaintext $x \in \mathsf{X}$ to a ciphertext c under a public key pk: $c \leftarrow \mathsf{Encrypt}(\mathsf{pk}, x)$. Decrypt decrypts a ciphertext c to a plaintext x using the secret key sk: $x \leftarrow \mathsf{Decrypt}(\mathsf{sk}, c)$. Evaluate applies a function $f : \mathsf{X}^l \to \mathsf{X}$ (given as an arithmetic circuit) to ciphertexts $c_1, \ldots, c_l$ and outputs a new ciphertext c_f using the evaluation key evk : $c_f \leftarrow \mathsf{Evaluate}(\mathsf{evk}, f, c_1, \ldots, c_l)$.

A homomorphic encryption scheme HE is *L-homomorphic* for $L = L(\lambda)$ if for any function $f : \mathsf{X}^l \to \mathsf{X}$ given as an arithmetic circuit of depth L and for any l plaintexts $x_1, \ldots, x_l \in \mathsf{X}$, it holds that

$$\mathsf{Decrypt}_{\mathsf{sk}}(\mathsf{Evaluate}_{\mathsf{evk}}(f, c_1, \ldots, c_l)) = f(x_1, \ldots, x_l)$$

for $c_i \leftarrow \mathsf{Encrypt}_{\mathsf{pk}}(x_i)$ $(i = 1, \ldots, l)$ except with a negligible probability (i.e., $\mathsf{Decrypt}_{\mathsf{sk}}$ is ring homomorphic). A homomorphic encryption scheme is called *fully homomorphic* if it is L-homomorphic for any polynomial function $L = poly(\lambda)$.

2.2 Gaussian Distributions and Subgaussian Random Variables

For a positive real $s > 0$, the n-dimensional (spherical) Gaussian function $\rho_s : \mathbb{R}^n \to (0, 1]$ is defined as

$$\rho_s(x) = \exp(-\pi \|x\|_2^2 / s^2).$$

It defines the continuous Gaussian distribution D_s with density $s^{-n}\rho_s(x)$.

A random variable X over $\mathbb{R}$ is called *subgaussian with parameter* s (> 0) if $\mathbb{E}[\exp(2\pi t X)] \leq \exp(\pi s^2 t^2)$ $(\forall t \in \mathbb{R})$. A random variable X over $\mathbb{R}^n$ is called subgaussian with parameter s if $\langle X, u \rangle$ is subgaussian with parameter s for any unit vector $u \in \mathbb{R}^n$. A random variable X according to Gaussian distribution D_s is subgaussian with parameter s. A bounded random variable X (as $|X| \leq B$) with $\mathbb{E}[X] = 0$ is subgaussian with parameter $B\sqrt{2\pi}$.

A subgaussian random variable with parameter s satisfies the tail inequality:

$$\Pr[|X| \geq t] \leq 2\exp\left(-\pi \frac{t^2}{s^2}\right) \quad (\forall t \geq 0). \tag{2}$$

2.3 Lattices

For n linearly independent vectors $B = \{b_j\}_{j=1}^n \subset \mathbb{R}^n$, $\Lambda = \mathcal{L}(B) = \left\{\sum_{j=1}^n z_j b_j \mid z_j \in \mathbb{Z} \ \ (\forall j)\right\}$ is called an n-dimensional *lattice*. For a lattice $\Lambda \subset \mathbb{R}^n$, its *dual lattice* is defined by $\Lambda^\vee = \left\{y \in \mathbb{R}^n \mid \langle x, y \rangle \in \mathbb{Z} \ (\forall x \in \Lambda)\right\}$. The dual lattice is itself a lattice. The dual of dual lattice is the same as the original lattice: $(\Lambda^\vee)^\vee = \Lambda$. For a countable subset $A \subset \mathbb{R}^n$, the sum $D_s(A) \stackrel{\text{def}}{=} \sum_{x \in A} D_s(x)$ is well-defined. The discrete Gaussian distribution $D_{\Lambda+c,s}$ on a (coset of) lattice Λ is defined by restricting the continuous Gaussian distribution D_s on the (coset of) lattice Λ:

$$D_{\Lambda+c,s}(x) \stackrel{\text{def}}{=} \frac{D_s(x)}{D_s(\Lambda + c)} \quad (x \in \Lambda + c).$$

2.4 Number Fields

A complex number $\alpha \in \mathbb{C}$ is called an *algebraic number* if it satisfies $f(\alpha) = 0$ for some nonzero polynomial $f(X) \in \mathbb{Q}[X]$ over $\mathbb{Q}$. For an algebraic number α, the monic and irreducible polynomial $f(X)$ satisfying $f(\alpha) = 0$ is uniquely determined and called the *minimum polynomial* of α. An algebraic number α generates a *number field* $K = \mathbb{Q}(\alpha)$ over $\mathbb{Q}$, which is isomorphic to $\mathbb{Q}[X]/(f(X))$, via $g(\alpha) \mapsto g(X)$. The dimension of K as a $\mathbb{Q}$-vector space is called the *degree* of K and denoted as $[K : \mathbb{Q}]$. By the isomorphism, $[K : \mathbb{Q}] = \deg f$.

An algebraic number α is called an *algebraic integer*

if its minimum polynomial belongs to $\mathbb{Z}[X]$. All algebraic integers belonging to a number field $K = \mathbb{Q}(\alpha)$ constitutes a ring R, called an *integer ring* of K.

A number field $K = \mathbb{Q}(\alpha)$ has $n (= [K : \mathbb{Q}])$ isomorphisms ρ_i $(i = 1, \ldots, n)$ to subfields of the complex number field $\mathbb{C}$. The trace map $\mathrm{Tr}_{K|\mathbb{Q}} : K \to \mathbb{Q}$ is defined by $\mathrm{Tr}_{K|\mathbb{Q}}(a) = \sum_{i=1}^{n} \rho_i(a)$ $(\in \mathbb{Q})$. If all of the isomorphisms ρ_i induce automorphisms of K (i.e., $\rho_i(K) = K$ for any i), the field K is called a *Galois extension* of $\mathbb{Q}$ and the set of isomorphisms $\mathrm{Gal}(K|\mathbb{Q}) \stackrel{\text{def}}{=} \{\rho_1, \ldots, \rho_n\}$ constitutes a group, called the *Galois group* of K over $\mathbb{Q}$. By the Galois theory, all subfields L of K and all subgroups H of $G = \mathrm{Gal}(K|\mathbb{Q})$ corresponds to each other one-to-one:

$$\begin{aligned} L \;&\mapsto\; H = G_L = \{\rho \in G \mid \rho(a) = a \text{ for any } a \in L\} \\ &\quad : \text{the stabilizer group of } L \\ H \;&\mapsto\; L = K^H = \{a \in K \mid \rho(a) = a \text{ for any } \rho \in H\} \\ &\quad : \text{the fixed field by } H. \end{aligned}$$

Here, K is also a Galois extension of L with Galois group $\mathrm{Gal}(K|L) = H$ (since any isomorphism (of K into $\mathbb{C}$) that fixes L sends K to K). Especially, $[K : L] = |H|$. The trace map of K over L is defined by $\mathrm{Tr}_{K|L}(a) = \sum_{\rho \in H} \rho(a)$ $(\in L)$ for $a \in K$.

2.5 Cyclotomic Fields and Rings

Let m be a positive integer. A primitive m-th root of unity $\zeta = \exp(2\pi\sqrt{-1}/\mathsf{m})$ has the minimum polynomial $\Phi_{\mathsf{m}}(X) \in \mathbb{Z}[X]$ of degree $n = \phi(\mathsf{m})$ that belongs to $\mathbb{Z}[X]$, called the *cyclotomic polynomial*. Especially, ζ is an algebraic integer. A number field $K = \mathbb{Q}(\zeta)$ generated by ζ is called the m-th *cyclotomic field* and its elements are called *cyclotomic numbers*. The integer ring R of the cyclotomic field $K = \mathbb{Q}(\zeta)$ is known to be $R = \mathbb{Z}[\zeta] = \mathbb{Z}[X]/\Phi_{\mathsf{m}}(X)$. In particular, as a $\mathbb{Z}$-module, R has a basis (called *power basis*) $\{1, \zeta, \ldots, \zeta^{n-1}\}$, i.e., $R = \mathbb{Z} \cdot 1 + \mathbb{Z} \cdot \zeta + \cdots + \mathbb{Z} \cdot \zeta^{n-1}$. The integer ring R is called the m-th *cyclotomic ring* and its elements are called *cyclotomic integers*. For a positive integer q, $R_q = R/qR = \mathbb{Z}_q[X]/\Phi_{\mathsf{m}}(X)$ is a ring of *cyclotomic integers mod* q.

The cyclotomic field $K = \mathbb{Q}(\zeta)$ is a Galois extension over $\mathbb{Q}$ since it has $n = [K : \mathbb{Q}]$ automorphisms ρ_i defined by $\rho_i(\zeta) = \zeta^i$ for $i \in \mathbb{Z}_{\mathsf{m}}^*$. Its Galois group $G = \mathrm{Gal}(K|\mathbb{Q})$ is isomorphic to $\mathbb{Z}_{\mathsf{m}}^*$ by corresponding ρ_i to i. Note that $\rho_i(\overline{b}) = \overline{\rho_i(b)}$, since $\overline{a} = \rho_{-1}(a)$.

The trace of ζ for the prime index m is simple:

Lemma 1: If the index m is prime, we have

$$\mathrm{Tr}_{K|\mathbb{Q}}(\zeta^i) = \begin{cases} \mathsf{m} - 1 & (i \equiv 0 \pmod{\mathsf{m}}) \\ -1 & (i \not\equiv 0 \pmod{\mathsf{m}}). \end{cases}$$

2.5.1 Structure of R_p

Let p be a prime that does not divide m. Although the cyclotomic polynomial $\Phi_{\mathsf{m}}(X)$ is irreducible over $\mathbb{Z}$, by taking mod p, it will be factored into a product of several polynomials $F_i(X)$'s:

$$\Phi_{\mathsf{m}}(X) \equiv F_0(X) \cdots F_{g-1}(X) \pmod{p}, \tag{3}$$

where all of $F_i(X)$ are irreducible mod p, and have the same degree $d = \mathrm{ord}_{\mathsf{m}}^{\times}(p)$ which is a divisor of n. The number of factors is equal to $g = n/d$.

It is known that there are g prime ideals $\mathfrak{P}_0, \ldots, \mathfrak{P}_{g-1}$ of R lying over p : $\mathfrak{P}_i \cap \mathbb{Z} = p\mathbb{Z}$ $(i = 0, \ldots, g-1)$ and p decomposes into a product of those prime ideals in R:

$$pR = \mathfrak{P}_0 \cdots \mathfrak{P}_{g-1}. \tag{4}$$

This decomposition of the prime p reflects the factorization of $\Phi_{\mathsf{m}}(X)$ mod p (Eq. (3)). In fact, each prime factor $\mathfrak{P}_i$ is generated by p and $F_i(\zeta)$ as ideals of R, $\mathfrak{P}_i = (p, F_i(\zeta))$ for $i = 0, \ldots, g-1$. The corresponding residual fields are given by

$$R/\mathfrak{P}_i \;\simeq\; \mathbb{Z}_p[X]/F_i(X) \;\simeq\; \mathrm{GF}(p^d)$$

for $i = 0, \ldots, g-1$. Thus, we have

$$R_p \;\simeq\; R/\mathfrak{P}_0 \oplus \cdots \oplus R/\mathfrak{P}_{g-1} \;\simeq\; \mathrm{GF}(p^d) \oplus \cdots \oplus \mathrm{GF}(p^d).$$

In the Ring-HE schemes such as [4], [5], [9], plaintexts are encoded by cyclotomic integers $x \in R_p$ modulo some *small prime* p $(\nmid \mathsf{m})$. By the factorization of R_p above, g plaintexts $x_0, \ldots, x_{g-1}$ belonging to $\mathrm{GF}(p^d)$ are encoded into a single cyclotomic integer $x \in R_p$. The place of each plaintext $x_i \in \mathrm{GF}(p^d)$ is called a *plaintext slot*. Thus, in the Ring-HE schemes, one can encrypt g plaintexts into a single ciphertext by setting them on corresponding plaintext slots and can evaluate or decrypt the g encrypted plaintexts at the same time using arithmetic of cyclotomic integers [22]. Gentry, Halevi, and Smart [12] homomorphically evaluates AES circuit on HE-encrypted AES-ciphertexts in the SIMD manner, using such plaintext slot structure for $p = 2$, which fits to the underlying $\mathrm{GF}(2^d)$-arithmetic of the AES cipher.

2.5.2 Geometry of Numbers

Using the n automorphisms ρ_i $(i \in \mathbb{Z}_{\mathsf{m}}^*)$, the cyclotomic field K is embedded into an n-dimensional complex vector space $\mathbb{C}^{\mathbb{Z}_{\mathsf{m}}^*}$, called the *canonical embedding* $\sigma : K \to H (\subset \mathbb{C}^{\mathbb{Z}_{\mathsf{m}}^*})$: $\sigma(a) = (\rho_i(a))_{i \in \mathbb{Z}_{\mathsf{m}}^*}$. Its image $\sigma(K)$ is contained in the space H defined as

$$H \stackrel{\text{def}}{=} \{x \in \mathbb{C}^{\mathbb{Z}_{\mathsf{m}}^*} \;:\; x_i = \overline{x}_{m-i} \quad (\forall i \in \mathbb{Z}_{\mathsf{m}}^*)\}.$$

Since $H = B\mathbb{R}^n$ with the unitary matrix $B = \frac{1}{\sqrt{2}} \begin{pmatrix} I & \sqrt{-1}J \\ J & -\sqrt{-1}I \end{pmatrix}$, the space H is isomorphic to $\mathbb{R}^n$ as an inner product $\mathbb{R}$-space (where J is the reversal matrix of the identity matrix I).

By the canonical embedding σ, one can regard R (or its (fractional) ideals of R) as lattices in the n-dimensional real vector space H, called *ideal lattices*. Inner products or

norms of elements $a \in K$ are defined through the embedding σ:

$$\langle a, b\rangle \stackrel{\text{def}}{=} \langle \sigma(a), \sigma(b)\rangle = \mathrm{Tr}_{K|\mathbb{Q}}(a\overline{b}),$$
$$\|a\|_2 \stackrel{\text{def}}{=} \|\sigma(a)\|_2,\ \|a\|_\infty \stackrel{\text{def}}{=} \|\sigma(a)\|_\infty.$$

3. Decomposition Rings and Their Properties

To realize plaintext structure composed of slots of mod-p^l integers for some small prime p, we use decomposition rings R_Z w.r.t. p instead of cyclotomic rings R.

3.1 Decomposition Field

Let $G = \mathrm{Gal}(K|\mathbb{Q})$ be the Galois group of the m-th cyclotomic field $K = \mathbb{Q}(\zeta)$ over $\mathbb{Q}$. Let p be a prime that does not divide m. Recall such p has the prime ideal decomposition of Eq. (4). The *decomposition group* G_Z of K w.r.t. p is the subgroup of G defined as

$$G_Z \stackrel{\text{def}}{=} \{\rho \in G \mid \mathfrak{P}_i^\rho = \mathfrak{P}_i \ \ (i = 0, \ldots, g-1)\}.$$

That is, G_Z is the subgroup of automorphisms ρ of K that stabilize each prime ideal $\mathfrak{P}_i$ lying over p. Recall the Galois group $G = \mathrm{Gal}(K|\mathbb{Q})$ is isomorphic to $\mathbb{Z}_\mathsf{m}^*$ via ρ^{-1}. Since p does not divide m, $p \in \mathbb{Z}_\mathsf{m}^*$. It is known that the decomposition group G_Z is generated by the automorphism ρ_p corresponding to the prime p, called the Frobenius map w.r.t. p: $G_Z = \langle \rho_p\rangle \simeq \langle p\rangle \subseteq \mathbb{Z}_\mathsf{m}^*$. The order of G_Z is equal to $d = \mathrm{ord}_\mathsf{m}^\times(p)$. The fixed field $Z = K^{G_Z}$ by G_Z is called the *decomposition field* of K (w.r.t. p). The decomposition field Z can be characterized as the smallest subfield Z of K such that $\mathfrak{P}_i \cap Z$ does not split in K, so that the prime p factorizes into prime ideals in Z in much the same way as in K. By the Galois theory, $G_Z = \mathrm{Gal}(K|Z)$. For degrees, we have $[K : Z] = |G_Z| = d$, $[Z : \mathbb{Q}] = n/d = g$. The decomposition field Z is itself the Galois extension of $\mathbb{Q}$ and its Galois group $\mathrm{Gal}(Z|\mathbb{Q}) = G/G_Z$ is given by $\mathrm{Gal}(Z|\mathbb{Q}) \simeq \mathbb{Z}_\mathsf{m}^*/\langle p\rangle$.

3.2 Decomposition Ring

The integer ring $R_Z = R \cap Z$ of the decomposition field Z is called the *decomposition ring*. Primes ideals over p in the decomposition ring R_Z are given by $\mathfrak{p}_i = \mathfrak{P}_i \cap Z$ for $i = 0, \ldots, g-1$, and the prime p factors into the product of those prime ideals in much the same way as in K:

$$pR_Z = \mathfrak{p}_0 \cdots \mathfrak{p}_{g-1}. \tag{5}$$

This leads to the decomposition of $(R_Z)_p$: $(R_Z)_p \simeq R_Z/\mathfrak{p}_0 \oplus \cdots \oplus R_Z/\mathfrak{p}_{g-1}$.

For each prime ideal $\mathfrak{P}_i$ (of R) lying over $\mathfrak{p}_i$, the Frobenius map ρ_p acts as the p-th power automorphism $\mathrm{pow}_p(x) = x^p$ on $R/\mathfrak{P}_i$:

$$\begin{array}{ccc} R & \longrightarrow & R/\mathfrak{P}_i \\ {\scriptstyle \rho_p}\downarrow & & {\scriptstyle \mathrm{pow}_p}\downarrow \\ R & \longrightarrow & R/\mathfrak{P}_i \end{array}$$

Then, by definition of $R_Z = R^{\langle \rho_p\rangle}$, any element in $R_Z/\mathfrak{p}_i$ must be fixed by pow_p, which means:

$$R_Z/\mathfrak{p}_i = (R/\mathfrak{P}_i)^{\langle \mathrm{pow}_p\rangle} = \mathbb{Z}_p.$$

Thus, we see that all slots of $(R_Z)_p$ must be one-dimensional: $(R_Z)_p \simeq \mathbb{Z}_p \oplus \cdots \oplus \mathbb{Z}_p$.

Here we recall Hensel Lifting:

Lemma 2 ([11] Lemma 3, Hensel Lifting): Let p be a prime, let $i \geq 1$ be an integer, and let $F, G, \Phi \in \mathbb{Z}[X]$ be monic integer polynomials, such that F, G are co-prime modulo p, and $F \cdot G = \Phi \pmod{p^i}$. Then there exist monic polynomials $\overline{F}, \overline{G} \in \mathbb{Z}[X]$ such that $F \equiv \overline{F} \pmod{p^i}$ and $G \equiv \overline{G} \pmod{p^i}$ and $\overline{F} \cdot \overline{G} = \Phi \pmod{p^{i+1}}$.

By Hensel-lifting of the factorization of $\Phi_\mathsf{m}(X)$ mod p (Eq. (3)) to modulus p^l, we get factorization of $\Phi_\mathsf{m}(X)$ mod p^l: $\Phi_\mathsf{m}(X) \equiv \overline{F}_0(X) \cdots \overline{F}_{g-1}(X) \pmod{p^l}$. Here, note that the number g of irreducible factors and the degree d of each factor remain unchanged in the lifting. According to this factorization, the ideal p^lR of R is factored as $p^lR = \mathfrak{Q}_0 \cdots \mathfrak{Q}_{g-1}$ with ideals $\mathfrak{Q}_i = (p^l, \overline{F}_i(\zeta))$ of R.

Then, on the decomposition ring, we get

$$p^lR_Z = \mathfrak{q}_0 \cdots \mathfrak{q}_{g-1} \tag{6}$$
$$(R_Z)_{p^l} \simeq \mathbb{Z}_{p^l} \oplus \cdots \oplus \mathbb{Z}_{p^l} \tag{7}$$

with $\mathfrak{q}_i = \mathfrak{Q}_i \cap Z$ and $R_Z/\mathfrak{q}_i \simeq \mathbb{Z}_{p^l}$. This structure of the decomposition ring $(R_Z)_{p^l}$ brings us the plaintext structure of our decomposition ring homomorphic encryption scheme, being composed of g mod-p^l integer slots.

3.3 Bases of the Decomposition Ring R_Z

To construct homomorphic encryption schemes using some ring R, we will need two types of bases of the ring R over $\mathbb{Z}$, one for decoding elements in $R \otimes \mathbb{R}$ into its nearest element in R, and another one that enables FFT-like fast computations among elements in R. In addition, we also need some quasi-linear time transformations among vector representations with respect to the two types of bases. Here, *assuming the index* m *of cyclotomic ring* R *is prime*, we construct such two types of bases for the decomposition ring R_Z, following the cyclotomic ring case given by Lyubashevsky, Peikert and Regev [21].

3.3.1 The η-Basis

Let m be a prime and $K = \mathbb{Q}(\zeta)$ be the m-th cyclotomic field. For a prime $p\,(\neq \mathsf{m})$, let Z be the decomposition field of K with respect to p.

Fix any set of representatives $\{t_0, \ldots, t_{g-1}\}$ of $\mathbb{Z}_\mathsf{m}^*/\langle p\rangle \simeq \mathrm{Gal}(Z|\mathbb{Q})$. For $i = 0, \ldots, g-1$, define

$$\eta_i \stackrel{\text{def}}{=} \mathrm{Tr}_{K|Z}(\zeta^{t_i}) = \sum_{a \in \langle p\rangle} \zeta^{t_i a} \ \ (\in R_Z).$$

Lemma 3: For $i = 0, \ldots, g-1$, we have $\mathrm{Tr}_{Z|\mathbb{Q}}(\eta_i) = \sum_{i=0}^{g-1} \eta_i = -1$, $\mathrm{Tr}_{Z|\mathbb{Q}}(\overline{\eta}_i) = \sum_{i=0}^{g-1} \overline{\eta}_i = -1$.

Lemma 4: For the prime index m, the set $\{\eta_0, \ldots, \eta_{g-1}\}$ is a basis of the decomposition ring R_Z (w.r.t. $p\,(\neq \mathsf{m})$) over $\mathbb{Z}$, i.e., $R_Z = \mathbb{Z}\eta_0 + \cdots + \mathbb{Z}\eta_{g-1}$.

Definition 1: We call the basis $\boldsymbol{\eta} := (\eta_0, \ldots, \eta_{g-1})$ *η-basis* of R_Z. For any $a \in R_Z$, there exists unique $\boldsymbol{a} \in \mathbb{Z}^g$ satisfying $a = \boldsymbol{\eta}^T \boldsymbol{a}$. We call such $\boldsymbol{a} \in \mathbb{Z}^g$ *η-vector* of $a \in R_Z$.

3.3.2 The ξ-Basis

By the choice of t_i's, the Galois group $\mathrm{Gal}(Z|\mathbb{Q})$ of Z is given by

$$\mathrm{Gal}(Z|\mathbb{Q}) = \{\rho_{t_0}, \ldots, \rho_{t_{g-1}}\}.$$

Elements $a \in Z$ in the decomposition field are regarded as g-dimensional $\mathbb{R}$-vectors through the canonical embedding $\sigma_Z : Z \to H_Z\,(\subset \mathbb{C}^{\mathbb{Z}^*_{\mathsf{m}}/\langle p\rangle})$ defined as $\sigma_Z(a) = (\rho_i(a))_{i \in \mathbb{Z}^*_{\mathsf{m}}/\langle p\rangle}$. The g-dimensional $\mathbb{R}$-subspace H_Z is as

$$H_Z \stackrel{\mathrm{def}}{=} \{x \in \mathbb{C}^{\mathbb{Z}^*_{\mathsf{m}}/\langle p\rangle} \,:\, x_i = \overline{x}_{\mathsf{m}-\mathsf{i}} \quad (\forall i \in \mathbb{Z}^*_{\mathsf{m}}/\langle p\rangle)\}.$$

Define a $g \times g$ matrix Ω_Z over R_Z as

$$\Omega_Z = \left(\rho_{t_i}(\eta_j)\right)_{0 \le i,j < g} \quad (\in R_Z^{g\times g}).$$

Note that each column of Ω_Z is the canonical embedding $\sigma_Z(\eta_j)$ of η_j. Since the index m is prime, the Galois group $\mathrm{Gal}(Z|\mathbb{Q})$ is cyclic and we can take the representatives $\{t_0, \ldots, t_{g-1}\}$ so that $t_j \equiv t^j \pmod{\langle p\rangle}$ with some $t \in \mathbb{Z}^*_{\mathsf{m}}$ for $j = 0, \ldots, g-1$. Setting $\eta = \mathrm{Tr}_{K|Z}(\zeta)$, for any i and j,

$$\rho_{t_i}(\eta_j) = \rho_{t_i}(\rho_{t_j}(\eta)) = \rho_{t_i \cdot t_j}(\eta) = \rho_{t_{i+j}}(\eta) = \eta_{i+j}.$$

In particular, Ω_Z is symmetric. We can show that:

Lemma 5: $\Omega_Z^* \Omega_Z = (\mathrm{Tr}_{Z|\mathbb{Q}}(\overline{\eta}_i \eta_j))_{0 \le i,j < g} = \mathsf{m}\mathsf{I}_{\mathsf{g}} - \mathsf{d}\mathbf{1}\cdot\mathbf{1}^{\mathsf{T}}$ $(\in \mathbb{Z}^{\mathsf{g}\times\mathsf{g}})$.

Corollary 1: The set $\{\mathsf{m}^{-1}(\eta_0 - \mathsf{d}), \cdots, \mathsf{m}^{-1}(\eta_{\mathsf{g}-1} - \mathsf{d})\}$ is the dual basis of conjugate η-basis $\{\overline{\eta_0}, \cdots, \overline{\eta_{g-1}}\}$, i.e. for any $0 \le i, j < g$,

$$\mathrm{Tr}_{Z|\mathbb{Q}}\Big(\frac{\eta_i - d}{\mathsf{m}} \cdot \overline{\eta_j}\Big) = \delta_{ij}.$$

In particular, $R_Z^\vee = \mathbb{Z}\frac{\eta_0 - d}{\mathsf{m}} + \cdots + \mathbb{Z}\frac{\eta_{g-1} - d}{\mathsf{m}}$.

Define a $g \times g$ matrix Γ_Z over Z as

$$\Gamma_Z \stackrel{\mathrm{def}}{=} \left(\rho_{t_i}\Big(\frac{\overline{\eta_j} - d}{\mathsf{m}}\Big)\right)_{0 \le i,j < g} \quad (\in Z^{g\times g}).$$

Corollary 1 means that $\overline{\Gamma}_Z^T \overline{\Omega}_Z = I$. Since Ω_Z is symmetric,

$$\Gamma_Z \Omega_Z = \Omega_Z \Gamma_Z = I. \tag{8}$$

Lemma 6: For any $\boldsymbol{b} = \Omega_Z \boldsymbol{a}$, we have

$$\boldsymbol{a} = \Gamma_Z \boldsymbol{b} = \frac{1}{\mathsf{m}}\Big(\overline{\Omega}_Z \boldsymbol{b} - d\Big(\sum_j b_j\Big) \cdot \mathbf{1}\Big).$$

Let q be a power of the prime p. (Later we will use $q = p^l$ for the plaintext modulus and $q = p^r$ for the ciphertext modulus of the FV-type scheme.) Let $\mathfrak{q} = \mathfrak{q}_0$ be the first ideal that appears in the factorization of qR_Z (Eq. (6)). Recall that $R_Z/\mathfrak{q} \simeq \mathbb{Z}_q$.

Let

$$\Omega_Z^{(q)} \stackrel{\mathrm{def}}{=} \Omega_Z \bmod \mathfrak{q} \quad (\in (R_Z)_{\mathfrak{q}}^{g\times g} \simeq \mathbb{Z}_q^{g\times g})$$

$\Omega_Z^{(q)}$ is invertible mod $\mathfrak{q}$.

Definition 2: Define $\boldsymbol{\xi} = (\xi_0, \ldots, \xi_{g-1}) \in (R_Z)_{\mathfrak{q}}^g$ by $\boldsymbol{\eta}^T \equiv \boldsymbol{\xi}^T \Omega_Z^{(q)} \pmod{\mathfrak{q}}$. We call the basis $\boldsymbol{\xi}$ of $(R_Z)_{\mathfrak{q}}$ over $\mathbb{Z}_q$ *ξ-basis* of R_Z (with respect to $\mathfrak{q}$). For any $a \in (R_Z)_{\mathfrak{q}}$, there exists unique $\boldsymbol{b} \in \mathbb{Z}_q^g$ satisfying that $a = \boldsymbol{\xi}^T \boldsymbol{b}$. We call such $\boldsymbol{b} \in \mathbb{Z}_q^g$ as *ξ-vector* of $a \in (R_Z)_{\mathfrak{q}}$.

Lemma 7: For any $a \in R_Z$ it holds that

$$a \equiv \boldsymbol{\eta}^T \cdot \boldsymbol{a} \;\Leftrightarrow\; a \equiv \boldsymbol{\xi}^T \cdot (\Omega_Z^{(q)} \cdot \boldsymbol{a}) \pmod{\mathfrak{q}}$$
$$a = \boldsymbol{\eta}^T \cdot \boldsymbol{a} \;\Leftrightarrow\; \sigma_Z(a) = \Omega_Z \boldsymbol{a}$$
$$a \equiv \boldsymbol{\xi}^T \cdot \boldsymbol{b} \pmod{\mathfrak{q}} \;\Leftrightarrow\; \sigma_Z(a) \equiv \boldsymbol{b} \pmod{\mathfrak{q}}$$

Lemma 8: If $a_1 = \boldsymbol{\xi}^T \cdot \boldsymbol{b}_1$ and $a_2 = \boldsymbol{\xi}^T \cdot \boldsymbol{b}_2$, then $a_1 a_2 = \boldsymbol{\xi}^T \cdot (\boldsymbol{b}_1 \odot \boldsymbol{b}_2)$.

3.4 Conversion between η- and ξ-Vectors

3.4.1 Resolution of Unity in R_Z mod $\mathfrak{q}$

As stated before, by Hensel-lifting the factorization of $\Phi_{\mathsf{m}}(X) \bmod p$ (Eq. (3)) to modulus q which is a power of p, we get factorization of $\Phi_{\mathsf{m}}(X) \bmod q$: $\Phi_{\mathsf{m}}(X) \equiv \overline{F}_0(X) \cdots \overline{F}_{g-1}(X) \pmod{q}$. According to this factorization, the ideal qR of R is factored as $qR = \mathfrak{Q}_0 \cdots \mathfrak{Q}_{g-1}$ with ideals $\mathfrak{Q}_i = (q, \overline{F}_i(\zeta))$ of R.

For each $i = 0, \ldots, g-1$, let $G_i(X) \stackrel{\mathrm{def}}{=} \prod_{j \ne i} \overline{F}_j(X) \pmod{q}$ and $P_i(X) \stackrel{\mathrm{def}}{=} (G_i(X)^{-1} \bmod (q, \overline{F}_i(X))) \cdot G_i(X) \pmod{q}$. It is verified that the set $\{\tau_i = P_i(\zeta)\}_{i=0}^{g-1}$ constitutes a *resolution of unity* in R mod q, i.e.

$$\tau_i \equiv \begin{cases} 1 \pmod{\mathfrak{Q}_i} & (i = 0, \ldots, g-1) \\ 0 \pmod{\mathfrak{Q}_j} & (j \ne i) \end{cases}$$

and it holds that

$$\sum_{i=0}^{g-1} \tau_i \equiv 1,\ \tau_i^2 \equiv \tau_i,\ \tau_i \tau_j \equiv 0 \pmod{q}\,(j \ne i).$$

By the Chinese remainder theorem, the resolution of unity $\{\tau_i\}_{i=0}^{g-1}$ is uniquely determined mod qR. In the following we take coefficients of each τ_i from $[-q/2, q/2)$ over the basis $B' = \{\zeta, \zeta^2, \ldots, \zeta^{\mathsf{m}-1}\}$ of R.

Lemma 9: For any $0 \le i < g$ it is that $\tau_i \in R_Z$, and $\{\tau_i\}_{i=0}^{g-1}$

is also a resolution of unity in R_Z mod q.

Using the resolution of unity $\{\tau_i\}_{i=0}^{g-1}$ in R_Z, we can compute $a_i \in \mathbb{Z}_q$ satisfying $a \equiv a_i \pmod{\mathfrak{q}_i}$ given $a \in R_Z$, as follows:

$$a \bmod \mathfrak{q}_i = a\tau_i \bmod q \;=\; a_i\tau_i \bmod q \overset{\text{dividing by } \tau_i}{\mapsto} a_i.$$

3.4.2 Computation of $\Omega_Z^{(q)}$

Now we can compute the matrix $\Omega_Z^{(q)} = \left(\eta_{i+j} \bmod \mathfrak{q}\right)_{0\le i,j<g}$ ($\in \mathbb{Z}_q^{g\times g}$) by computing the entities η_{i+j} in Ω_Z as cyclotomic integers and reducing them modulo $\mathfrak{q}$ ($= \mathfrak{q}_0$) using the resolution of unity $\{\tau_i\}_{i=0}^{g-1}$. Since the matrix $\Omega_Z^{(q)}$ has cyclic structure (the $(i+1)$-th row is a left shift of the i-th row), it is sufficient to compute its first row. Here, we remark that once we have computed the matrix $\Omega_Z^{(q)}$, we can totally forget the original structure of cyclotomic ring R, and all we need is doing various computations among η- and ξ-vectors (of elements in R_Z) with necessary conversion between them using the matrix $\Omega_Z^{(q)}$.

3.4.3 Computation of $\boldsymbol{b} = \Omega_Z^{(q)} \cdot \boldsymbol{a}$

To convert η-vector $\boldsymbol{a}$ of an element $a = \boldsymbol{\eta}^T \cdot \boldsymbol{a} \in R_Z$ to its corresponding ξ-vector $\boldsymbol{b}$ (satisfying $a = \boldsymbol{\xi}^T \cdot \boldsymbol{b}$), by Lemma 7, we need to compute a matrix-vector product $\boldsymbol{b} = \Omega_Z^{(q)} \cdot \boldsymbol{a}$. By Lemma 6, the inverse conversion from ξ-vector $\boldsymbol{b}$ to its corresponding η-vector $\boldsymbol{a} = \Gamma_Z \cdot \boldsymbol{b}$ also can be computed using a similar matrix-vector product $\overline{\Omega}_Z^{(q)} \cdot \boldsymbol{b}$. Here, $\overline{\Omega}_Z^{(q)} \overset{\text{def}}{=} \overline{\Omega_Z} \bmod \mathfrak{q}$.

By definition of $\Omega_Z^{(q)}$, the j-th component b_j of the product $\boldsymbol{b} = \Omega_Z^{(q)} \cdot \boldsymbol{a}$ is $b_j = \sum_{i=0}^{g-1} a_i\eta_{i+j}$ (where indexes are mod g and we omit mod $\mathfrak{q}$). This means that $\boldsymbol{b}$ is the convolution product of vector $\boldsymbol{\eta}$ and the reversal vector $(a_0, a_{g-1}, a_{g-2}, \cdots, a_1)$ of $\boldsymbol{a}$, where $\boldsymbol{\eta} = (\eta_i)_{i=0}^{g-1}$ is the first row of $\Omega_Z^{(q)}$.

Define two polynomials $f(X) = \sum_{i=0}^{g-1} \eta_i X^i$ and $g(X) = a_0 + \sum_{i=1}^{g-1} a_{g-i}X^i$ over $\mathbb{Z}_q$. Since $\boldsymbol{b}$ is the convolution product of $\boldsymbol{\eta}$ and the reversal vector of $\boldsymbol{a}$, it holds that $f(X)g(X) = \sum_{i=0}^{g-1} b_i X^i \pmod{X^g - 1}$. The polynomial product $f(X)g(X) \pmod{X^g - 1}$ can be computed in quasi-linear time $\tilde{O}(g)$ using the FFT multiplication. Thus, we know that conversions between η-vectors $\boldsymbol{a}$ and ξ-vectors $\boldsymbol{b}$ can be done in quasi-linear time $\tilde{O}(g)$.

Remark : In the BGV-type scheme, the ciphertext modulus makes a chain which contains L modulus $q_0, \cdots, q_{L-1}$ using L primes $p_0, \cdots, p_{L-1}$ s.t. $q_i = \Pi_{j=0}^{i} p_j$. For each modulus q_i, we generate a matrix $\Omega_Z^{(q_i)}$ in $\mathbb{Z}_{q_i}^{g\times g}$ to convert between η-vector and ξ-vector efficiently. More precisely, the reason why the method of Sects. 3.4.1 and 3.4.2 work is that the modulus q factors completely on the decomposition ring R_Z with respect to the prime p. When we choose each prime p_j to satisfy $p_j \equiv 1 \pmod{\mathsf{m}}$ then p_j factors completely on the cyclotomic ring and thus also factors completely on the decomposition ring R_Z. Therefore we can generate $\Omega_Z^{(p_j)}$ ($\in \mathbb{Z}_{p_j}^{g\times g}$) for such primes p_j using the method of Sects. 3.4.1 and 3.4.2, and we get $\Omega_Z^{(q_i)}$ by CRT-lifting the matrices $\Omega_Z^{(p_j)} (j = 0, \cdots, i)$ entity-wise:

$$\Omega_Z^{(q_i)} = CRT(\Omega_Z^{(p_0)}, \cdots, \Omega_Z^{(p_i)}) \in \mathbb{Z}_{q_i}^{g\times g}.$$

4. Homomorphic Encryption Based on Decomposition Ring

Now we construct two types of homomorphic encryption schemes over the decomposition ring: DR-FV and DR-BGV.

4.1 The Ring-LWE Problem on the Decomposition Ring

For security of our homomorphic encryption scheme over the decomposition ring, we need hardness of a variant of the decisional Ring-LWE problem over the decomposition ring. Let m be a prime. Let R_Z be the decomposition ring of the m-th cyclotomic ring R with respect to some prime p ($\neq \mathsf{m}$). Let q be a positive integer. For an element $s \in R_Z$ and a distribution χ over R_Z, define a distribution $A_{s,\chi}$ on $(R_Z)_q \times (R_Z)_q$ as follows: First choose an element a uniformly from $(R_Z)_q$ and sample an element e according to the distribution χ. Then return the pair $(a, b = as + e \bmod q)$.

Definition 3 (LWE problem on the decomposition ring): Let q, χ be as above. *The* R-DLWE$_{q,\chi}$ *problem on the decomposition ring* R_Z asks to distinguish samples from $A_{s,\chi}$ with $s \xleftarrow{\cup} \mathbb{Z}_q$ and (the same number of) samples uniformly chosen from $(R_Z)_q \times (R_Z)_q$.

Recall the search version of Ring-LWE problem is already proved to have a quantum polynomial time reduction from the approximate shortest vector problem of ideal lattices in *any number field* by Lyubashevsky, Peikert, and Regev [20]. They proved equivalence between the search and the decisional versions of the Ring-LWE problems only for cyclotomic rings. The key of their proof of equivalence is the existence of prime modulus q for Ring-LWE problem which totally decomposes into n prime ideal factors: $qR = \mathfrak{Q}_0 \cdots \mathfrak{Q}_{n-1}$. (Their residual fields $R/\mathfrak{Q}_i$ have polynomial order q and we can guess the solution of the Ring-LWE problem modulo ideal $\mathfrak{Q}_i$, and then we can verify validity of the guess using the assumed oracle for the decisional Ring-LWE problem.) Since the decomposition ring R_Z is a subring of the cyclotomic ring R, such modulus q totally decomposes into g prime ideals also in the decomposition ring R_Z: $qR_Z = \mathfrak{q}_0 \cdots \mathfrak{q}_{g-1}$. Using this decomposition, the proof of equivalence by [20] holds also over the decomposition rings R_Z, essentially as it is.

4.2 Parameters

Let m be a prime index of cyclotomic ring R. Choose a

(small) prime p, distinct from m. Let $d = \mathrm{ord}_{\mathsf{m}}^{\times}(p)$ be the multiplicative order of p mod m, and $g = (\mathsf{m}-1)/\mathsf{d}$ be the degree of the decomposition ring R_Z of R with respect to p. The plaintext modulus $t = p^l$ is a power of p and the ciphertext modulus q will be chosen below.

4.3 Sampling

We will use the following three types of sampling algorithms regarding as R_Z.

The uniform distribution $\mathcal{U}_q$: This is uniform distribution over $(\mathbb{Z}/q\mathbb{Z})^g$ identified with $(\mathbb{Z} \cap (-q/2, q/2])$.

The discrete gaussian distribution $\mathcal{G}_q(\sigma^2)$: The discrete gaussian distribution $\mathcal{G}_q(\sigma^2)$ draws a g dimension integer vector which rounds a normal gaussian distribution with zero-mean and variance σ^2 to the nearest integer and outputs that integer vector reduced modulo q (into interval $(-q/2, q/2]$).

The $\mathcal{HWT}(h)$ distribution: The $\mathcal{HWT}(h)$ distribution chooses a vector uniformly at random from $\{0, \pm 1\}^g$ that has h nonzero entries.

4.3.1 Bounds on Noises

We analyze bounds of noises according to the way of [8].

Let $a = \sum_{i=0}^{g} a_i \eta_i$ denote a random element of R_Z. The canonical embedding of each η_i has approximately Euclidean norm $\sqrt{\phi(\mathsf{m})}$ $(= \sqrt{gd})$. Letting V_a be the variance of each coefficient of $\boldsymbol{a}$, the variance of a is estimated as $V_a gd$.

When $\boldsymbol{a} \leftarrow \mathcal{U}_q$ then V_a is $(q-1)^2/12 \approx q^2/12$, hence the variance of a is $V_U = q^2 gd/12$. When $\boldsymbol{a} \leftarrow \mathcal{G}_q(\sigma^2)$ then the variance of a is $V_G = \sigma^2 gd$. When choosing $\boldsymbol{a} \leftarrow \mathcal{HWT}(h)$, the variance of a is $V_H = hd$.

The random element of R_Z is a sum of many independent distributed random variables, hence its distribution is approximated by a gaussian distribution of the same variance. We use six standard deviations as a bound on the size of a: $\|a\|_\infty \leq 6\sqrt{V}$. We use 16 $\sigma_a \sigma_b$ as a bound of product ab of two random variables both distributed closely to gaussian distributions of variances σ_a and σ_b, respectively. For a product of three random variables with variances σ_a, σ_b and σ_c, we use 40 $\sigma_a \sigma_b \sigma_c$.

4.4 Encoding Methods of Elements in R_Z

Basically, we use η-vectors $\boldsymbol{a} \in \mathbb{Z}^g$ to encode elements $a = \boldsymbol{\eta}^T \cdot \boldsymbol{a}$ in R_Z. To multiply two elements encoded by η-vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ modulo q, first we convert those η-vectors to corresponding ξ-vectors modulo q. We can multiply resulting ξ-vectors component-wise, and then re-convert the result into its corresponding η-vector modulo q. The functions eta_to_xi and xi_to_eta use the matrix $\Omega_Z^{(q)}$ computed in advance. $(\eta_i)_{i=0}^{g-1}$ denotes the first row of $\Omega_Z^{(q)}$.

mult_eta $(\boldsymbol{a}, \boldsymbol{b}, q)$:

$\boldsymbol{\alpha}_\xi = \mathsf{eta\_to\_xi}(\boldsymbol{a}, q)$
$\boldsymbol{\beta}_\xi = \mathsf{eta\_to\_xi}(\boldsymbol{b}, q)$
$\gamma_i = \alpha_i \beta_i \bmod q \ (i = 0, \ldots, g-1)$
return $\boldsymbol{c} = \mathsf{xi\_to\_eta}(\boldsymbol{\gamma}_\xi, q)$

eta_to_xi$(\boldsymbol{a}, q)$:

$a(X) = a_0 + \sum_{i=1}^{g-1} a_{g-i} X^i$
$c(X) = \sum_{i=0}^{g-1} \eta_i X^i$
$b(X) = a(X)c(X) \bmod (q, X^g - 1)$
return $\boldsymbol{b}_\xi = (b_0, \ldots, b_{g-1})$

xi_to_eta$(\boldsymbol{b}_\xi, q)$:

$b(X) = b_0 + \sum_{i=1}^{g-1} b_{g-i} X^i$
$c(X) = \sum_{i=0}^{g-1} \overline{\eta}_i X^i$
$a(X) = b(X)c(X) \bmod (q, X^g - 1)$
$t = b_0 + \cdots + b_{g-1} \bmod q$
return $\boldsymbol{a} = (m^{-1}(a_i - dt) \bmod q)_{i=0}^{g-1}$

These algorithms can be computed by $O(g \log g)$ operations of underlying mod q integers by using the FFT multiplications of degree g polynomials (see Sect. 3.4.3).

We regard plaintext vectors $\boldsymbol{m} \in \mathbb{Z}_t^g$ as ξ-vectors of corresponding elements $m_\xi = \boldsymbol{\xi}^T \boldsymbol{m} \in (R_Z)_t$. By Lemma 8 their products $m_\xi m'_\xi \in (R_Z)_t$ encodes the plaintext vector $\boldsymbol{m} \odot \boldsymbol{m}' \in \mathbb{Z}_t^g$. For a fixed integer base w, let $l_w = \lfloor \log_w(q) \rfloor + 1$. Any vector $\boldsymbol{a} \in \mathbb{Z}_q^g$ can be written as $\boldsymbol{a} = \sum_{k=0}^{l_w - 1} \boldsymbol{a}_k w^k$ with vectors $\boldsymbol{a}_k \in \mathbb{Z}_w^{l_w}$ of small entries. Define $\mathsf{WD}(\boldsymbol{a}) \stackrel{\text{def}}{=} (\boldsymbol{a}_k)_{k=0}^{l_w - 1}$ $(\in (\mathbb{Z}_w^g)^{l_w})$.

4.5 Two Types of Homomorphic Encryption Schemes

We construct two types of homomorphic encryption schemes based on the decomposition ring. The first scheme DR-FV is based on the FV scheme proposed by Fan and Vercauteren [9], in which a plaintext is placed in the most significant digits of ciphertext modulus (called MSD form). The second scheme DR-BGV is based on the BGV scheme proposed by Brakerski, Gentry, and Vaikuntanathan [5], in which a plaintext is placed in the lowest significant digits of ciphertext modulus (called LSD form).

FV-type

$$a + b \cdot s \equiv \left\lfloor \frac{q}{t} \right\rfloor f \cdot m + e \pmod{q}$$

BGV-type

$$a + b \cdot s \equiv f \cdot m + t \cdot e \pmod{q}$$

Here, (a, b) denotes a ciphertext, s denotes a secret key, f is a factor of plaintext m, e denotes a noise, l is ciphertext

level, and t and q are modulus of plaintext and ciphertext.

4.6 Scheme Description

Each scheme is given as symmetric key version. The public key version is easily derived as usual.

Our schemes have two functions for HE.Gen. SecretKeyGen() generates a secret key sk. Using the secret key sk, SwitchKeyGen($\boldsymbol{x}$, sk) generates a switching key swk from $\boldsymbol{x}$ to sk. Especially it gives the evaluation key evk taking $\boldsymbol{x} = \mathsf{sk}^2$: evk $\leftarrow$ SwitchKeyGen(sk^2, sk). Encrypt(sk, $\boldsymbol{m}$) encrypts a message $\boldsymbol{m}$ under the secret key sk and outputs a ciphertext ct, and Decrypt(sk, ct) decrypts the ciphertext ct under the secret key sk and outputs a message $\boldsymbol{m}$. Given an arithmetic circuit of function f, HE.Evaluate evaluates the circuit of f on given ciphertexts homomorphically. It uses Add(ct_1, ct_2) when evaluating an addition gate and uses Mult(swk, ct_1, ct_2) when evaluating a multiplication gate.

4.6.1 DR-FV Scheme

We use a ciphertext modulus $q = p^r$ and a plaintext modulus $t = p^l$ $(r > l)$. Let $\Delta = \frac{q}{t} = p^{r-l}$.

(1) Key Generation

The SecretKeyGen generates a secret key $\boldsymbol{s}_\xi$ as ξ-vector.

$\mathsf{SecretKeyGen}^{\mathsf{DR-FV}}(\mathsf{prm})$:

$\boldsymbol{s} \leftarrow \chi_{key}$
$\boldsymbol{s}_\xi \leftarrow \mathsf{eta\_to\_xi}(\boldsymbol{s}, q)$
return $\mathsf{sk} = \boldsymbol{s}_\xi \in \mathbb{Z}^g$

This can be computed by $O(g \log g)$ operations of underlying mod q integers.

The SwitchKeyGen takes as input a η-vector $\boldsymbol{x}$ and a secret key $\boldsymbol{s}_\xi$, and encrypts $\boldsymbol{x}$ under $\boldsymbol{s}_\xi$. The output switching key swk is used to linearize s^2 to s in homomorphic multiplication. $\mathsf{B}_{\mathsf{lin}}^{\mathsf{FV}}$ is the upper bound of canonical embedding norm of the noise included in the swk. The base-w decomposition technique is used to make swk be of less noise.

$\mathsf{SwitchKeyGen}^{\mathsf{DR-FV}}(\boldsymbol{x}, \mathsf{sk} = \boldsymbol{s}_\xi, \mathsf{prm})$:

For $j = 0$ to $l_w - 1$

$\boldsymbol{e}_j \leftarrow \chi_{err}$
$\boldsymbol{h}_j = w^j \boldsymbol{x} + \boldsymbol{e}_j \bmod q$
$\boldsymbol{B}_j \xleftarrow{\mathsf{u}} \mathbb{Z}_q^g$
$\boldsymbol{A}_j = -\boldsymbol{B}_j \odot \boldsymbol{s}_\xi + \mathsf{eta\_to\_xi}(\boldsymbol{h}_j, q) \bmod q$

return $\mathsf{swk} = ((\boldsymbol{A}_j, \boldsymbol{B}_j)_{j=0}^{l_w-1}, \nu = \mathsf{B}_{\mathsf{lin}}^{\mathsf{FV}})$

This can be computed by $O(l_w g \log g)$ operations of underlying mod q integers.

(2) Encryption

The input plaintext $\boldsymbol{m}$ is interpreted as a ξ-vector. In the resulting ciphertext, $\mathsf{B}_{\mathsf{clean}}^{\mathsf{FV}}$ denotes the bound of its noise.

$\mathsf{Encrypt}^{\mathsf{DR-FV}}(\mathsf{sk} = \boldsymbol{s}_\xi \in \mathbb{Z}^g, \boldsymbol{m} \in \mathbb{Z}_t^g, \mathsf{prm})$:

$\boldsymbol{e} \leftarrow \chi_{err}$
$\boldsymbol{b}_\xi \xleftarrow{\mathsf{u}} \mathbb{Z}_q^g$
$\boldsymbol{a}_\xi = -\boldsymbol{b}_\xi \odot \boldsymbol{s}_\xi + \Delta \boldsymbol{m} + \mathsf{eta\_to\_xi}(\boldsymbol{e}, q) \bmod q$
return $ct = (\boldsymbol{a}_\xi, \boldsymbol{b}_\xi, \nu = \mathsf{B}_{\mathsf{clean}}^{\mathsf{FV}})$

This can be computed by $O(g \log g)$ operations of underlying mod q integers.

(3) Decryption

Decryption removes the noise in the ciphertext by taking its right shift (by $\frac{1}{\Delta}$) and rounding.

$\mathsf{Decrypt}^{\mathsf{DR-FV}}(\mathsf{sk} = \boldsymbol{s}_\xi \in \mathbb{Z}^g, ct = (\boldsymbol{a}_\xi, \boldsymbol{b}_\xi, \nu), \mathsf{prm})$:

$\boldsymbol{h}_\xi = \boldsymbol{a}_\xi + \boldsymbol{b}_\xi \odot \boldsymbol{s}_\xi \bmod q$
$\boldsymbol{h} = \mathsf{xi\_to\_eta}(\boldsymbol{h}_\xi, q)$
$z = \left\lfloor \frac{\boldsymbol{h}}{\Delta} \right\rceil \bmod t$
$\boldsymbol{m} = \mathsf{eta\_to\_xi}(z, t)$
return $\boldsymbol{m}$

This can be computed by $O(g \log g)$ operations of underlying mod q integers.

(4) Addition

Addition algorithm takes as input two ciphertext ct_1 and ct_2 and outputs a ciphertext ct encrypting the sum of underlying plaintexts.

$\mathsf{Add}^{\mathsf{DR-FV}}(ct_1 = (\boldsymbol{a}_1, \boldsymbol{b}_1, \nu_1), ct_2 = (\boldsymbol{a}_2, \boldsymbol{b}_2, \nu_2), \mathsf{prm})$:

$\boldsymbol{a} = \boldsymbol{a}_1 + \boldsymbol{a}_2 \bmod q$
$\boldsymbol{b} = \boldsymbol{b}_1 + \boldsymbol{b}_2 \bmod q$

$\nu = \nu_1 + \nu_2$
return $ct = (\boldsymbol{a}, \boldsymbol{b}, \nu)$

This can be computed by $O(g)$ operations of underlying mod q integers.

(5) Multiplication

Multiplication algorithm takes as input two ciphertext ct_1 and ct_2 and outputs a ciphertext ct encrypting the product of underlying plaintexts.

$\mathsf{Mult}^{\mathsf{DR-FV}}(\mathsf{swk}, ct_1 = (\boldsymbol{a}_1, \boldsymbol{b}_1, \nu_1), ct_2 = (\boldsymbol{a}_2, \boldsymbol{b}_2, \nu_2), \mathsf{prm})$:

$\boldsymbol{\alpha} = \left\lfloor \frac{1}{\Delta} \cdot \mathsf{xi\_to\_eta}(\boldsymbol{a}_1 \odot \boldsymbol{a}_2 \bmod q^2/t) \right\rceil$
$\boldsymbol{\beta} = \left\lfloor \frac{1}{\Delta} \cdot \mathsf{xi\_to\_eta}(\boldsymbol{a}_1 \odot \boldsymbol{b}_2 + \boldsymbol{a}_2 \odot \boldsymbol{b}_1 \bmod q^2/t) \right\rceil$
$\boldsymbol{\gamma} = \left\lfloor \frac{1}{\Delta} \cdot \mathsf{xi\_to\_eta}(\boldsymbol{b}_1 \odot \boldsymbol{b}_2 \bmod q^2/t) \right\rceil$
$\boldsymbol{\alpha}' = \mathsf{eta\_to\_xi}(\boldsymbol{\alpha}, q)$
$\boldsymbol{\beta}' = \mathsf{eta\_to\_xi}(\boldsymbol{\beta}, q)$
$\boldsymbol{\gamma}' = \mathsf{eta\_to\_xi}(\boldsymbol{\gamma}, q)$

$\nu = \mathsf{B}_{\mathsf{direct\_mult}}^{\mathsf{FV}}(\nu_1, \nu_2)$
$ct = \mathsf{Linearize}^{\mathsf{DR-FV}}(\mathsf{swk}, (\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}', \nu), \mathsf{prm})$

return ct

Here, Linearize switches the key in the ciphertext from s^2 to s using swk.

$\mathsf{Linearize}^{\mathsf{DR-FV}}(\mathsf{swk}, ct = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \nu), \mathsf{prm})$:

$(\boldsymbol{a}, \boldsymbol{b}) = \mathsf{KeySwitch}^{\mathsf{DR-FV}}(\mathsf{swk}, \boldsymbol{\gamma}, \mathsf{prm})$
$\boldsymbol{a}' = \boldsymbol{\alpha} + \boldsymbol{a} \bmod q$
$\boldsymbol{b}' = \boldsymbol{\beta} + \boldsymbol{b} \bmod q$
$\nu' = \nu + \mathsf{B}_{\mathsf{lin}}^{\mathsf{FV}}$
return $ct = (\boldsymbol{a}', \boldsymbol{b}', \nu')$

KeySwitch takes a switching key swk of $\boldsymbol{y}$ and a vector $\boldsymbol{x}$ as input, and returns a ciphertext encrypting the product $\boldsymbol{x} \odot \boldsymbol{y}$. In the multiplication, swk is an encryption of s^2 and $\boldsymbol{x} = \gamma$ so the output is a ciphertext of $s^2\gamma$.

$\mathsf{KeySwitch}^{\mathsf{DR-FV}}(\mathsf{swk} = (\{\boldsymbol{A}_j, \boldsymbol{B}_j\}_{j=0}^{l_w-1}, \nu), \boldsymbol{x}, \mathsf{prm})$:

$\boldsymbol{d} = \mathsf{xi\_to\_eta}(\boldsymbol{x}, q)$
Decompose $\boldsymbol{d} = \Sigma_{j=0}^{l_w-1} \boldsymbol{d}_j w^j$
$\boldsymbol{d}'_j = \mathsf{eta\_to\_xi}(\boldsymbol{d}_j, q)$ for $j = 0 \cdots l_w - 1$
$\boldsymbol{A} = \Sigma_{j=0}^{l_w-1} \boldsymbol{A}_j \odot \boldsymbol{d}'_j \bmod q$
$\boldsymbol{B} = \Sigma_{j=0}^{l_w-1} \boldsymbol{B}_j \odot \boldsymbol{d}'_j \bmod q$
return $(\boldsymbol{A}, \boldsymbol{B})$

As easily verified, the total complexity of $\mathsf{Mult}^{\mathsf{DR-FV}}$ is $O(l_w g \log g)$ of underlying mod q integers.

4.6.2 DR-BGV Scheme

Ciphertext modulus chain: For scaling down ciphertext, the ciphertext modulus makes a chain which contains L modulus $q_0, \cdots, q_{L-1}$ using L primes $p_0, \cdots, p_{L-1}$ s.t. $q_i = \Pi_{j=0}^{i} p_j$ and primes are $p_i \equiv 1 \pmod{\mathsf{m}}$. We call a modulus q_i ciphertext as a level-i ciphertext. The level of a fresh ciphertext is $L-1$ and one level is consumed by one multiplication. In the scheme, we use another special modulus q_s to reduce noise.

(1) Key Generation

The SecretKeyGen generates a secret key of the maximum level $L-1$.

$\mathsf{SecretKeyGen}^{\mathsf{DR-BGV}}(\mathsf{prm})$:

$\boldsymbol{s} \leftarrow \chi_{key}$
$\boldsymbol{s}_\xi \leftarrow \mathsf{eta\_to\_xi}(\boldsymbol{s}, q_{L-1})$
return $\mathsf{sk} = \boldsymbol{s}_\xi \in \mathbb{Z}^g$

This can be computed by $O(g \log g)$ operations of underlying mod q_{L-1} integers.

In the SwitchingKeyGen, we temporarily enlarge the modulus by multiplying q_s, to reduce the noise occurring.

$\mathsf{SwitchKeyGen}^{\mathsf{DR-BGV}}(\boldsymbol{x}, \mathsf{sk} = \boldsymbol{s}_\xi, \mathsf{prm})$:

For $j = 0$ to $l_w - 1$

$\boldsymbol{e}_j \leftarrow \chi_{err}$
$\boldsymbol{h}_j = q_s w^j \boldsymbol{x} + t\boldsymbol{e}_j \bmod q_{L-1} q_s$
$\boldsymbol{B}_j \xleftarrow{\mathsf{u}} \mathbb{Z}_{q_{L-1}q_s}^g$
$\boldsymbol{A}_j = -\boldsymbol{B}_j \odot \boldsymbol{s}_\xi + \mathsf{eta\_to\_xi}(\boldsymbol{h}_j, q_{L-1}q_s) \bmod q_{L-1}q_s$

return $\mathsf{swk} = ((\boldsymbol{A}_j, \boldsymbol{B}_j)_{j=0}^{l_w-1}, \nu = \mathsf{B}_{\mathsf{lin}}^{\mathsf{BGV}})$

This can be computed by $O(l_w g \log g)$ operations of underlying mod $q_{L-1}q_s$ integers.

(2) Encryption

The level of a fresh ciphertext is the maximum level $L-1$.

$\mathsf{Encrypt}^{\mathsf{DR-BGV}}(\mathsf{sk} = \boldsymbol{s}_\xi \in \mathbb{Z}^g, \boldsymbol{m} \in \mathbb{Z}_t^g, \mathsf{prm})$:

$\boldsymbol{e} \leftarrow \chi_{err}$
$\boldsymbol{e}' = t\boldsymbol{e} \bmod q_{L-1}$
$\boldsymbol{b}_\xi \xleftarrow{\mathsf{u}} \mathbb{Z}_{q_{L-1}}^g$
$\boldsymbol{a}_\xi = -\boldsymbol{b}_\xi \odot \boldsymbol{s}_\xi + \boldsymbol{m} + \mathsf{eta\_to\_xi}(\boldsymbol{e}', q_{L-1}) \bmod q_{L-1}$
return $ct = (\boldsymbol{a}_\xi, \boldsymbol{b}_\xi, f = 1, l = L-1, \nu = \mathsf{B}_{\mathsf{clean}}^{\mathsf{FV}})$

This can be computed by $O(g \log g)$ operations of underlying mod q_{L-1} integers.

(3) Decryption

Decryption algorithm removes the noise placed in upper digits by taking residue with respect to plaintext modulus t.

$\mathsf{Decrypt}^{\mathsf{DR-BGV}}(\mathsf{sk} = \boldsymbol{s}_\xi \in \mathbb{Z}^g, ct = (\boldsymbol{a}_\xi, \boldsymbol{b}_\xi, f, l, \nu), \mathsf{prm})$:

$\boldsymbol{h}_\xi = \boldsymbol{a}_\xi + \boldsymbol{b}_\xi \odot \boldsymbol{s}_\xi \bmod q_l$
$\boldsymbol{h} = \mathsf{xi\_to\_eta}(\boldsymbol{h}_\xi, q_l)$
$z = f^{-1}\boldsymbol{h} \bmod t$
$\boldsymbol{m} = \mathsf{eta\_to\_xi}(z, t)$
return $\boldsymbol{m}$

This can be computed by $O(g \log g)$ operations of underlying mod q_l integers.

(4) Rescale

Rescale scales down a modulus of a given ciphertext from q_l to $q_{l'}$ ($q_l > q_{l'}$), to reduce its noise. Setting P be $q_l/q_{l'}$, the noise is scaled down by a factor of $1/P$, however an additional noise term $\mathsf{B}_{\mathsf{scale}}^{\mathsf{BGV}}$ appears.

$\mathsf{Rescale}(ct = (\boldsymbol{a}_\xi, \boldsymbol{b}_\xi, f, l, \nu), l')$:

$\boldsymbol{a} = \mathsf{xi\_to\_eta}(\boldsymbol{a}_\xi, q_l)$
$\boldsymbol{b} = \mathsf{xi\_to\_eta}(\boldsymbol{b}_\xi, q_l)$
Fix $\boldsymbol{\delta_a}$ s.t. $\boldsymbol{\delta_a} \equiv \boldsymbol{a} \pmod{P}$ and $\boldsymbol{\delta_a} \equiv 0 \pmod{t}$,
$\boldsymbol{\delta_b}$ s.t. $\boldsymbol{\delta_b} \equiv \boldsymbol{b} \pmod{P}$ and $\boldsymbol{\delta_b} \equiv 0 \pmod{t}$
$\boldsymbol{a}' = (\boldsymbol{a} - \boldsymbol{\delta_a})/P$, $\boldsymbol{b}' = (\boldsymbol{b} - \boldsymbol{\delta_b})/P$
$\boldsymbol{a}'_\xi = \mathsf{eta\_to\_xi}(\boldsymbol{a}', q_{l'})$
$\boldsymbol{b}'_\xi = \mathsf{eta\_to\_xi}(\boldsymbol{b}', q_{l'})$
$f' = f/P \bmod t$
$\nu' = \nu/P + \mathsf{B}_{\mathsf{scale}}^{\mathsf{BGV}}$
return $ct = (\boldsymbol{a}'_\xi, \boldsymbol{b}'_\xi, f', l', \nu')$

This can be computed by $O(g \log g)$ operations of underlying mod q_l and mod $q_{l'}$ integers.

(5) Addition

Assume the level of ct_2 is not greater than the level of ct_1.

$\mathsf{Add}^{\mathsf{DR-BGV}}(ct_1 = (\boldsymbol{a}_1, \boldsymbol{b}_1, f_1, l_1, \nu_1), ct_2 = (\boldsymbol{a}_2, \boldsymbol{b}_2, f_2, l_2, \nu_2), \mathsf{prm})$:

$ct'_2 = (\boldsymbol{a}'_2, \boldsymbol{b}'_2, f'_2, l_1, \nu'_2) = \mathsf{Rescale}(ct_{2\_msd}, l_1)$
$\boldsymbol{a} = f'_2\boldsymbol{a}_1 + f_1\boldsymbol{a}'_2 \bmod q_{l_1}$
$\boldsymbol{b} = f'_2\boldsymbol{b}_1 + f_1\boldsymbol{b}'_2 \bmod q_{l_1}$
$f = f_1 f'_2$
$\nu = \nu_1 + \nu'_2$
return $ct = (\boldsymbol{a}, \boldsymbol{b}, f, l_1, \nu)$

This can be computed by $O(g \log g)$ operations of underlying mod q_{l_1} and mod q_{l_2} integers.

(6) Multiplication

Assume the level of ct_2 is not greater than the level of ct_1. At the last step in Mult, one level is consumed to reduce the incurred noise.

$\mathsf{Mult}^{\mathsf{DR-BGV}}(\mathsf{swk}, ct_1 = (\boldsymbol{a}_1, \boldsymbol{b}_1, f_1, l_1, \nu_1), ct_2 = (\boldsymbol{a}_2, \boldsymbol{b}_2, f_2, l_2, \nu_2), \mathsf{prm})$:

$ct'_2 = (\boldsymbol{a}'_2, \boldsymbol{b}'_2, f'_2, l_1, \nu'_2) = \mathsf{Rescale}(ct_2, l_1)$
$\boldsymbol{\alpha} = \boldsymbol{a}_1 \odot \boldsymbol{a}'_2 \bmod q_{l_1}$
$\boldsymbol{\beta} = \boldsymbol{a}_1 \odot \boldsymbol{b}'_2 + \boldsymbol{a}'_2 \odot \boldsymbol{b}_1 \bmod q_{l_1}$
$\boldsymbol{\gamma} = \boldsymbol{b}_1 \odot \boldsymbol{b}'_2 \bmod q_{l_1}$
$f = f_1 f'_2$
$\nu = \mathsf{B}^{\mathsf{BGV}}_{\mathsf{direct\_mult}}(\nu_1, \nu'_2)$
$(\boldsymbol{a}', \boldsymbol{b}', f, l_1, \nu'') = \mathsf{Linearize}^{\mathsf{DR-BGV}}(\mathsf{swk}, (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, f, l_1, \nu), \mathsf{prm})$
$ct = \mathsf{Rescale}((\boldsymbol{a}', \boldsymbol{b}', f, l_1, \nu''), l_1 - 1)$
return ct

Here, the key switching procedure is described below.

$\mathsf{Linearize}^{\mathsf{DR-BGV}}(\mathsf{swk}, ct = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, f, l, \nu), \mathsf{prm})$:

$(\boldsymbol{a}, \boldsymbol{b}, f, l, \nu') = \mathsf{KeySwitch}^{\mathsf{DR-BGV}}(\mathsf{swk}, (\boldsymbol{\gamma}, f, l), \mathsf{prm})$
$\boldsymbol{a}' = \boldsymbol{\alpha} + \boldsymbol{a} \bmod q_l$
$\boldsymbol{b}' = \boldsymbol{\beta} + \boldsymbol{b} \bmod q_l$
$\nu'' = \nu + \nu' + \mathsf{B}_{\mathsf{lin}}$
return $ct = (\boldsymbol{a}', \boldsymbol{b}', f, l, \nu'')$

In the last step of KeySwitch, the size of modulus is lowered from $q_l q_s$ to q_l to reduce noise in the key.

$\mathsf{KeySwitch}^{\mathsf{DR-BGV}}(\mathsf{swk} = (\{\boldsymbol{A}_j, \boldsymbol{B}_j\}_{j=0}^{l_w-1}, \nu), (\boldsymbol{x}, f, l), \mathsf{prm})$:

$\boldsymbol{d} = \mathsf{xi\_to\_eta}(\boldsymbol{x}, q_l)$
Decompose $\boldsymbol{d} = \Sigma_{j=0}^{l_w-1} \boldsymbol{d}_j w^j$
$\boldsymbol{d}'_j = \mathsf{eta\_to\_xi}(\boldsymbol{d}_j, q_l)$ for $j = 0 \cdots l_w - 1$
$\boldsymbol{A} = \Sigma_{j=0}^{l_w-1} \boldsymbol{A}_j \odot \boldsymbol{d}'_j \bmod q_l q_s$
$\boldsymbol{B} = \Sigma_{j=0}^{l_w-1} \boldsymbol{B}_j \odot \boldsymbol{d}'_j \bmod q_l q_s$
$(\boldsymbol{a}, \boldsymbol{b}, f, l, \nu') = \mathsf{Rescale}((\boldsymbol{A}, \boldsymbol{B}, f, l + s, \nu), l)$
return $(\boldsymbol{a}, \boldsymbol{b}, f, l, \nu')$

The total complexity of $\mathsf{Mult}^{\mathsf{DR-BGV}}$ is sum of $O(g \log g)$ operations with respect to the ciphertext modulus q_2 and $q_1 q_s$, and $O(l_w g \log g)$ operations with respect to q_1.

It is direct to see that:

Theorem 1: The decomposition ring homomorphic encryption schemes DR-FV and DR-BGV are indistinguishably secure under the chosen plaintext attack if the R-DLWE$_{q,\chi_{key},\chi_{err}}$ problem on the decomposition ring R_Z is hard.

For correctness we have the following theorem. (The proof is in Appendix C)

Theorem 2: The decomposition ring homomorphic encryption schemes DR-FV and DR-BGV will be fully homomorphic under circular security assumption (i.e., an encryption of secret key $\boldsymbol{s}$ does not leak any information about $\boldsymbol{s}$) by taking ciphertext modulus $q = O(\lambda^{\log \lambda})$ for DR-FV , and $p_i = \Omega(\sqrt{\lambda})$ $(i = 1, \ldots, L = \Omega(\log \lambda))$ and $q_s = \Omega(\sqrt{\lambda})$ for DR-BGV.

5. Benchmark Results

We implemented our two decomposition ring homomorphic encryption schemes DR-FV and DR-BGV, using the C++ language and performed several experiments using different parameters, comparing efficiency of our implementation of DR-FV, DR-BGV and the homomorphic encryption library HElib by Halevi and Shoup [14], which is based on the BGV scheme over ordinal cyclotomic rings [5]. SEAL [19] is a homomorphic encryption library of FV-type. Recall that our target plaintext space is a power of small prime since we think many applications will use such plaintext modulus (e.g. 2^l), however in the CRT batching of SEAL, the plaintext modulus t is required to be $t \equiv 1 \pmod{\mathsf{m}}$ and cannot be a power of small prime. For this reason, we do not compare our schemes with SEAL.

For notation of parameters, see Sect. 4.2. As common parameters, we choose four values of prime m so that the m-th cyclotomic ring R will have as many number of plaintext slots (i.e., large g and small d values) as possible. The plaintext modulus $t = 2^l$ is fixed as $l = 8$. The noise parameter $s_{err} = \sqrt{2\pi}\sigma_{err}$ is fixed as $\sigma_{err} = 3.2$. The ciphertext modulus q of bit-length r is chosen as small as possible so that it enables homomorphic evaluation of exponentiation by 2^8 (i.e., $\mathsf{Enc}(\boldsymbol{s}, \boldsymbol{m})^{2^8}$) with respect to each implementation. In the DR-FV, the modulus is $q = 2^r$ with r in Table 1. In the BGV-type schemes (DR-BGV and HElib), the modulus is $q_8 = p_0 \cdots p_8 q_s$ and we describe the bit-length r of q_8 in Table 1. Table 1 summarizes the chosen parameters.

Assuming that there is no special attack utilizing the particular algebraic structure of involving rings, corresponding security parameters λ are estimated using the lwe-estimator-9302d4204b4f by [2], [3].

Table 2 shows timing results for HElib in milliseconds on Intel Celeron(R) CPU G1840 @ 2.80 GHz × 2. (We

Table 1 Chosen parameters.

	m	g	d	l	r (DR-FV)	r (DR-BGV)	r (HElib)
par-127	127	18	7	8	162	189	135
par-8191	8191	630	13	8	210	247	250
par-43691	43691	1285	34	8	234	258	256
par-131071	131071	7710	17	8	242	261	-

Table 2 Timing results of HElib on mod-2^l integer plaintexts.

	λ	Enc	Dec	Add	Mult	Exp-by-2^8
par-127	26	0.23	0.18	0.00	0.66	4.78
par-8191	92	30.45	210.77	0.84	107.53	512.64
par-43691	237	268.00	5158.44	4.74	634.69	4187.81
par-131071	-	-	-	-	-	-

could not perform the test for par-131071 due to shortage of memory.)

The secret key is chosen uniformly random among binary vectors of Hamming weight 64 over the power basis (default of HElib) and we encrypt g number of mod-2^l integer plaintexts into a single HElib ciphertext using plaintext slots. As seen in Sect. 2.5, HElib basically realizes $GF(2^d)$ arithmetic in each of g slots. If we want to encrypt mod-2^l integer plaintexts on slots and to homomorphically evaluate on them, we can use only 1-dimensional constant polynomials in each $d(= n/g)$-dimensional slots. This should cause certain waste in time and space. In fact, for example, timings for par-43691 ($g = 1285$) is much larger than two times of those for par-8191 ($g = 630$) while being the ratio of g is $1285/630 \approx 2$. This indicates that the HElib scheme cannot handle many mod-2^l integer slots with high parallelism. So, to encrypt large number of mod-2^l integer plaintexts using HElib, we have no choice but to prepare many ciphertexts, each of which encrypts a divided set of small number of plaintexts on their slots.

On the other hand, Table 3 and Table 4 shows timing results (also in milliseconds on Intel Celeron(R) CPU G1840 @ 2.80GHz × 2) for our DR-FV scheme and DR-BGV scheme, respectively.

The secret key is chosen uniformly random among binary vectors of Hamming weight 64 over η-basis and we encrypt g number of mod-2^l integer plaintexts into a single DR-FV or DR-BGV ciphertext. As seen, DR-BGV scheme is a little bit faster than DR-FV scheme, due to the effect of rescaling ciphertext modulus to the smaller ones after linearization. In both schemes, timings are approximately linear with respect to the number of slots g. This means that the DR-FV and DR-BGV schemes can handle many mod-2^l slots with high parallelism, as expected. We can encrypt large number of mod-2^l integer plaintexts into a single DR-FV or DR-BGV ciphertext using mod-2^l slots without waste, and can homomorphically compute on them with high parallelism.

Then, which is faster to encrypt many number of mod-2^l integer plaintexts between the following two cases?

(1) A single DR-BGV ciphertext with many plaintext slots.

Table 3 Timing results of DR-FV on mod-2^l integer plaintexts.

	λ	Enc	Dec	Add	Mult	Exp-by-2^8
par-127	-	0.14	0.12	0.00	0.57	4.47
par-8191	29	7.39	7.37	0.03	39.43	318.65
par-43691	32	17.38	17.19	0.11	92.14	741.42
par-131071	91	104.33	103.93	0.97	574.44	4620.22

Table 4 Timing results of DR-BGV on mod-2^l integer plaintexts.

	λ	Enc	Dec	Add	Mult	Exp-by-2^8
par-127	-	0.06	0.08	0.00	0.54	3.53
par-8191	29	2.49	2.35	0.24	21.23	127.34
par-43691	32	5.17	5.19	0.59	50.85	293.52
par-131071	84	30.14	29.35	3.70	282.11	1678.52

(2) Many HElib ciphertexts with small number of plaintext slots.

The result for par-131071 of Table 4 shows we can encrypt 7710 mod-2^l integer slots in a single DR-BGV ciphertext with security parameter $\lambda = 84$ with timing:

$$(30.14, 29.35, 3.70, 282.11, 1678.52)$$

On a while, the result for par-8191 of Table 2 shows one can encrypt the same number of 7710 mod-2^l integer slots using $\lceil 7710/630 \rceil = 13$ ciphertexts with security parameter $\lambda = 92$. The 13 times of the line par-8191 of Table 2 is

$$(395.85, 2740.01, 10.92, 1397.89, 6664.32).$$

Thus, our benchmark indicates that Case (1) (a single DR-BGV ciphertext with many slots) is significantly faster than Case (2) (many HElib ciphertexts with small number of plaintext slots) under the similar level of security parameters.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR1503. This work is further supported by the JSPS KAKENHI Grant Number 17K05353.

References

[1] S. Arita and S. Handa, "Subring homomorphic encryption," Information Security and Cryptology - ICISC 2017, H. Kim and D.C. Kim, eds., pp.112–136, Springer International Publishing, Cham, 2018.
[2] M.R. Albrecht, "On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL," Advances in Cryptology - EUROCRYPT 2017, J.S. Coron and J.B. Nielsen, eds., pp.103–129, Springer International Publishing, Cham, 2017.
[3] M.R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of learning with errors," J. Math. Cryptol., vol.9, no.3, pp.169–203, 2015.
[4] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," Advances in Cryptology - CRYPTO 2012, R. Safavi-Naini and R. Canetti, eds., pp.868–886, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
[5] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully

homomorphic encryption without bootstrapping," Proc. 3rd Innovations in Theoretical Computer Science Conference, ITCS'12, pp.309–325, New York, NY, USA, ACM, 2012.
[6] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," Proc. 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS'11, pp.97–106, IEEE Computer Society, Washington, DC, USA, 2011.
[7] J.H. Cheon, M. Kim, and K. Lauter, "Homomorphic computation of edit distance," Financial Cryptography and Data Security 2015, LNCS 8976, pp.194–212, 2015.
[8] A. Costache and N.P. Smart, "Which ring based somewhat homomorphic encryption scheme is best?," Proc. RSA Conference on Topics in Cryptology - CT-RSA 2016 - vol.9610, pp.325–340, Springer-Verlag New York, New York, NY, USA, 2016.
[9] J. Fan and F. Vercauteren,"Somewhat practical fully homomorphic encryption," Cryptology ePrint Archive, Report 2012/144, 2012. https://eprint.iacr.org/2012/144
[10] C. Gentry, "Fully homomorphic encryption using ideal lattices," Proc. Forty-first Annual ACM Symposium on Theory of Computing, STOC'09, pp.169–178, ACM, New York, NY, USA, 2009.
[11] C. Gentry, S. Halevi, and N.P. Smart, "Better bootstrapping in fully homomorphic encryption," Public Key Cryptography – PKC 2012, M. Fischlin, J. Buchmann, and M. Manulis, eds., pp.1–16, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
[12] C. Gentry, S. Halevi, and N.P. Smart, "Homomorphic evaluation of the AES circuit," Advances in Cryptology - CRYPTO 2012, R. Safavi-Naini and R. Canetti, eds., pp.850–867, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
[13] T. Graepel, K. Lauter, and M. Naehrig, "ML confidential: Machine learning on encrypted data," Information Security and Cryptology - ICISC 2012, T. Kwon, M.K. Lee, and D. Kwon, eds., pp.1–21, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
[14] S. Halevi and V. Shoup, "Algorithms in HElib," Advances in Cryptology - CRYPTO 2014, J.A. Garay and R. Gennaro, eds., pp.554–571, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
[15] D. Kim and Y. Song, "Approximate homomorphic encryption over the conjugate-invariant ring," Information Security and Cryptology - ICISC 2018, K. Lee, ed., pp.85–102, Springer International Publishing, Cham, 2019.
[16] J. Liu, J. Li, S. Xu, and B.C. Fung, "Secure outsourced frequent pattern mining by fully homomorphic encryption," Big Data Analytics and Knowledge Discovery, S. Madria and T. Hara, eds., pp.70–81, Springer International Publishing, Cham, 2015.
[17] K. Lauter, A. Lopez-Alt, and M. Naehrig, "Private computation on encrypted genomic data," Progress in Cryptology - LATINCRYPT 2014, D.F. Aranha and A. Menezes, eds., pp.3–27, Springer International Publishing, Cham, 2015.
[18] W. Lu, S. Kawasaki, and J. Sakuma, "Using fully homomorphic encryption for statistical analysis of categorical, ordinal and numerical data," Network and Distributed System Security Symposium (NDSS), Feb. 2017.
[19] K. Laine and R. Player, "Simple encrypted arithmetic library - SEAL (v2.0)," Technical Report, Microsoft Research, MSR-TR-2016-52, Sept. 2016.
[20] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," Advances in Cryptology EUROCRYPT 2010, H. Gilbert, ed., pp.1–23, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
[21] V. Lyubashevsky, C. Peikert, and O. Regev, "A toolkit for ring-LWE cryptography," EUROCRYPT 2013, LNCS 7881, pp.35–54, Springer, 2013.
[22] N.P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations," Des. Codes Cryptogr., vol.71, no.1, pp.57–81, April 2014.
[23] S. Terada, H. Nakano, S. Okumura, and A. Miyaji, "On the security of Ring-LWE with homomorphic encryption," SCIS2018, 2018.

Appendix A: Proofs of Lemma

(1) Proof of Lemma 3.

$\mathrm{Tr}_{Z|\mathbb{Q}}(\eta_i) = \mathrm{Tr}_{Z|\mathbb{Q}}(\mathrm{Tr}_{K|Z}(\zeta^{t_i})) = \mathrm{Tr}_{K|\mathbb{Q}}(\zeta^{t_i})$. So, by Lemma 1, $\mathrm{Tr}_{Z|\mathbb{Q}}(\eta_i) = -1$ for any i. Similarly, $\mathrm{Tr}_{Z|\mathbb{Q}}(\overline{\eta}_i) = \mathrm{Tr}_{Z|\mathbb{Q}}(\mathrm{Tr}_{K|Z}(\zeta^{-t_i})) = \mathrm{Tr}_{K|\mathbb{Q}}(\zeta^{-t_i}) = -1$. □

(2) Proof of Lemma 4

Since the index m is prime, the cyclotomic ring R has a basis $B = \{1, \zeta, \ldots, \zeta^{\mathsf{m}-2}\}$ over $\mathbb{Z}$. Since ζ is a unit of R, $B' := \zeta B = \{\zeta, \zeta^2, \ldots, \zeta^{\mathsf{m}-1}\}$ is also a basis of R over $\mathbb{Z}$. The fixing group $G_Z = \langle \rho_p \rangle$ of Z acts on B' and decomposes it into g orbits $\zeta^{t_i\langle p\rangle} = \{\zeta^{t_i}, \zeta^{t_i p}, \ldots, \zeta^{t_i p^{d-1}}\}$ $(i = 0, \ldots, g-1)$. An element $z = \sum_{i=1}^{\mathsf{m}-1} z_i \zeta^i \in R_Z$ that is stable under the action of G_Z must have constant integer coefficients over the each orbits $\zeta^{t_i\langle p\rangle}$. Hence, z is a $\mathbb{Z}$-linear combination of $\{\eta_1, \ldots, \eta_g\}$ □

(3) Proof of Lemma 5

For $0 \le i, j < g$,

$$\begin{aligned}\overline{\eta}_i\eta_j &= \Big(\sum_{a\in\langle p\rangle} \zeta^{-at_i}\Big)\Big(\sum_{b\in\langle p\rangle} \zeta^{bt_j}\Big) = \sum_{a,b\in\langle p\rangle} \zeta^{-at_i+bt_j} \\ &= \sum_{a\in\langle p\rangle}\sum_{b\in\langle p\rangle} \rho_a(\zeta^{-t_i+ba^{-1}t_j}) \\ &= \sum_{a\in\langle p\rangle}\sum_{b\in\langle p\rangle} \rho_a(\zeta^{-t_i+bt_j}) \\ &= \sum_{b\in\langle p\rangle} \mathrm{Tr}_{K|Z}(\zeta^{-t_i+bt_j}).\end{aligned}$$

Here, Suppose $i \neq j$. Then, $-t_i + bt_j \not\equiv 0 \pmod{\mathsf{m}}$ for any $b \in \langle p\rangle$. Hence, by Lemma 1,

$$\mathrm{Tr}_{Z|\mathbb{Q}}(\overline{\eta}_i\eta_j) = \sum_{b\in\langle p\rangle} \mathrm{Tr}_{K|\mathbb{Q}}(\zeta^{-t_i+bt_j}) = |\langle p\rangle| \cdot (-1) = -d.$$

If $i = j$, since $\mathrm{Tr}_{K|\mathbb{Q}}(\zeta^{-t_i+bt_i}) = \mathsf{m} - 1$ only if $b = 1$ and -1 otherwise by Lemma 1,

$$\begin{aligned}\mathrm{Tr}_{Z|\mathbb{Q}}(\overline{\eta}_i\eta_i) &= \sum_{b\in\langle p\rangle} \mathrm{Tr}_{K|\mathbb{Q}}(\zeta^{-t_i+bt_i}) \\ &= \mathsf{m} - 1 + (\mathsf{d} - 1)\cdot(-1) \qquad = m - d\end{aligned}$$

□

(4) Proof of Corollary 1

For any i, by Lemma 3 and 5 we have

$$\mathrm{Tr}_{Z|\mathbb{Q}}\Big(\frac{\eta_i - d}{\mathsf{m}} \cdot \overline{\eta_i}\Big) = \frac{1}{\mathsf{m}}(\mathsf{m} - \mathsf{d}) - \frac{\mathsf{d}}{\mathsf{m}} \cdot (-1) = 1.$$

Similarly, for any $i \neq j$ we have

$$\mathrm{Tr}_{Z|\mathbb{Q}}\Big(\frac{\eta_i - d}{\mathsf{m}} \cdot \overline{\eta_j}\Big) = \frac{-d}{\mathsf{m}} - \frac{d}{\mathsf{m}} \cdot (-1) = 0$$

□

(5) Proof of Lemma 6

$$\begin{aligned}\boldsymbol{a} &= \Gamma_Z \boldsymbol{b} = \Big(\rho_{t_i}(\frac{\overline{\eta}_j - d}{\mathsf{m}})\Big)_{ij}\boldsymbol{b} \\ &= \Big(\frac{1}{\mathsf{m}}\sum_j \rho_{t_i}(\overline{\eta}_j - d)b_j\Big)_i \\ &= \frac{1}{\mathsf{m}}\Big(\sum_j \rho_{t_i}(\overline{\eta}_j)b_j - d\sum_j b_j\Big)_i \\ &= \frac{1}{\mathsf{m}}\Big(\overline{\Omega}_Z\boldsymbol{b} - d(\sum_j b_j)\cdot\mathbf{1}\Big)\end{aligned}$$

□

(6) Proof of Lemma 7

The first claim is the definition of $\boldsymbol{\xi}$.

Since $\Omega_Z = \Big(\sigma_Z(\eta_j)\Big)_{0\le j<g}$, $a = \boldsymbol{\eta}^T\cdot\boldsymbol{a}$ if and only if $\sigma_Z(a) = \Omega_Z\boldsymbol{a}$.

Next,

$$\begin{aligned}a = \boldsymbol{\xi}^T\cdot\boldsymbol{b} &\Leftrightarrow a \equiv \boldsymbol{\eta}^T(\Omega_Z^{(q)})^{-1}\cdot\boldsymbol{b} \pmod{\mathfrak{q}} \\ &\Leftrightarrow \sigma_Z(a) \equiv \Omega_Z(\Omega_Z^{(q)})^{-1}\cdot\boldsymbol{b} \equiv \boldsymbol{b} \pmod{\mathfrak{q}}\end{aligned}$$

□

(7) Proof of Lemma 8

$\sigma_Z(a_1a_2) = \sigma_Z(a_1)\odot\sigma_Z(a_2) = \boldsymbol{b}_1\odot\boldsymbol{b}_2$ □

(8) Proof of Lemma 9

The ideal qR_Z factors in R_Z as

$$qR_Z = \mathfrak{q}_0\mathfrak{q}_1\cdots\mathfrak{q}_{g-1}$$

where $\mathfrak{q}_i = \mathfrak{Q}_i \cap R_Z$ for any i.

Let $\{\tau'_i\}_{i=0}^{g-1}$ be a resolution of unity in R_Z mod q. Here, we take the coefficients of each τ'_i from $[-q/2, q/2)$ over the η-basis $\{\eta_0,\ldots,\eta_{g-1}\}$ of R_Z.

Then,

$$\tau'_i \equiv \begin{cases} 1 \pmod{\mathfrak{q}_i} & (i = 0,\ldots,g-1) \\ 0 \pmod{\mathfrak{q}_j} & (j \ne i).\end{cases}$$

Since $\mathfrak{q}_i \subset \mathfrak{Q}_i$ for any i, $\{\tau'_i\}_{i=0}^{g-1}$ is also a resolution of unity in R mod q. Since the coefficients of each τ'_i over the η-basis are in $[-q/2, q/2)$, by definition of $\eta_i = \sum_{a\in\langle p\rangle}\zeta^{t_i a}$, their coefficients over the basis B' are trivially also in $[-q/2, q/2)$. Hence, by the uniqueness of resolution, it must be that $\tau'_i = \tau_i$ for all i □

Appendix B: Norms on the Decomposition Ring

Let $Z = Q(R_Z)$ be the quotient field of the decomposition ring R_Z. Norms of $a \in Z$ are defined by

$$\|a\|_2 \stackrel{\text{def}}{=} \|\sigma_Z(a)\|_2,\ \|a\|_\infty \stackrel{\text{def}}{=} \|\sigma_Z(a)\|_\infty.$$

Lemma 10: For any $a, b \in Z$, we have

$$\|ab\|_\infty \le \|a\|_\infty\cdot\|b\|_\infty.$$

Proof: $\|ab\|_\infty = \|\sigma_Z(ab)\|_\infty = \|\sigma_Z(a)\odot\sigma_Z(b)\|_\infty \le \|\sigma_Z(a)\|_\infty\cdot\|\sigma_Z(b)\|_\infty = \|a\|_\infty\cdot\|b\|_\infty$. □

In the following, $\boldsymbol{a}$ means the η-vector of given $a = \boldsymbol{\eta}^T\cdot\boldsymbol{a} \in R_Z$.

Lemma 11: (1) For any $a \in Z$, $\|a\|_2 \le \sqrt{\mathsf{m}}\|\boldsymbol{a}\|_2$.

(2) For any $\boldsymbol{a} \in \mathbb{R}^g$, $\|\boldsymbol{a}\|_2 \le \|a\|_2$.

(3) If $\boldsymbol{a} \in \mathbb{R}^g$ is far from being proportional to vector $\mathbf{1}$ (*far from constants* in short), we have $\|\boldsymbol{a}\|_2 \approx \frac{1}{\sqrt{\mathsf{m}}}\|a\|_2$.

Proof: (1) By Lemma 7, $\sigma_Z(a) = \Omega_Z\boldsymbol{a}$ and by Lemma 5

$$\Omega_Z^*\Omega_Z = \mathsf{m}\mathsf{I}_\mathsf{g} - \mathsf{d}\mathbf{1}\cdot\mathbf{1}^\mathsf{T}.$$

The right-hand side matrix has eigenvalues $g-1$ times of m and 1 with corresponding eigenvectors $(1,-1,0,\cdots,0)$, $(1,0,-1,0,\cdots,0)$, …, $(1,0,\cdots,0,-1)$, $(1,1,\cdots,1)$. So, the symmetric matrix $\Omega_Z^*\Omega_Z$ can be diagonalized to $\mathrm{Diag}(\mathsf{m},\cdots,\mathsf{m},1)$ by an orthogonal transformation, and we have $s_1(\Omega_Z) = \sqrt{\mathsf{m}}$. This means $\|a\|_2 \le \sqrt{\mathsf{m}}\|\boldsymbol{a}\|_2$.

(2), (3) Conversely, $\boldsymbol{a} = (\Omega_Z)^{-1}\sigma_Z(a) = \Gamma_Z\sigma_Z(a)$. Similarly as above, the matrix $\Gamma_Z^*\Gamma_Z$ can be diagonalized to $\mathrm{Diag}(1/\mathsf{m},\cdots,1/\mathsf{m},1)$ by the orthogonal transformation. Hence, $s_1(\Gamma_Z) = 1$ and $\|\boldsymbol{a}\|_2 \le \|a\|_2$. Since almost all of the eigenvalues of $\Gamma_Z^*\Gamma_Z$ are $1/m$, except 1 for eigenvector $(1,1,\cdots,1)$, if $\boldsymbol{a}$ is far from being proportional to the eigenvector $(1,1,\cdots,1)$, $\|\boldsymbol{a}\|_2 \approx \frac{1}{\sqrt{\mathsf{m}}}\|a\|_2$ □

Lemma 12: (1) For any $a \in Z$, $\|a\|_\infty \le \sqrt{\mathsf{mg}}\|\boldsymbol{a}\|_\infty$.

(2) For any $\boldsymbol{a} \in \mathbb{R}^g$, $\|\boldsymbol{a}\|_\infty \le \sqrt{g}\|a\|_\infty$.

(3) If a is far from constants, we have $\|\boldsymbol{a}\|_\infty \lessapprox \sqrt{g/\mathsf{m}}\|a\|_\infty$.

Proof: (1) By Lemma 11-(1), $\|a\|_\infty \le \|a\|_2 \le \sqrt{\mathsf{m}}\|\boldsymbol{a}\|_2 \le \sqrt{\mathsf{mg}}\|\boldsymbol{a}\|_\infty$.
(2) By Lemma 11-(2), $\|\boldsymbol{a}\|_\infty \le \|\boldsymbol{a}\|_2 \le \|a\|_2 \le \sqrt{g}\|a\|_\infty$.
(3) By Lemma 11-(3), $\|\boldsymbol{a}\|_\infty \le \|\boldsymbol{a}\|_2 \approx \frac{1}{\sqrt{\mathsf{m}}}\|a\|_2 \le \sqrt{g/\mathsf{m}}\|a\|_\infty$. □

Subgaussian elements

We call a random variable $a \in Z$ *subgaussian with parameter* s if corresponding random variable $\sigma_Z(a)$ on H_Z is subgaussian with parameter s.

Lemma 13 (Claim 2.1, Claim 2.4 [21]): Let a_i be independent subgaussian random variables over Z with parameter s_i $(i = 1, 2)$. Then,

1. The sum $a_1 + a_2$ is subgaussian with parameter $\sqrt{s_1^2 + s_2^2}$.
2. For any a_2 fixed, the product $a_1\cdot a_2$ is subgaussian with parameter $\|a_2\|_\infty s_1$.

Lemma 14: Let $\boldsymbol{a}$ be a subgaussian random variable over $\mathbb{R}^g$ of parameter s. Then, $a = \boldsymbol{\eta}^T \cdot \boldsymbol{a}$ is subgaussian over Z of parameter $\sqrt{\mathsf{m}}s$.

Proof : By Lemma 7 $\sigma_Z(a) = \Omega_Z \boldsymbol{a}$. As seen in the proof of Lemma 11, $s_1(\Omega_Z) = \sqrt{\mathsf{m}}$. Hence, $\sigma_Z(a)$ is subgaussian of parameter $\sqrt{\mathsf{m}}s$ □

Appendix C: Correctness of Our Decomposition Ring Homomorphic Encryption Scheme

We evaluate sizes of noises using their canonical embedding norms.

C.1 FV-Type Scheme DR-FV

Definition 4 (The noise term in FV-type scheme): Let $(a, b) \in R_Z^2$ be a ciphertext pair designed for a message $m \in R_Z$ under a secret key $s \in R_Z$. When given $((a, b), s, m)$, the smallest noise term $e \in R_Z$ satisfying

$$a + bs = \Delta m + e + kq$$

for some $k \in R_Z$ called the inherent noise term of (a, b) for a message m.

(1) Noise bound for correctness

Set $y = a + bs = \Delta m + e \pmod q$. Then,

$$\frac{y}{\Delta} = \frac{t}{q}(\frac{q}{t}m + e) = m + \frac{t}{q}e \pmod q.$$

If $\left\|\frac{t}{q}e\right\| < \frac{1}{2}$ then decryption works correctly. By 12-(3), to satisfy this inequation,

$$\sqrt{\frac{g}{\mathsf{m}}}\frac{t}{q}\left\|e\right\|_\infty < \frac{1}{2}$$

is required. We define $\mathsf{B}^{\mathsf{FV}}_{\mathsf{correct}} \stackrel{\text{def}}{=} \frac{q}{2t}\sqrt{\frac{\mathsf{m}}{g}}$. If the noise term e in a given ciphertext satisfies $\left\|e\right\|_\infty < \mathsf{B}^{\mathsf{FV}}_{\mathsf{correct}}$ then the ciphertext can be decrypted correctly.

(2) Estimate of $\mathsf{B}^{\mathsf{FV}}_{\mathsf{clean}}$

Let e be a noise sampled in $\mathsf{Encrypt}^{\mathsf{DR-FV}}$. Then

$$\mathsf{B}^{\mathsf{FV}}_{\mathsf{clean}} = \left\|e\right\|_\infty = \sigma\sqrt{gd}.$$

(3) Estimate of $\mathsf{B}^{\mathsf{FV}}_{\mathsf{direct\_mult}}$

Let (a', b') be the resulting ciphertext of $\mathsf{Mult}^{\mathsf{DR-FV}}(\mathsf{swk}, (a_1, b_1, \nu_1), (a_2, b_2, \nu_2))$, where swk is $((A_j, B_j)_{j=0}^{l_w-1}, \nu = \mathsf{B}^{\mathsf{FV}}_{\mathsf{lin}}) = \mathsf{SwitchKeyGen}^{\mathsf{DR-FV}}(s^2, \mathsf{sk} = s)$ satisfies $\Sigma_{j=0}^{l_w-1}(A_j + B_j s) = \Sigma_{j=0}^{l_w-1}(w^j s^2 + e_j) \bmod q$. Let α, β, γ be as in $\mathsf{Mult}^{\mathsf{DR-FV}}$ and $\epsilon_\alpha, \epsilon_\beta, \epsilon_\gamma$ be their rounding noises in $(-\frac{1}{2}, \frac{1}{2}]$. Set $(d_0, \cdots, d_{l_w-1}) = \mathsf{WD}(\gamma)$. Then,

$$\begin{aligned}
\alpha + \beta s + \gamma s^2 &= \frac{1}{\Delta}(a_1 a_2 \bmod q^2/t) + \epsilon_\alpha \\
&\quad + (\frac{1}{\Delta}(a_1 b_2 + a_2 b_1 \bmod q^2/t) + \epsilon_\beta)s \\
&\quad + (\frac{1}{\Delta}(b_1 b_2 \bmod q^2/t) + \epsilon_\gamma)s^2 \\
&= \frac{1}{\Delta}(a_1 + b_1 s)(a_2 + b_2 s) + (\epsilon_\alpha + \epsilon_\beta s + \epsilon_\gamma s^2) \\
&= \frac{1}{\Delta}(\Delta m_1 + e_1)(\Delta m_2 + e_2) + (\epsilon_\alpha + \epsilon_\beta s + \epsilon_\gamma s^2) \\
&= \Delta m_1 m_2 + (m_1 e_2 + m_2 e_1) + e_1 e_2/\Delta \\
&\quad + (\epsilon_\alpha + \epsilon_\beta s + \epsilon_\gamma s^2)
\end{aligned}$$

Setting $e' = (m_1 e_2 + m_2 e_1) + e_1 e_2/\Delta + (\epsilon_\alpha + \epsilon_\beta s + \epsilon_\gamma s^2)$, we have $\mathsf{B}^{\mathsf{FV}}_{\mathsf{direct\_mult}}(\nu_1, \nu_2) = t\sqrt{3gd}(\nu_1 + \nu_2) + \frac{t}{q}\nu_1\nu_2 + \sqrt{3gd} + 8d\sqrt{\frac{gh}{3}} + 20hd\sqrt{\frac{gd}{3}}$, since

$$\begin{aligned}
\left\|e'\right\|_\infty &\le (\left\|m_1\right\|_\infty \nu_2 + \left\|m_2\right\|_\infty \nu_1) + \nu_1\nu_2/\Delta \\
&\quad + (\left\|\epsilon_\alpha\right\|_\infty + \left\|\epsilon_\beta s\right\|_\infty + \left\|\epsilon_\gamma s^2\right\|_\infty) \\
&\le t\sqrt{3gd}(\nu_1 + \nu_2) + \frac{t}{q}\nu_1\nu_2 \\
&\quad + \sqrt{3gd} + 8d\sqrt{\frac{gh}{3}} + 20hd\sqrt{\frac{gd}{3}}.
\end{aligned}$$

(4) Estimate of $\mathsf{B}^{\mathsf{FV}}_{\mathsf{lin}}$

Suppose an input ciphertext (α, β, γ) of $\mathsf{Linearize}^{\mathsf{DR-FV}}$ satisfies $\alpha + \beta s + \gamma s^2 = \Delta m + e \bmod q$ and its noise bound is ν. Let (a', b') be the output ciphertext. Generate a switching key as $\mathsf{swk} = ((A_j, B_j)_{j=0}^{l_w-1}, \nu_{swk}) = \mathsf{SwitchKeyGen}^{\mathsf{DR-FV}}(s^2, \mathsf{sk} = s)$, which satisfies $\Sigma_{j=0}^{l_w-1}(A_j + B_j s) = \Sigma_{j=0}^{l_w-1}(w^j s^2 + e_j) \bmod q$. Let $(d_0, \cdots, d_{l_w-1}) = \mathsf{WD}(\gamma)$. Then,

$$\begin{aligned}
a' + b's &= (\alpha + \Sigma_{j=0}^{l_w-1} A_j d_j) + (\beta + \Sigma_{j=0}^{l_w-1} B_j d_j)s \bmod q \\
&= \alpha + \beta s + \Sigma_{j=0}^{l_w-1}(A_j + B_j s)d_j \bmod q \\
&= \alpha + \beta s + \Sigma_{j=0}^{l_w-1}(w^j s^2 + e_j)d_j \bmod q \\
&= \alpha + \beta s + \gamma s^2 + \Sigma_{j=0}^{l_w-1} e_j d_j \bmod q
\end{aligned}$$

So, it satisfies that

$$\left\|a' + b's\right\|_\infty \le \nu + \left\|\Sigma_{j=0}^{l_w-1} e_j d_j\right\|_\infty.$$

Therefore,
$\mathsf{B}^{\mathsf{FV}}_{\mathsf{lin}} = \left\|\Sigma_{j=0}^{l_w-1} e_j d_j\right\|_\infty = 16 l_w \sigma\sqrt{gd} w\sqrt{gd/12} = \frac{8}{\sqrt{3}} l_w \sigma g d w.$

(5) Noise Bound for $\mathsf{Mult}^{\mathsf{DR-FV}}$

The noise size of the output ciphertext of $\mathsf{Mult}^{\mathsf{DR-FV}}$ is

$$\begin{aligned}
\nu^{DR-FV}_{mult} &= \mathsf{B}^{\mathsf{FV}}_{\mathsf{direct_mult}} + \mathsf{B}^{\mathsf{FV}}_{\mathsf{lin}} \\
&= t\sqrt{3gd}(\nu_1 + \nu_2) + \frac{t}{q}\nu_1\nu_2 \\
&\quad + \sqrt{3gd} + 8d\sqrt{\frac{gh}{3}} + 20hd\sqrt{\frac{gd}{3}}
\end{aligned}$$

$$+ \frac{8}{\sqrt{3}} l_w \sigma g d w.$$

(6) Proof of Theorem 2 for DR-FV

By Lemma 4 of [4], we can implement $\mathsf{Decrypt}^{\mathsf{DR-BGV}}$ algorithm by some circuit of level $L_{dec} = O(\log \lambda)$.

Let ct' be the ciphertext after L_{dec} times multiplications and ν' be the bound of the canonical embedding noise of ct'. Then if $\nu' \le \mathsf{B}^{\mathsf{FV}}_{\mathsf{correct}}$, then the scheme can homomorphically evaluate its own $\mathsf{Decrypt}^{\mathsf{DR-FV}}$ circuit and will be fully homomorphic under circular security assumption.

Now we show $\nu' \le \mathsf{B}^{\mathsf{FV}}_{\mathsf{correct}}$ as follows: The size of each parameter is $g = O(\lambda), h = O(1), t = O(1)$, and we suppose $d = O(\log \lambda) = \tilde{O}(1)$. Two fresh ciphertexts have noises of size $\nu_1 = O(\sqrt{\lambda}), \nu_2 = O(\sqrt{\lambda})$. Then, by repeating the multiplication, the noise bound of the resulting ciphertext becomes as

$$\begin{aligned}\nu_{mult}^{DR-FV} &= \tilde{O}(\sqrt{\lambda})(\nu_1 + \nu_2) + \frac{t}{q}\nu_1\nu_2 \\ &\quad + \tilde{O}(\sqrt{\lambda}) + \tilde{O}(\sqrt{\lambda}) + \tilde{O}(\sqrt{\lambda}) + \tilde{O}(\lambda)\end{aligned}$$

This shows that the increase ratio of ν_{mult}^{DR-FV} by one multiplication is $\tilde{O}(\sqrt{\lambda})$. It is because, the second term is $\frac{t}{q}\nu_1\nu_2 \le \frac{t}{q}\mathsf{B}^{\mathsf{FV}}_{\mathsf{correct}}\nu_i = \frac{1}{2}\sqrt{\frac{\mathsf{m}}{g}}\nu_i$ ($i = 1$ or 2), so the increase ratio of second term is $\tilde{O}(1)$ and it is smaller than that of the first term $\tilde{O}(\sqrt{\lambda})$.

Thus ν_{mult}^{DR-FV} increases by the factor of $\tilde{O}(\sqrt{\lambda})$ for each multiplication, and the factor is of $\log_2(\sqrt{\lambda}) = O(\log \lambda)$ bits. After L_{dec} times multiplication, the noise bound ν' is $O(\log(\lambda^{\log \lambda}))$ bit.

On the other hand, taking $q = O(\lambda^{\log \lambda})$ as assumption of the Theorem 2, $\mathsf{B}^{\mathsf{FV}}_{\mathsf{correct}} = \frac{q}{2t}\sqrt{\frac{\mathsf{m}}{g}} = O(\lambda^{\log \lambda})$ and it is $O(\log(\lambda^{\log \lambda}))$ bit. Therefore, we can take the modulus to satisfy $\nu' \le \mathsf{B}^{\mathsf{FV}}_{\mathsf{correct}}$. □

C.2 BGV-Type Scheme DR-BGV

Definition 5 (The noise term in BGV-type scheme): Let $(a, b) \in R_Z^2$ be a ciphertext pair designed for a message $m \in R_Z$ under a secret key $s \in R_Z$. When given $((a, b), s, m, l)$ where l is level, a noise $e \in R_Z$ is uniquely determined by the equation

$$a + bs = m + te + kq_l$$

for some $k \in R_Z$. Note that $m + te$ is not necessarily lower than q_l. We define the value $m + te$ as the noise term of $((a, b), s, m, l)$.

(1) Noise bound for correctness

Let $y = m + te$. If $\|y\| < \frac{q_l}{2}$ then decryption works correctly. By 12-(3), to satisfy this inequation,

$$\sqrt{\frac{g}{\mathsf{m}}}\|y\|_\infty < \frac{q_l}{2}$$

is required. We define $\mathsf{B}^{\mathsf{BGV}}_{\mathsf{correct}} \stackrel{\text{def}}{=} \frac{q_l}{2}\sqrt{\frac{\mathsf{m}}{g}}$. If the inherent noise y in a given level-l ciphertext satisfies $\|y\|_\infty < \mathsf{B}^{\mathsf{BGV}}_{\mathsf{correct}}$ then the ciphertext can be decrypted correctly.

(2) Estimate of $\mathsf{B}^{\mathsf{BGV}}_{\mathsf{clean}}$

For a fresh ciphertext, the upper bound of its inherent noise is $\mathsf{B}^{\mathsf{BGV}}_{\mathsf{clean}} = t\sqrt{gd}(\sqrt{3} + 6\sigma)$ since

$$\begin{aligned}\|m + te\|_\infty &\le \|m\|_\infty + t\|e\|_\infty \\ &\le t\sqrt{3gd} + t6\sigma\sqrt{gd} = t\sqrt{gd}(\sqrt{3} + 6\sigma).\end{aligned}$$

(3) Estimate of $\mathsf{B}^{\mathsf{BGV}}_{\mathsf{scale}}$

Let a scaled ciphertext from q_l to $q_{l'}$ be $(a', b', f, l', \nu') = \mathsf{Rescale}((a, b, f, l, \nu), l')$, and write its inherent noise as y. Set $P = \frac{q_l}{q_{l'}}$ and fix δ_a and δ_b s.t. $\delta_a \equiv a \pmod P$ and $\delta_a \equiv 0 \pmod t$, $\delta_b \equiv b \pmod P$ and $\delta_b \equiv 0 \pmod t$. Then, we have

$$\begin{aligned}\|a' + b's\|_\infty &\le \frac{1}{P}\|a + bs\|_\infty + \frac{1}{P}\|\delta_a + \delta_b s\|_\infty \\ &\le \frac{\nu}{P} + \frac{1}{P}(6\sqrt{P^2 gd/12} + 16\sqrt{P^2 gd/12}\sqrt{hd}) \\ &\le \frac{\nu}{P} + t(\sqrt{3gd} + 8d\sqrt{gh/3}).\end{aligned}$$

Thus, $\mathsf{B}^{\mathsf{BGV}}_{\mathsf{scale}} = t(\sqrt{3gd} + 8d\sqrt{gh/3})$.

(4) Estimate of $\mathsf{B}^{\mathsf{BGV}}_{\mathsf{direct\_mult}}$

Let (a_1, b_1, f_1, ν_1) and (a_2, b_2, f_2, ν_2) be input ciphertexts of $\mathsf{Mult}^{\mathsf{DR-BGV}}$ and compute α, β, γ according to $\mathsf{Mult}^{\mathsf{DR-BGV}}$. Then,

$$\begin{aligned}\alpha + \beta s + \gamma s^2 &= a_1 a_2 + (a_1 b_2 + a_2 b_1)s + b_1 b_2 s^2 \bmod q_l \\ &= (a_1 + b_1 s)(a_2 + b_2 s) \bmod q_l\end{aligned}$$

This means that $\mathsf{B}_{\mathsf{direct\_mult}}(\nu_1, \nu_2) = \nu_1\nu_2$.

(5) Estimate of $\mathsf{B}^{\mathsf{BGV}}_{\mathsf{lin}}$

Suppose an input ciphertext (α, β, γ) of $\mathsf{Linearize}^{\mathsf{DR-BGV}}$ satisfies $\alpha + \beta s + \gamma s^2 \equiv m + te \pmod{q_l}$ and its noise is bound by ν. Let (a', b') be the output ciphertext of $\mathsf{Linearize}^{\mathsf{DR-BGV}}$ with input (α, β, γ). Generate a switching key $\mathsf{swk} = ((A_j, B_j)_{j=0}^{l_w-1}, \nu_{swk}) = \mathsf{SwitchKeyGen}^{\mathsf{DR-BGV}}(s^2, \mathsf{sk} = s)$, which satisfies $\Sigma_{j=0}^{l_w-1}(A_j + B_j s) = \Sigma_{j=0}^{l_w-1}(q_s w^j s^2 + te_j) \bmod q_{L-1}q_s$. Let $(d_0, \cdots, d_{l_w-1}) = \mathsf{WD}(\gamma)$ and $A = \Sigma_{j=0}^{l_w-1} A_j d_j \bmod q_l q_s$, $B = \Sigma_{j=0}^{l_w-1}$ and $B_j d_j \bmod q_l q_s$. Then,

$$\begin{aligned}A + Bs &= \Sigma_{j=0}^{l_w-1}(A_j + B_j s)d_j \bmod q_l q_s \\ &= \Sigma_{j=0}^{l_w-1}(q_s w^j s^2 + te_j)d_j \bmod q_l q_s \\ &= q_s\gamma s^2 + t\Sigma_{j=0}^{l_w-1} e_j d_j \bmod q_l q_s\end{aligned}$$

After scaling from $q_l q_s$ to q_l, i.e. $(a, b, f, l, \nu') = \mathsf{Rescale}((A, B, f, l + s, \nu), l)$, it satisfies that

$$a + bs = \gamma s^2 + \frac{t}{q_s}\Sigma_{j=0}^{l_w-1} e_j d_j + e_k \bmod q_l.$$

where e_k is a rounding noise added by Rescale satisfying $\|e_k\|_\infty < \mathsf{B}_{\mathsf{scale}}^{\mathsf{BGV}}$. Now we see that

$$\begin{aligned} a' + b's &= \alpha + a + (\beta + b)s \bmod q_l \\ &= \alpha + \beta s + a + bs \bmod q_l \\ &= \alpha + \beta s + \gamma s^2 + \frac{t}{q_s}\Sigma_{j=0}^{l_w-1} e_j d_j + e_k \bmod q_l \end{aligned}$$

Then,

$$\|a' + b's\|_\infty \le \nu + \frac{t}{q_s}\|\Sigma_{j=0}^{l_w-1} e_j d_j\|_\infty + \|e_k\|_\infty,$$

where

$$\begin{aligned} \frac{t}{q_s}\|\Sigma_{j=0}^{l_w-1} e_j d_j\|_\infty &\le \frac{t}{q_s} 16 l_w \sigma \sqrt{gd} w \sqrt{gd/12} \\ &\le 8 t l_w \sigma g d w / \sqrt{3} q_s. \end{aligned}$$

Thus, we have $\mathsf{B}_{\mathsf{lin}} = 8 t l_w \sigma g d w / \sqrt{3} q_s$.

(6) Noise Bound for $\mathsf{Mult}^{\mathsf{DR-BGV}}$

In $\mathsf{Mult}^{\mathsf{DR-BGV}}$, the noise bound of output ciphertext of $\mathsf{Linearize}^{\mathsf{DR-BGV}}$ is $\nu'' = \nu_1 \nu_2' + \mathsf{B}_{\mathsf{lin}}^{\mathsf{BGV}} + \mathsf{B}_{\mathsf{scale}}^{\mathsf{BGV}}$, where ν_2' is a noise bound of ct_2 rescaled from q_{l_2} to q_{l_1}. After linearization, the noise is reduced by $\mathsf{Rescale}((a', b', f, l_1, \nu''), l_1 - 1)$. Thus, the noise bound of one multiplication is as follows:

$$\begin{aligned} \nu_{mult}^{DR-BGV} &= (\nu_1 (\frac{q_{l_1}}{q_{l_2}} \nu_2 + t(\sqrt{3gd} + 8d\sqrt{gh/3})) \\ &\quad + \frac{8 t l_w \sigma g d w}{\sqrt{3} q_s} + t(\sqrt{3gd} + 8d\sqrt{gh/3})) \frac{q_{l_1 - 1}}{q_{l_1}} \\ &\quad + t(\sqrt{3gd} + 8d\sqrt{gh/3}). \end{aligned}$$

(7) Proof of Theorem 2 for DR-BGV

By Lemma 4 of [4], we can implement $\mathsf{Decrypt}^{\mathsf{DR-BGV}}$ algorithm by some circuit of level $L_{dec} = O(\log \lambda)$.

Let ct' be the ciphertext after L_{dec} times multiplications, L be the maximum level in the system parameter, $l' = L - L_{dec}$ be the level of ct' and ν' be the bound of the canonical embedding noise of ct'. Then if $\nu' \le \mathsf{B}_{\mathsf{correct}}^{\mathsf{BGV}}(l')$, then the scheme can homomorphically evaluate its own $\mathsf{Decrypt}^{\mathsf{DR-BGV}}$ circuit and will be fully homomorphic under circular security assumption.

Now we show $\nu' \le \mathsf{B}_{\mathsf{correct}}^{\mathsf{BGV}}(l')$ as follows:

The size of each parameter is $g = O(\lambda), h = O(1), t = O(1)$, and we suppose $d = O(\log \lambda) = \tilde{O}(1)$, then the size of noise of a fresh ciphertext is $3.2\sqrt{gd} = \tilde{O}(\sqrt{\lambda})$.

From assumption of Theorem 2, $p_i = \Omega(\sqrt{\lambda})$ (i.e. $\frac{q_{l-1}}{q_l} = \frac{1}{\Omega(\sqrt{\lambda})}$) and $q_s = \Omega(\sqrt{\lambda})$. Two fresh ciphertexts have noises of size $\nu_1 = \tilde{O}(\sqrt{\lambda}), \nu_2 = \tilde{O}(\sqrt{\lambda})$. Then, by repeating the multiplication, the noise bound of the resulting ciphertext becomes as

$$\begin{aligned} \nu_{mult}^{DR-BGV} &= (\tilde{O}(\sqrt{\lambda})(\frac{q_{l_1}}{q_{l_2}} \tilde{O}(\sqrt{\lambda}) + \tilde{O}(\sqrt{\lambda})) + \frac{\tilde{O}(\lambda)}{\Omega(\sqrt{\lambda})} \\ &\quad + \tilde{O}(\sqrt{\lambda})) \frac{1}{\Omega(\sqrt{\lambda})} + \tilde{O}(\sqrt{\lambda}) \\ &= \tilde{O}(\sqrt{\lambda}). \end{aligned}$$

This shows that after multiple multiplications the noise bound of result ciphertext always keeps $\tilde{O}(\sqrt{\lambda})$.

We denote the bound for correctness of level-l ciphertext $\mathsf{B}_{\mathsf{correct}}^{\mathsf{BGV}}(l)$ $(= \frac{q_l}{2}\sqrt{\frac{\mathfrak{m}}{g}})$. Since $\sqrt{\frac{\mathfrak{m}}{g}} = \tilde{O}(1)$ and $q_0 = O(\sqrt{\lambda})$, $\mathsf{B}_{\mathsf{correct}}^{\mathsf{BGV}}(0) = \tilde{O}(\sqrt{\lambda})$. Thus, $\nu_{mult}^{DR-BGV} = \tilde{O}(\sqrt{\lambda}) < \tilde{O}(\sqrt{\lambda}) = \mathsf{B}_{\mathsf{correct}}^{\mathsf{BGV}}(0) \le \mathsf{B}_{\mathsf{correct}}^{\mathsf{BGV}}(l)$ for any level l. Therefore $\nu' \le \mathsf{B}_{\mathsf{correct}}^{\mathsf{BGV}}(l')$. □

Seiko Arita received his B.E. and M.E. from Kyoto University, and Ph.D. from Chuo University. He has been interested in prime numbers, algebraic curves and cryptographic protocols. He is with Institute of Information Security (IISEC), Japan.

Sari Handa received her B.E. from Meiji University and M.E. from Institute of Information Security. She is interested in algebraic number theory and lattice-based cryptography. She is with Institute of Information Security (IISEC), Japan.