Fully Homomorphic Encryption on the Ring of Gaussian Periods

Yimeng He[1] San Ling[1], Yimin Shi[1*], Benjamin Hong Meng Tan[2], Huaxiong Wang[1] and Allen Siwei Yang[1,2]

[1] School of Physical & Mathematical Sciences, Nanyang Technological University, Singapore
yimeng002@e.ntu.edu.sg lingsan@ntu.edu.sg, yimin005@e.ntu.edu.sg, hxwang@ntu.edu.sg, yang0788@e.ntu.edu.sg

[2] Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore
benjamin_tan@a-star.edu.sg

Abstract. In Geelen and Vercauteren (Eurocrypt 2025), a Generalized BFV (GBFV) fully homomorphic encryption scheme was proposed. Here, a plaintext space of form $\mathbb{Z}[x]/(\Phi(x), t(x))$ was utilized to reduce the number of Single Instruction Multiple Data (SIMD) slots within the initial BFV plaintext space. This lowered its dimension and thus enabled lower latencies as well as greater flexibility in parameter selection. However, to obtain slots of degree 1, the methods of Geelen and Vercauteren limit the choice of plaintext modulus to that of large primes, which can be unnecessary for various use cases.

To resolve this, we propose a generalized method to perform FHE based on a subring of the plaintext polynomial ring. We utilize the decomposition ring $\mathcal{O}_{\mathbf{K}}$, with which when taking quotient with a rational prime p, already factors into residual fields of dimension 1. From here, we develop methods to perform FHE on subrings of the decomposition ring $\mathcal{O}_{\mathbf{K}}$, which we refer to as the decomposition subring $\mathcal{O}_{\mathbf{M}}$. We introduce novel methods to enable both encoding and decoding maps within the decomposition subring $\mathcal{O}_{\mathbf{M}} \subset \mathcal{O}_{\mathbf{K}}$. By utilizing $\mathcal{O}_{\mathbf{M}}$, we further lower the dimension of the underlying ring, improving upon efficiency while retaining sufficient security. In experiments, we provide a proof-of-concept implementation, demonstrating up to a 5.06× improvement in the latency of operations for selected parameters. This approach offers enhanced flexibility in the selection of parameters for FHE with the subring dimension being any suitable divisor of r. This direction also represents the first generalization of the subring approach for FHE.

Keywords: Fully Homomorphic Encryption, Decomposition Rings.

*Corresponding author.

1 Introduction

Fully Homomorphic Encryption (FHE) constitutes a type of encryption that enables arbitrary computations on encrypted data [8]. Its robustness in preserving privacy lends itself to an extensive range of applications both inside and outside cryptography. A substantial portion of FHE-related research has been devoted to enhancing efficiency. This ranges from works that focus on accelerating the bootstrapping procedure [17, 28] to works that target more fundamental components such as homomorphic matrix multiplication operations [4]. Homomorphic encryption (HE) schemes typically consist of two types. The first includes schemes that support Single Instruction Multiple Data (SIMD) [25] operations on the encryption of multiple elements. Examples of this include BGV [8], BFV [7, 15] and CKKS [10]. The second prioritizes lower latencies with smaller parameters. Representatives for this type include FHEW [14] and TFHE [11, 12]. The focus of this work is on the first category of FHE schemes.

1.1 Contributions

Most relevant to our work is that of GBFV [16]. Here, they attempted to reduce the number of slots in the plaintext space to lower its dimension. This was applied to the BFV cryptosystem [7, 15]. To do so, they extended the work of [9] from a linear polynomial plaintext modulus to that of a binomial modulus and eventually to a general non-linear polynomial plaintext modulus. The work of [16] computes a specific quotient of the cyclotomic ring which then enables (SIMD) operations over a subset of the slots. Specifically, they obtained the quotient of the cyclotomic ring, by this polynomial modulus, and set this to be the plaintext space. This was shown to correspond to a subset of the slots in the initial BFV plaintext space under certain conditions. This enabled lower latencies as well as improved flexibility in the parameter selection. Specifically, new, lower latency parameters could be obtained where typical BFV parameters would first be selected followed by selecting a subset of the slots.

In most use cases, which require slots of degree 1, a large prime modulus plaintext space corresponding to a large number of slots of degree 1 would first have to be selected. From there, a specified quotient of the plaintext ring would be applied to reduce the dimension and obtain the desired number of slots as the subset. However, this limits the choice of plaintext modulus to that of large primes, which can be unnecessary for various use cases.

Motivated by the work of [16], we attempt to reduce both the degree of each slot as well as the slot count to expand the options available for parameter selection. We find that the decomposition ring [2] exhibits some special properties that support our idea. To obtain slots of degree 1, the work of [2] had attempted to first select parameters whose slots had degree larger than 1. From here, by moving down to the decomposition ring, this would then decrease the degree of each slot to 1 as desired. However, in the context of the decomposition ring, the ideas of [16] do not necessarily apply as the ring utilized in [2] will not be a polynomial quotient ring. Drawing upon the works of [2, 16], we therefore

instead propose a form of FHE on the ring of Gaussian periods. Here, we propose methods to go further down the decomposition ring $\mathcal{O}_{\mathbf{K}}$, yielding subrings of smaller dimension. These subrings are referred to as decomposition subrings $\mathcal{O}_{\mathbf{M}}$ and their structure is discussed in Section 2.2. However, the methods of [2] are not applicable in this context, as the CRT basis elements τ_i are no longer idempotents. To overcome this, we introduce novel methods for encoding and decoding. Here, we obtain an element F_i in the decomposition subring $\mathcal{O}_{\mathbf{M}}$ such that every factor of a given prime p is contained in (F_i) except for a specified prime ideal. We demonstrate that this construction of $F_i \in \mathcal{O}_{\mathbf{M}}$ enables us to find the desired encoding and decoding maps to work over the decomposition subring $\mathcal{O}_{\mathbf{M}}$. Our work could therefore be viewed as the first instance of a generalized subring FHE. This is detailed in Section 3.

Following this, the novel encoding and decoding maps introduced are utilized as the building blocks for the operations of an FHE scheme. A comparison between the operations performed over the decomposition subring $\mathcal{O}_{\mathbf{M}}$ and the decomposition ring $\mathcal{O}_{\mathbf{K}}$ is conducted. Let r denote the dimension of the decomposition ring $\mathcal{O}_{\mathbf{K}}$ and $n < r$ denote the dimension of the decomposition subring $\mathcal{O}_{\mathbf{M}}$. It is observed that the time complexity of the operations is decreased from, for encryption and decryption, $O(r \log r)$ to $O(n \log n)$, for addition, $O(r)$ to $O(n)$ and for multiplication, $O(l_w \cdot r \log r)$ to $O(l_w \cdot n \log n)$ in Section 4. Concretely, this implies that greater efficiency is obtained for the operations performed, lowering latency. We further verify this by providing a proof-of-concept implementation in Section 5. Here, the implementation demonstrates up to a $5.06\times$ improvement in the latency of operations for selected parameters. We emphasize that this latency improvement demonstrates that previously available parameters can now be instead utilized in a lower latency setting, while still retaining acceptable security levels. We further note that while the FHE scheme proposed is a BFV-variant, our techniques are not restricted to the BFV cryptosystem. The methods introduced could also be applied to BGV-styled cryptosystems, directly following from the work of [3].

Finally, the utilization of low-dimension decomposition subrings for FHE provides flexibility in the choice of parameters. Compared to [16], the choice of our prime number p could be small while matching the number of slots in the parameters of [16]. This optimizes for use-cases where a large dimension and a large plaintext integer modulus are not required. Our scheme therefore unlocks new sets of parameters that can be utilized for FHE, offering greater flexibility in the choice of parameters in practice. This technique could additionally be applied to optimize key FHE operations, such as blind rotations and bootstrapping [21]. For instance, for 2048 many slots at 128-bit security, the smallest plaintext integer modulus of the recommended parameters from [16] is a 16-bit prime 65537. On the other hand, with our techniques we achieve approximately this number of slots and security level with a plaintext integer modulus of 113. This is discussed in Table 6 of Section 5. This fine-grained selection of the plaintext modulus consequently also applies when compared with the typical power-of-two cyclotomic ring utilized in BFV. Unlike the case involving the power-of-two cyclotomic rings,

the smaller plaintext modulus from our techniques would be better suited for applications such as circuit-oriented equality comparisons [24, 26].

1.2 Related Work

Alongside the aforementioned works, there have been several others that focus on obtaining subring structures or on applications of the subring plaintext spaces. [18] explored the use of the conjugate-invariant subring, which was specific to the CKKS cryptosystem [10]. [3] extended the work of [2], providing both refinements to the original BFV-based cryptosystem [7,15] and an added BGV-based variant. [29] further explored conditions in which subfields of the cyclotomic field could be utilized for cryptosystems. Recently, [21] utilized the subring technique of [2] to efficiently perform slot-wise blind rotation in the context of bootstrapping.

2 Preliminaries

2.1 Notation

For a positive integer number m, $\Phi_m(x)$ refers to the m-th cyclotomic polynomial. Denote $\mathbb{Z}$ as the integer ring and $\mathbb{Z}_p := \mathbb{Z}/p\mathbb{Z}$. Denote $\mathcal{R}$ to be the polynomial ring $\mathbb{Z}[x]/f(x)$ where $f(x)$ is irreducible and $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$. By default, bold uppercase letters such as $\mathbf{A}$ will be interpreted as matrices. The symbol $||\cdot||_\infty$ refers to the L_∞ norm and $||\cdot||_2$ refers to the L_2 norm. Denote $\mathbb{Q}$ as the field of rational numbers and $\mathbf{K}$, $\mathbf{L}$, $\mathbf{M}$ etc. as its extension fields. The relative trace map $\mathbf{Tr}^{\mathbf{L}}_{\mathbf{K}}(\cdot)$ for any two number fields $\mathbf{K} \subset \mathbf{L}$ is defined to be the sum $\sum_{i=1}^{n} \sigma_i(\cdot)$, where each σ_i is one of the $n = [\mathbf{L} : \mathbf{K}]$ embeddings of $\mathbf{L}$ in $\mathbb{C}$ which fixes $\mathbf{K}$ pointwise. This takes as input an element of $\mathbf{L}$. Exact division of ideals is denoted by $\|$ and is defined as $P^k \parallel u \iff (P^{k+1}, u) = P^k$, for ideals P, u and some integer $k \geq 0$.

2.2 Gaussian Periods

Gaussian periods serve as the building blocks of and characterize a large class of number fields which possess a normal integral basis. Consequently, fields of Gaussian periods are well-suited for instantiating cryptographic protocols founded on the Ring-LWE assumption, offering a viable alternative to the traditional reliance on cyclotomic fields. Specifically, the Ring-LWE assumption on the decomposition subring $\mathcal{O}_{\mathbf{M}}$ is utilized to instantiate our FHE scheme, as opposed to being over $\mathbb{Z}[\zeta_m]$. Here, we describe how the decomposition subrings as well as their respective period-basis are obtained. More generally, we first define the decomposition field and its corresponding decomposition subfields. From the decomposition subfield, we then obtain the corresponding decomposition subring $\mathcal{O}_{\mathbf{M}}$ and period-basis.

Let $q > 2$ be an odd prime, $g > 0$ be a multiplicative generator of $\mathbb{Z}_q^\times$. Select positive integers $n, f > 0$ such that $q = nf + 1$. Let ζ_q denote a primitive q^{th} root of unity.

Definition 1 (Gaussian Periods). *The Gaussian periods of degree n and length f are algebraic integers $\eta_0, \ldots, \eta_{n-1}$ satisfying:*

$$\forall 0 \leq i < n : \ \eta_i \coloneqq \sum_{k=0}^{f-1} \zeta_q^{g^{i+nk}}$$

For ease of notation, we denote $\eta_i \coloneqq \eta_{i \bmod n}$ for $i \in \mathbb{Z}$.

2.2.1 Decomposition Rings. We first define the decomposition ring. Suppose $m > 0$ and let $\mathbf{L} = \mathbb{Q}(\zeta_m)$, $\mathcal{O}_{\mathbf{L}} = \mathbb{Z}[\zeta_m]$. Choose a prime $p \nmid m$, then p is unramified and the ideal $p\mathcal{O}_{\mathbf{L}}$ factorizes as:

$$p\mathcal{O}_{\mathbf{L}} = \prod_{i=1}^{r} \mathfrak{P}_i$$

In the factorization above, each prime ideal $\mathfrak{P}_i$ has inertial degree $d = \mathsf{ord}_m^{\times}(p)$, the multiplicative order of $p \in \mathbb{Z}_m^{\times}$. The number of prime ideals $r = \frac{\varphi(m)}{d}$ where $\varphi(\cdot)$ denotes the Euler totient function.

The *Decomposition Group* $\mathbf{H} = \langle \phi_p \rangle \subset \mathsf{Gal}\left(\mathbb{Q}(\zeta_m) \,/\, \mathbb{Q}\right)$ of p is the subgroup generated by the Frobenius map $\phi_p : \zeta_m \mapsto \zeta_m^p$. The subgroup $\mathbf{H}$ is the (unique) subgroup of order d in $\mathbb{Z}_q^{\times}$. Let $\mathbf{K} = \mathbf{L}^{\langle \phi_p \rangle}$ and $\mathcal{O}_{\mathbf{K}}$ be its ring of integers. A standard result in algebraic number theory tells us that the prime ideal $p\mathcal{O}_{\mathbf{K}}$ "splits completely" in $\mathcal{O}_{\mathbf{K}}$:

$$p\mathcal{O}_{\mathbf{K}} = \prod_{i=1}^{r} \mathfrak{p}_i$$

i.e. each $\mathfrak{p}_i$ is a prime ideal in $\mathcal{O}_{\mathbf{K}}$ with inertial degree 1. Equivalently,

$$\mathcal{O}_{\mathbf{K}}/p\mathcal{O}_{\mathbf{K}} \cong \prod_{i=1}^{r} \mathcal{O}_{\mathbf{K}}/\mathfrak{p}_i \cong \prod^{r} \mathbb{F}_p$$

where $\mathbb{F}_p$ is the finite field of cardinality p.

The ring $\mathcal{O}_{\mathbf{K}}$ is therefore referred to as the *Decomposition ring*. The splitting structure described above represents the single instruction, multiple data (SIMD) slot packing [25], with each $\mathcal{O}_{\mathbf{K}}/\mathfrak{p}_i$ representing a slot. In [2,3], the left and right ends of the isomorphism above have the basis elements η_i and ξ_i respectively. The conversion between the two bases is performed via multiplication by matrix $\Omega_Z^{(p)}$. Further information on the conversion can be found in [2, 3]. It is noted that the decomposition ring $\mathcal{O}_{\mathbf{K}}$ is not the only subring of $\mathbb{Z}[\zeta_m]$ in which the prime p splits completely, but is however the largest. This is detailed in Theorem 2.

Theorem 2 ([20]). *Under the notation above, let $\mathbf{M} \subset \mathbf{L}$ be any subfield of $\mathbf{L}$, with ring of integers $\mathcal{O}_{\mathbf{M}}$. The prime p splits completely in $\mathcal{O}_{\mathbf{M}}$ iff $\mathbf{M} \subset \mathbf{K}$ is a subfield of the Decomposition field $\mathbf{K}$.*

2.2.2 Decomposition Subrings. Following the notation of Theorem 2, we obtain the decomposition subring $\mathcal{O}_{\mathbf{M}}$ by first selecting a subfield $\mathbf{M} \subset \mathbf{K}$ and obtaining the corresponding ring of integers. Specifically, this is performed below in terms of the Gaussian periods, obtaining the decomposition subring $\mathcal{O}_{\mathbf{M}} = \mathbb{Z}[\eta_0, \ldots, \eta_{n-1}]$. The splitting behavior of the decomposition subring $\mathcal{O}_{\mathbf{M}}$ is also described below. Set $m = q$ to be an odd prime, then $\varphi(q) = q - 1$, $q - 1 = rd$, where $d = ord_q^{\times}(p)$. Let $k \,|\, r$ be any divisor of r and define $n = \frac{r}{k}$, $f = dk$. Since $q - 1 = nf$, the Gaussian periods of degree n and length f, $\eta_0, \ldots, \eta_{n-1}$ (cf. Definition 1), forms a subfield $\mathbb{Q}(\eta_0) \subset \mathbb{Q}(\zeta_q)$ whose Galois group $\mathsf{Gal}\left(\mathbb{Q}(\zeta_q) \,/\, \mathbb{Q}(\eta_0)\right) \rhd \mathbf{H}$ has order $f = dk$. By the Galois correspondence, $\mathbb{Q}(\eta_0) \subset \mathbf{K}$ is a subfield and by Theorem 2 the prime p also splits completely in $\mathbb{Z}[\eta_0, \ldots, \eta_{n-1}]$. This is described in Proposition 3.

Proposition 3. *Let $q > 2$ be an odd prime. Choose $n, f > 0$ such that $q = nf + 1$. Let $\eta_0, \ldots, \eta_{n-1}$ be the Gaussian periods of degree n and length f. Suppose p is a prime whose multiplicative order $\bmod q$ divides f, then the ideal $p\mathbb{Z}[\eta_0, \ldots, \eta_{n-1}]$ splits completely in $\mathbb{Z}[\eta_0, \ldots, \eta_{n-1}]$.*

Basis Elements. For a given decomposition subring $\mathcal{O}_{\mathbf{M}}$, the associated Gaussian period forms a basis of $\mathcal{O}_{\mathbf{M}}$. This is referred to as the period-basis. By considering both the decomposition rings and their subrings as subfields of a cyclotomic field, this is shown in Proposition 4. Let $\mathbb{Q}(\zeta_q)$ be the cyclotomic field of prime conductor $q > 2$. The Galois group $\mathbf{G} \coloneqq \mathsf{Gal}\left(\mathbb{Q}(\zeta_q) \,/\, \mathbb{Q}\right) \cong \mathbb{Z}_q^{\times}$ is cyclic of order $q-1$. Let g be a generator of $\mathbb{Z}_q^{\times}$. The automorphism $\sigma : \mathbb{Q}(\zeta_q) \to \mathbb{Q}(\zeta_q)$ sending $\zeta_q \mapsto \zeta_q^g$ is a generator of the Galois group $\mathbf{G}$. The relation between the Gaussian periods in Definition 1 and the cyclotomic field $\mathbb{Q}(\zeta_q)$ is captured in the following proposition.

Proposition 4. *Let $\mathbf{H}$ be the (unique) subgroup of order f in $\mathbf{G}$. Then $\mathbf{H} = \langle \sigma^n \rangle$, and the following is true:*

1. $\forall i \in \mathbb{Z} : \sigma^n(\eta_i) = \eta_i$.
2. $\forall i \in \mathbb{Z} : \mathbb{Q}(\eta_i) = \mathbb{Q}(\eta_0)$. $\mathbb{Q}(\eta_0)$ *is the fixed field corresponding to* $\mathbf{H}$.
3. $\forall i \in \mathbb{Z} : \eta_i = \mathbf{Tr}_{\mathbb{Q}(\eta_0)}^{\mathbb{Q}(\zeta_q)}(\zeta_q^i)$. *The minimal polynomial of* η_0 *is* $p_{\eta_0}(x) = \prod_{i=0}^{n-1}(x - \eta_i) \in \mathbb{Z}[x]$.
4. *The vector* $(\eta_0, \ldots, \eta_{n-1})$ *forms a normal, integral basis of* $\mathbb{Q}(\eta_0)$. *In particular,* $\mathbb{Z}[\eta_0, \ldots, \eta_{n-1}]$ *is the ring of integers of* $\mathbb{Q}(\eta_0)$.

Proof. The proof is included in the Appendix.

In summary we have,

$$\underset{\text{Decomposition Subring}}{\mathcal{O}_{\mathbf{M}}} \subset \underset{\text{Decomposition Ring}}{\mathcal{O}_{\mathbf{K}}} \subset \underset{\text{Ring of Integers}}{\mathcal{O}_{\mathbf{L}}}$$

with $\mathcal{O}_{\mathbf{L}} = \mathbb{Z}[\zeta_m]$ corresponding to the typical BGV/BFV plaintext space. An overview of the notation introduced is also provided in Table 1.

Symbol	Definition
r	Number of slots/η_i period-basis elements under Decomposition Ring $\mathcal{O}_{\mathbf{K}}$.
n	Number of slots/η_i period-basis elements under Decomposition Ring $\mathcal{O}_{\mathbf{M}}$.
q	Gaussian period root of unity parameter, set to be equal to cyclotomic m and is an odd prime.
p	Plaintext prime modulus whose multiplicative order $\bmod q$ divides f.
f	Length parameter of Gaussian periods.

Table 1. A summary of the main notation utilized.

2.3 Fully Homomorphic Encryption

Modern realizations of practical FHE are built upon lattice assumptions, particularly the decisional variants of the learning with errors (LWE) [23] and ring learning with errors (RLWE) [19] problems. The RLWE problem is based on the LWE problem. Both of them are commonly used to construct homomorphic encryption schemes. In this work, we employ the ring-variant problem.

Definition 5 (Ring-LWE Distribution). *For any secret* $\mathbf{s} \in \mathcal{R}_Q$*, error distribution* Φ *over* $\mathcal{R}_Q$*, the Ring-LWE distribution* $A_{\mathbf{s},\Phi}$ *over* $\mathcal{R}_Q^2$ *is defined by sampling* $\mathbf{a} \leftarrow \mathcal{U}(\mathcal{R}_Q)$, $\mathbf{e} \leftarrow \Phi$ *independently and output* $(\mathbf{a}, \mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \mathbf{e} \bmod q)$.

Definition 6 (Ring-LWE Problem, average-case decision). *Let* Υ *be a family of error distributions over* $\mathcal{R}_Q$*. Ring-LWE*$_{Q,\Upsilon}$ *asks to distinguish, with non-negligible advantage, between independent samples* $A_{\mathbf{s},\Phi}$*, where* $\mathbf{s}, \Phi \leftarrow \mathcal{R}_Q, \Upsilon$ *are randomly chosen; and the same number of uniform, independent samples from* $\mathcal{U}\left(\mathcal{R}_Q^2\right)$.

In practice each component of $\mathbf{s}$, over some fixed basis, is uniformly and independently sampled from a small set $\mathcal{S}$. When the secret $\mathbf{s}$ is sampled this way, we refer to the Ring-LWE problem as the small-secret Ring-LWE.

A homomorphic encryption (HE) scheme is a collection of algorithms - (Setup, KeyGen, Enc, Dec, EvalKeyGen, Evaluate). Setup gives the parameter set for following procedures. KeyGen generates a secret key sk. Enc encrypts a plaintext element from the plaintext space to a ciphertext from the ciphertext space using the public key. Dec decrypts a ciphertext ct to a plaintext using the secret key. EvalKeyGen generates an evaluation key evk. Evaluate allows the evaluation of ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_n$ on a function f and outputs a new ciphertext encrypting $f(\mathsf{ct}_1, , \ldots, \mathsf{ct}_n)$ using the evaluation key evk.

2.3.1 BFV Cryptosystem.

The BFV cryptosystem is an FHE scheme where the message and the noise are separated by storing the message in the higher bits and the noise in the lower bits (LSB). We present the BFV scheme here as a variant is utilized in Section 2.4. The BFV encryption scheme is as follows:

-BFV.Setup(1^λ): Given the security parameter λ, set ring $\mathcal{R}$, plaintext and ciphertext moduli t and q, the secret key distribution ψ and the error distribution χ, output the parameter set $\mathsf{pp} := (t,\ q,\ \psi,\ \chi)$.

-BFV.KeyGen(pp): Given the parameter set pp, sample secret key $\mathsf{sk} \leftarrow \psi$.

-BFV.Enc(m, sk): Given message $m \in \mathcal{R}_t$, sample $a \xleftarrow{\$} \mathcal{R}_q$ and $e \leftarrow \chi$. Define the scaling factor Δ to be $\lfloor \frac{q}{t} \rceil$. Encrypt it using secret key sk and output $\mathsf{ct} := (\mathsf{ct}_0, \mathsf{ct}_1)$ where $\mathsf{ct}_0 := a,\ \mathsf{ct}_1 := \mathsf{ct}_0 \cdot \mathsf{sk} + \Delta \cdot m + e$.

-BFV.Dec(ct, sk): Given the ciphertext $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1)$ and secret key sk, recover the message $m := \lfloor \frac{\mathsf{ct}_1 - \mathsf{ct}_0 \cdot \mathsf{sk}}{\Delta} \rceil$.

-BFV.EvalKeyGen(sk): Given the secret key sk, output the evaluation key evk:= BFV.Enc($\mathsf{sk}^2, \mathsf{sk}$).

-BFV.Evaluate($f, \mathsf{ct}^{(1)},\ \ldots, \mathsf{ct}^{(n)}, \mathsf{evk}$): Given a sequence of ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_n$, a function f and evaluation key evk, output a ciphertext $\mathsf{ct}_{\mathsf{eval}}$ which satisfies $\mathsf{Dec}(\mathsf{ct}_{\mathsf{eval}}, \mathsf{sk}) = f(\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(n)})$.

2.4 Subring Homomorphic Encryption

In BFV-based FHE schemes, the plaintext is set to be a vector of slots. Each slot represents an element of the Galois field GF(p^d) of degree d [25]. Addition and multiplication in plaintext space corresponds to slot-wise addition and multiplication over the Galois field GF(p^d). In [2], it was noted that in practice, a plaintext slot space of form $\mathbb{Z}_{p^l}$ is often preferred. Subring homomorphic encryption was therefore presented in [2]. The idea was to first go down from $\mathbb{Z}[\zeta_q]$ to $\mathcal{O}_K$ as discussed in Section 2.2.1, take the quotient of the ring under modulo p and then perform Hensel lifting over to $q' = p^l$. This decreased the slot degree from d to 1. In addition, a BFV-based homomorphic scheme [25] over the decomposition ring $\mathcal{O}_K$ was realized. The encoding and decoding algorithms, along with the complete scheme, are summarized in the following sections.

2.4.1 Encoding/Decoding. The encoding and decoding techniques of [2] involve transformations between the η-vectors and ξ-vectors. The η-vectors form the basis of the decomposition ring $\mathcal{O}_K$ and the ξ-vectors form the basis for the other end of the isomorphism (slot space) of the decomposition ring described in Section 2.2.1. The conversions are analogous to conversions between the power(ful) and Chinese Remainder Theorem(CRT) bases respectively. As before, let q be a prime index of the cyclotomic ring R. Let $p \neq q$ be a prime and $d = ord_q^\times(p)$ and the number of slots $r = \frac{q-1}{d}$. Let the integer modulus $q' = p^m$ and $t = p^l$, for $l < m$. Let $\vec{a} \in \mathbb{Z}^r$ denote an η-vector and let its encoding be denoted as $a = \vec{\eta}^T \cdot \vec{a} \in \mathcal{O}_{\mathbf{K}}$.

We present the transformations, between the η basis and the ξ basis, in Algorithm 1 and Algorithm 2. These perform the transformations between the ring $\mathcal{O}_{\mathbf{K}}/p^m\mathcal{O}_{\mathbf{K}}$ and the slots $\prod_{i=1}^{r}\mathbb{Z}_{p^m}$, mapping the ring elements to their slots and vice versa. We refer the reader to Section 3 of [2] for the definitions of the matrices $\Omega_Z^{(q')}$ and $\bar{\Omega}_Z^{(q')}$ as these are not utilized for the rest of the paper.

Algorithm 1 `eta_to_xi`

Input: η-vector $\vec{a}$, prime power q'.
Output: ξ-vector $\vec{b}$.
1: Set $a(X) = a_0 + a_{r-1}X + \cdots + a_1X^{r-1}$.
2: Set $c(X) = \eta_0 + \eta_1 X + \cdot + \eta_{r-1}X^{r-1}$ where $(\eta_i)_{i=0}^{r-1}$ is the first row of $\Omega_Z^{(q')}$.
3: Compute $b(X) = a(X)c(X) \bmod (q', X^{r-1})$.
4: **Return** $\vec{b} = (b_0, \ldots, b_{r-1})$.

Algorithm 2 `xi_to_eta`

Input: ξ-vector $\vec{b}$, prime power q'.
Output: η-vector $\vec{a}$.
1: Set $b(X) = b_0 + b_{r-1}X + \cdots + b_1X^{r-1}$.
2: Set $c(X) = \bar{\eta}_0 + \bar{\eta}_1 X + \cdots + \bar{\eta}_{r-1}X^{r-1}$ where $(\bar{\eta}_i)_{i=0}^{r-1}$ is the first row of $\bar{\Omega}_Z^{(q')}$.
3: Compute $a(X) = b(X)c(X) \bmod (q', X^{r-1})$.
4: Compute $t = b_0 + \cdots + b_{r-1} \bmod q'$.
5: **Return** $\vec{a} = (q^{-1}(a_i - dt) \bmod q')_{i=0}^{r-1}$.

To multiply two vectors encoded by η-vectors $\vec{a}$ and $\vec{b}$ modulo $q' = p^m$, we first convert the η-vectors to ξ-vectors modulo q', that is, they become elements of the slot space. From here, we multiply the resulting ξ-vectors component-wise and convert the result back to an η-vector modulo q'. The procedure is detailed in Algorithm 3. This is utilized in their proposed scheme to perform multiplication between elements of the ring $\mathcal{O}_{\mathbf{K}}/p^m\mathcal{O}_{\mathbf{K}}$, forming the building block of the scheme. That is, for two elements of the ring $\mathcal{O}_{\mathbf{K}}/p^m\mathcal{O}_{\mathbf{K}}$, their η-vectors are obtained and converted to the slot space, with the component-wise multiplication then applied and finally converted back.

2.4.2 Scheme. The subring homomorphic encryption scheme is based on the decomposition ring $\mathcal{O}_{\mathbf{K}}$ and the BFV scheme [7,15]. We describe the symmetric key variant below.

-SubringHE.SecretKeyGen(): Sample secret key from the key distribution $\vec{s} \leftarrow \chi_{key}$. Return $\mathsf{sk} = \vec{s} \in \mathbb{Z}^r$.

Algorithm 3 `mult_eta`

Input: η-vectors $\vec{a}, \vec{b}$, prime power q'.
Output: Multiplication product η-vector $\vec{c}$.
1: Set $\vec{\alpha} = \text{eta_to_xi}(\vec{a}, q')$.
2: Set $\vec{\beta} = \text{eta_to_xi}(\vec{b}, q')$.
3: **for** i in range $0 \leq i \leq r-1$ **do**
4: Compute $\gamma_i = \alpha_i \beta_i \bmod q'$.
5: **end for**
6: **Return** $\vec{c} = \text{xi_to_eta}(\vec{\gamma}, q')$.

-SubringHE.Enc($\mathsf{sk} = \vec{s} \in \mathbb{Z}^r$, $\vec{m} \in \mathbb{Z}_t^r$): Sample $\vec{a} \xleftarrow{\$} \mathbb{Z}_{q'}^r$, $\vec{e} \leftarrow \chi_{err}$, $\vec{n} =$ `xi_to_eta` $(\vec{m}, t)$. Calculate $\vec{b} :=$ `mult_eta`$(\vec{a}, \vec{s}, q') + \Delta \cdot \vec{n} + \vec{e} \bmod q'$. Return $\mathsf{ct} = (\vec{a}, \vec{b})$.

-SubringHE.Dec($\mathsf{sk} = \vec{s} \in \mathbb{Z}^r$, $\mathsf{ct} = (\vec{a}, \vec{b})$): Calculate $\vec{n} = \lfloor \frac{1}{\Delta}(\vec{b} - $`mult_eta`$(\vec{a}, \vec{s}, q') \bmod q') \rceil$. Define message vector as $\vec{m} =$ `eta_to_xi`$(\vec{n}, t)$. Return $\vec{m}$.

-SubringHE.Add($\mathsf{ct}_1 = (\vec{a}_1, \vec{b}_1)$, $\mathsf{ct}_2 = (\vec{a}_2, \vec{b}_2)$): Given two pairs of ciphertexts, calculate $\vec{a} := (\vec{a}_1 + \vec{a}_2) \mod q'$, $\vec{b} := (\vec{b}_1 + \vec{b}_2) \bmod q'$. Return $\mathsf{ct}_{\text{add}} = (\vec{a}, \vec{b})$.

-SubringHE.EvalKeyGen($\vec{s}$): Given secret key vector $\vec{s}$, calculate $\vec{\gamma} =$ `mult_eta` $(\vec{s}, \vec{s}, q')$. For k from 0 to $l_w - 1$, sample $\vec{\alpha}_k \xleftarrow{\$} \mathbb{Z}_{q'}^r$, $\vec{x}_k \leftarrow \chi_{err}$ and calculate $\vec{\beta}_k =$ `mult_eta`$(\vec{\alpha}_k, \vec{s}, q') + w^k \vec{\gamma} + \vec{x}_k \bmod q'$. Return an evaluation key $\mathsf{evk} = ((\vec{\alpha}_k, \vec{\beta}_k))_{k=0}^{l_w - 1}$.

-SubringHE.Mult($\mathsf{ct}_1 = (\vec{a}_1, \vec{b}_1)$, $\mathsf{ct}_2 = (\vec{a}_2, \vec{b}_2)$, $\mathsf{evk} = ((\vec{\alpha}_k, \vec{\beta}_k))_k$): Given two pairs of ciphertexts and evaluation key ek, calculate $\vec{e} = \lfloor \frac{1}{\Delta} \cdot$`mult_eta`$(\vec{b}_1, \vec{b}_2, \frac{q'^2}{t}) \rceil$, $\vec{c} = \lfloor \frac{1}{\Delta} \cdot$`mult_eta`$(\vec{a}_1, \vec{b}_2, \frac{q'^2}{t}) +$`mult_eta`$(\vec{a}_2, \vec{b}_1, \frac{q'^2}{t}) \rceil$, $\vec{d} = \lfloor$`mult_eta`$(\vec{a}_1, \vec{a}_2, \frac{q'^2}{t}) \rceil$. Define $(\vec{d}_0, \cdots, \vec{d}_{l_w - 1}) = \mathtt{WD}(\vec{d})$, which satisfies $l_w = \lfloor \log_w(q') \rceil + 1$, $\vec{d} = \sum_{k=0}^{l_w - 1} w^k \vec{d}_k$, $\vec{d}_k \in \mathbb{Z}_w^r$. Calculate $\vec{a} = \vec{c} + \sum_{k=0}^{l_w - 1}$`mult_eta`$(\vec{d}_k, \vec{\alpha}_k, q') \mod q'$, $\vec{b} = \vec{e} + \sum_{k=0}^{l_w - 1}$`mult_eta`$(\vec{d}_k, \vec{\beta}_k, q') \bmod q'$. Return $\mathsf{ct}_{\text{mult}} = (\vec{a}, \vec{b})$.

3 Decomposition Subring

To perform FHE more efficiently, we utilize the decomposition subring $\mathcal{O}_{\mathbf{M}}$ introduced in Section 2.2.2, to perform FHE over a ring of smaller dimension n. This is as opposed to performing FHE over the decomposition ring $\mathcal{O}_{\mathbf{K}}$ with dimension $r > n$. To enable this, encoding and decoding maps specific to the structure of $\mathcal{O}_{\mathbf{M}}$ are introduced in this section. From here, the encoding and decoding maps are later utilized for the operations within FHE. By decreasing the dimension, significant gains in efficiency are observed. This is later discussed in Section 4.

In the context of [2], encoding and decoding serve as fundamental operations in their scheme as described in Section 2.4.2. Crucially, they are utilized to define Algorithm 3 which is then heavily used in the subsequent homomorphic operations, forming the building block of their scheme. In general, to perform encoding and decoding, the fundamental idea involves a conversion between two bases. One is of the decomposition (sub)ring and the other, the slots form. This is analogous to the power(ful) and CRT bases in FHE.

Let $q > 2$ be an odd prime, $\Phi_q(x)$ be the q^{th} cyclotomic polynomial, $\mathbb{Q}(\zeta_q) \cong \mathbb{Q}[x]/(\Phi_q(x))$ the q^{th} cyclotomic field with the ring of integers $\mathbb{Z}[\zeta_q] \cong \mathbb{Z}[x]/(\Phi_q(x))$ and p be another prime distinct from q. Note that from this section onward, m is utilized to denote the power of the plaintext prime modulus and not the cyclotomic parameter. The famous Kummer-Dedekind theorem tells us exactly how the prime ideal $p\mathbb{Z}[\zeta_q]$ factorizes:

$$\mathbb{Z}[\zeta_q]/p\mathbb{Z}[\zeta_q] \cong \mathbb{Z}[x]/(p, \Phi_q(x)) \cong \prod_{i=1}^{r} \mathbb{Z}[x]/(p, f_i(x))$$

where $f_1(x) \dots f_r(x) \equiv \Phi_q(x) \pmod{p}$ are the irreducible divisors of $\Phi_q(x)$ over $\mathbb{F}_p[x]$. $\forall i : \deg(f_i) \coloneqq f = \mathsf{ord}_q^{\times}(p)$ and $r = \frac{\phi(q)}{f} = \frac{q-1}{f}$.

Prior Approach. In [2], the authors consider the decomposition ring $\mathbf{K} = \mathbb{Q}(\eta_0)$, $\mathcal{O}_{\mathbf{K}} = \mathbb{Z}[\eta_0, \dots, \eta_{r-1}]$ where $\forall i : \eta_i \coloneqq \sum_{t=0}^{f-1} \zeta_q^{g^{i+rt}}$, g is a multiplicative generator of $\mathbb{Z}_q^{\times}$, r, f are as in the previous paragraph. As discussed in Section 2.2.1, the prime ideal factorization of $p\mathcal{O}_{\mathbf{K}}$ is given by:

$$p\mathcal{O}_{\mathbf{K}} = \prod_{i=1}^{r} \mathfrak{P}_i, \quad \mathcal{O}_{\mathbf{K}}/p\mathcal{O}_{\mathbf{K}} \cong \prod_{i=1}^{r} \mathcal{O}_{\mathbf{K}}/\mathfrak{P}_i \cong \prod_{i=1}^{r} \mathbb{Z}_p$$

where each $\mathfrak{P}_i$ is a prime ideal factor of $p\mathcal{O}_{\mathbf{K}}$. In addition, $\forall m \geq 1$:

$$\mathcal{O}_{\mathbf{K}}/p^m\mathcal{O}_{\mathbf{K}} \cong \prod_{i=1}^{r} \mathcal{O}_{\mathbf{K}}/\mathfrak{P}_i^m \cong \prod_{i=1}^{r} \mathbb{Z}_{p^m}$$

The LHS of the equation above corresponds to the plaintext/ciphertext space of the BFV scheme. It has a basis $(\eta_0, \dots, \eta_{r-1})$ and multiplication follows the ring arithmetic of $\mathcal{O}_{\mathbf{K}} = \mathbb{Z}[\eta_0, \dots, \eta_{r-1}]$. The RHS corresponds to the slot space of the BFV scheme. It possesses "standard basis" $(\varepsilon_0, \dots, \varepsilon_{r-1})$ and follows the point-wise multiplication rule. We note that the authors of [2] refer to the basis on the RHS as the ξ basis. They define the ξ basis in terms of the canonical embedding but we find their formalism slightly extraneous. We deviate from their notation and emphasize that the isomorphism above is simply an application of the Chinese Remainder Theorem.

An explicit conversion between the slot space and the plaintext/ciphertext space can be found in Section 3.4 of [2]. Their general strategy can be summarized as follows:

– From the factorization $\Phi_q(x) \equiv f_1(x) \dots f_r(x) \pmod{p}$, calculate a CRT basis

$$\tau_i \coloneqq \left(\prod_{j \neq i} f_j(x)^{-1} \bmod (p, f_i(x)) \right) \prod_{j \neq i} f_j(x) \bmod p$$

Identify τ_i with an element of $\mathbb{Z}[\zeta_q]$. It turns out that $\tau_i \in \mathcal{O}_{\mathbf{K}}$ and τ_i is an idempotent element with respect to the ideal $\mathfrak{P}_i$.
– Hensel lift the idempotents to $\bar{\tau_i} \in \mathcal{O}_{\mathbf{K}}$ such that $\bar{\tau_i}$ is an idempotent element with respect to the ideal $\mathfrak{P}_i^m$
– For each i, calculate an $a_i \in \mathbb{Z}_{p^m}$ satisfying $a_i \cdot \bar{\tau_i} \equiv \eta_0 \cdot \bar{\tau_i} \pmod{p^m}$ over $\mathcal{O}_{\mathbf{K}}$
– Finally, to translate the basis component vector $\mathbf{v}$ of the LHS to the slot component vector $\mathbf{w}$, compute $\mathbf{w} = \mathbf{A}\mathbf{v}$ where $\mathbf{A} \in \mathbb{Z}_{p^m}^{r \times r}$ is a circulant matrix, with first row $(a_j)_{j=1}^r$.

Our Approach. The approach in [2] depends essentially on $q - 1 = rf$, where $f = \mathsf{ord}_q^{\times}(p)$, $r = \frac{q-1}{f}$. In our case in which we wish to decrease r and work in a subring of $\mathcal{O}_{\mathbf{K}}$, which we refer to as the decomposition subring $\mathcal{O}_{\mathbf{M}}$. This method breaks down because the τ_i terms are no longer idempotents in our setting. Therefore, we propose a new method that handles both the decomposition ring and decomposition subring cases. Furthermore, this approach is simpler than the previous one. To overcome the difficulty of finding the idempotents, we attempt to find a member, F_i, of the decomposition subring $\mathcal{O}_{\mathbf{M}}$, such that a principal ideal (F_i) contains every factor of a prime p except a specific prime ideal $\mathbf{P}_i$. In a sense F_i represents the complement of $\mathbf{P}_i$. This F_i is then used to compute the decoding map as first discussed in Section 3.1. From here the encoding map is computed from the decoding map. The details are discussed in Section 3.2.

3.1 Decoding map

Given the period-basis elements $\eta_0, \dots, \eta_{n-1}$ of the decomposition subring $\mathcal{O}_{\mathbf{M}}$, we consider the prime ideal factorization of $\mathcal{O}_{\mathbf{M}}$ and show that, up to permutation, any prime ideal $\mathbf{P}_i = (p, f_i(\zeta_q))$ from $p\mathbb{Z}[\zeta_q] = \prod_{i=1}^r (p, f_i(\zeta_q))$ can be used to calculate decoding kernel $(a_i)_{i=0}^{n-1}$. This is detailed in Algorithm 4. Here the decoding kernel corresponds to the first row of a circulant matrix $\mathbf{A}$. This decoding kernel is subsequently employed to compute the decoding map, mapping elements from $\mathcal{O}_{\mathbf{M}}/p^m\mathcal{O}_{\mathbf{M}} \to \prod_{i=1}^n \mathbb{Z}_{p^m}$. This is performed via taking the product of a polynomial containing the coefficients to be decoded and a polynomial containing decoding kernel as its coefficients. The details are presented in Algorithm 5.

To give a technical overview, a factor $\mathfrak{P}_i$ of $p\mathcal{O}_{\mathbf{M}} = p\mathbb{Z}[\eta_0, \dots, \eta_{n-1}] = \prod_{i=0}^{n-1} \mathfrak{P}_i$ is first selected. Any factor can be chosen, and without loss of generality, denote this factor as $\mathfrak{P}_i$. From here, we consider $\mathfrak{P}_i\mathbb{Z}[\zeta_q] = \prod_k \mathbf{P}_{i_k}$ and select one of the $\mathbf{P}_{i_k}$. Again, any factor can be chosen and without loss of generality, denote this factor as $\mathbf{P}_i$. A special polynomial $F_i(x)$ is constructed such

that for any element u, $\mathfrak{P}_i$ exactly divides u if and only if $\mathbf{P}_i$ exactly divides u. This is as per Proposition 7. Furthermore this is if and only if p exactly divides $uF_i(x)^2$. This polynomial enables us to find the a_i such that $\eta_i \equiv a_i (\mathrm{mod} \mathfrak{P}_i^m)$, by the proof of correctness for Algorithm 4.

Let $q-1 = nf$, now f can be a multiple of $\mathsf{ord}_q^\times(p)$. Again designate $\mathbf{M} = \mathbb{Q}(\eta_0)$, $\mathcal{O}_\mathbf{M} = \mathbb{Z}[\eta_0, \ldots, \eta_{n-1}]$ where $\forall i: \ \eta_i \coloneqq \sum_{t=0}^{f-1} \zeta_q^{g^{i+tn}}$, again g is a fixed multiplicative generator of $\mathbb{Z}_q^\times$. Let $m \geq 1$, the ideal $p^m\mathcal{O}_\mathbf{M}$ factorizes as:

$$p^m\mathcal{O}_\mathbf{M} = \prod_{i=1}^{n} \mathfrak{P}_i^m, \quad \forall i: \ \mathcal{O}_\mathbf{M}/\mathfrak{P}_i^m \cong \mathbb{Z}_{p^m}$$

Each $\mathfrak{P}_i$ is a prime ideal divisor of $p\mathcal{O}_\mathbf{M}$. Fix any prime ideal divisor, say $\mathfrak{P}_0$. For each $0 \leq i < n$ our goal is to calculate the unique $a_i \in \mathbb{Z}_{p^m}$ satisfying

$$\eta_i \equiv a_i \pmod{\mathfrak{P}_0^m}$$

Because the Galois group $\mathsf{Gal}\left(\mathbb{Q}(\eta_0) \,/\, \mathbb{Q}\right) \cong C_n$ is cyclic of order n, the Chinese Remainder isomorphism can be captured by a circulant matrix $\mathbf{A}$ with a first row of form $(a_i)_{i=0}^{n-1}$.

To compute a_i, recall that over the cyclotomic ring $\mathbb{Z}[\zeta_q]$, $p\mathbb{Z}[\zeta_q] = \prod_{i=1}^{r}(p, f_i(\zeta_q))$ where $\Phi_q(x) \equiv \prod_{i=1}^{r} f_i(x) \pmod p$ and $r = \mathsf{ord}_q^\times(p)$.

Proposition 7. *Fix an index i, and let $\mathbf{P}_i = (p, f_i(\zeta_q))$ be a prime ideal divisor of $p\mathbb{Z}[\zeta_q]$. Define $F_i(x) \coloneqq \Phi_q(x) \cdot f_i(x)^{-1} \bmod p$. The element $F_i(\zeta_q) \in \mathbb{Z}[\zeta_q]$ separates the prime ideal $\mathbf{P}_i$ in the following sense:*

For any element $u \in \mathbb{Z}[\zeta_q]$, $\mathbf{P}_i^k \parallel u$ if and only if $p^k \parallel u \cdot F_i(\zeta_q)^{k+1}$ over $\mathbb{Z}[\zeta_q]$. i.e. $\mathbf{P}_i^k$ exactly divides u if and only if p^k exactly divides $u \cdot F_i(\zeta_q)^{k+1}$ over $\mathbb{Z}[\zeta_q]$.

Proof. The proof is provided in the Appendix.

As $\mathcal{O}_\mathbf{M} \subset \mathbb{Z}[\zeta_q]$, for any fixed index i, the lifted ideal $\mathfrak{P}_i\mathbb{Z}[\zeta_q]$ factorizes as $\mathfrak{P}_i\mathbb{Z}[\zeta_q] = \prod_l (p, f_{i_l}(\zeta_q))$ for some distinct indices i_l over $\mathbb{Z}[\zeta_q]$. We can use the valuation index of any prime ideal $(p, f_{i_l}(\zeta_q))$ to calculate the valuation index over $\mathfrak{P}_i$. In other words, for any $u \in \mathbb{Z}[\eta_0, \ldots, \eta_{n-1}]$

$$\mathfrak{P}_i^k \parallel u \text{ over } \mathcal{O}_\mathbf{M} \iff (p, f_{i_l}(\zeta_q))^k \parallel u \text{ over } \mathbb{Z}[\zeta_q] \tag{1}$$

Up to permutation, any prime ideal $\mathfrak{P}_i$ can serve as an "anchor" as $\mathcal{O}_\mathbf{M}/\mathfrak{P}_i^m$ are all isomorphic to $\mathbb{Z}_{p^m}$. The discussion above implies that we may essentially use any prime ideal $(p, f_i(\zeta_q))$ to calculate $(a_i)_{i=0}^{n-1}$. Specifically, denote a chosen $f_i(X)$ as an irreducible factor $f(x)$ of $\Phi_q(x) \bmod p$. Similarly, denote the corresponding $\mathfrak{P}_i$ as $\mathfrak{P}$. Also, denote the coefficient of x^i in a given the polynomial $H(x)$ as $H[i]$. Moreover, all indices i are implicitly taken to be $i \bmod n$. Algorithm 4 is then used to compute the decoding kernel. This then enables the use of the decoding map, detailed in Algorithm 5. This produces the map of the left-to-right direction of the Chinese Remainder isomorphism discussed above.

Algorithm 4 `DecodeKerGen`

Input: Prime q, period-basis $\eta_0, \ldots, \eta_{n-1}$, modulus p^m.
Output: Decoding kernel $(a_i)_{i=0}^{n-1} \in \mathbb{Z}_{p^m}^n$, where $\eta_i \equiv a_i \pmod{\mathfrak{P}^m}$ for some prime ideal $\mathfrak{P}|p$.

1: Obtain an irreducible factor $f(x)$ of $\Phi_q(x) \bmod p$.
2: Compute $F(x) = \Phi_q(x) f(x)^{-1} \bmod p$.
3: Compute $\phi(x) = F(x)^m \bmod (p^m, \Phi_q(x))$. ▷ Taken to power of m to fulfill the requirement/congruence on page 12.
4: Obtain an index idx where $gcd(\phi[\mathsf{idx}], p) = 1$. ▷ To allow for computation of the inverse in Step 8.
5: **for** each basis element η_i **do**
6: Represent η_i as a polynomial $\eta_i(x) = \sum_{t=0}^{f-1} x^{g^{i+nt} \bmod q} \in \mathbb{Z}[x]$. ▷ Replacing ζ_q with a variable x.
7: Compute $A_i(x) = \eta_i(x)\phi(x) \bmod (p^m, \Phi_q(x))$.
8: Compute $a_i = A_i[\mathsf{idx}] \cdot \phi[\mathsf{idx}]^{-1} \bmod p^m$.
9: **end for**
10: **Return** $(a_i)_{i=0}^{n-1} \in \mathbb{Z}_{p^m}^n$

Algorithm 4 takes as its input a prime p where p^m forms the modulus to be utilized. It then takes in prime q corresponding to the cyclotomic polynomial $\Phi_q(x)$ as discussed above as well as the period basis. The irreducible factor $f(x)$ is selected and the relevant polynomials are computed as previously discussed. From here, as $\eta_i \in \mathbb{Z}[\zeta_q]$ we then express it as the polynomial $\eta_i(x)$. The polynomial $A_i(x)$ is then computed as per Proposition 7. Furthermore, according to Proposition 7 and the proof below, the polynomial $A_i(x)$ should be an a_i multiple of $\phi(x) \bmod p^m$. This implies that the degree j coefficient of $A_i(x)$ is also an a_i multiple of the degree j coefficient of $\phi(x)$ for all $0 \leq j \leq \deg(\phi(x))$. We therefore choose any "good" index idx such that the degree idx coefficient of $\phi(x)$ is coprime to p, which then uniquely determines the constant multiple a_i using the modular division in Step 8 of Algorithm 4. Additionally, an example demonstrating how Algorithm 4 works is provided in the Appendix.

Correctness of Algorithm 4: Let $f(x)$ be the mod p irreducible polynomial chosen by the Algorithm 4, and let $\mathfrak{P}$ be the unique prime ideal in $\mathcal{O}_{\mathbf{M}} = \mathbb{Z}[\eta_0, \ldots, \eta_{n-1}]$ such that $\mathfrak{P} \mid (p, f(\zeta_q))$ over $\mathbb{Z}[\zeta_q]$. The correctness is obtained from the correctness of Steps 7 and 8 of the algorithm.

For the correctness of Step 7,

$$
\begin{aligned}
&\eta_i \equiv a_i \pmod{\mathfrak{P}^m} \iff \mathfrak{P}^m \mid (\eta_i - a_i) \\
&\iff (p, f(\zeta_q))^m \mid (\eta_i - a_i) \text{ over } \mathbb{Z}[\zeta_q] \text{ by Equation (1)} \\
&\iff p^m \mid (\eta_i - a_i) F(\zeta_q)^m \text{ by Proposition 7} \\
&\iff (\eta_i(x) - a_i) F(x)^m \equiv 0 \bmod (p^m, \Phi_q(x)) \text{ lifting cyclotomic integers to } \mathbb{Z}[x] \\
&\iff \eta_i(x) F(x)^m \bmod (p^m, \Phi_q(x)) \text{ is an } a_i \text{ multiple of } F(x)^m \bmod (p^m, \Phi_q(x))
\end{aligned}
$$

The ratio mod p^m of an appropriate pair of coefficients will yield the constant $a_i \in \mathbb{Z}_{p^m}$.

For the correctness of Step 8, denote the result of Step 7 after modular reduction to be of the form

$$A_i(x) = \sum_{j=0}^{q-2} r_j x^j = a_i \sum_{j=0}^{q-2} t_j x^j \bmod (p^m, \Phi_q(x)).$$

Here, we denote $\phi(x) = \sum_{j=0}^{q-2} t_j x^j$ and denote by r_j the coefficients of the reduced $A_i(x)$. Again, note that this form is due to the result of Step 7 being an a_i multiple of $\phi(x) = F(x)^m \bmod (p^m, \Phi_q(x))$. Choose index idx such that $gcd(\phi[\mathsf{idx}], p) = 1$. By equating the coefficients of the reduced equation, we obtain $a_i \cdot t_{\mathsf{idx}} x^{\mathsf{idx}} = r_{\mathsf{idx}} x^{\mathsf{idx}}$. As $\phi[\mathsf{idx}]$ is invertible under modulo p by the choice of idx, we can obtain $a_i = r_{\mathsf{idx}} \cdot t_{\mathsf{idx}}^{-1}$.

□

Algorithm 5 `Decode`

Input: Encoded polynomial $c(X) = \sum_{i=0}^{n-1} c_i X^i \in \mathbb{Z}_{p^m}[X]$, decoding kernel $(a_i)_{i=0}^{n-1} \in \mathbb{Z}_{p^m}^n$, modulus p^m.
Output: Polynomial $c'(X)$ containing the slot entries in its coefficients.
1: Initialize decoding polynomial $t(X) = \sum_{i=0}^{n-1} t_i X^i$.
2: **for** i in range $0 \le i \le n-1$ **do**
3: Set $t_i \leftarrow a_{-i \pmod n}$
4: **end for**
5: Compute $c'(X) = t(X) \cdot c(X) \bmod (X^n - 1, p^m)$.
6: **Return** $c'(X)$

The resulting decoding map is described in Algorithm 5. Here, it utilizes the decoding kernel computed using Algorithm 4, first converting it into a polynomial $t(X)$. This is then multiplied with an encoded polynomial, resulting in the desired decoded polynomial $c'(X)$. This contains the slot-wise entries as its coefficients. The correctness of Algorithm 5 follows directly from the discussion of Section 3.1, the correctness of Algorithm 4 and the standard properties of circulant matrices. It is routine to see that the computation in Algorithm 5 corresponds to multiplication by the circulant matrix $\mathbf{A}$. The time complexity of Algorithm 5 is $O(n \log n)$. This arises from the multiplication of the two polynomials in Step 5 of the algorithm, which is performed using the Fast Fourier Transformation(FFT).

3.2 Encoding map

The encoding map, which performs the map from $\prod_{i=1}^{n} \mathbb{Z}_{p^m} \to \mathcal{O}_{\mathbf{M}}/p^m\mathcal{O}_{\mathbf{M}}$, is subsequently derived from the decoding map. Here, we first observe that for a circulant matrix $\mathbf{A}$ corresponding to the decoding map, the circulant matrix $\mathbf{B}$ corresponding to the encoding map can be derived. From there, Algorithm 6 is utilized to compute the first row of the matrix $\mathbf{B}$, referred to as the encoding

kernel $(b_j)_{j=0}^{n-1} \in \mathbb{Z}_{p^m}^n$. The encoding kernel is then utilized in Algorithm 7, which details the encoding procedure.

For consistency, we use the same notation and variables as Section 3.1. To recap, let $\mathbf{v} \in \mathbb{Z}_{p^m}^n$ be a component vector of $\mathcal{O}_{\mathbf{M}}/p^m\mathcal{O}_{\mathbf{M}}$ with the basis $(\eta_0, \ldots, \eta_{n-1})$. The corresponding component vector $\mathbf{w} \in \mathbb{Z}_{p^m}^n$ in the slot space $\prod_{i=1}^n \mathbb{Z}_{p^m}$ with "standard basis" $(\varepsilon_0, \ldots, \varepsilon_{n-1})$ is determined by the Chinese Remainder isomorphism

$$\varphi : \mathcal{O}_{\mathbf{M}}/p^m\mathcal{O}_{\mathbf{M}} \to \prod_{i=1}^{n} \mathbb{Z}_{p^m}$$
$$\eta_i \mapsto \sum_{j=0}^{n-1} (\eta_i \bmod \mathfrak{P}_j^m)\varepsilon_j = \sum_{j=0}^{n-1} \sigma^j(\eta_{i-j} \bmod \mathfrak{P}_0^m)\varepsilon_j = \sum_{j=0}^{n-1} a_{i-j}\varepsilon_j$$

Or equivalently $\mathbf{w} = \mathbf{A}\mathbf{v}$, $\mathbf{A}_{i,j} = a_{j-i}$ where $\mathfrak{P}$ is the unspecified prime ideal and $(a_i)_{i=0}^{n-1}$ the output of Algorithm 4. The generator of the Galois group is $\langle\sigma\rangle = \mathsf{Gal}\left(\mathbb{Q}(\eta_0) \,/\, \mathbb{Q}\right)$ with $\sigma(\eta_i) = \eta_{i+1}$.

According to Equations (8) and (20) of [13],

$$\sum_{i=0}^{n-1} \eta_i = -1 \tag{*}$$
$$\sum_{i=0}^{n-1} \eta_i\eta_{i+k} = q \cdot d_k - f \text{ where} \tag{**}$$
$$d_k = \delta_{0,k} \text{ if } f \text{ is even, } d_k = \delta_{e/2,k} \text{ if } f \text{ is odd}$$

The inverse isomophism φ^{-1} can be easily obtained from these 2 facts.

Proposition 8. *The inverse of the matrix $\mathbf{A} \in \mathbb{Z}_{p^m}^{n\times n}$ given above is a circulant matrix $\mathbf{B} \in \mathbb{Z}_{p^m}^{n\times n}$, whose first row is given by $(b_j)_{j=0}^{n-1}$ such that for any $0 \le j \le n-1$:*

- *If f is even, $b_j = q^{-1} \cdot (a_{-j} - f) \bmod p^m$*
- *If f is odd, $b_j = q^{-1} \cdot (a_{e/2-j} - f) \bmod p^m$*

Proof. The proof is provided in the Appendix.

To summarize the computation for the encoding map described above, we present Algorithm 6. This computes the first row of the circulant matrix $\mathbf{B}$, which we refer to as the encoding kernel $(b_j)_{j=0}^{n-1} \in \mathbb{Z}_{p^m}^n$. Here, the identities obtained from Proposition 8 are utilized to compute the respective coefficients of the encoding kernel. The correctness of Algorithm 6 follows directly from Proposition 8.

The encoding map is described in Algorithm 7. Here, it utilizes the encoding kernel computed using Algorithm 6. The encoding kernel is first converted into a polynomial $t(X)$. This is then multiplied with a polynomial containing the slot

Algorithm 6 `EncodeKerGen`

Input: Decoding kernel $(a_i)_{i=0}^{n-1} \in \mathbb{Z}_{p^m}^n$,, modulus p^m, prime q.
Output: Encoding kernel $(b_j)_{j=0}^{n-1} \in \mathbb{Z}_{p^m}^n$.

Compute $f = \frac{q-1}{e}$.
Compute inverse $q^{-1} \bmod p^m$.
if f is even **then**
    for j in range $0 \leq j \leq n-1$ **do**
        $b_j \leftarrow (a_{-j \bmod n} - f) \cdot q^{-1} \bmod p^m$
    end for
else
    for j in range $0 \leq j \leq n-1$ **do**
        $b_j \leftarrow (a_{\frac{e}{2}-j} - f) \cdot q^{-1} \bmod p^m$
    end for
end if
Return $(b_j)_{j=0}^{n-1} \in \mathbb{Z}_{p^m}^n$

Algorithm 7 `Encode`

Input: Polynomial $c(X) = \sum_{j=0}^{n-1} c_j X^j \in \mathbb{Z}_{p^m}[X]$ with slot entries in coefficients, encoding kernel $(b_j)_{j=0}^{n-1} \in \mathbb{Z}_{p^m}^n$, modulus p^m.
Output: Polynomial $c'(X)$ containing c_js in slots.

Initialize encoding polynomial $t(X) = \sum_{j=0}^{n-1} t_j X^j$.
for j in range $0 \leq j \leq n-1$ **do**
    Set $t_j \leftarrow b_{-j \pmod n}$
end for
Compute $c'(X) = t(X) \cdot c(X) \bmod (X^n - 1, p^m)$
Return $c'(X)$

entries in the coefficients, resulting in the desired encoded polynomial $c'(X)$. The correctness of Algorithm 7 follows directly from the discussion above in Section 3.2, the correctness of Algorithm 6 and the standard properties of circulant matrices. Similar to Algorithm 5, the computation in Algorithm 7 corresponds to multiplication by the circulant matrix $\mathbf{B}$. The time complexity of Algorithm 7 is $O(n \log n)$. This arises from the multiplication of the polynomials in Step 5 of the algorithm, which is performed using FFT.

4 Homomorphic Operations

Here, the encoding and decoding techniques of Section 3 are applied to develop a homomorphic scheme. For the purpose of comparing the effects of using the decomposition subring $\mathcal{O}_{\mathbf{M}}$ against using the decomposition ring $\mathcal{O}_{\mathbf{K}}$, this scheme will be similar to the one in Section 2.4.2. The `Encode` and `Decode` functions enable the use of the decomposition subring, which has a lower dimension compared to the decomposition ring. We show that this therefore provides greater efficiency.

Algorithm 8 `GaussMult`

Input: Polynomials $c_1(X) = \sum_{j=0}^{n-1} c_{1,j} X^j \in \mathbb{Z}_{p^m}[X]$, $c_2(X) = \sum_{j=0}^{n-1} c_{2,j} X^j \in \mathbb{Z}_{p^m}[X]$ where the coefficients are the components in the period-basis, encoding kernel $(b_j)_{j=0}^{n-1} \in \mathbb{Z}_{p^m}^n$, decoding kernel $(a_i)_{i=0}^{n-1} \in \mathbb{Z}_{p^m}^n$, modulus p^m.

Output: Encoded polynomial $t(X)$ containing in its slots the slot-wise product of $c_1(X), c_2(X)$.

1: Set $c_1'(X) \leftarrow$ `Decode`$(c_1(X), (a_i)_{i=0}^{n-1}, p^m)$
2: Set $c_2'(X) \leftarrow$ `Decode`$(c_2(X), (a_i)_{i=0}^{n-1}, p^m)$
3: Initialize polynomial $t'(X) = \sum_{j=0}^{n-1} t_j' X^j$.
4: **for** j in range $0 \leq j \leq n-1$ **do**
5: Set $t_j' \leftarrow c_1'(X)[j] \cdot c_2'(X)[j] \bmod p^m$
6: **end for**
7: Set $t(X) \leftarrow$ `Encode`$(t'(X), (b_j)_{j=0}^{n-1}, p^m)$
8: **Return** $t(X)$

To construct the scheme, first recall that the function described in Section 2.4.1 as Algorithm 3 forms the main building block of the homomorphic scheme described in Section 2.4.2. It is heavily utilized, being a component of almost all the FHE operations in Section 2.4.2. Here, we provide a variant of Algorithm 3, labeled Algorithm 8, based on the encoding and decoding maps described in Section 3. From there, we utilize it in the scheme described in Section 2.4.2 and analyze the effects on the complexity of the FHE operations. It is shown that the operations of the proposed scheme exhibit a lower complexity due to the lower dimension of the decomposition subring. Following this, the security analysis of the scheme is performed. It is shown that the proposed scheme is secure assuming the hardness of the small-secret Ring-LWE$_{Q,\Phi}$. Finally, the correctness of the scheme is shown. Here, error bounds are described for each operation.

Algorithm 8 describes the slot-wise multiplication procedure. It functions similar to Algorithm 3, with the main difference being the calls to our encoding and decoding functions. Specifically, Algorithm 5 is utilized in the first two steps while Algorithm 7 is utilized in the last step. Correctness follows directly from Section 2.4.1 and the correctness of Algorithms 5 and 7. Crucially, we note that the modulus p^m utilized changes depending on which operation Algorithm 8 is utilized for in Section 2.4.2. As Algorithms 5 and 7 are both performed with time complexity $O(n \log n)$, Algorithm 8 has time complexity $O(n \log n)$.

Complexity Analysis. From here, we directly replace Algorithm 3 with Algorithm 8 and apply it to the FHE scheme presented in Section 2.4.2. Similarly Algorithms 1, 2 are replaced with Algorithms 5 and 7, our `Decode` and `Encode` algorithms, respectively. Following the notation introduced at the beginning of Section 3, we denote n as the dimension of our decomposition subring $\mathcal{O}_{\mathbf{M}}$ and we let r be the dimension of the decomposition ring $\mathcal{O}_{\mathbf{K}}$ utilized in [2].

For encryption, as shown in Section 2.4.2, a single call to the `Encode` and `GaussMult` functions is invoked. This dominates the complexity of encryption as both functions are of $O(n \log n)$ complexity, from the discussion above as well as in Section 3.2. Encryption therefore requires $O(n \log n)$ time complexity in

Multiplication Algorithm	Encryption	Decryption	Addition	Multiplication
Algorithm 3	$O(r \log r)$	$O(r \log r)$	$O(r)$	$O(l_w \cdot r \log r)$
Algorithm 8	$O(n \log n)$	$O(n \log n)$	$O(n)$	$O(l_w \cdot n \log n)$

Table 2. Comparison of efficiency between the use of Algorithm 3 of [2] against the use of our Algorithm 8 in the homomorphic scheme described in Section 2.4.2.

our context and $O(r \log r)$ in theirs. Similarly, decryption invokes a single call to the `Decode` and `GaussMult` functions which both require $O(n \log n)$ complexity due to the discussion above as well as Section 3.1. Therefore decryption requires $O(n \log n)$ time complexity in our context and $O(r \log r)$ in theirs. For addition as described in Section 2.4.2, it is trivial to see that ours requires $O(n)$ time complexity while theirs requires $O(r)$. For multiplication, the procedure is dominated by the sum of l_w calls to `GaussMult`, where l_w is the parameter selected for decomposition. Its complexity is therefore $O(l_w \cdot n \log n)$ time complexity in our context and $O(l_w \cdot r \log r)$ in theirs. The complexity analysis is summarized in Table 2. By choosing the parameters such that $n < r$, our framework therefore outperforms theirs in terms of complexity.

4.1 Analysis of the Scheme

Security Analysis. The security of our scheme is based on a variant of the decisional version of the Ring-LWE assumption, defined on an arbitrary number field. Compared to the standard assumption, this assumes that the underlying ring can be an arbitrary ring of integers and that the coefficients of the secret key are binary. This is described in Theorem 9 below.

Theorem 9 (Security). *Assume the hardness of the small-secret Ring-DLWE$_{Q,\Phi}$ [6, 22] where Q is the ciphertext modulus and Φ a distribution on $\mathbf{K}_\mathbb{R}$ specified in the BFV scheme. The scheme is* IND-CPA *secure.*

Proof. The proof is provided in the Appendix.

Correctness.

Here, we provide the noise analysis for the decryption as well as the addition and multiplication operations for our scheme. A bound[3] for the multiplication depth supported by the scheme is also provided. The results are presented in Theorem 11.

Let $\mathbf{K} = \mathbb{Q}[\eta_0]$, $Q > 0$ be the ciphertext modulus, $0 < t < Q$ be the plaintext modulus and $\Delta := \lfloor \frac{Q}{t} \rceil$ be the scaling factor. Define $\|a\|_{\eta,\infty} := \max_{0 \le i < n} |a_i|$, where a_i are the components of a in the period basis and let δ be the multiplication expansion factor defined as the smallest $\delta > 0$ such that $\forall x, y \in \mathbf{K}_\mathbb{R}$:

[3] Further derivations of these bounds will be included in the full version of the paper.

$\|xy\|_{\eta,\infty} \leq \delta \|x\|_{\eta,\infty} \|y\|_{\eta,\infty}$. Let $w > 0 \in \mathbb{Z}$ be the relinearization modulus, and $l = \lfloor \log_w Q \rfloor + 1$ be the relinearization depth.

For decryption correctness, let $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1)$ be a ciphertext satisfying $\mathsf{ct}_1 - s \cdot \mathsf{ct}_0 \equiv \Delta m + e \bmod Q$ for some message $m = \sum m_i \eta_i$, $\|m\|_{\eta,\infty} \leq \frac{t}{2}$ and error $e \in \mathbf{K}_{\mathbb{R}}$. Then decrypting ct correctly recovers m if $\|e\|_{\eta,\infty} < \frac{Q}{2t} - \frac{Q - \Delta t}{2}$.

For addition, suppose $\mathsf{ct}^{(1)}$ has an inherent noise $e^{(1)}_{\mathsf{inh}}$ corresponding to $m_1 \in \mathbb{R}_t$, $\mathsf{ct}^{(2)}$ has an inherent noise $e^{(2)}_{\mathsf{inh}}$ corresponding to $m_1 \in \mathbb{R}_t$. Then the ciphertext resulting from homomorphic addition $\mathsf{ct}^{(1+2)}$ has an inherent noise $e^{(1+2)}_{\mathsf{inh}}$ satisfying $\|e^{(1+2)}_{\mathsf{inh}}\|_{\eta,\infty} \leq |Q - \Delta t| + \|e^{(1)}_{\mathsf{inh}}\|_{\eta,\infty} + \|e^{(2)}_{\mathsf{inh}}\|_{\eta,\infty}$.

For multiplication, assume each evaluation key has the same noise bound $\|e_{\mathsf{ev}}\|_{\eta,\infty}$.

Lemma 10. *For each $A \in \mathcal{R} = \mathbb{Z}[\eta_0]$, write the w-decomposition of A as $A = \sum_{i=0}^{l-1} A^{(i)} w^i$, $A^{(i)} \in \mathcal{R}$, $\|A^{(i)}\|_{\eta,\infty} < w$. To obtain the decomposition we decompose each coefficient into w-ary representation and then collect the terms with the same arity into each $A^{(i)}$. For any "relinearization" $\mathsf{ct}^{\mathsf{rel}} = (\mathsf{ct}^{\mathsf{rel}}_0, \mathsf{ct}^{\mathsf{rel}}_1)$ defined by:*

$$\mathsf{ct}^{\mathsf{rel}}_0 \coloneqq \sum_{i=0}^{l-1} A^{(i)} \cdot \mathsf{ev}^{(i)}_0 \bmod Q, \quad \mathsf{ct}^{\mathsf{rel}}_1 \coloneqq \sum_{i=0}^{l-1} A^{(i)} \cdot \mathsf{ev}^{(i)}_1 \bmod Q$$

There exists $e^{(\mathsf{rel})}_{\mathsf{inh}}$ such that $\mathsf{ct}^{\mathsf{rel}}_1 - s \cdot \mathsf{ct}^{\mathsf{rel}}_0 \equiv A \cdot s^2 + e^{(\mathsf{rel})}_{\mathsf{inh}} \pmod Q$ with an inherent noise bound $\|e^{(\mathsf{rel})}_{\mathsf{inh}}\|_{\eta,\infty} \leq w\delta \sum_{i=0}^{l-1} \|e^{(i)}_{\mathsf{ev}}\|_{\eta,\infty}$.

Let $\mathsf{ct}^{\mathsf{rel}}$ be the relinearization ciphertext defined in Lemma 10. Let the ciphertext $\mathsf{ct}^{(1\times 2)} = (\mathsf{ct}^{(1\times 2)}_0, \mathsf{ct}^{(1\times 2)}_1) \in \mathcal{R}^2_Q$ be the result of homomorphic multiplication defined by:

$$\begin{aligned}
d_0 &\coloneqq \lfloor \frac{t}{Q} \mathsf{ct}^{(1)}_0 \mathsf{ct}^{(2)}_0 \rceil \bmod Q \\
d_1 &\coloneqq \lfloor \frac{t}{Q} (\mathsf{ct}^{(1)}_0 \mathsf{ct}^{(2)}_1 + \mathsf{ct}^{(1)}_1 \mathsf{ct}^{(2)}_0) \rceil \bmod Q \\
d_2 &\coloneqq \lfloor \frac{t}{Q} \mathsf{ct}^{(1)}_1 \mathsf{ct}^{(2)}_1 \rceil \bmod Q \\
\mathsf{ct}^{\mathsf{rel}} = (\mathsf{ct}^{\mathsf{rel}}_0, \mathsf{ct}^{\mathsf{rel}}_1) &\coloneqq \left(\sum_i d_0{}^{(i)} \cdot \mathsf{ev}^{(i)}_0 \bmod Q, \sum_i d_0{}^{(i)} \cdot \mathsf{ev}^{(i)}_1 \bmod Q \right) \\
\mathsf{ct}^{(1\times 2)} = (\mathsf{ct}^{(1\times 2)}_0, \mathsf{ct}^{(1\times 2)}_1) &\coloneqq (d_1 + \mathsf{ct}^{\mathsf{rel}}_0 \bmod Q, d_2 + \mathsf{ct}^{\mathsf{rel}}_1 \bmod Q)
\end{aligned}$$

Then $\mathsf{ct}^{(1\times 2)}$ has an inherent noise $e^{(1\times 2)}_{\mathsf{inh}}$ satisfying $\mathsf{ct}^{(1\times 2)}_1 - s \cdot \mathsf{ct}^{(1\times 2)}_0 \equiv \Delta(m_1 m_2 \bmod t) + e^{(1\times 2)}_{\mathsf{inh}} \pmod Q$, upper bounded by:

$$\begin{aligned}
\|e^{(1\times 2)}_{\mathsf{inh}}\|_{\eta,\infty} \leq \Bigg\{ &\frac{\delta t}{2}(\delta + t)(Q - \Delta t) + \frac{\delta^2 t}{2}(\|e^{(1)}_{\mathsf{inh}}\|_{\eta,\infty} + \|e^{(2)}_{\mathsf{inh}}\|_{\eta,\infty}) + \\
&\frac{2\delta t}{Q} \|e^{(1)}_{\mathsf{inh}}\|_{\eta,\infty} \|e^{(2)}_{\mathsf{inh}}\|_{\eta,\infty} + w\delta l \|e_{\mathsf{ev}}\|_{\eta,\infty} \Bigg\} (1 + o(1))
\end{aligned}$$

For the multiplication depth, under the additional assumptions that $Q = \Delta t$ and that each multiplicand has roughly the same inherent noise magnitude: $\|e^{(1)}_{\mathsf{inh}}\|_{\eta,\infty} \approx \|e^{(1)}_{\mathsf{inh}}\|_{\eta,\infty}$, the BFV scheme supports repeated homomorphic multiplication with the multiplication depth being

$$L \lesssim \frac{\log \frac{Q}{2t} - \log \|e^{\mathsf{init}}_{\mathsf{inh}}\|_{\eta,\infty} - \log w\delta l}{\log 2t + 2\log \delta}$$

where $\|e^{\mathsf{init}}_{\mathsf{inh}}\|_{\eta,\infty}$ is the magnitude of the inherent noise in a fresh ciphertext.

Theorem 11. *Let* $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1)$ *be a ciphertext of the FHE scheme discussed. Let the error term of* ct *satisfy* $\|e\|_{\eta,\infty} < \frac{Q}{2t} - \frac{Q-\Delta t}{2}$. *The decryption of* ct *then yields the correct output. Furthermore, assuming the bound on multiplicative depth provided above holds, our scheme correctly evaluates any circuit of multiplication depth* L. *The correctness of the FHE scheme discussed therefore follows.*

5 Experiments

We provide a proof-of-concept implementation of the BFV-based FHE scheme on the ring of Gaussian periods as discussed in Section 4. This was implemented and benchmarked using SageMath [27] on an Intel(R) Core(TM) i7-7700HQ CPU @ (2.80GHz, 2.80GHz).

Parameter Set	Cyclotomic q	Ring Dimension n	Plaintext Modulus	Ciphertext Modulus	Error Var.	Security (bits)
1-A	45319	6474	11^2	$(11^2)^8$	4	444.2
1-B		2158				127.6
2-A	41761	5220	17	17^8		-
2-B		1044				103.4
3-A	44983	6426	113	113^8		447.6
3-B		2142				128.4
4-A	41761	5220	17	17^{16}	8	292.4
4-B		1740				88.7
5-A	28057	9352	167	167^{16}		290.2
5-B		4676				133.1
6-A	41203	4578	29	29^{16}		206.7
6-B		2289				98

Table 3. Parameters for our FHE framework described in Section 4.

Experiments were conducted with the six sets of parameters listed in Table 3. For each parameter set, the cyclotomic q value is first chosen. Following this, for a given prime, the corresponding decomposition ring dimension(A) and the smallest decomposition subring dimension(B) we can decrease it to, are listed.

The ring dimension is decreased as much possible while still retaining at least 80-bit security. Parameter sets 1, 3 and 5 correspond to approximately 128-bit security. The security estimates were obtained from the Lattice-Estimator [1]. The binary secret key was sampled from a uniform distribution. The error terms were sampled from a discrete Gaussian distribution, with the variance listed in Table 3. Parameter sets 1 to 3 correspond to having a smaller ciphertext modulus while the parameter sets 4 to 6 correspond to a larger ciphertext modulus. The benchmark timings of the experiments are listed in Table 4 for each operation.

Parameter Set	Encryption(ms)	Decryption(ms)	Addition(ms)	Multiplication(ms)
1-A	106.65	126.83	5.83	1829.98
1-B	38.58	42.19	2.05	683.28
2-A	108.43	100.41	4.44	1982.82
2-B	21.42	21.13	0.91	414.07
3-A	161.85	149.97	5.59	4449.5
3-B	52.35	54.5	2.09	1665.39
4-A	110.07	102.46	4.58	2203.22
4-B	36.95	45.94	1.91	629.92
5-A	289.81	272.52	9.12	8767.55
5-B	125.70	126.40	4.49	4157.89
6-A	94.73	105.63	4.33	2198.2
6-B	47.83	49.21	2.21	1109.54

Table 4. Timings for our FHE framework described in Section 4.

As observed in Tables 4, 5, significant latency improvements are obtained. Specifically, for the parameter set 2, we obtain up to a 5.06× speed-up with our techniques. Based on the data in Table 5, it is also observed that the proportion of the latency improvement for each operation is roughly similar to the proportion of the decrease in the ring dimension n. It is noted that, compared to the other parameters, the parameter set 5 exhibits a marginally greater degree of improvement in Table 5. This may be attributed to the SageMath libraries employed within the implementation for our scheme. It is also noted that the size of the parameters utilized was restricted by the capabilities of our hardware. If larger parameters were employed, it is likely that the latency would be further improved through greater reductions in the dimension. Tables 4 and 5 also represent a form of comparison between the techniques of [2, 3] (A parameter sets) with our work (B parameter sets). Note that the works of [2, 3] utilize similar techniques to form their plaintext spaces. However, a direct comparison with their proposed parameters is not provided here, as applying our techniques to their proposed parameter sets would result in there being less than sufficient security due to the decrease in the dimension.

Parameter Set	Change in Dimension	Change in Encryption	Change in Decryption	Change in Addition	Change in Multiplication
1-A $\leftrightarrow$ 1-B	$\times 3$	$\times 2.76$	$\times 3.01$	$\times 2.84$	$\times 2.68$
2-A $\leftrightarrow$ 2-B	$\times 5$	$\times 5.06$	$\times 4.75$	$\times 4.88$	$\times 4.79$
3-A $\leftrightarrow$ 3-B	$\times 3$	$\times 3.09$	$\times 2.75$	$\times 2.67$	$\times 2.67$
4-A $\leftrightarrow$ 4-B	$\times 3$	$\times 2.98$	$\times 2.23$	$\times 2.40$	$\times 3.50$
5-A $\leftrightarrow$ 5-B	$\times 2$	$\times 2.31$	$\times 2.16$	$\times 2.03$	$\times 2.11$
6-A $\leftrightarrow$ 6-B	$\times 2$	$\times 1.98$	$\times 2.15$	$\times 1.96$	$\times 1.98$

Table 5. Relation between change in dimension and change in timings for our FHE framework described in Section 4.

A comparison between our work and that of [16] is performed in Table 6. Amongst the recommended parameters of [16], the parameters with the smallest plaintext modulus are selected. We compare them with our parameters while approximately equating for the number of slots and security. As observed in Table 6, for a similar number of slots and security, our methods can attain plaintext modulus values that are significantly smaller than theirs. This demonstrates an accessibility to new parameters not previously available with the techniques of [16], which instead restricts their plaintext modulus to large primes.

Parameter Set	2-B	[16]	3-B	[16]	5-B	[16]
Plaintext Modulus	17	$2^{16}+1$	113	$2^{16}+1$	167	$2^{16}+1$
Number of slots	1044	1024	2142	2048	4676	4096
Security	103.4	128	128.4	128	133.1	128
Multiplication levels	< 4	~ 25	< 4	~ 25	< 6	~ 25

Table 6. A comparison between our parameters and the proposed parameters of [16] with their smallest plaintext modulus $2^{16}+1 = 65537$.

It is noted that for our proof-of-concept parameters, the number of multiplication levels available is lower than that of [16], as observed in Table 6. This is likely due to the implementation-level differences between the schemes of the two works. However, it is crucial to note that the primary contribution of our work is the construction of a new plaintext space which allows for parameter selection through a reduction of both the slot degree and the number of slots, and not the scheme itself. We leave it to future works to adapt and optimize our techniques within the typical BFV framework, which would provide a more exact comparison to the work of [16].

Additionally, although our parameters support fewer levels, our scheme exhibits a significantly improved communication cost. This is detailed in Table 7 which compares the communication cost of our scheme to the estimated communication cost of [16]. Specifically, for both the size of the ciphertexts and public keys in our proposed parameters, an improvement of up to $172\times$ is obtained

Parameter Set	2-B	3-B	5-B	[16]
Ciphertext size (MB)	0.01	0.03	0.14	1.72
Public key size (MB)	0.01	0.03	0.14	1.72
Evaluation key size (MB)	0.18	1.02	10.36	37.84
Total size (MB)	0.20	1.08	10.64	41.28

Table 7. A comparison of communication cost between our parameters and the proposed parameters of [16] with their smallest plaintext modulus $2^{16} + 1 = 65537$.

over the work of [16]. For the size of the evaluation key, an improvement of up to 210.2× is observed. Overall, this translates into an improvement of up to 206.4× in communication cost. Again, optimizing and implementing our plaintext space within the standard BFV framework is beyond the scope of this work and is left to future research, where a more precise comparison with [16] may be conducted, for factors such as the communication cost.

6 Conclusion

To conclude, we introduced FHE on the ring of Gaussian periods. By proceeding further down the decomposition ring and utilizing the decomposition subring, the dimension is lowered from r to n, therefore lowering the latency. To achieve this, we introduced novel methods for both encoding and decoding. These methods were then applied to the operations within FHE, decreasing the time complexity from, for encryption and decryption, $O(r \log r)$ to $O(n \log n)$, for addition, $O(r)$ to $O(n)$ and for multiplication, $O(l_w \cdot r \log r)$ to $O(l_w \cdot n \log n)$. This was further validated in a proof-of-concept implementation, achieving up to a 5.06× latency improvement for selected parameters. To summarize, we introduced a novel method which is optimized for lower-dimension use-cases and offers an increased flexibility in the selection of parameters. Future studies could explore the application of these techniques within the context of bootstrapping. Our techniques may also be applied to the work of [21], allowing for greater flexibility in their bootstrapping procedure. Furthermore, additional algebraic constructions could be explored. For instance, we could take the minimal polynomial $p_{\eta_0}(x) \in \mathbb{Z}[x]$ of η_0 and work over the order $\mathbb{Z}[x]/(p_{\eta_0}(x)) \subsetneq \mathbb{Z}[\eta_0, \ldots, \eta_{n-1}]$. Under Order-LWE assumption [5], this order would support FHE instantiations based on polynomial arithmetic. This direction of research would be non-trivial as the Gaussian periods $\eta_1, \ldots, \eta_{n-1}$ do not generally lie inside $\mathbb{Z}[x]/(p_{\eta_0}(x))$. We leave a more detailed comparison between the ring of integers and its suborder to future work.

Acknowledgments.

This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative, Trust Tech Funding Initiative (DTC-IGC-01) and Singapore Ministry of

Education Academic Research Fund Tier 2 Grant MOE-T2EP20223-0016. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority. Allen Siwei Yang is supported by the Institute for Infocomm Research of Agency for Science, Technology and Research (A*STAR). The authors would like to thank Dr Khoa Nguyen for his insightful discussions on these topics. We also thank the anonymous CT-RSA reviewers for their helpful comments.

References

1. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology 9(3), 169–203 (2015)
2. Arita, S., Handa, S.: Subring homomorphic encryption. In: Kim, H., Kim, D.C. (eds.) ICISC 17. LNCS, vol. 10779, pp. 112–136. Springer, Cham (Nov / Dec 2018)
3. Arita, S., Handa, S.: Fully homomorphic encryption scheme based on decomposition ring. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 103(1), 195–211 (2020)
4. Bae, Y., Cheon, J.H., Hanrot, G., Park, J.H., Stehlé, D.: Plaintext-ciphertext matrix multiplication and FHE bootstrapping: Fast and fused. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part III. LNCS, vol. 14922, pp. 387–421. Springer, Cham (Aug 2024)
5. Bolboceanu, M., Brakerski, Z., Perlman, R., Sharma, D.: Order-LWE and the hardness of ring-LWE with entropic secrets. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part II. LNCS, vol. 11922, pp. 91–120. Springer, Cham (Dec 2019)
6. Boudgoust, K., Jeudy, C., Roux-Langlois, A., Wen, W.: On the hardness of module-LWE with binary secret. In: Paterson, K.G. (ed.) CT-RSA 2021. LNCS, vol. 12704, pp. 503–526. Springer, Cham (May 2021)
7. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 868–886. Springer, Berlin, Heidelberg (Aug 2012)
8. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: Goldwasser, S. (ed.) ITCS 2012. pp. 309–325. ACM (Jan 2012)
9. Chen, H., Laine, K., Player, R., Xia, Y.: High-precision arithmetic in homomorphic encryption. In: Smart, N.P. (ed.) CT-RSA 2018. LNCS, vol. 10808, pp. 116–136. Springer, Cham (Apr 2018)
10. Cheon, J.H., Kim, A., Kim, M., Song, Y.S.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 409–437. Springer, Cham (Dec 2017)
11. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 3–33. Springer, Berlin, Heidelberg (Dec 2016)
12. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 377–408. Springer, Cham (Dec 2017)

13. Dickson, L.E.: Cyclotomy, higher congruences, and waring's problem. American Journal of Mathematics 57(2), 391–424 (1935)
14. Ducas, L., Micciancio, D.: FHEW: Bootstrapping homomorphic encryption in less than a second. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 617–640. Springer, Berlin, Heidelberg (Apr 2015)
15. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144 (2012), https://eprint.iacr.org/2012/144
16. Geelen, R., Vercauteren, F.: Fully homomorphic encryption for cyclotomic prime moduli. In: Fehr, S., Fouque, P.A. (eds.) EUROCRYPT 2025, Part III. LNCS, vol. 15603, pp. 366–397. Springer, Cham (May 2025)
17. Ho, M.C., Ku, Y.T., Xiao, Y., Liu, F.H., Hsu, C.F., Chang, M.C., Hung, S.H., Chen, W.C.: Invited paper: Efficient design of fhew/tfhe bootstrapping implementation with scalable parameters. In: Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design. Association for Computing Machinery (2025)
18. Kim, D., Song, Y.: Approximate homomorphic encryption over the conjugate-invariant ring. In: Lee, K. (ed.) ICISC 18. LNCS, vol. 11396, pp. 85–102. Springer, Cham (Nov 2019)
19. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Berlin, Heidelberg (May / Jun 2010)
20. Marcus, D.A., Sacco, E.: Number fields, vol. 1995. Springer (1977)
21. Min, S., Song, Y.: Carousel: Fully homomorphic encryption from slot blind rotation technique. Cryptology ePrint Archive, Report 2024/2032 (2024), https://eprint.iacr.org/2024/2032
22. Peikert, C., Regev, O., Stephens-Davidowitz, N.: Pseudorandomness of ring-LWE for any ring and modulus. In: Hatami, H., McKenzie, P., King, V. (eds.) 49th ACM STOC. pp. 461–473. ACM Press (Jun 2017)
23. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC. pp. 84–93. ACM Press (May 2005)
24. Ren, S.Q., Meng, T.H., Yibin, N., Mi Aung, K.M.: Privacy-preserved multi-party data merging with secure equality evaluation. In: 2016 International Conference on Cloud Computing Research and Innovations (ICCCRI). pp. 34–41 (2016)
25. Smart, N.P., Vercauteren, F.: Fully homomorphic SIMD operations. DCC 71(1), 57–81 (2014)
26. Tan, B.H.M., Lee, H.T., Wang, H., Ren, S., Aung, K.M.M.: Efficient private comparison queries over encrypted databases using fully homomorphic encryption with finite fields. IEEE Transactions on Dependable and Secure Computing 18(6), 2861–2874 (2021)
27. The Sage Developers: SageMath, the Sage Mathematics Software System (Version 10.5) (2024), https://www.sagemath.org
28. Xia, H., Liu, F.H., Wang, H.: More efficient functional bootstrapping for general functions in polynomial modulus. In: Boyle, E., Mahmoody, M. (eds.) TCC 2024, Part IV. LNCS, vol. 15367, pp. 130–163. Springer, Cham (Dec 2024)
29. Yamada, R., Okumura, S., Miyaji, A.: Consideration on defining field for efficient ring-lwe. In: 2024 19th Asia Joint Conference on Information Security (AsiaJCIS). pp. 25–32. IEEE (2024)

7 Appendix

Proof of Proposition 4.

Proof. Items 1 to 3 are immediate consequences of Galois theory and cyclotomic fields. For Item 4, first observe that all the summands within η_i, $0 \leq i < n$ form a partition of all the primitive q^{th} roots of unity $\{\zeta_q^j : 0 \leq j < q\}$. Since the latter set is linearly independent over $\mathbb{Q}$, $\{\eta_i, 0 \leq i < n\}$ are also linearly independent over $\mathbb{Q}$. It is a basis of $\mathbb{Q}(\eta_0)$ because of dimensionality.

Suppose $a \in \mathbb{Q}(\eta_0)$ is integral. Viewing $\mathbb{Q}(\eta_0) \subset \mathbb{Q}(\zeta_q)$ as a subfield and recalling that $\{\zeta_q^j : 0 \leq j < q\}$ is a normal, integral basis of $\mathbb{Q}(\zeta_q)$, we may write $a = \sum_{j=0}^{q-1} a_j \zeta_q^j$ where all $a_j \in \mathbb{Z}$. By applying the automorphisms σ^{nk}, $1 \leq k < f$ to a and comparing the coefficients with a, we see that all ζ_q^j that appear as summands in the same Gaussian period share the same coefficient. Consequently, a can be written as integer combinations of the periods. □

Proof of Proposition 7.

Proof. Since $F_i(x) = \prod_{j \neq i} f_j(x) \bmod p$, the ideal $(p, F_i(x)) = (p, \prod_{j \neq i} f_j(x))$ holds over $\mathbb{Z}[x]$. Next we claim that $(p, \prod_{j \neq i} f_j(x)) = \prod_{j \neq i}(p, f_j(x))$ over $\mathbb{Z}[x]$.

On one hand, $\prod_{j \neq i}(p, f_j(x)) \subseteq (p, \prod_{j \neq i} f_j(x))$ because each ideal generator of the LHS belongs to the RHS. On the other hand, $p \in \prod_{j \neq i}(p, f_j(x))$ since the $f_j(x)$ are pairwise relatively prime mod p. Therefore equality holds.

Hence $\gcd(p, F_i(\zeta_q)) = \prod_{j \neq i} \mathbf{P}_j$, $\mathbf{P}_j \coloneqq (p, f_j(\zeta_q))$ over $\mathbb{Z}[\zeta_q]$. In other words, $F_i(\zeta_q)$ is divisible by all the prime ideal factors of $p\mathbb{Z}[\zeta_q]$ except $\mathbf{P}_i$. Note that when considering just division when restricted to the case of k,

$$\mathbf{P}_i^k \mid u \iff \mathbf{P}_i^k \cdot F_i(\zeta_q)^k \mid u \cdot F_i(\zeta_q)^k \iff p^k \mid u \cdot F_i(\zeta_q)^k$$

can be obtained. For exact division in the case of $k+1$ as per the theorem, since $\mathbb{Z}[\zeta_q]$ is a Dedekind domain and integral ideals have a unique factorization,

$$\begin{aligned} \mathbf{P}_i^k \parallel u &\iff (\mathbf{P}_i^{k+1}, u) = \mathbf{P}_i^k \\ &\iff (\mathbf{P}_i^{k+1} \cdot F_i(\zeta_q)^{k+1}, u \cdot F_i(\zeta_q)^{k+1}) = \mathbf{P}_i^k \cdot F_i(\zeta_q)^{k+1} = (p)^k \cdot F_i(\zeta_q) \\ &\iff p^k \parallel u \cdot F_i(\zeta_q)^{k+1}. \end{aligned}$$

For the forward direction of the last step, $p^k \mid u \cdot F_i(\zeta_q)^{k+1}$ follows directly from the previous step. To show exact division, suppose not, that is, suppose $p^{k+1} \mid u \cdot F_i(\zeta_q)^{k+1}$. This implies $\mathbf{P}_i^{k+1} \cdot F_i(\zeta_q)^{k+1} \mid u \cdot F_i(\zeta_q)^{k+1}$ which implies $\mathbf{P}_i^{k+1} \mid u$, leading to a contradiction.

□

An example for Algorithm 4. As an example of how Algorithm 4 works, consider the thirteenth cyclotomic field with cyclotomic polynomial $\Phi_{13}(x) = x^{12} + x^{11} + \cdots + x^2 + x + 1$. Now let $p = 3$. It is not difficult to compute:

$$\Phi_{13}(x) \equiv (x^3 + 2x + 2)(x^3 + x^2 + 2)(x^3 + x^2 + x + 2)(x^3 + 2x^2 + 2x + 2) \bmod 3.$$

By the CRT we have:

$$\mathbb{Z}[x]/(3, \Phi_{13}(x)) \cong \prod_{i=1}^{4} \mathbb{F}_{27}.$$

In the BFV terminology the RHS contains 4 slots, each with degree 3 over $\mathbb{F}_3$. By looking at a subring $\mathcal{O}_{\mathbf{M}} \subset \mathbb{Z}[x]/(\Phi_{13}(x))$ and using structure theorems from algebraic number theory, it is possible to obtain a simpler CRT factorization:

$$\mathcal{O}_{\mathbf{M}}/3\mathcal{O}_{\mathbf{M}} \cong \prod_{i=1}^{2} \mathbb{F}_3.$$

Both the degree of each slot and the number of slots have shrunk in the subring.

A priori, $\mathcal{O}_{\mathbf{M}}$ contains an integral normal basis $\{\eta_0, \eta_1\}$, and each element of $\mathcal{O}_{\mathbf{M}}/3\mathcal{O}_{\mathbf{M}}$ can be represented by $a\eta_0 + b\eta_1$, $a, b \in \mathbb{Z}_3$. It is not immediately clear though, for example, which element in $\mathbb{F}_3 \times \mathbb{F}_3$ corresponds to $\eta_0 \bmod 3$ in the CRT factorization.

The idea is therefore to leverage our knowledge of the explicit factorization $\Phi_{13}(x) \bmod 3$. This is what is utilized by Proposition 7 and Algorithm 4. In this particular example, the procedure becomes the following:

1. Use the unique prime ideal $\mathbf{P} \subset \mathcal{O}_{\mathbf{M}}$ lying under $(3, \zeta_{13}^3 + 2\zeta_{13} + 2)$ as an anchor.
2. Compute the polynomial $F(x) = \Phi_{13}(x)/(x^3 + 2x + 2) \bmod 3 = x^9 + x^8 + 2x^7 + x^5 + 2x^3 + 2x^2 + 2$.
3. Use the definition of η_0 to represent it as a polynomial $\eta_0(x) = x + x^4 + x^3 + x^{12} + x^9 + x^{10}$.
4. Compute $\eta_0(x)F(x) \bmod \Phi_{13}(x) \bmod 3 = 2x^9 + 2x^8 + x^7 + 2x^5 + x^3 + x^2 + 1$, and notice that this is $2F(x) \bmod 3$. Therefore, $\eta_0 \equiv 2 \bmod \mathbf{P}$.
5. Similarly, we obtain $\eta_1 \equiv 0 \bmod \mathbf{P} \iff \eta_0 \equiv 0 \bmod \sigma^{-1}(\mathbf{P})$, where σ is a generator of the Galois group. Specifically, $\sigma(\zeta_{13}) \to \zeta_{13}^2$.
6. We conclude that up to permutation, $\eta_0 \bmod 3$ corresponds to $(2, 0) \in \mathbb{F}_3 \times \mathbb{F}_3$ under the CRT map above.

Proof of Proposition 8.

Proof. In the case where f is even,

$$\begin{aligned}
\mathbf{BA}_{i,j} &= \sum_k \mathbf{B}_{i,k}\mathbf{A}_{k,j} \bmod p^m = \sum_k b_{k-i}a_{j-k} \bmod p^m \\
&= q^{-1} \sum_k (a_{i-k} - f)a_{j-k} \bmod p^m \\
&= q^{-1} \left(\sum_k \eta_{i-k}\eta_{j-k} - f \sum_k \eta_{j-k} \right) \bmod \mathfrak{P}^m \text{ as } \eta_i \equiv a_i \bmod \mathfrak{P}^m. \\
&= q^{-1}(q\delta_{i,j} - f + f) \text{ by (*) and (**)} \ = \delta_{i,j}.
\end{aligned}$$

In the case where f is odd,

$$\begin{aligned}
\mathbf{BA}_{i,j} &= \sum_k \mathbf{B}_{i,k}\mathbf{A}_{k,j} \bmod p^m = \sum_k b_{k-i}a_{j-k} \bmod p^m \\
&= q^{-1}\sum_k (a_{\frac{e}{2}+i-k} - f)a_{j-k} \bmod p^m \\
&= q^{-1}\left(\sum_k \eta_{\frac{e}{2}+i-k}\eta_{j-k} - f\sum_k \eta_{j-k}\right) \bmod \mathfrak{P}^m \text{ as } \eta_i \equiv a_i \bmod \mathfrak{P}^m. \\
&= q^{-1}(q\delta_{i,j} - f + f) \text{ by (*) and (**)} \ = \delta_{i,j}.
\end{aligned}$$

□

Proof of Theorem 9.

Proof. The proof is the same as the standard BFV scheme instantiated over power-of-two cyclotomic fields. Below we sketch the basic idea:

1. The public key $(\mathsf{pk}_0, \mathsf{pk}_1)$ is pseudorandom. $\mathsf{pk}_0 = a \xleftarrow{\$} \mathcal{R}_Q$, $\mathsf{pk}_1 = as + e$ and $s \xleftarrow{\$} \mathcal{S}$, $e \xleftarrow{\$} \Phi$. $(\mathsf{pk}_0, \mathsf{pk}_1)$ therefore comes from the small secret Ring-LWE$_{Q,\Phi}$ sample and hence is indistinguishable from being random by the small-secret Ring-DLWE assumption.
2. Assume $(\mathsf{pk}_0, \mathsf{pk}_1)$ is truly random, then $(\mathsf{ct}_0, \mathsf{ct}_1)$ is pseudorandom.
   Note that $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1) = (\mathsf{pk}_0 \cdot u + e_1 \bmod Q, \mathsf{pk}_1 \cdot u + \Delta m + e_2 \bmod Q)$ where u, e_1, e_2 are independent and u has the same distribution as s, $e_1, e_2 \xleftarrow{\$} \Phi$ are from the error distribution.
   Under the small secret Ring-DLWE assumption, $(\mathsf{pk}_0 \cdot u + e_1, \mathsf{pk}_1 \cdot u + e_2)$ is pseudorandom. It therefore completely masks the scaled message Δm, so ct is pseudorandom. □