Breaking the power-of-two barrier: noise estimation for BGV in NTT-friendly rings

Andrea Di Giusto[1,2] and Chiara Marcolla[2]

[1] Eindhoven University of Technology, Eindhoven, the Netherlands
a.di.giusto@tue.nl
[2] Technology Innovation Institute, Abu Dhabi, United Arab Emirates
chiara.marcolla@gmail.com

The Brakerski-Gentry-Vaikuntanathan (BGV) scheme is a Fully Homomorphic Encryption (FHE) cryptosystem based on the Ring Learning With Error (RLWE) problem. Ciphertexts in this scheme contain an error term that grows with operations and causes decryption failure when it surpasses a certain threshold. For this reason, the parameters of BGV need to be estimated carefully, with a trade-off between security and error margin. The ciphertext space of BGV is the ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(\Phi_m(x))$, where usually the degree n of the cyclotomic polynomial $\Phi_m(x)$ is chosen as a power of two for efficiency reasons. However, the jump between two consecutive powers-of-two polynomials also causes a jump in the security, resulting in parameters that are much bigger than what is needed.

In this work, we explore the non-power-of-two instantiations of BGV. Although our theoretical research encompasses results applicable to any cyclotomic ring, our main investigation is focused on the case of $m = 2^s \cdot 3^t$, i.e., cyclotomic polynomials with degree $n = \phi(m) = 2^s \cdot 3^{t-1}$. We provide a thorough analysis of the noise growth in this new setting using the canonical norm and compare our results with the power-of-two case, considering practical aspects like NTT algorithms. We find that in many instances, the parameter estimation process yields better results for the non-power-of-two setting.

1 Introduction

Fully Homomorphic Encryption (FHE) is a revolutionary field that enables computations on encrypted data without the need for decryption. Namely, a set of operations can be performed over ciphertexts such that these operations are reflected as additions and multiplications on the corresponding plaintexts. This capability presents a powerful tool for privacy-preserving data processing, offering solutions for different applications such as machine learning, cloud services, and secure computation outsourcing.

Several FHE schemes were proposed after Gentry's breakthrough thesis [33]. Among all FHE schemes, the most practical, efficient and widely adopted are BGV [13], BFV [12,31], TFHE [18,19] which improves the FHEW scheme [30], and CKKS [17,16]. The reader interested in FHE and its applications will find some introductory material in [50,1,15,49].

In this work, we focus on the Brakerski-Gentry-Vaikuntanathan (BGV) [13] scheme. BGV can be instantiated using either the integers or cyclotomic rings, yielding a scheme based on Learning with Errors [56] (LWE) or its Ring variant [46] (RLWE), respectively; the latter version is often preferred for efficiency reasons.

Roughly speaking, the (decision version of) RLWE problems consist of distinguishing equations perturbed by small noise from uniformly random systems. The issue arising from this construction is noise growth. Indeed, to guarantee a correct decryption, the error added has to be small. Unfortunately, it increases when homomorphic operations are computed, and to allow a larger number of supported operations, we have to increase the ciphertext modulus. However, a higher modulus also decreases the security level of the underlying scheme. On the other hand, to increase the security level, we can adopt a higher polynomial degree n at the cost of efficiency. This balancing process called *parameter estimation*, is one of the main issues that need to be tackled in order to make FHE practical. See [35,24,42,52,25] for more details on BGV parameter estimation and [2,52,6,8,11,43] regarding frameworks for efficiently selecting parameters.

The ciphertext space of RLWE-based schemes is the ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(\Phi_m(x))$, where $\Phi_m(x)$ is the cyclotomic polynomial of degree $n = \phi(m)$. In general, n is chosen as a power of two because $\Phi_m(x) = x^n + 1$ and the ring has a nice algebraic structure, exploitable in many ways. The main example is polynomial multiplication, which is one of the main computational bottlenecks in lattice based cryptography. To address this problem, fast algorithms are necessary for efficient computation; when n is a power of two, we can use the powerful radix-2 Number Theoretic Transform (NTT) [7] algorithm.

Powers-of-two are sparse, and this can turn out to be a problem: it can happen that we are forced to choose *non-optimal* instantiations of BGV only because we have to increase the degree n and the jump between two consecutive powers-of-two is too big. Due to this significant gap, researchers have started to explore the idea of studying non-power-of-two cyclotomic polynomials [4]. Promising results have been obtained by applying it to NTRU, as demonstrated in [48].

Our contribution. In this work, we investigate non power-of-two BGV, meaning we choose the cyclotomic index m to be different from a power-of-two, and in particular, we consider $m = 2^s \cdot 3^t$. The main change coming with this idea is that now $\Phi_m(x) = x^n - x^{n/2} + 1$, where $n = m/3$, which influences many different aspects of the BGV cryptosystem. The most important ones are 1) the algorithms for the NTT; 2) how modular reductions affect the computation of polynomial products and 3) how reductions modulo the quotient cyclotomic polynomial $\Phi_m(x)$ impact the error bounds.

The first topic has been recently addressed in [48], showing how it is possible to find algorithms that are competitive with the radix-2 NTT also in this framework, and in our work we explore the latter two aspects thorougly.

Regarding the second subject, we make a significant contribution by demonstrating how to compute the full covariance matrix of the product of two random polynomials modulo $\Phi_m(x)$ when $m = 2^s \cdot 3^t$ (Theorem 3). The proof is based on a particular factorization of the polynomial $\Phi_m(x)$, suggesting possible generalizations to scenarios where m is the product of other prime powers.

Concerning the third topic, we provide a comprehensive worst-case analysis using the canonical norm to estimate the parameters. Specifically, we compute noise bounds for all homomorphic operations and investigate how to effectively combine different operations within the BGV scheme to perform complex computations in specific homomorphic circuits. Moreover, using Ljapunov's Central Limit Theorem [9], we give a rigorous proof of the widely used fact that the canonical embedding of a random polynomial yields vectors whose components have distribution well approximated by a complex Gaussian (Theorem 2). This result is independent from the factorization of m, hence it is valid for any cyclotomic ring. We believe that the theorems and technical tools developed in this paper hold the potential to be of independent interest for various other applications beyond the scope of this work, especially when we consider the amount of attention drawn by lattice based cryptography in the context of post-quantum standardization.

On the applied side, we compare our results with the power-of-two setting and find that there are many scenarios where it is preferable to use non-power-of-two BGV. In fact, our examples demonstrate that, while maintaining a similar modulus size q and comparable performance in NTT algorithms, it is possible to achieve a 25% reduction in the vector length n. This discovery represents a significant advancement towards more feasible applications of BGV and suggests that similar techniques can also be applied to other FHE constructions.

This work is structured as follows.

- In Section 2, we introduce the mathematical notions serving as foundations to BGV and parameter estimation.
- In Section 3, we describe the BGV scheme; the relation between polynomial products and the new algebraic structure is discussed in Section 3.3.
- In Section 4 we present our techniques for the parameter estimation, including the tools for the non-power-of-two framework. In particular, we prove our theoretical results (Theorems 2 and 3) which describe the underlying mathematical structure of the rings we work into.
- In Section 5 and Section 6, we study the noise growth in each operation and through the circuits formed by the operations.
- In Section 7, we present our results for non-power-of-two parameter estimation, and draw comparisons with the power-of-two instantiations.
- Finally, in Section 8, we draw our conclusions and propose future research directions.

2 Preliminaries

2.1 Notation

We begin by fixing some notation.

- $\mathbb{C}$ and $\mathbb{Q}$ are the complex and rational fields respectively, $\mathbb{Z}$ is the ring of integers, and for $a \in \mathbb{Z}_{>0}$ we let $\mathbb{Z}_a = \mathbb{Z}/a\mathbb{Z}$, and $[a] = \{0, 1, \ldots, a-1\}$.
- Integer modular reductions modulo odd numbers q are symmetric with respect to the origin: the notation $[x]_q$ refers to the representative of the class of x that is contained in $\left[-\lfloor \frac{q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor\right]$.
- For any ring R, R^* denotes the units of R; for $a, b \in \mathbb{N}$, $R^{a\times b}$ is the set of $a \times b$ matrices with elements in R.
- Coordinate vectors with respect to some basis are indicated by bold letters: e.g., $\mathbf{a} = (a_0, \ldots, a_{n-1})$ where each a_i lies in some ring, $[\mathbf{a}]_q$ indicates the vector $([a_0]_q, \ldots, [a_{n-1}]_q)$. $||\mathbf{a}||$ is shorthand for the infinity norm of $\mathbf{a}$.
- Given a vector $\mathbf{X} = (X_0, \ldots, X_{n-1})$ of random variables, $\mathrm{E}[\mathbf{X}] = (\mathrm{E}[X_0], \ldots, \mathrm{E}[X_{n-1}])$ is its expected value and $\mathrm{Var}(\mathbf{X})$ is the vector of variances; for a random vector $\mathbf{Y}$, $\mathrm{CovM}(\mathbf{X}, \mathbf{Y}) = (\mathrm{Cov}(X_i, Y_i))_{i,j=1,\ldots,n}$ is their covariance matrix.
- Given a distribution χ on some set S, $s \leftarrow \chi$ means sampling $s \in S$ according to χ, and this generalizes to vectors in a coefficient-wise fashion, where each coefficient is sampled independently. χ_s and χ_e will refer to the secret and error distributions for RLWE samples.
- $\Re(z)$ and $\Im(z)$ denote real and imaginary part of $z \in \mathbb{C}$.
- Given an integer r we call $\mathcal{R}_r = \mathcal{R}/(r\mathcal{R})$. We denote the plaintext and ciphertext moduli with t and q, respectively. The plaintext space is $\mathcal{R}_t = \mathbb{Z}_t[x]/(\Phi_m(x))$, while the ciphertext space is $\mathcal{R}_q = \mathbb{Z}_q[x]/(\Phi_m(x))$, where $\Phi_m(x)$ is the cyclotomic polynomial (see Section 2.2). Moreover, we set t and q coprime and q a chain of primes, such that

$$q = q_{L-1} = \prod_{j=0}^{L-1} p_j,$$

  where $p_j \equiv 1 \mod m$ [35]. The multiplicative depth M of the circuit determines the number of primes $L = M + 1$. Thus, for any level ℓ, we have $q_\ell = \prod_{j=0}^{\ell} p_j$.
- $\phi(\cdot)$ denotes Euler's totient function.

2.2 Mathematical Background

Cyclotomic polynomials For $m \in \mathbb{N}$, an m^{th} root of unity in a field F is any element $\zeta \in F$ such that $\zeta^m = 1$; if $\zeta^k \neq 1$ for any $k < m$ then ζ is called *primitive*. The set of primitive m^{th} roots of unity is $\{\zeta^i \ : \ i \in \mathbb{Z}_m^*\}$. Finally, the m^{th} cyclotomic polynomial $\Phi_m(x)$ is

$$\Phi_m(x) = \prod_{i\in\mathbb{Z}_m^*} (x - \zeta^i)$$

and it has degree $n = \phi(m)$. Let $m = \prod_{i=1}^{l} p_i^{\alpha_i}$ be a natural number, where p_i are distinct primes. Then the radical $\mathrm{rad}(m)$ is the product of its prime factors, namely, $\mathrm{rad}(m) = \prod_{i=1}^{l} p_i$.

Lemma 1. *[26] For any $m \in \mathbb{N}$ we have $\Phi_m(x) = \Phi_{\mathrm{rad}(m)}(x^{m/\mathrm{rad}(m)})$.*

This result implies that for $m = 2^s 3^t$ we have $\Phi_m(x) = x^n - x^{\frac{n}{2}} + 1$, where $s, t \geq 1$ and $n = m/3$.

The following result describes how cyclotomic polynomials factorize over finite fields. This factorization is a crucial finding with significant implications for polynomial multiplication algorithms.

Lemma 2. *[45, Theorem 2.47] For any $m \in \mathbb{N}$ the polynomial $\Phi_m(x)$ has $\phi(m)/d$ factors of same degree d over $\mathbb{F}_q$, where d is the multiplicative order of q modulo m.*

The quotient ring $K_m = \mathbb{Q}[x]/(\Phi_m(x))$ is the m^{th} cyclotomic field. This extension has degree $n = \phi(m)$ over the rationals.

Lemma 3. *[44, Chapter IV, Theorem 3] The ring of integers of $\mathbb{Q}(\zeta_m)$ is $\mathcal{R} = \mathbb{Z}[\zeta_m] = \mathbb{Z}[x]/(\Phi_m(x))$.*

Canonical embedding and norm The canonical embedding of a polynomial $a(x) \in K_m$ is the vector $\sigma(a(x)) = (a(\zeta^i) \ : \ i \in \mathbb{Z}_m^*)$. Ordering the roots appropriately we have $\sigma : K_m \to H$, with

$$H = \{(x_1, \ldots, x_n) \in \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2} : x_{s_1+i} = \overline{x_{s_1+s_2+i}} \text{ for } i = 1, \ldots, s_2\} \subset \mathbb{C}^n \ , \quad (1)$$

where s_1 is the number of real embeddings and s_2 is the number of conjugate pairs of embeddings of K_m in $\mathbb{C}$ [46]. The canonical embedding is a ring homomorphism; by identifying the conjugate pairs, we have $H \cong \mathbb{R}^{s_1+s_2}$. We recall that the infinity norm of a polynomial $a \in K$ is defined as $||a||_\infty = \max\{|a_i| : 0 \leq i \leq \phi(m) - 1\}$. The canonical norm $|| \cdot ||^{can}$ is the pull-back of the infinity norm via the canonical embedding σ, namely $||a||^{can} := ||\sigma(a)||_\infty$. It is sub-multiplicative: $\forall\ a, b \in K$

$$||ab||^{can} \leq ||a||^{can} ||b||^{can} \ . \quad (2)$$

The following two results establish a connection between the infinity norm and its canonical counterpart. For full proofs and a more extensive background, we refer to [27].

Lemma 4. *Let K be the m^{th} cyclotomic field, $\mathcal{R}$ be its ring of integers, and σ the canonical embedding of K. There exists is a constant c_m such that for any $\alpha \in \mathcal{R}$ we have*

$$||\alpha||_\infty \leq c_m ||\alpha||^{can}.$$

The constant c_m is called the ring's expansion factor and enjoys the following properties.

Lemma 5. *Let $m \geq 2$, then*

1. *for $r = \mathrm{rad}(m)$ we have $c_m \leq c_r$;*
2. *if m is odd then $c_{2m} = c_m$;*

3. for $m = p$ prime we have

$$c_p = \frac{2\sin(\pi/p)}{p(1-\cos(\pi/p))} \ .$$

A straightforward application of the properties above is that for $m = 2^s 3^t$, we can bound the value of c_m with

$$c_3 = \frac{2\sin(\pi/3)}{3(1-\cos(\pi/3))} = \frac{2}{\sqrt{3}} \tag{3}$$

from which we deduce the bound $c_m \leq 1.1547$.

Since we will focus on the canonical norm, we will omit the superscript from the notation in most of the paper and write $||\cdot||$ to indicate $||\cdot||^{can}$.

Probability theory We assume the reader is familiar with the basic properties of expected value and covariance, including the complex case; a basic reference including proofs for the following results is [40]. All distributions in this work are *centred*, meaning they are symmetric around the origin. This implies the mean μ of the distributions is always zero. We will use the following widely known distributions:

– the uniform distribution $\mathcal{T}$ over the ternary set $\{\pm 1, 0\}$, having variance $\sigma^2_{\mathcal{T}} = 2/3$;
– for an odd $q \in \mathbb{N}$, the uniform centered discrete distribution $\mathcal{U}_q$ over $\mathbb{Z}_q$ with variance is $q^2/12$;
– The continuous Gaussian distribution on $\mathbb{R}$ with variance σ^2, denoted as $\mathcal{N}_r = \mathcal{N}(0, \sigma^2)$, and its discretized version $\mathcal{DG}(0, \sigma^2)$ obtained by rounding to the closest integer/rational number.

In our work, we use $\chi_s = \mathcal{T}$.

A multivariate normal vector is defined as an affine transformation of a standard normal vector, that is a vector of independent Gaussian random variables with mean 0 and variance 1. We have the following equivalent definition.

Lemma 6. *A random vector $(X_0, \dots, X_{n-1})$ is Gaussian if and only if each linear combination over $\mathbb{R}$ of its components is a Gaussian random variable.*

Moreover, we have the following property.

Lemma 7. *If the components of a Gaussian random vector $(X_0, \dots, X_{n-1})$ are uncorrelated, then they are also independent.*

We also recall the statement of Lyapunov's Central Limit Theorem, to justify some theoretical results needed for our estimates. Let $\xrightarrow{\mathrm{d}}$ denote convergence in distribution.

Theorem 1. *(Lyapunov CLT) Let $X_0, X_1, \ldots, X_j, \ldots$ be a sequence of independent random variables each with mean μ_j and variance σ_j^2 both finite for each j, and let $s_n^2 = \sum_{j=0}^{n-1} \sigma_j^2$. Assume the existence of a strictly positive real number δ such that*

$$\lim_{n\to\infty} \frac{1}{s_n^{2+\delta}} \sum_{j=0}^{n-1} \mathrm{E}[|X_j - \mu_j|^{2+\delta}] = 0. \tag{4}$$

Then

$$\frac{1}{s_n} \sum_{j=0}^{n-1} (X_j - \mu_j) \xrightarrow{d} \mathcal{N}(0,1).$$

Gaussian sampling In order to obtain secure RLWE cryptosystems, one key aspect is the choice of a secure error distribution χ_e. The error polynomial $e \in \mathcal{R}_q$ must have a spherical Gaussian distribution in the canonical embedding. While for the power-of-two constructions, it is sufficient to use $\chi_e = \mathcal{DG}(0, \sigma^2)$ where $\sigma = 3.19$ and each component is independent, things get more complicated when $m \neq 2^k$ due to the geometry of the canonical embedding. In [29] the authors tackle this issue by showing an efficient way to sample error polynomials securely. They first sample an error polynomial $\bar{e}$ in the ring $\mathbb{Z}_q[x]/(\Theta(x))$, where $\Theta(x) = x^{m/2} + 1$, and then reduce $\bar{e}$ modulo $\Phi_m(x)$ to obtain an error polynomial $e \in \mathcal{R}_q$. Each coefficient of the polynomial modulo $\Theta(x)$ is sampled independently according to a Gaussian distribution of variance $m\sigma^2$, where $\sigma = 3.19$. For every $m \neq 2^k$ we will denote this distribution by χ_e. By looking closely at the reduction modulo $\Phi_m(x)$, we see that each coefficient of e is the sum of two independent coefficients of $\bar{e}$; for this reason, its variance will be $V_e = 2m\sigma^2$.

Lattices We assume the reader is familiar with the basic definitions concerning lattices.

The BGV cryptosystem relies on the hardness of the LWE [56] and RLWE [46] problems; we will focus on the latter version as it is more efficient. This problem is based on the RLWE distribution, obtained as follows: sample uniformly at random $a \in \mathcal{R}_q$, then an RLWE sample is $(a, a \cdot s + e)$ where s and e are sampled from two distributions χ_s and χ_e. The search and decision versions of the RLWE (S-RLWE and D-RLWE respectively) are then defined as follows:

Definition 1. *(S-RLWE) Recover a fixed secret s from a given number of* RLWE *samples with a non-negligible advantage.*

Definition 2. *(D-RLWE) For a fixed secret s, distinguish with non-negligible advantage between a certain number of independent* RLWE *samples and independent uniform samples.*

Under certain assumptions, these problems are as hard as some well studied lattice problems such as GAPSVP or SIVP [56,55], and it is widely believed that quantum computers have no significant advantage over classical ones in solving them [54].

3 The BGV Scheme

In this section, we present the BGV scheme [13]. BGV functionalities can be divided into two main categories: the basic encryption scheme, including key generation, encryption and decryption, and the homomorphic operations.

3.1 Basic encryption scheme

The three basic algorithms of BGV are as follows:

- Key generation (KeyGen(λ)): sample $s \leftarrow \chi_s$, $a \leftarrow \mathcal{U}_{q_L}$ and $e \leftarrow \chi_e$ in $\mathcal{R}q_L$, output the secret key $\mathsf{sk} = s$ and the public key $\mathsf{pk} = (b, a) = [(-a \cdot s + te, a)]_{q_l}$;
- Encryption ($\mathrm{Enc}_{\mathsf{pk}}(m)$): given a plaintext $m \in \mathcal{R}_t$ and the public key $\mathsf{pk} = (b, a)$, sample $u \leftarrow \chi_s$, $e_0, e_1 \leftarrow \chi_e$ and output $\mathfrak{c} = (\mathbf{c}, l, \nu)$ where the ciphertext is

$$\mathbf{c} = (c_0, c_1) = [(b \cdot u + te_0 + m, a \cdot u + te_1]_{q_L}, \tag{5}$$

  and l and ν are quantities related to noise management, whose role we explain below. The triad $\mathfrak{c}$ is called the *extended ciphertext*. It is worth noting that the ciphertext $\mathbf{c} = (c_0, c_1)$ can be seen as the polynomial $c_0 + c_1 x \in \mathcal{R}_{q_l}[x]$.
- Decryption ($\mathrm{Dec}_{\mathsf{sk}}(\mathfrak{c})$): given the secret key sk and the ciphertext $\mathbf{c} = (c_0, c_1)$ output

$$m = [[c_0 + c_1 \cdot s]_{q_l}\ ]_t\ .$$

The plaintext modulus t is chosen accordingly with the specific purpose of the implementation of BGV (e.g., $t = 64$). The plaintext $m \in \mathcal{R}_t$ is treated as a uniformly random polynomial with independent coefficients for the purposes of our analysis.

The first part of the decryption can be seen as the polynomial evaluation of $c_0 + c_1 x \in \mathcal{R}_{q_l}[x]$ in the secret key s. For this reason, we will often write $c(s)$ in the place of $c_0 + c_1 \cdot s$; this notation extends to triples of polynomials in an obvious way, namely $c(s) = c_0 + c_1 \cdot s + c_2 \cdot s^2$, where s^2 stands for the repeated product of s as a polynomial (see Section 3.2).
In the extended ciphertext, l is the current multiplicative level, while $\nu = [c(s)]_{q_l}$ is the *critical quantity*. For a fresh ciphertext, we have

$$\nu = m + t(e \cdot u + e_1 \cdot s + e_0) = m + tE\ , \tag{6}$$

and this quantity increases through homomorphic operations [22]. The importance of ν lies in the fact that as long as its coefficients do not *wrap around* modulo q_l, the decryption is correct. For this reason, we need to study the canonical norm of the critical quantity $||\nu|| = ||\nu||^{can}$ (Section 2.2), called *noise*.

This work considers the Residue Number System (RNS) representation of the ciphertext space. Since the modulus $q = p_0 \dots p_{L-1}$ is the product of distinct primes, applying the Chinese Remainder Theorem, we get the isomorphism

$$\mathcal{R}_q \cong \mathcal{R}_{p_0} \times \dots \times \mathcal{R}_{p_{L-1}}\ . \tag{7}$$

This representation allows the use of native data types for integers because the p_i can be chosen to fit into 32 or 64 bits. When using BGV with the RNS representation, we need to change the modulus of the ring in use, switching from $\mathcal{R}_A$ to $\mathcal{R}_B$, where $A = a_0 \cdots a_k$, $B = b_0 \cdots b_{k'}$ are the two product decompositions used for RNS. For this purpose, we need a Fast Base Extension (FBE) algorithm [32]. Namely, if $\mathbf{a} \in \mathcal{R}_{a_0} \times \ldots \times \mathcal{R}_{a_k}$, then

$$\mathrm{FBE}(\mathbf{a}, A, B) = \left(\left[\sum_{j=0}^{k} \left[\mathbf{a} \left(\frac{A}{a_j} \right)^{-1} \right]_{a_j} \frac{A}{a_j} \right]_{b_i} \right)_{i=0,\ldots,k'} . \tag{8}$$

3.2 Homomorphic operations

We introduce the three homomorphic ring operations (addition, multiplication and constant multiplication) and two key subroutines (key and modulus switching).

– The addition $\mathsf{Add}(\mathfrak{c}, \mathfrak{c}')$ is defined as

$$\mathsf{Add}(\mathfrak{c}, \mathfrak{c}') = (([c_0 + c'_0]_{q_l}\ ,\ [c_1 + c'_1]_{q_l}), l, \nu + \nu') = ([\mathbf{c} + \mathbf{c}']_{q_l}, l, \nu_{\mathsf{Add}}). \tag{9}$$

The critical quantity ν_{Add} is $\nu + \nu'$ since we have

$$[(c + c')(s)]_{q_l} = [[c(s)]_{q_l} + [c'(s)]_{q_l}]_{q_l} = [m + tE + m' + tE']_{q_l}\ .$$

– The ciphertext multiplication $\mathsf{Mul}(\mathfrak{c}, \mathfrak{c}')$ outputs

$$\begin{aligned} \mathsf{Mul}(\mathfrak{c}, \mathfrak{c}') &= ((c_0 \cdot c'_0\ ,\ c_0 \cdot c'_1 + c_1 \cdot c'_0\ ,\ c_1 \cdot c'_1), l, \nu \cdot \nu') \\ &= ((c''_0, c''_1, c''_2), l, \nu_{\mathsf{Mul}}) \end{aligned} \tag{10}$$

where $\mathbf{c}'' = (c''_0, c''_1, c''_2)$ represents the coefficients vector of the product among the two polynomials $c(x)$ and $c'(x)$ in $\mathcal{R}_{q_l}[x]$ (which has degree 2, and hence nonzero 3 coefficients). This means that to recover the message hidden in $\mathbf{c}''$, we would actually need to calculate

$$[\ [c''(s) = c''_0 + c''_1 \cdot s + c''_2 \cdot s^2]_{q_l}\ ]_t.$$

However, instead of using this *special* decryption, we will use a relinearization procedure to convert the ciphertext $\mathbf{c}'' = (c''_0, c''_1, c''_2) \in \mathcal{R}^3_{q_l}$ back to a ciphertext $\bar{\mathbf{c}} = (\bar{c}_0, \bar{c}_1) \in \mathcal{R}^2_{q_l}$ (see Equation (15)). Since we have

$$\begin{aligned} [c''(s)]_{q_l} &= [c(s) \cdot c'(s)]_{q_l} = [[c(s)]_{q_l} \cdot [c'(s)]_{q_l}]_{q_l} \\ &= [(m + tE)(m' + tE')]_{q_l}\ , \end{aligned}$$

the critical quantity for the Mul operation is $\nu_{\mathsf{Mul}} = \nu\nu'$. We point out that in this operation, the noise growth is multiplicative, which is the worst case among basic operations.

– The constant multiplication $\mathsf{ConstMul}(\alpha, \mathfrak{c})$ defined as

$$\mathsf{ConstMul}(\alpha, \mathfrak{c}) = ((\alpha \cdot c_0\ ,\ \alpha \cdot c_1), l, \alpha \cdot \nu) = (\alpha \cdot \mathbf{c}, l, \nu_{\mathsf{ConstMul}})\ , \tag{11}$$

where $\alpha \in \mathcal{R}_t$. The critical quantity is correct because

$$[\alpha \cdot c(s)]_{q_l} = [\alpha \cdot [c(s)]_{q_l}]_{q_l} = [\alpha \cdot (m + tE)]_{q_l}\ .$$

The main novelty separating the BGV scheme from its predecessors is the *modulus switching*. This operation allows sacrificing one or more of the primes p_i that compose the ciphertext moduli q_l to obtain a noise reduction.

– Let $\mathfrak{c} = (\mathbf{c}, l, \nu)$ be the extended ciphertext and let $l' = l - \kappa$ be a target level, where κ is a positive integer. Then

$$\mathsf{ModSw}(\mathfrak{c}, l') = \left(\mathbf{c}' = \left[\frac{q_{l'}}{q_l} (\mathbf{c} + \boldsymbol{\delta}) \right]_{q_{l'}}, l', \nu_{\mathsf{ModSw}} \right)\ .$$

The polynomial $\boldsymbol{\delta}$ is a correction term computed as

$$\boldsymbol{\delta} = t[-t^{-1}\mathbf{c}]_{q_l/q_{l'}} = t[(t^{-1}c_0, t^{-1}c_1)]_{q_l/q_{l'}}\ , \tag{12}$$

and it is formulated (i) to affect the errors, since it is a multiple of t, and (ii) to adjust the ciphertext to be divisible by $q_l/q_{l'}$, indeed $\boldsymbol{\delta} \equiv -\mathbf{c} \bmod q_l/q_{l'}$. In this way, it allows to descend in the moduli ladder from q_l to $q_{l'}$. The formal proof of why this procedure reduces the noise is in [13, Lemma 5]. If we consider only one-step modulus switching, i.e., $\kappa = 1$ and $l' = l - 1$, then $q_l/q_{l'} = 1/p_l$ and we have

$$[c'(s)]_{q_{l'}} = c'(s) - kq_{l'} = \frac{c(s) + kq_l + \delta(s)}{p_l} - kq_{l'} = \frac{c(s) + \delta(s)}{p_l}\ .$$

Note that we actually decrypt to the plaintext $p_l^{-1} m \bmod t$, but we can multiply a plaintext by p_l either before encryption or after decryption. This issue does not exist for $p_l \equiv 1 \bmod t$, but finding such p_l can be difficult in practice.
Hence, the critical quantity for modulus switching is

$$\nu_{\mathsf{ModSw}} = \frac{\nu + \delta(s)}{p_l}\ . \tag{13}$$

The last procedure that we are going to analyze is the subroutine called *key switching*. This procedure is used for (i) reducing the degree of a ciphertext polynomial, usually the output of multiplication, or (ii) changing the key after a ciphertext rotation.

Rotations are special ring automorphisms used to operate with packed plaintexts and improve efficiency [36,34]. To rotate an encrypted vector, we apply a permutation rot on the ciphertext as $\mathrm{rot}(c(x)) = \mathrm{rot}(c_0) + \mathrm{rot}(c_1)\,\mathrm{rot}(x)$ Thus, after a rotation, the ciphertext can be decrypted computing $\mathrm{rot}(c(s))$. This means that to recover the message hidden in $\mathrm{rot}(c(x))$, we would actually need to calculate the rotation of the secret key. Thus, we apply key switching to convert the ciphertext term $\mathrm{rot}(c_1)\,\mathrm{rot}(s)$ to a polynomial $c_0^{\mathsf{ks}} + c_1^{\mathsf{ks}} \cdot s$. In a similar way, for multiplication, we convert the ciphertext term $c_2'' \cdot s^2$ to a polynomial $c_0^{\mathsf{ks}} + c_1^{\mathsf{ks}} \cdot s$. In the following, we will only analyze multiplication and more specifically, we will output $\mathbf{c}' = (c_0 + c_0^{\mathsf{ks}}, c_1 + c_1^{\mathsf{ks}})$ and denote the ciphertext term we want to remove by c_2. This also covers rotations as one only has to consider the term we want to remove as c_1 and an output of $(c_0 + c_0^{\mathsf{ks}}, c_1^{\mathsf{ks}})$. Intuitively, the basic idea of this method is to using the encryption of s' (that can be s^2 or $\mathrm{rot}(s)$) under s [49]. Namley, $\mathrm{Enc}_s(s') = (\beta, -\alpha) = (-us + te + s', u + te_1) \approx (s' + \alpha s, -\alpha)$. Thus, $s' \approx \beta - \alpha s$ and then for the multiplication we get that the extended ciphertext $c_0'' + c_1'' s + c_2'' s^2 = c_0'' + c_1'' s + c_2''(\beta - \alpha s)$ becomes a *normal* ciphertext $\tilde{c}_0 + \tilde{c}_1 s$ encrypting the same plaintext. Similarly, for the rotation, we have that $\mathrm{rot}(c_0) + \mathrm{rot}(c_1)\,\mathrm{rot}(s) = \tilde{c}_0 + \tilde{c}_1 s$. For this reason, we treat the noise coming from rotations as if it was coming from key switching.

The key switching procedure can be divided into two parts: a key generation ($\mathsf{KeySwGen}$) that *somehow* encrypts s^2 under s itself (notice the similarity between Equation (14) and Equation (5)) and the actual key switching operation (KeySw).

- The key generation takes as input s and s^2, samples $a \leftarrow \mathcal{U}_{q_l}$ and $e \leftarrow \chi_e$ and outputs

$$\mathsf{KeySwGen}(s, s^2) = \mathbf{ks} = (\mathsf{ks}_0, \mathsf{ks}_1) = [(-a \cdot s + te + s^2, a)]_{q_l} \ . \tag{14}$$

- the key switching operation takes as input an extended ciphertext $\mathfrak{c} = (\mathbf{c}, l, \nu) = ((c_0, c_1, c_2), l, \nu)$ and the relative key switching key $\mathbf{ks} = (\mathsf{ks}_0, \mathsf{ks}_1)$, computes

$$\mathbf{c}' = (c_0', c_1') = [(c_0 + c_2 \cdot \mathsf{ks}_0, c_1 + c_2 \cdot \mathsf{ks}_1)]_{q_l}$$

  and outputs

$$\mathsf{KeySw}(\mathbf{ks}, \mathfrak{c}) = \mathfrak{c}' = (\mathbf{c}', l, \nu_{\mathsf{KeySw}}) \ . \tag{15}$$

The critical quantity after this operation is $\nu_{\mathsf{KeySw}} = \nu + tc_2 \cdot e$. Unfortunately, if we tried to compute $||\nu_{\mathsf{KeySw}}||$, even after only one homomorphic operation has been performed, it becomes evident that the noise growth introduced by the term $tc_2 \cdot e$ in the critical quantity is too big, as c_2 is pseudorandom. Several variations of the KeySw procedure have been developed to effectively address this issue, aiming to control the growth of noise introduced during computations. We focus on the Hybrid variant presented in [35], called so because it is a mix of the BV [14] and the GHS [35] variants. From the former we need the following decompositions: let $b \in \mathbb{N}$ be a basis, then for $k = \lfloor \log_b q_l \rfloor + 1$ and any $\alpha \in \mathcal{R}_{q_l}$, if we define

$$D_b(\alpha) = ([\alpha]_b, [\lfloor \alpha/b \rceil]_b, [\lfloor \alpha/b^2 \rceil]_b, \ldots, [\lfloor \alpha/b^{k-1} \rceil]_b)$$
$$P_b(\alpha) = ([\alpha]_{q_l}, [b\alpha]_{q_l}, [b^2\alpha]_{q_l}, \ldots, [b^{k-1}\alpha]_{q_l}) \ .$$

Then for any $\alpha, \beta \in \mathcal{R}_q$ we obtain $\langle D_b(\alpha), P_b(\beta)\rangle = \alpha \cdot \beta$ [42]. The GHS variant instead limits the noise growth by performing the key switching with respect to a bigger ciphertext modulus and then going back to the original q_l via modulus switching. A number C coprime with q_l is chosen, and the key switching takes place in $\mathcal{R}_{Q_l}$ where $Q_l = q_l C$. Then, the Hybrid key switching is performed as follows: with the above notations, the key generation is given by

$$\mathsf{KeySwGen}^{\mathrm{Hybrid}}(s, s^2) = \mathsf{ks}^{\mathrm{Hybrid}} = [(-\mathbf{a} \cdot s + t\mathbf{e} + CP_b(s^2), \mathbf{a})]_{Q_l}$$

and the new ciphertext is computed in two steps: first, let

$$\mathbf{c}' = [(Cc_0 + \langle D_b(c_2), \mathsf{ks}_0^{\mathrm{Hybrid}}\rangle, Cc_1 + \langle D_b(c_2), \mathsf{ks}_1^{\mathrm{Hybrid}}\rangle)]_{Q_l};$$

and then set $\boldsymbol{\delta} = t[-t^{-1}c']_C$ and modulus switch back to q_l:

$$\mathbf{c}'' = \left[\frac{\mathbf{c}' + \boldsymbol{\delta}}{C}\right]_{q_l}$$

Finally, the output of the Hybrid key switching is

$$\mathsf{KeySw}^{\mathrm{Hybrid}}(\mathsf{ks}^{\mathrm{Hybrid}}, \mathfrak{c}) = (\mathbf{c}'', l, \nu_{\mathsf{ks}}^{\mathrm{Hybrid}}) \tag{16}$$

with critical quantity given by putting together the BV and GHS ones: we set

$$\nu_{\mathsf{KeySw}}^{\mathrm{Hybrid}} = \nu + \frac{t\langle D_b(c_2), \mathbf{e}\rangle + \delta(s)}{C} \; . \tag{17}$$

The Hybrid key switching achieves better efficiency than the BV and better noise management than GHS, and, for this reason, it is the preferred one when it comes to implementations [37].

3.3 Impact of the algebraic structure on polynomial multiplication

The BGV cryptosystem can be implemented over any cyclotomic polynomial ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(\Phi_m(x))$. The issue arises, though, when one wants to multiply two polynomials efficiently, which is the main computational bottleneck of FHE and, more in general, of lattice-based cryptosystems. One of the most elegant solutions to this problem is the Number Theoretic Transform (NTT), the discrete counterpart of the Fast Fourier Transform. When the cyclotomic index m is a power-of-two, the NTT can be implemented using the Cooley-Tukey/radix-2 butterfly operations [21], yielding an implementation-friendly and straightforward solution. The absence of similar methods in non-power-of-two settings has remarkable consequences: for example, the library HElib implements a special algorithm for non-power-of-two instantiations, mixing Bluestein's algorithm [10] with NTT [36]. This results in a heavy computational burden, and alternatives are being explored that include hardware acceleration [28]. For cyclotomic rings where the index has the form $m = 2^s 3^t$, a new NTT algorithm has been proposed in [48] that exploits the algebraic structure of the ring in a way that resembles very closely the radix-2 butterfly operations. We recap briefly this construction to highlight its portability into our context, since the authors proposed it specifically for $m = 2304$.

Let us consider the cyclotomic ring $\mathcal{R}_q$ with index $m = 2^s 3^t$ and $q = 1 \mod m$. Then, given a sixth primitive root of unity ζ, we have the factorization

$$\Phi_m(x) = x^n - x^{n/2} + 1 = (x^{n/2} - \zeta)(x^{n/2} - \zeta^5) \ .$$

Using the fact that $\zeta^5 = \bar{\zeta} = 1 - \zeta$, the corresponding radix-6 butterfly operation requires only 1 extra addition w.r.to the Cooley-Tukey, meaning an NTT layer requires only $n/2$ extra additions (see Figure 1). As described in [48], after this

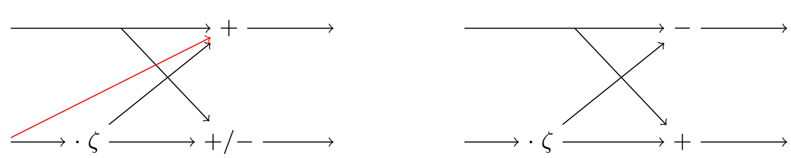

Fig. 1. Butterfly operation of [48] (left) vs Cooley-Tukey's [21] (right); in red the extra addition.

step we can proceed with $s - 1$ Cooley-Tukey layers (extracting the square roots of ζ and $1 - \zeta$, and then the fourth roots and so on), obtaining 2^s rings of degree 3^{t-1}. These have the same cost as the corresponding power-of-two counterparts. We will consider especially the case of $t = 2$, where at this point one can choose whether to compute the product of 2^s polynomials of degree less than 3 (as done in [48]) or to use radix-3 NTT layer [7]. The latter option exploits the full factorization of $\Phi_m(x)$ (see Lemma 2), giving a result that is more similar to the DCRT representation used e.g. in HElib; it requires 2 multiplications and 11 additions per polynomial [7], and there are $n/3$ polynomials. To compare the two algorithms, we keep track of the required additions and multiplications over $\mathbb{Z}_q$ (first part of Table 1). The cost of an NTT algorithm is given by the sum of the costs of its layers:

- $u - 1$ radix-2 layers for $m = 2^u$;
- 1 radix-6, $s - 1$ radix-2 and 1 radix-3 for the non-power-of-two.

A fair comparison then can be made by considering e.g. the two powers of 2 that are closest to a non-power-of-two instantiation: this amounts to choosing $u = s + 2$ and $u = s + 3$, as then we have $2^{s+2} < 2^s \cdot 3 < 2^{s+3}$. By plugging these values in the above equations, we get the results in the second part of Table 1. We can see that the non-power-of-two incurs in an overhead in terms of additions, as $s + 25/6 > s + 1, s + 2$, while as far as multiplications go, its performance is aligned with the power-of-two counterpart, as $s + 1 < s + 4/3 < s + 2$. Because multiplications over $\mathbb{Z}_q$ are much more expensive than additions, this tells us

m	$n=\phi(m)$	additions	multiplications
2^u	2^{u-1}	$(u-1)n$	$(u-1)\frac{n}{2}$
$2^s \cdot 3^2$	$2^s \cdot 3$	$\underbrace{\frac{3}{2}n}_{\text{radix-6}} + \underbrace{(s-1)n}_{\text{radix-2 layers}} + \underbrace{\frac{11}{3}n}_{\text{radix-3}}$	$\underbrace{\frac{n}{2}}_{\text{radix-6}} + \underbrace{(s-1)\frac{n}{2}}_{\text{radix-2 layers}} + \underbrace{\frac{2}{3}n}_{\text{radix-3}}$
	2^{s+2}	$(s+1)n$	$(s+1)\frac{n}{2}$
	$2^s \cdot 3$	$(s+\frac{25}{6})n$	$(s+\frac{4}{3})\frac{n}{2}$
	2^{s+3}	$(s+2)n$	$(s+2)\frac{n}{2}$

Table 1. Comparisons between NTT algorithms.

that the non-power-of-two NTT algorithm has roughly the same computational impact as its power-of-two counterpart.

We point out that this algorithm is only available when cyclotomic index m has the specified non-power-of-two form, as it relies on the algebraic relations between the roots of unity to build a fast radix-6 butterfly operation.

4 Theoretical results for noise estimation

Since the encryption process in BGV involves randomization and we need to estimate the canonical norm of the ciphertexts, we focus on estimating the canonical norm of random polynomials.

Remark 1. In this section, and especially in Theorem 2 and Theorem 3, by random polynomial we always mean a polynomial whose coefficients are sampled independently from some distribution. For a discussion of this assumption (also in light of Theorem 3), see Section 4.4.

4.1 Advantages of the canonical norm for BGV: comparison with other norms

Over the past few years, several methods to compute a bound for the error have been proposed for power-of-two BGV, from the Euclidean [13] and infinity norms [31,5,42] to the canonical norm (called *worst-case* analysis) [34,35,39,41,52,36].

The most promising lines of research are those developing new techniques to replace the canonical norm worst-case analysis. Several recent papers propose alternative approaches like *average-case* analysis, where the coefficients of the polynomial error are treated as random variables. This approach was first employed in the TFHE scheme [18], and then has been studied for the CKKS [23], the BGV [53], and the BFV [8] schemes.

Interest in this method grew due to a recognized discrepancy between the estimates based on worst-case technique and experimental data, as highlighted in [24]. The introduction of the average-case approach, as seen in [8,25], offers a potential resolution to these disparities, indeed, with this method, it is possible to compute a tight *probabilistic* upper bound by considering the Gaussian distribution of the error coefficients, their mean, and variance. This topic is fascinating, and any progress makes FHE easier to deploy in real-life applications.

However, this approach is limited to cases where the ciphertexts are computed independently and it holds accurately for TFHE [19,20] and BFV [8]. The differences among the FHE schemes are as follows. In TFHE, as pointed out in [24], gate bootstrapping enables the implementation of elementary operations on a linear combination of ciphertexts. Therefore, due to its linearity, the noise in TFHE ciphertexts can be modelled as subgaussian, allowing for a straightforward analysis of the variances. On the other hand, in BGV, BFV, and CKKS, the noise grows non-linearly during multiplication, which makes the error analysis more intricate. While the authors of [8] provide accurate bounds for BFV, in the case of the BGV [53] and the CKKS [23] schemes, the heuristics tend to *underestimate* noise growth in many scenarios. This issue arises due to the assumption of independence of the noise coefficients, which leads to *imprecise bounds* [23,53,8].

In light of this, we want to emphasize that applying the infinity [42] or canonical norm [39,22,24], results in looser but safer bounds. The main difference regarding the bounds between these two norms is that if we consider two polynomials $a, b \in \mathcal{R}$, the infinity norm of their product is bounded by $||ab||_\infty \leq \delta_R ||a||_\infty ||b||_\infty$, where $\delta_\mathcal{R}$ is the *expansion factor* depending on $\mathcal{R}$. In contrast, as proven in [46], the canonical norm does not have this *expansion* for multiplication, achieving tighter bounds and offering an improvement over the infinity norm. Moreover, the canonical embedding provides a better and more precise method for managing the geometry of cyclotomic rings (for more details, see [46,47]).

For these reasons, in our study, we use the canonical norm, which provides the most precise and correct bound according to the current state-of-the-art for the BGV scheme. This approach also enables us to make more accurate and clear comparisons between the power-of-two and non-power-of-two cases.

4.2 Canonical norm of random polynomials

The main result in this section is Theorem 2, stating a probabilistic bound on the canonical norm of a random polynomial that depends on the variance of its coefficients. Previous works stated similar bounds (e.g. [24,39]) without proof, and we provide a comprehensive one.

Theorem 2. *Let $a(x) = \sum_{i=0}^{n} a_i x^i \in \mathcal{R}$ be a random polynomial whose coefficients are zero-mean, identically distributed with finite variance V_a. Furthermore assume there is some $\delta > 0$ and a constant $\gamma_1 \in \mathbb{R}_{>0}$ such that for any j*

$$\mathrm{E}[|a_j|^{2+\delta}] < \gamma_1. \tag{18}$$

Then for any primitive m^{th} root of unity $\zeta = \cos(\alpha) + i\sin(\alpha) \in \mathbb{C}$, the distribution of $a(\zeta)$ is well approximated by centered Gaussian distribution with variance nV_a.

Proof. Note that by the independence of the coefficients of $a(x)$, we have

$$\mathrm{E}[a(\zeta)] = \sum_j \mathrm{E}[a_j]\zeta^j = 0 \text{ and } \mathrm{Var}(a(\zeta)) = \mathrm{E}[a(\zeta)\overline{a(\zeta)}]$$

Since the product of a root of unity and its conjugate is 1, then

$$\mathrm{Var}(a(\zeta)) = \sum_{j_1,j_2=0}^{n-1} \mathrm{E}[a_{j_1}a_{j_2}\zeta^{j_1}\overline{\zeta^{j_2}}] = \sum_{j_1,j_2=0}^{n-1} \mathrm{Cov}(a_{j_1},a_{j_2})\zeta^{j_1}\overline{\zeta^{j_2}} = nV_a.$$

We show that $a(\zeta)$ has a Gaussian distribution. To prove that, we consider $a(\zeta)$ as a random vector $Z = (X, Y) = (\Re(a(\zeta)), \Im(a(\zeta)))$ over $\mathbb{C} \cong \mathbb{R}^2$ and, by Lemma 6, we prove that it is a Gaussian vector. The trigonometric expressions of X and Y are

$$X = \Re(a(\zeta)) = \sum_{j=0}^{n-1} a_j \cos(\alpha j) \quad \text{and} \quad Y = \Im(a(\zeta)) = \sum_{j=0}^{n-1} a_j \sin(\alpha j)$$

and for any given $\eta, \rho \in \mathbb{R}$ we have

$$\eta X + \rho Y = \eta \sum_{j=0}^{n-1} a_j \cos(\alpha j) + \rho \sum_{j=0}^{n-1} a_j \sin(\alpha j) = \sum_{j=0}^{n-1} (\eta\cos(\alpha j) + \rho\sin(\alpha j))a_j\ .$$

We can approximate the distribution of $\eta X + \rho Y$ using Lyapunov's CLT (Theorem 1), treating the coefficients $\eta\cos(\alpha j) + \rho\sin(\alpha j)$ as constants and hence applying the theorem to the random variables $W_j = (\eta\cos(\alpha j) + \rho\sin(\alpha j))a_j$, which have mean 0 and variance

$$\mathrm{Var}(W_j) = (\eta\cos(\alpha j) + \rho\sin(\alpha j))^2\,\mathrm{Var}(a_j)$$

This implies

$$s_n^2 = \sum_{j=0}^{n-1} \mathrm{Var}(W_j) = \sum_{j=0}^{n-1} (\eta\cos(\alpha j) + \rho\sin(\alpha j))^2\, V_a$$

and this quantity can be bounded from below by considering that the equation

$$\eta\cos(x) + \rho\sin(x) = 0$$

has at most two solutions in $[0, 2\pi]$ for any η, ρ. This tells us that the set $J = \{j \in [n] \,:\, \eta\cos(\alpha j) + \rho\sin(\alpha j) = 0\}$ has cardinality at most two. Then for all $j \in [n] \setminus J$ we have $(\eta\cos(\alpha j) + \rho\sin(\alpha j))^2 > 0$, implying that there exists $\gamma_2 \in \mathbb{R}_{>0}$ such that

$$\gamma_2 < (\eta\cos(\alpha j) + \rho\sin(\alpha j))^2$$

for these values of j. Since we have $\eta\cos(\alpha j) + \rho\sin(\alpha j) = 0$ for any $j \in J$, we get the bound $s_n^2 > (n-2)\gamma_2 V_a$. Let $\delta > 0$ be such that $\mathrm{E}[|a_j|^{2+\delta}] < \gamma_1$, then for each j we can bound $|(\eta\cos(\alpha j) + \rho\sin(\alpha j))|^{2+\delta} < \gamma_3$ for some $\gamma_3 \in \mathbb{R}_{>0}$ and get

$$\begin{aligned}\sum_{i=0}^{n-1} \mathrm{E}[|(\eta\cos(\alpha j) + \rho\sin(\alpha j))a_j|^{2+\delta}] &= \sum_{i=0}^{n-1} |(\eta\cos(\alpha j) + \rho\sin(\alpha j))|^{2+\delta}\, \mathrm{E}[|a_j|^{2+\delta}] \\ &< n\gamma_1\gamma_3.\end{aligned}$$

Now we are ready to check that Lyapunov's condition (Equation (4)) holds: we have

$$\frac{1}{s_n^{2+\delta}} \sum_{i=0}^{n-1} \mathrm{E}[|(\eta\cos(\alpha j) + \rho\sin(\alpha j))a_j|^{2+\delta}] \leq \frac{n\gamma_1\gamma_3}{((\gamma_2(n-2))^{\frac{1}{2}})^{2+\delta}} = \mathcal{O}\left(\frac{1}{n^{\delta/2}}\right)$$

and hence

$$\lim_{n\to\infty} \frac{1}{s_n^{2+\delta}} \sum_{i=0}^{n-1} \mathrm{E}[|(\eta\cos(\alpha j) + \rho\sin(\alpha j))a_j|^{2+\delta}] = 0.$$

Then by Lyapunov's CLT, the distribution of $\eta X + \rho Y$ is very well approximated by a Gaussian $\forall\ \eta, \rho \in \mathbb{R}$. But then it follows from Lemma 6 that the distribution of (X, Y) is well approximated by a Gaussian random vector, meaning the random vector Z is approximately Gaussian.

If we take a random polynomial $a \in \mathcal{R}_q$, the condition in Equation (18) is easily satisfied since the distributions of the coefficients are bounded. Then we can apply Theorem 2 to get the following result.

Corollary 1. *Let $a \in \mathcal{R}_q$ be a random polynomial with coefficient variance V_a and ζ be a primitive m^{th} root of unity, then the distribution of $a(\zeta)$ is well approximated by a centred Gaussian distribution with variance nV_a.*

We can use this result to derive a bound on $||a||^{can}$ in the following way. Given a complex centred Gaussian random variable $Z = (X, Y)$ with variance V_Z, we have that $|Z|$ follows a Hoyt distribution [38]. As a consequence, for any $B \in \mathbb{R}_{>0}$ we have $|Z| > B$ with probability $\mathrm{erf}(-B/\sqrt{2}V_Z) \approx 1 - e^{B^2/2V_Z^2}$. In our case then $Z = a(z)$, $V_Z = nV_a$; let $B = D\sqrt{V_Z}$ for some integer D, then we get

$$P(|a(z)| \geq D\sqrt{V_Z}) \approx e^{-D^2/2}\ .$$

This immediately translates into a bound on the canonical norm of a: by definition $||a||^{can} = \max|a(\zeta)|$ with ζ ranging among primitive m^{th} roots of unity. It follows that the inequality

$$||a||^{can} < D\sqrt{nV_a} \tag{19}$$

holds with probability $(1 - e^{-D^2/2})^n \approx 1 - ne^{-D^2/2}$, meaning it fails with negligible probability. In our work, we use $D = 6$.

4.3 Variance of random polynomials

Since we can estimate the canonical norm of a random polynomial using its variance, we study the behaviour of the variance with respect to ring operations. For the sum of random polynomials and the multiplication for a constant in $\mathbb{Z}_q$, the results do not differ from the power-of-two case (e.g. see [24,52]) and are widely known.
What changes in our new case is the coefficient variance of the product of two random polynomials $c(x) = a(x)b(x)$. In the power-of-two case, in [39], it is shown that $V_c = nV_aV_b$, where n is the degree of the ring $\mathcal{R}_q$. Finding a similar result for the case where the cyclotomic index is $m = 2^s3^t$ is not trivial because the reduction modulo $\Phi_m(x)$ is more complex. In fact for $m = 2^s$ we have $\Phi_m(x) = x^n + 1$, while $m = 2^s3^t$ implies $\Phi_m(x) = x^n - x^{n/2} + 1$ (where in both cases $n = \phi(m)$), and this affects the computations. In [48, Section 3.2], the authors give a bound on the variance by making some considerations on the behaviour of the product, finding

$$V_c \leq \frac{3}{2}nV_aV_b \; . \tag{20}$$

We show an alternative way to obtain the same bound, with the difference that we compute the full covariance matrix of the vector of coefficients of the product $c(x)$. This is a generalization of the result in [48], as we compute all the variances exactly and not only an upper bound, giving deep insight on random polynomials' behaviour. These computations only concern the reduction modulo the cyclotomic polynomial, not the one modulo q; hence, we consider the product of two random polynomials in $\mathcal{R}$ instead of $\mathcal{R}_q$. The formal way to compute such a product is in two steps: let

$$a(x) = \sum_{i=0}^{n-1} a_ix^i,\; b(x) = \sum_{i=0}^{n-1} b_ix^i \in \mathbb{Z}[x]/(\Phi_m(x)) = \mathcal{R} \; .$$

First we consider a and b as if they were in $\mathbb{Z}[x]$, and multiply them to obtain

$$g(x) = \sum_{i=0}^{2n-1} g_lx^l = a(x)b(x) \in \mathbb{Z}[x] \; .$$

After this, we compute $c(x)$ by reducing $g(x)$ modulo $\Phi_m(x)$. General formulas for the coefficients of c can be computed, yielding

$$c_k = \begin{cases} g_k - g_{n+k} - g_{n+n/2+k} & k = 0, \dots, n/2-2 \\ g_k - g_{n+k} & k = n/2-1 \\ g_k + g_{n/2+k} & k = n/2, \dots, n-1 \end{cases}$$

which expands to

$$c_k = \begin{cases} \sum_{j=0}^{k} a_j b_{k-j} - \sum_{j=k+1}^{n-1} a_j b_{n+k-j} - \sum_{j=\frac{n}{2}+1+k}^{n-1} a_j b_{n+\frac{n}{2}+k-j} & k = 0, \dots, \frac{n}{2}-2 \\ \sum_{j=0}^{k} a_j b_{k-j} - \sum_{j=k+1}^{n-1} a_j b_{n+k-j} & k = \frac{n}{2}-1 \\ \sum_{j=0}^{k} a_j b_{k-j} - \sum_{j=k-\frac{n}{2}+1}^{n-1} a_j b_{\frac{n}{2}+k-j} & k = \frac{n}{2}, \dots, n-1 \end{cases} .$$

These equations lack the same regularity observed in their power-of-two counterparts: we need three distinct cases, whereas in [39], one formula is sufficient to express all the coefficients. For this reason, straightforward substitution does not enable us to compute $\mathrm{Cov}(c_i, c_j)$, so we need the following theorem.

Theorem 3. *Let $m = 2^i 3^j$ for $i, j \in \mathbb{N}_{>0}$ and $\mathcal{R} = \mathbb{Z}[x]/\Phi_m(x)$ where $\Phi_m(x) = x^n - x^{n/2} + 1$ is the m^{th} cyclotomic polynomial ($n = \phi(m)$). Let $c(x) = a(x)b(x)$ be the product of two random polynomials in $\mathcal{R}$ with coefficient variances V_a and V_b respectively, and let $\mathbf{c} = (c_0, \dots, c_{n-1})$ be the vector of coefficients of $c(x)$. Then the covariance matrix of $\mathbf{c}$ is formed by four diagonal blocks of size $n/2$:*

$$\mathrm{CovM}(\mathbf{c}) = \begin{pmatrix} \mathrm{Diag}(\alpha_0, \dots, \alpha_{n/2-1}) & \mathrm{Diag}(\beta_0, \dots, \beta_{n/2-1}) \\ \mathrm{Diag}(\beta_0, \dots, \beta_{n/2-1}) & \mathrm{Diag}(\alpha_{n/2}, \dots, \alpha_{n-1}) \end{pmatrix}$$

where

$$\alpha_k = \begin{cases} \left(\frac{3}{2}n - (k+1)\right) V_a V_b & \text{if } 0 \le k < n/2 \\ \frac{3}{2} n V_a V_b & \text{if } n/2 \le k < n \end{cases}$$
$$\beta_k = (k+1-n) V_a V_b \qquad 0 \le k < n/2\ .$$

Notice how the bound in Equation (20) follows immediately from the theorem: the variances of the coefficients are the values α_i in the matrix above.

Proof. The fundamental tool in this proof is the radix-6 NTT isomorphism (Section 3.3)

$$\Psi : \mathcal{R} \to \mathbb{Z}[x]/(x^{n/2} - \zeta) \times \mathbb{Z}[x]/(x^{n/2} - \zeta^5) = \mathcal{R}^\ell \times \mathcal{R}^r$$

where $\zeta = 1/2 + \sqrt{3}/2i$ is a complex primitive 6^{th} root of unity. The idea is to consider the images $a(x)$, $b(x)$ via this isomorphism and perform the multiplication in the factor rings where it is easier to keep track of the correlations.

Since $\mathbb{Z}$ does not contain a sixth primitive root of unity, we have to embed $\mathbb{Z}[x]$ identically into the polynomial ring $\mathbb{Z}[\zeta][x]$. By doing so, we obtain a CRT isomorphism represented by:

$$\Psi : \mathbb{Z}[\zeta][x]/(x^n - x^{n/2} + 1) \to \mathbb{Z}[\zeta][x]/(x^{n/2} - \zeta) \times \mathbb{Z}[\zeta][x]/(x^{n/2} - \zeta^5),$$

whose restriction to $\mathbb{Z}$ yields exactly the desired isomorphism. In practice, this transformation is given by reductions modulo the quotienting polynomials of the factor rings, and it can be computed on 2 coefficients simultaneously using a radix-6 butterfly operation (Section 3.3). Using this isomorphism, we compute $c(x) = a(x)b(x) \in \mathcal{R}$ as $c(x) = \Psi^{-1}(\Psi(a(x))\Psi(b(x)))$. The advantage of multiplying in $\mathcal{R}^\ell$ and $\mathcal{R}^r$ is that their quotienting polynomial is of the form $x^\alpha +$ constant, which makes the modular reduction again similar to the power-of-two case. In other words, using the radix-6 split takes care of the repetition of coefficients introduced by the reduction modulo $x^n - x^{n/2} + 1$ mentioned above. We proceed now by examining each of the three steps in more detail: the direct isomorphism Ψ, the product in $\mathcal{R}^\ell$ and $\mathcal{R}^r$ (which are essentially the same) and finally, the inverse isomorphism Ψ^{-1}.
Recall that ζ satisfies $\bar{\zeta} = \zeta^5 = 1 - \zeta$ and $\zeta^2 - \zeta + 1 = 0$; furthermore for any $z \in \mathbb{C}$ we have $z\bar{z} = |z|^2$ where $|\cdot|$ is the complex modulus.

1. The isomorphism Ψ. Let $a(x) \in \mathcal{R}$, then $\Psi(a) = (a^\ell(x), a^r(x))$ where

$$a_i^\ell = a_i + \zeta a_{i+n/2} \quad \text{and} \quad a_i^r = a_i + \zeta^5 a_{i+n/2} = a_i + (1-\zeta)a_{i+n/2}$$

for any $i = 0, \ldots, n/2-1$. Since $\zeta(1-\zeta) = 1$ and since all coefficients of $a(x) \in \mathcal{R}$ are uncorrelated, with mean 0 and variance V_a, we have

$$\begin{aligned}
\mathrm{E}[a_i^\ell] &= \mathrm{E}[a_i^r] = 0 \\
\mathrm{Var}(a_i^\ell) = \mathrm{Var}(a_i^r) &= \mathrm{E}[(a_i + \zeta a_{i+n/2})\overline{(a_i + \zeta a_{i+n/2})}] \\
&= \mathrm{E}[a_i^2 + \zeta a_i a_{i+n/2} + (1-\zeta)a_i a_{i+n/2} + \zeta(1-\zeta)a_{i+n/2}^2] \\
&= \mathrm{E}[a_i^2] + \mathrm{E}[a_i a_{i+n/2}] + \mathrm{E}[a_{i+n/2}^2] = 2V_a\ .
\end{aligned} \tag{21}$$

Moreover, each coefficient of a is used to construct exactly one coefficient of a^ℓ and one of a^r. Then, by the independence of the a_is, it follows that for any $i \neq j$ we have that each of a_i^ℓ and a_i^r is independent of both a_j^ℓ and a_j^r. Namely, for all $i \neq j$, $\mathrm{Cov}(a_i^\ell, a_j^\ell) = \mathrm{Cov}(a_i^\ell, a_j^r) = \mathrm{Cov}(a_i^r, a_j^r) = 0$ and the only nonzero covariances are given by

$$\begin{aligned}
\mathrm{Cov}(a_i^\ell, a_i^r) &= \mathrm{E}[(a_i + \zeta a_{i+n/2})\overline{(a_i + (1-\zeta)a_{i+n/2})}] \\
&= \mathrm{E}[(a_i + \zeta a_{i+n/2})^2] = (1+\zeta^2)V_a = \zeta V_a\ .
\end{aligned} \tag{22}$$

Obviously, the same formulas hold for $b(x)$ with V_b in place of V_a.

2. Product in $\mathcal{R}^\ell$ and $\mathcal{R}^r$. Consider the two left images $a^\ell(x)$ and $b^\ell(x)$ in $\mathcal{R}^l$. We compute the coefficients of $c^\ell(x) = a^\ell(x)b^\ell(x)$ by first calculating the product as if we were working in $\mathbb{Z}[x]$ and then reducing modulo $x^{n/2} - \zeta$. Let

V_{a^ℓ} and V_{b^ℓ} be the coefficient variances of the two factors. We have

$$g^\ell(x) = \sum_{l=0}^{n-2} g_l^\ell x^l = a^\ell(x) b^\ell(x) \in \mathbb{Z}[x] \text{ with } g_l^\ell = \sum_{i+j=l} a_i^\ell b_j^\ell \ .$$

It is clear that all the g_l^ℓ are uncorrelated and have mean 0, and $c_k^\ell = g_k^\ell + \zeta g_{k+n/2}^\ell$. Again no g_l^ℓ is repeated in any two distinct c_k^ℓs, implying

$$\mathrm{Cov}(c_{k_1}^\ell, c_{k_2}^\ell) = \begin{cases} \mathrm{E}[(g_k^\ell + \zeta g_{k+n/2}^\ell)(g_k^\ell + \bar{\zeta} g_{k+n/2}^\ell)] = \frac{n}{2} V_{a^\ell} V_{b^\ell} & \text{if } k_1 = k_2 \\ 0 & \text{otherwise} \end{cases} .$$

The same reasoning holds for $\mathcal{R}^r$: we have

$$g^r(x) = \sum_{l=0}^{n-2} g_l^r x^l = a^r(x) b^r(x) \in \mathbb{Z}[x] \text{ with } g_l^r = \sum_{i+j=l} a_i^r b_j^r \ .$$

Since for any $i = 0, \dots, n/2 - 1$ we have $c_k^r = g_k^r + (1-\zeta) g_{k+n/2}^r$, we get also for the right side

$$\mathrm{Cov}(c_{k_1}^r, c_{k_2}^r) = \begin{cases} \frac{n}{2} V_{a^r} V_{b^r} & \text{if } k_1 = k_2 \\ 0 & \text{otherwise} \end{cases} .$$

Regarding the cross-side covariance $\mathrm{Cov}(c_{k_1}^\ell, c_{k_2}^r)$, its computation reduces by linearity to many terms of the form $\mathrm{Cov}(a_{i_1}^\ell b_{j_1}^\ell, a_{i_2}^r b_{j_2}^r)$. As before, we have

$$\mathrm{Cov}(a_{i_1}^\ell b_{j_1}^\ell, a_{i_2}^r b_{j_2}^r) \neq 0 \iff i_1 = i_2 \text{ and } j_1 = j_2.$$

Since no product $a_i^\ell b_j^\ell$ ($a_i^r b_j^r$) is repeated in two different g_l^ℓ (g_l^r), and no g_l^ℓ (g_l^r) is repeated in any two distinct c_k^ℓ (c_k^r), the condition above can be realized only when $k_1 = k_2$, meaning that we also have

$$\mathrm{Cov}(c_{k_1}^\ell, c_{k_2}^r) \begin{cases} \neq 0 & \text{if } k_1 = k_2 \\ = 0 & \text{otherwise} \end{cases} .$$

Furthermore for $k = 0, \dots, n/2 - 1$ we have

$$\begin{aligned} \mathrm{Cov}(c_k^\ell, c_k^r) &= \mathrm{Cov}(g_k^\ell + \zeta g_{k+n/2}^\ell, g_k^r + (1-\zeta) g_{k+n/2}^r) \\ &= \mathrm{Cov}(g_k^\ell, g_k^r) + \zeta \overline{(1-\zeta)} \, \mathrm{Cov}(g_{k+n/2}^\ell, g_{k+n/2}^r) \\ = (k+1) \, \mathrm{Cov}&(a_i^\ell, a_i^r) \, \mathrm{Cov}(b_i^\ell, b_i^r) + \zeta^2 \left(\frac{n}{2} - (k+1) \right) \mathrm{Cov}(a_i^\ell, a_i^r) \, \mathrm{Cov}(b_i^\ell, b_i^r) \ . \end{aligned}$$

Thus, we can substitute Equations (21) and (22) obtaining

$$\begin{aligned} V_{a^l} &= V_{a^r} = \mathrm{Var}(a_i^l) = \mathrm{Var}(a_i^r) = 2V_a \\ V_{b^l} &= V_{b^r} = \mathrm{Var}(b_i^l) = \mathrm{Var}(b_i^r) = 2V_b \\ \mathrm{Cov}&(a_i^\ell, a_i^r) = \zeta V_a \text{ and } \mathrm{Cov}(b_i^\ell, b_i^r) = \zeta V_b. \end{aligned}$$

Hence

$$\mathrm{Var}(c_k^l) = \mathrm{Var}(c_k^r) = \frac{n}{2} \cdot 2V_a \cdot 2V_b = 2nV_aV_b \tag{23}$$

and

$$\begin{aligned}\mathrm{Cov}(c_k^l, c_k^r) &= (k+1)\zeta V_a \zeta V_b + \zeta^2(\frac{n}{2} - (k+1))\zeta V_a \zeta V_b \\ &= (\zeta^2 + \zeta)(k+1)V_aV_b - \zeta\frac{n}{2}V_aV_b\ . \end{aligned}\tag{24}$$

3. The isomorphism Ψ^{-1}. The inverse NTT butterfly operation in [48] is given by the following matrix-vector product: for any $k = 0, \ldots, n/2 - 1$

$$\begin{pmatrix} c_k \\ c_{k+n/2} \end{pmatrix} = \frac{1}{(1-2\zeta)} \begin{pmatrix} 1-\zeta & -\zeta \\ -1 & 1 \end{pmatrix} \begin{pmatrix} c_k^\ell \\ c_k^r \end{pmatrix}$$

Note that, for any $\bar{k}_1, \bar{k}_2 = 0, \ldots, n-1$, the computation of $\mathrm{Cov}(c_{\bar{k}_1}, c_{\bar{k}_2})$ reduces by linearity to calculate a linear combination of the terms $\mathrm{Cov}(c_{k_1}^\ell, c_{k_2}^r)$ where

$$k_j = \begin{cases} \bar{k}_j & \text{if } j < \frac{n}{2} \\ \bar{k}_j - \frac{n}{2} & \text{if } j \geq \frac{n}{2} \end{cases}$$

for any $j = 1, 2$. As seen previously, $\mathrm{Cov}(c_{k_1}^\ell, c_{k_2}^r) \neq 0$ if and only if $k_1 = k_2$, and this implies either $\bar{k}_1 = \bar{k}_2$ or $\bar{k}_1 = \bar{k}_2 \pm \frac{n}{2}$; hence

$$\mathrm{Cov}(c_{\bar{k}_1}, c_{\bar{k}_2}) \neq 0 \Rightarrow \bar{k}_1 = \bar{k}_2 \text{ or } \bar{k}_1 = \bar{k}_2 \pm \frac{n}{2}\ .$$

Regarding the exact formulas for the nonzero terms in $\mathrm{CovM}(\mathbf{c})$, we have different cases according to k. Notice that

$$\frac{1}{1-2\zeta}\overline{\left(\frac{1}{1-2\zeta}\right)} = \frac{1}{|1-2\zeta|^2} = 1/3$$

moreover, for any $z \in \mathbb{C}$ we have $z + \overline{z} = 2\Re(z)$, and by the properties of covariance $\mathrm{Cov}(X, Y) = \overline{\mathrm{Cov}(Y, X)}$.

For $0 \leq k < n/2$ we have

$$\mathrm{Var}(c_k) = \mathrm{Var}\left(\frac{(1-\zeta)c_k^l - \zeta c_k^r}{1-2\zeta}\right) = \frac{1}{3}(\mathrm{Var}(c_k^l) + \mathrm{Var}(c_k^r) - 2\Re((1-\zeta)^2\,\mathrm{Cov}(c_k^l, c_k^r))).$$

Substituting Equations (21) and (22), we get

$$\begin{aligned}\mathrm{Var}(c_k) &= \frac{1}{3}\left(4nV_aV_b - 2\Re\left((1-\zeta)^2[(\zeta^2+\zeta)(k+1)V_aV_b - \zeta\frac{n}{2}V_aV_b]\right)\right) \\ &= \left(\frac{3}{2}n - (k+1)\right)V_aV_b\ .\end{aligned}$$

For $k \geq n/2$, instead, the behaviour of the variance is constant:

$$\mathrm{Var}(c_k) = \mathrm{Var}\left(\frac{-c_k^l + c_k^r}{1-2\zeta}\right) = \frac{1}{3}(\mathrm{Var}(c_k^l) + \mathrm{Var}(c_k^r) - 2\Re(\mathrm{Cov}(c_k^l, c_k^r)))\ .$$

Since $\zeta^2 + \zeta = \sqrt{3}i$ has the real part equal to 0 and thanks to Equations (21) and (22), we have:

$$\begin{aligned}\mathrm{Var}(c_k) &= \frac{1}{3}\left(4nV_aV_b - 2\Re\left((\zeta^2+\zeta)(k+1)V_aV_b - \zeta\frac{n}{2}V_aV_b\right)\right) = \\ &= \frac{1}{3}\left(4nV_aV_b + \frac{n}{2}V_aV_b\right) = \frac{3}{2}nV_aV_b \ .\end{aligned}$$

Finally, regarding the nonzero covariances for $0 \leq k < n/2$ we find

$$\begin{aligned}\mathrm{Cov}(c_k, c_{k+n/2}) &= \mathrm{Cov}\left(\frac{(1-\zeta)c_k^l - \zeta c_k^r}{1-2\zeta}, \frac{-c_k^l + c_k^r}{1-2\zeta}\right) = \\ &= \frac{1}{3}\left(-(1-\zeta)\,\mathrm{Var}(c_k^l) - \zeta\,\mathrm{Var}(c_k^r) + 2\Re((1-\zeta)\,\mathrm{Cov}(c_k^l, c_k^r))\right)\end{aligned}$$

and substituting Equations (23) and (24) we get

$$\begin{aligned}\mathrm{Cov}(c_k, c_{k+n/2}) &= \frac{1}{3}(-(1-\zeta)2nV_aV_b - \zeta 2nV_aV_b + \\ &\quad + 2\Re\left((1-\zeta)[(\zeta^2+\zeta)(k+1)V_aV_b - \zeta\frac{n}{2}V_aV_b]\right) \\ &= \frac{1}{3}\left(-3nV_aV_b + 3(k+1)V_aV_b\right) = (k+1-n)V_aV_b\end{aligned}$$

and this concludes the proof.

The following result is the analogue of Theorem 3 for the power-of-two case; the proof is much simpler and does not require the use of NTT butterflies. The results are consistent with the bound on the variance of a random product in [39, Section 2.8].

Theorem 4. *Let $m = 2^i$ for $i \in \mathbb{N}_{>0}$ and $\mathcal{R} = \mathbb{Z}[x]/\Phi_m(x)$ where $\Phi_m(x) = x^n + 1$ is the m^{th} cyclotomic polynomial ($n = \phi(m)$). Let $c(x) = a(x)b(x)$ be the product of two random polynomials in $\mathcal{R}$ and let $\mathbf{c} = (c_0, \ldots, c_{n-1})$ be the vector of coefficients of $c(x)$. Then the covariance matrix of $\mathbf{c}$ is diagonal, and in particular* $\mathrm{CovM}(\mathbf{c}) = \mathrm{Diag}(nV_aV_b)$.

4.4 Independence of the coefficients

In order to use the bound given in Equation (19) on a product of random polynomials $c(x) = \sum_{i=0}^{n-1} c_i x^i = a(x)b(x) \in \mathcal{R}_q$, we need the coefficients c_i to be independent. This is also needed to some extent to use Theorem 3 iteratively to bound the variance of the coefficients, although some dependence could be allowed in that case. As we have seen in previous section, expressing the coefficient of a product in $\mathcal{R}$ or $\mathcal{R}_q$ starting from the factors requires quite complex formulas, determined by the reduction modulo $\Phi_m(x)$. If we try to formally check the independence of the coefficients of the product in $\mathcal{R}$, we obtain a negative result (Theorem 3). For the power of two case we have linear independence [39], but one can easily infer that there is a higher degree relation by looking at the second moments. BGV anyway is based on using polynomial with coefficients $\mathbb{Z}_q$, hence we wonder what is the effect of working in $\mathcal{R}_q$ on the dependence relations between the coefficients. As in the power-of-two case, we wish that the reduction modulo q wipes out the dependence relations, as formally stated in the following conjecture.

Conjecture 1. Let $a(x)$ and $b(x)$ be random polynomials in $\mathcal{R}_q$, then the coefficients of $c(x) = \sum_{i=0}^{n-1} c_i x^i = a(x)b(x)$ are statistically independent.

Such a conjecture is reasonable to make, because the reduction modulo q is hard to predict probabilistically. To support this idea, we performed chi-squared independence test using the Python module *stats*. We sampled large amounts of products of random polynomials, stored the coefficients in a contingency table and performed the chi-squared test $stats.chi2\_contingency$. The implementation of such test is straightforward, but due to computational constraints we could only test for small sets of parameters. This is mainly due to the fact that the size of the set of samples has to grow with the parameters to maintain statistical significance. Anyway the algebraic structure of $\mathcal{R}_q$ does not change when scaling up n and q, and we believe these tests are a good indicator of the general behaviour. The formulas determining the c_is only depend on the nonzero coefficients of the cyclotomic polynomial $\Phi_m(x)$ in use, which in turn are determined essentially by the radical of m (see Lemma 1). We set the statistical significance threshold at 5%, a common choice when performing the chi-squared test. The interested reader can find below the results for two of the biggest instances we could test. For a fixed cyclotomic index we tested different prime moduli q: in the tables below, 'sample size' is the number of random products that was generated in a test instance, 'rejections' is the average number of times the independence hypothesis was rejected and 'rejections %' is the rate of rejections over the number of tests. We point out that the couples of coefficients that get rejected are not always the same in different test instances, meaning the rejections are generated by the nature of the test itself and not by a cause related to our hypothesis. By the very nature of the test itself, 5% of all instances should be rejected; anyway sometimes we observed a higher percentage (e.g. greater than 6%). As we found out, this depends on the modulus q and the sample size: the bigger q gets with respect to n, the bigger samples we need to have a statistically significant result. In fact we can see how increasing the sample size brings the percentage of rejections closer to 5%, therefore failing to reject the independence hypothesis. We stress the fact that the more complex of these tests already require several minutes on a dedicated server, meaning that testing on realistic BGV rings would require a huge amount of computational power and time.

5 Noise estimates for homomorphic operations

In this section, we develop the noise bounds for the operations described in Section 3 with the aid of the results in Section 4. The main properties we use are the following.

q	number of samples	sample size	rejections (avg.)	rejections %
433	5	5000	67.8	6.2%
433	5	10000	51.8	4.8%
1009	5	15000	70.8	6.5%
1009	5	20000	58.8	5.4%

Table 2. Test 3: $m = 2^4 3^2 = 144$, $n = 48$.

q	number of samples	sample size	rejections (avg.)	rejections %
433	5	5000	154	6.2%
433	5	10000	134.4	5.4%
1297	5	20000	173	6.9%
1297	5	30000	145.8	5.8%

Table 3. Test 4: $m = 2^3 3^3 = 216$, $n = 72$.

– Lemma 4 and Equation (3) to bound the noise with the canonical norm of the critical quantity. We get

$$||\nu|| < c_m ||\nu||^{can} \text{ with } c_m = 2/\sqrt{3} .$$

– Equation (19) to bound the canonical norm of ν with the variance of its coefficients. We have

$$||\nu||^{can} \leq 6\sqrt{nV_\nu}, \text{ and so } ||\nu|| < \frac{2}{\sqrt{3}} 6\sqrt{nV_\nu} = 4\sqrt{3nV_\nu}$$

with probability $1 - ne^{-36}$.

– The properties of the coefficients variance of random polynomials, including Theorem 3. For two independent random polynomials a and b in $\mathcal{R}_q$ and a scalar $\gamma \in \mathbb{Z}_q$

$$V_{a+b} = V_a + V_b, \quad V_{\gamma a} = \gamma^2 V_a, \quad V_{ab} \leq \frac{3}{2} n V_a V_b .$$

This approach is also referred to as a *worst-case* canonical embedding analysis in the literature. A similar work for the power-of-two case is [52].

5.1 Encryption and ring operations

After the encryption, the critical quantity ν is given by Equation (6). Recalling that all errors have the same distribution with variance V_e, and u comes from the same distribution of the secret key s,

$$\begin{aligned} ||\nu||^{can} &\leq 4\sqrt{3nV_{m+t(e\cdot u+e_1\cdot s+e_0)}} \\ &\leq 4\sqrt{3n\left(\frac{t^2}{12}+t^2\left(\frac{3}{2}nV_eV_u+\frac{3}{2}nV_{e_1}V_s+V_{e_0}\right)\right)} \\ &\leq 4t\sqrt{3n\left(\frac{1}{12}+3nV_eV_s+V_e\right)} = \mathsf{B}_{\mathsf{clean}} \ . \end{aligned} \tag{25}$$

By this computation, we set $\mathsf{B}_{\mathsf{clean}}$ as our bound for the noise in a fresh ciphertext.

To estimate ν_{Add} (Equation (9)), we use the triangular inequality for the canonical norm: we have

$$||\nu_{\mathsf{Add}}||^{can} = ||\nu+\nu'||^{can} \leq ||\nu||^{can} + ||\nu'||^{can} \tag{26}$$

and this actually applies to any sum of polynomials.

Regarding polynomial multiplication (Equation (10)), we have $\nu_{\mathsf{Mul}} = \nu\nu'$, and we proceed using the sub-multiplicativity of the canonical norm (Equation (2)); we immediately obtain

$$||\nu_{\mathsf{Mul}}||^{can} \leq ||\nu||^{can}||\nu'||^{can} \tag{27}$$

which is used to estimate the noise.

Finally, for $\mathsf{ConstMul}$ (Equation (11)), the critical quantity is again a polynomial product $\nu_{\mathsf{ConstMul}} = \alpha\nu$. Note that the two factors are independent, as α can be seen as a uniformly random polynomial in $\mathcal{R}_t$. Thus, we can split the variance $V_{\alpha\nu}$ using Equation (20). Moreover, we have $V_\alpha = \frac{t^2}{12}$ and $||\nu||^{can} \approx 6\sqrt{nV_\nu}$. So

$$\begin{aligned} ||\nu_{\mathsf{ConstMul}}||^{can} &\leq 6\sqrt{nV_{\alpha\nu}} \leq 6\sqrt{n\frac{3}{2}nV_\alpha V_\nu} \\ &\leq \sqrt{\frac{3}{2}n\frac{t^2}{12}}6\sqrt{nV_\nu} = t\sqrt{\frac{1}{8}n}\ ||\nu||^{can} \ . \end{aligned} \tag{28}$$

This is an improvement on previous bounds (e.g. [51]), which used again the sub-multiplicativity of the canonical norm. It is worth noting that Equation (28) provides a generic bound. Thus, if we have a fixed constant $\alpha \in \mathbb{N}$, it is more convenient to use a *modified* version of the bound in Equation (9), namely $\alpha||\nu||$, since $\alpha\mathbf{c}$ is sum of α times the ciphertext $\mathbf{c}$.

5.2 Modulus switching

After the one-step modulus switching, the critical quantity is given by Equation (13) as $\nu_{\mathsf{ModSw}} = \nu + \delta(s)/p_l$, where $\delta(s)$ is as in Equation (12). By using the triangular inequality, we get

$$||\nu_{\mathsf{ModSw}}||^{can} \leq \frac{||\nu||^{can} + ||\delta(s)||^{can}}{p_l}.$$

Hence, we have to estimate the canonical norm of $\delta(s)$. The two polynomials δ_0 and δ_1 can be seen as random polynomials with coefficients in $\mathbb{Z}_{tp_l}$, as they are computed from ciphertexts c_0 and c_1 via modular reduction. Thus,

$$||\delta(s)||^{can} \leq 6\sqrt{nV_{\delta_0+\delta_1\cdot s}} = 6\sqrt{n\left(V_{\delta_0} + \frac{3}{2}nV_{\delta_1}V_s\right)} \leq p_l 6t\sqrt{n\left(\frac{1}{12} + \frac{1}{8}nV_s\right)}.$$

Namely,

$$||\nu_{\mathsf{ModSw}}||^{can} \leq \frac{||\nu||^{can}}{p_l} + \mathsf{B}_{\mathsf{scale}} \quad \text{where} \quad \mathsf{B}_{\mathsf{scale}} = 6t\sqrt{n\left(\frac{1}{12} + \frac{1}{8}nV_s\right)}. \tag{29}$$

Notice that the term $\mathsf{B}_{\mathsf{scale}}$ is independent of p_l.

In the general case of k-step modulus switching, we have to consider the RNS representation. If l is the starting level and $l' = l - k$ the arrival level, then using Equation (8) we have

$$\delta = t\,\mathrm{FBE}(-t^{-1}c, \frac{q_l}{q_{l'}}, q_{l'})$$

which implies the coefficient of the polynomials δ_0 and δ_1 have variance

$$V_{\delta_i} = t^2\frac{k}{12}\frac{q_l^2}{q_{l'}^2}\ .$$

As a consequence of this, we have $||\nu_{\mathsf{ModSw}}||^{can} \leq \frac{q_{l'}}{q_l}||\nu||^{can} + \sqrt{k}\,\mathsf{B}_{\mathsf{scale}}$.

Key switching Performing computations similar to those in [52], it is possible to find the following bounds for the noise in the BV and GHS variants. Specifically,

$$\begin{aligned}
||\nu_{\mathsf{KeySw}}^{\mathsf{BV}}||^{can} &\leq ||\nu + \langle D_b(c_2), \mathbf{e}\rangle||^{can} \\
&\leq ||\nu||^{can} + b\sqrt{(\lfloor \log_b q_l \rfloor + 1)}\,\mathsf{B}_{\mathsf{KeySw}} \quad \text{where} \quad \mathsf{B}_{\mathsf{KeySw}} = 6tn\sqrt{\frac{V_e}{8}} \\
||\nu_{\mathsf{KeySw}}^{\mathsf{GHS}}||^{can} &\leq \left|\left|\nu + \frac{tc_2 \cdot e + \delta(s)}{C}\right|\right|^{can} \leq ||\nu||^{can} + \frac{q_l}{C}\,\mathsf{B}_{\mathsf{KeySw}} + \mathsf{B}_{\mathsf{scale}}\ .
\end{aligned}$$

Instead, for the Hybrid variant, by Equation (17) we have

$$\begin{aligned}
||\nu_{\mathsf{KeySw}}^{\text{Hybrid}}||^{can} &\leq \left|\left|\nu + \frac{t\langle D_b(c_2), \mathbf{e}\rangle + \delta(s)}{C}\right|\right|^{can} \\
&\leq ||\nu||^{can} + \frac{b\sqrt{\log_b q_l}}{C}\,\mathsf{B}_{\mathsf{KeySw}} + \mathsf{B}_{\mathsf{scale}}\ .
\end{aligned} \tag{30}$$

The RNS representation affects both the BV and the GHS key switching variants, and hence also the Hybrid one. In the BV variant, we substitute the decomposition with respect to a basis b with the one given by the CRT split in

Equation (7). This results in each element of $D(\alpha)$ having coefficients of the size of the various p_i composing the modulus q_l in use. Consequently, we have

$$||\nu_{\mathsf{KeySw}}^{\mathsf{BV}-\mathrm{RNS}}||^{can} \leq ||\nu||^{can} + \sqrt{L+1}\max(p_i)\,\mathsf{B}_{\mathsf{ks}}$$

Regarding the GHS variant, we have to factor in the effect of the base extension algorithm, which is used two times: once to extend c_2 from q_l to Q_l, the other to extend $\delta_0 + \delta_1 \cdot s$ from C to Q_l.

$$||\nu_{\mathsf{KeySw}}^{\mathsf{GHS}-\mathrm{RNS}}||^{can} \leq ||\nu||^{can} + \sqrt{L+1}\frac{q_l}{C}\,\mathsf{B}_{\mathsf{KeySw}} + \sqrt{k}\,\mathsf{B}_{\mathsf{scale}}\ .$$

Finally, by putting together these two analyses, we can find a bound for the noise after the Hybrid key switching: we have to account for the fact that the RNS is used to split the ciphertext in modulus h chunks $\tilde{q}_0,\ldots,\tilde{q}_{h-1}$. This affects the second summand in the GHS estimate, as we have to account for the BV-style decomposition of c_2: we have

$$||t\langle D(c_2), \mathbf{e}\rangle||^{can} \leq \sqrt{h}\max_{i\in[h]}(\tilde{q}_i)\,\mathsf{B}_{\mathsf{KeySw}}$$

and so

$$\begin{aligned} ||\nu_{\mathsf{KeySw}}^{\mathrm{Hybrid-RNS}}||^{can} &\leq ||\nu||^{can} + \frac{\sqrt{l+1}}{C}\sqrt{h}\max_{i\in[h]}(\tilde{q}_i)\,\mathsf{B}_{\mathsf{KeySw}} + \sqrt{k}\,\mathsf{B}_{\mathsf{scale}} \\ &\leq ||\nu||^{can} + \sqrt{h(L+1)}\frac{\max_{i\in[h]}(\tilde{q}_i)}{C}\,\mathsf{B}_{\mathsf{KeySw}} + \sqrt{k}\,\mathsf{B}_{\mathsf{scale}}\ . \end{aligned} \tag{31}$$

6 Analyzing Error in a Homomorphic Circuit

In this section, we study how to combine the different operations of the BGV scheme to perform complex computations. We need to model circuits involving homomorphic sums and products while controlling the noise growth using the modulus switching technique. Our approach performed modulus switching immediately after each polynomial product, thereby effectively mitigating the noise increase caused by the multiplication operation (Equation (27)). However, an exception arises at the final multiplicative layer, where no relinearization or modulus switching is performed. Instead, it is more convenient to decrypt the three-word ciphertext directly. Furthermore, the noise after encryption (Equation (25)) is already significant. Hence, a modulus switching is performed right after Enc.

Following these ideas, the number L of primes p_i needed to compose the ciphertext modulus is determined: if M is the multiplicative depth of the homomorphic circuit we want to evaluate, then $L = M + 1$.

Another thing to take into account when modelling a circuit is ciphertext rotations: these operations are useful from a practical standpoint, as they make key management easier. We do not go into detail regarding these procedures; we only mention them because, after each rotation, it is necessary to perform a key-switching step.

6.1 Building blocks

This work studies Model 1 [52, Section 3]: we assume to be working with η independent-computed ciphertexts $\mathfrak{c}_1, \dots, \mathfrak{c}_\eta$ in parallel and

1. perform on each ciphertext, a constant multiplication α_i: $\mathfrak{c}_i^{\mathrm{I}} = \mathsf{ConstMul}(\alpha_i, \mathfrak{c}_i)$;
2. followed by τ rotations: $\mathfrak{c}_i^{\mathrm{II}} = \mathrm{rot}_\tau(\dots \mathrm{rot}_1(\mathfrak{c}_i^{\mathrm{I}}))$.
3. Finally, we sum all the results of the previous steps:

$$\mathfrak{c}^{\mathrm{III}} = \mathsf{Add}(\mathfrak{c}_\eta^{\mathrm{II}}, \mathsf{Add}(\mathfrak{c}_{\eta-1}^{\mathrm{II}}, \mathsf{Add}(\dots, \mathsf{Add}(\dots, \mathsf{Add}(\mathfrak{c}_2^{\mathrm{II}}, \mathfrak{c}_1^{\mathrm{II}})))))$$

The resulting ciphertext is used as input to one multiplication $\mathsf{Mul}(\mathfrak{c}^{III}, \tilde{\mathfrak{c}}^{III})$.

We now compute a bound $\mathsf{B}_{\mathsf{block}}$ for the output noise of one such blocks. We analyze the noise growth by assuming that each of the η input ciphertexts $\mathfrak{c}_i = (\mathbf{c}_i, l, \nu_i)$ has noise $||\nu_i||^{can} < \mathsf{B}$. Then, by Equation (28), after the step 1.,

$$||\nu_i^{\mathrm{I}}||^{can} \leq \varepsilon\, \mathsf{B} \quad \text{where } \varepsilon = t\sqrt{n/8}.$$

For any rotation, we have to perform an Hybrid key switching. These introduce an additive growth in the error, and using the computations in Section 5.2, we get that the noise in $\mathfrak{c}_i^{\mathrm{II}}$ is bounded by

$$||\nu_i^{\mathrm{II}}||^{can} \leq \varepsilon\, \mathsf{B} + \tau v \quad \text{where} \quad v = \frac{\gamma_0}{C}\, \mathsf{B}_{\mathsf{KeySw}} + \gamma_1\, \mathsf{B}_{\mathsf{scale}}\,.$$

The values of γ_0 and γ_1 are given by either Equation (30) or Equation (31) if we are using the RNS representation. Namely, we have

$$(\gamma_0, \gamma_1) = \begin{cases} (b\sqrt{\log_b q_l}, 1) & \text{(Hybrid)} \\ (\sqrt{h(L+1)}\max_{i\in[h]}(\tilde{q}_i), \sqrt{k}) & (\mathrm{Hybrid} - \mathrm{RNS}) \end{cases}. \tag{32}$$

The next step in the block is the sum of the η ciphertexts $\mathfrak{c}_i^{\mathrm{II}}$; by Equation (26),

$$||\nu_i^{\mathrm{III}}||^{can} < \eta ||\nu_i^{\mathrm{II}}||^{can} < \eta(\varepsilon\, \mathsf{B} + \tau v)\,.$$

Finally, two ciphertexts computed as $\mathfrak{c}^{\mathrm{III}}$ are multiplied together in a building block. Then Equation (27) implies

$$\mathsf{B}_{\mathsf{block}} = \eta^2 \left(\varepsilon\, \mathsf{B} + \tau v\right)^2 \tag{33}$$

6.2 Moduli size

In this section, we analyze the size of the different moduli $p_0, \dots, p_{L-1}$ depending on their role in the scheme. All the *middle* moduli p_i, for $i = L-2, \dots, 1$, are associated with a building block like the one analyzed in the previous section. The idea is to move down the moduli ladder from $q_{L-1} = p_{L-1} \cdots p_0$ to $q_0 = p_0$, keeping in mind the function each prime modulus has.

- The *top modulus* does not have to support any homomorphic operations, as after encryption, we immediately use ModSw to reduce the noise $\mathsf{B}_{\mathsf{clean}}$ down to the base noise B. This implies p_{L-1} can be smaller than the other p_is.
- The *middle moduli* p_i, $i = L-2, \dots, 1$ are used to reduce the noise back to B after the corresponding building block has been performed.
- The *bottom modulus* needs to support decryption without counting on modulus switching to reduce the noise. This means we can still perform some homomorphic operations, but p_0 needs to be large enough to contain the corresponding noise growth.

We now analyze in detail each of the three different categories above.

Middle moduli The noise growth in a building block of the circuit is given by Equation (33). After the homomorphic product of ciphertexts concludes the block, we perform two more operations: a key switching to relinearize the product result and a modulus switching to reduce the noise. In the Hybrid variant, it is possible to merge these two because in $\mathsf{KeySw}^{\text{Hybrid}}$ (Equation (16)), it is already included a modulus switching: instead of switching down from Q_l to q_l, we can go directly to q_{l-1}. This decreases the noise by a multiplicative factor of $q_{l-1}/Q_l = 1/Cp_l$, and thanks to Equations (30) and (31) the condition on B is

$$\frac{\eta^2\left(\varepsilon\,\mathsf{B}+\tau v\right)^2}{p_l}+\frac{\gamma_0}{Cp_l}\,\mathsf{B}_{\mathsf{KeySw}}+\gamma_1\,\mathsf{B}_{\mathsf{scale}}<\mathsf{B}\ . \tag{34}$$

where γ_i are as in (32). Expanding the square in this inequality, we get

$$\frac{\eta^2\left(\varepsilon\,\mathsf{B}+\tau v\right)^2}{p_l}=\frac{\eta^2\varepsilon^2}{p_l}\,\mathsf{B}^2+\frac{2\eta^2\varepsilon\tau}{p_l}v\,\mathsf{B}+\frac{\eta^2\tau^2}{p_l}v^2.$$

Following [35], to isolate the terms in B we let

$$R_l=\frac{\eta^2\tau^2}{p_l}v^2+\frac{\gamma_0}{Cp_l}\,\mathsf{B}_{\mathsf{KeySw}}+\gamma_1\,\mathsf{B}_{\mathsf{scale}}$$

for each multiplicative level $l = 1, \dots, L-2$. This quantity increases with l, hence by bounding R_{L-2}, we bound all the other R_ls; moreover, we want this term to be as close as possible to $\mathsf{B}_{\mathsf{scale}}$ (notice that for sure $R_l > \gamma_1\,\mathsf{B}_{\mathsf{scale}}$). We can modify C to achieve this goal: letting

$$C>K\gamma_0\frac{B_{\mathsf{KeySw}}}{\mathsf{B}_{\mathsf{scale}}} \tag{35}$$

for some large $K \in \mathbb{N}$, e.g. $K = 100$, Equation (34) becomes the following inequality in B:

$$\frac{\eta^2\varepsilon^2}{p_l}\,\mathsf{B}^2+\left(\frac{2\eta^2\tau\varepsilon\gamma_1}{p_l}\,\mathsf{B}_{\mathsf{scale}}-1\right)\mathsf{B}+\frac{\eta^2\tau^2\gamma_1^2}{p_l}\,{\mathsf{B}_{\mathsf{scale}}}^2+\gamma_1\,\mathsf{B}_{\mathsf{scale}}<0\ .$$

Taking B as a variable, we get a quadratic expression, and we need its discriminant Δ to be positive. This implies

$$\begin{aligned}\Delta &= \left(\frac{2\eta^2\tau\varepsilon\gamma_1}{p_l}\,\mathsf{B}_{\mathsf{scale}} - 1\right)^2 - 4\frac{\eta^2\varepsilon^2}{p_l}\left(\frac{\eta^2\tau^2\gamma_1^2}{p_l}\,{\mathsf{B}_{\mathsf{scale}}}^2 + \gamma_1\,\mathsf{B}_{\mathsf{scale}}\right)\\ &= 1 - \frac{4\eta^2\varepsilon\gamma_1(\tau+\varepsilon)\,\mathsf{B}_{\mathsf{scale}}}{p_l} \geq 0\end{aligned}$$

which results in an estimate for the prime moduli:

$$p_1 \approx \ldots \approx p_{L-2} \approx 4\eta^2\varepsilon\gamma_1(\tau+\varepsilon)\,\mathsf{B}_{\mathsf{scale}}\ . \tag{36}$$

Setting p_l as Equation (36), for each l, we have the $\Delta = 0$. Thus, we recover B

$$\begin{aligned}\mathsf{B} &\approx -\frac{\left(\frac{2\eta^2\tau\varepsilon\gamma_1}{p_l}\,\mathsf{B}_{\mathsf{scale}} - 1\right)}{2\frac{\eta^2\varepsilon^2}{p_l}} = \frac{p_l}{2\eta^2\varepsilon^2} - \frac{\tau\gamma_1}{\varepsilon}\,\mathsf{B}_{\mathsf{scale}}\\ &\approx \frac{4\eta^2\varepsilon\gamma_1(\tau+\varepsilon)\,\mathsf{B}_{\mathsf{scale}}}{2\eta^2\varepsilon^2} - \frac{\tau\gamma_1}{\varepsilon}\,\mathsf{B}_{\mathsf{scale}} \approx \gamma_1\left(\frac{\tau}{\varepsilon}+2\right)\mathsf{B}_{\mathsf{scale}}\,.\end{aligned} \tag{37}$$

To conclude our estimates, we bound the constant C in the key switching by looking at the explicit values of γ_0 in Equation (35). For $l = 1, \ldots, L-2$ we have $b\sqrt{\log_b q_l} \leq b\sqrt{\log_b q_{L-2}}$ and $\sqrt{h(L+1)}\max_{i\in[h]}(\tilde{q}_i) \leq K p_{L-2}^{L/h}\sqrt{h(L-1)}$, implying that

$$C \geq \begin{cases} Kb\sqrt{\log_b q_{L-2}}\,\dfrac{B_{\mathsf{KeySw}}}{\mathsf{B}_{\mathsf{scale}}} & (\mathrm{Hybrid})\\[2ex] Kp_{L-2}^{L/h}\sqrt{h(L-1)}\,\dfrac{B_{\mathsf{KeySw}}}{\mathsf{B}_{\mathsf{scale}}} & (\mathrm{Hybrid-RNS})\end{cases} \tag{38}$$

where $K \approx 100$. According to [52], this is the smallest lower bound for C, and it is for this reason that the Hybrid key switching is preferred to the other two variants.

Top modulus After encryption, the noise is bounded by $\mathsf{B}_{\mathsf{clean}}$. We want the noise after ModSw to be smaller than a threshold B. Following Equation (29) the inequality determining the top modulus p_{L-1} is ${}^{\mathsf{B}_{\mathsf{clean}}}/_{p_{L-1}} + \mathsf{B}_{\mathsf{scale}} < \mathsf{B}$ and using the approximation in Equation (37) we get

$$p_{L-1} > \frac{\mathsf{B}_{\mathsf{clean}}}{((\frac{\tau}{\epsilon}+2)\gamma_1 - 1)\,\mathsf{B}_{\mathsf{scale}}}\ .$$

Bottom modulus At this level, the decryption condition is applied directly to the noise bound for the building block (Equation (33)), resulting in the bound $p_0 = q_0 > 2c_m\eta^2\left(\varepsilon\,\mathsf{B} + \tau v\right)^2$. Since the constant C is quite large, we have

$$v = \gamma_0\frac{\mathsf{B}_{\mathsf{KeySw}}}{C} + \gamma_1\,\mathsf{B}_{\mathsf{scale}} \approx \gamma_1\,\mathsf{B}_{\mathsf{scale}}\ .$$

Moreover, thanks to Equation (37), we have

$$\varepsilon \mathsf{B} + \tau v \approx \varepsilon\gamma_1 \left(\frac{\tau}{\varepsilon} + 2\right) \mathsf{B}_{\mathsf{scale}} + \tau\gamma_1 \mathsf{B}_{\mathsf{scale}} = 2\gamma_1(\tau + \varepsilon) \mathsf{B}_{\mathsf{scale}} \ .$$

Finally, we get the following condition on p_0:

$$p_0 > 2c_m\eta^2(2(\tau\gamma_1 + 1) \mathsf{B}_{\mathsf{scale}})^2 = 8c_m\eta^2\gamma_1^2(\tau + \varepsilon)^2 \mathsf{B}_{\mathsf{scale}}{}^2 \ .$$

6.3 Parameters specification

We briefly recall the conditions of the parameters.

- $m = 2^i3^j$ is the cyclotomic index. It comes with an expansion factor $c_m = 2/\sqrt{3}$ and $n = \phi(m) = m/3$.
- $q_l = \prod_{i=0}^{l} p_i$ are the ciphertext moduli, for $l = 0, \dots, L-1$; we need $p_i =_m 1$ to have efficient NTT, and the p_i need to be *word-sized primes* ([36]), meaning they need to fit the native data length of the machine we are using (usually 32 or 64 bits) to exploit the RNS representation fully.
- h is the number of blocks for the RNS decomposition in the Hybrid key switching, and we take $h = 3$.
- C is the auxiliary modulus for the key switching. For the RNS variant, we need $C = \prod_{j=1}^{k} C_j$ and $C_j =_m 1$ again for NTT related reasons. The size of C is determined using Equation (38).
- $V_e = 2m\sigma^2$, where $\sigma = 3.19$, and $V_s = 2/3$ are the variances of the errors and of the secret key.
- τ is the number or rotations, η is the number of summands in each block.
- $\varepsilon = t\sqrt{n/8}$ is a constant due to the multiplication by the ConstMul step in the circuit; if we wish to suppress this step, it is sufficient to set $\varepsilon = 1$.

In Table 4 and Table 5, we summarize all the results coming from previous sections.

$\mathsf{B}_{\mathsf{clean}}$	$\mathsf{B}_{\mathsf{scale}}$	$\mathsf{B}_{\mathsf{KeySw}}$	B
$4t\sqrt{3n\left(\frac{1}{12} + 3nV_eV_s + V_e\right)}$	$6t\sqrt{n\left(\frac{1}{12} + \frac{1}{8}nV_s\right)}$	$6tn\sqrt{\frac{V_e}{8}}$	$\gamma_1\left(\frac{\tau}{\varepsilon} + 2\right) \mathsf{B}_{\mathsf{scale}}$

Table 4. Intermediate noise bounds

7 Our Results

As displayed in Section 3.3, the algebraic structure of a cyclotomic ring with index $m = 2^s3^t$ can be exploited to deploy multiplication algorithms with performance comparable to the power-of-two case. Nonetheless, as seen in Sections 4

τ	p_0	p_l $(l = 1, \ldots, L-2)$	p_{L-1}
0	$8c_m\eta^2\varepsilon^2\,\mathsf{B_{scale}}^2$	$4\eta^2\varepsilon^2\gamma_1\,\mathsf{B_{scale}}$	$\dfrac{\mathsf{B_{KeySw}}}{(2\gamma_1-1)\,\mathsf{B_{scale}}}$
$\neq 0$	$8c_m\eta^2\gamma_1^2(\tau+\varepsilon)^2\,\mathsf{B_{scale}}^2$	$4\eta^2\varepsilon\gamma_1(\tau+\varepsilon)\,\mathsf{B_{scale}}$	$\dfrac{\mathsf{B_{clean}}}{\mathsf{B}-\mathsf{B_{scale}}}$

Table 5. Sizes of the prime moduli

to 6 there are some major modifications that we need to take into account when performing noise estimation. In the following, we compare the results of the parameter estimation process with their power-of-two counterparts. The estimates for this setting are based on the formulas in [52] and follow the same blueprint of Section 6; this way, we obtain comparable results between the two frameworks. To draw comparisons, we fix a security threshold λ (e.g. $\lambda = 128$) and look for the smallest possible parameters supporting a certain circuit with security λ. The security of our constructions is evaluated using the Lattice Estimator by Albrecht et al. [3]. In Tables 6 to 9, we report both the sizes of the ciphertext modulus q and the modulus qC used in the Hybrid key switching (Section 3). Although most of BGV works modulo q, the security needs to be assessed with respect to qC as part of the key switching is public.

To build circuits, we fix the parameters of a building block (Section 6.1) and then increase the number of multiplications M. Obviously, the security decreases as M grows, meaning that at some point, we will slip below the security threshold. When this happens, it is necessary to raise the cyclotomic index, giving us the margin to show our improvements with respect to the power-of-two case. We call the instances for which we improve the estimates *corner cases*.

Example 1. We consider a simple circuit where the building block has no constant multiplication, no rotations ($\tau = 0$), two summands for each block ($\eta = 2$), and plaintext modulus $t = 64$. We will refer to this construction again; thus, we name it Circuit 1. We run the computation for $m = 2^{13}$ and $m = 2^{14}$, meaning we work with lattices of dimension $n = 2^{12} = 4096$ and $n = 2^{13} = 8192$, respectively. The sizes of the ciphertext modulus and the security parameter are reported in Table 6. Now, assume we want to achieve 128 bits of security on a circuit with 3 multiplications. If we look at our power-of-two parameters, we can see that for $n = 2^{12}$ this cannot be done, as for $M = 3$ we have $\lambda < 128$ (the red cells in Table 6). Considering the power-of-two rings, the only option we have at this point is to *jump* to $n = 2^{13}$ where a solution with $\lambda = 209$ is available (the green cells in Table 6). Moreover, for this n, it is also possible to use the circuits with $\mathrm{M} = 4, 5$ (the blue cells in Table 6) and decrypt after the desired number of multiplications since they also feature $\lambda \geq 128$. Anyway, this approach is suboptimal, since it requires significantly larger ciphertext moduli ($\log q$). The main issue with all three constructions is that increasing the dimension does not

	M	1	2	3	4	5	6	7	8	9	10
	$\log q$	46	68	91	115	138	161	185	209	232	256
$n = 2^{12}$	$\log qC$	70	100	131	163	194	225	257	288	320	341
	λ	203	137	103	83	70	61	54	48	45	45
	$\log q$	48	71	96	119	144	168	193	208	240	267
$n = 2^{13}$	$\log qC$	72	104	137	169	202	235	268	301	334	367
	λ	436	286	209	165	136	116	101	90	81	75

Table 6. Power-of-two estimates for Circuit 1.

come for free. Indeed, n is also the degree of the quotient polynomial in the ring $\mathcal{R}_q$ where the cryptosystem lives. Hence, we are doubling the length of all the vectors involved by moving from $n = 2^{12}$ to $n = 2^{13}$. This affects the quantity of memory involved as well as the computational time required for the scheme to work. If, instead, we consider the case of $n = 2^{11} \cdot 3 = 6144$, using the formulas in Section 6, we get Table 7.

	M	1	2	3	4	5	6	7	8	9	10
	$\log q$	66	77	102	127	153	176	200	224	249	274
$n = 2^{11} \cdot 3$	$\log qC$	91	112	144	177	211	242	275	307	340	374
	λ	265	205	153	121	100	87	77	69	63	58

Table 7. Non power-of-two estimates

Similarly to the case of $n' = 2^{13}$, these estimates tell us that it is possible to support the circuit with three multiplications (the green cells in Table 7), only this time we have $\lambda = 153$ instead of 209. This happens because the dimension of the lattice n is smaller, as $2^{11} \cdot 3 = 6144 < 2^{13} = 8192$. Hence, Circuit 1 with $M = 3$ is our first corner case.

Our performance comparison is essentially a systematic extension of what is just seen in Example 1 to different circuits. We focus on rings with cyclotomic index $m = 2^s \cdot 3^2$, meaning the quotienting polynomial is of the form $\Phi_m(x) = x^{2^s \cdot 3} - x^{2^{s-1} \cdot 3} + 1$, for essentially two reasons. The first is that in this setting, we can always deploy the NTT algorithm described in Section 3.3, and hence, we can be competitive with the power-of-two setting in terms of computational costs. The second reason is that the degree of the polynomial (and hence the dimension of the lattice used for the security assessment) is exactly halfway through two consecutive powers of two, which is a reasonable starting point to look for corner cases. In fact for any s we have $2^{s+1} < 2^s \cdot 3 < 2^{s+2}$ and $2^s \cdot 3 - 2^{s+1} = 2^{s+2} - 2^s \cdot 3 = 2^s$, meaning we can expect the security of the construction with $n = 2^s \cdot 3$ to be halfway between the two neighbouring power-of-two constructions.

7.1 Circuit 1

We conclude the work started in Example 1 with a full comparison for Circuit 1: we recall this is a basic construction, with only an addition in each building block. We merge and extend Table 6 and Table 7 in Table 8, and get an extensive study involving all multiplicative levels from 1 to 10. For each value of M, we highlight in green the instances optimal with respect to the security threshold $\lambda = 128$. It can be seen how for $\mathrm{M} = 3, 6, 7, 8$, the optimal estimate is achieved by a non-power-of-two construction; hence, we have 4 corner cases.

	M	1	2	3	4	5	6	7	8	9	10
$n = 2^{11}$	$\log q$	**43**	65	87	109	131	154	176	199	222	245
	$\log qC$	**67**	96	126	156	185	216	246	276	307	338
	λ	**161**	104	78	63	53	45	44	44	44	44
$n = 2^{10} \cdot 3$	$\log q$	55	76	99	122	145	168	192	216	239	253
	$\log qC$	78	108	139	170	201	232	264	296	327	349
	λ	146	101	77	64	54	46	45	45	45	45
$n = 2^{12}$	$\log q$	46	**68**	91	115	138	161	185	209	232	256
	$\log qC$	70	**100**	131	163	194	225	257	288	320	341
	λ	203	**137**	103	83	70	61	54	48	45	45
$n = 2^{11} \cdot 3$	$\log q$	66	77	**102**	127	153	176	200	224	249	274
	$\log qC$	91	112	**144**	177	211	242	275	307	340	374
	λ	265	205	**153**	121	100	87	77	69	63	58
$n = 2^{13}$	$\log q$	48	71	96	**119**	**144**	168	193	208	240	267
	$\log qC$	72	104	137	**169**	**202**	235	268	301	334	367
	λ	436	286	209	**165**	**136**	116	101	90	81	75
$n = 2^{12} \cdot 3$	$\log q$	59	83	108	133	158	**183**	**209**	**235**	260	286
	$\log qC$	85	117	151	185	218	**252**	**287**	**321**	355	389
	λ	652	436	318	249	205	**173**	**149**	**132**	119	108
$n = 2^{14}$	$\log q$	49	74	99	124	150	175	201	227	**235**	**278**
	$\log qC$	76	108	142	176	210	244	279	314	**330**	**382**
	λ	925	618	449	350	285	240	206	180	**171**	**145**

Table 8. Study of the estimates for Circuit 1.

7.2 Circuit 2

As a second example, we consider a more complex circuit: we allow for constant multiplication, followed by eight rotations ($\tau = 8$). We also increase the number of sums from one to eight with respect to Circuit 1 ($\eta = 9$) while we leave the plaintext modulus unchanged ($t = 64$). We obtain the estimates in Table 9. Again, for each value of M, we highlight in green the optimal instances with respect to the security threshold $\lambda = 128$. This time, we find 7 corner cases.

	M	1	2	3	4	5	6	7	8	9	10
$n = 2^{12}$	$\log q$	80	136	193	250	307	364	421	479	536	594
	$\log qC$	127	202	278	354	431	507	583	660	737	814
	λ	107	68	51	46	45	45	45	45	45	45
$n = 2^{11} \cdot 3$	$\log q$	82	131	182	232	283	334	384	435	487	538
	$\log qC$	124	191	259	326	394	462	529	597	667	735
	λ	182	111	81	65	55	47	46	46	46	46
$n = 2^{13}$	$\log q$	83	141	200	259	318	377	436	496	555	615
	$\log qC$	131	209	288	367	446	525	604	684	762	842
	λ	220	131	95	75	63	54	47	46	46	46
$n = 2^{12} \cdot 3$	$\log q$	86	137	190	242	295	348	401	454	507	560
	$\log qC$	130	204	269	339	410	481	552	622	693	764
	λ	383	222	161	124	102	87	76	68	62	57
$n = 2^{14}$	$\log q$	86	146	207	268	329	390	451	513	575	636
	$\log qC$	135	216	298	379	461	542	624	707	789	870
	λ	476	276	191	147	119	101	88	78	71	65
$n = 2^{13} \cdot 3$	$\log q$	89	142	197	251	306	361	416	471	526	581
	$\log qC$	134	206	279	352	425	499	572	645	719	793
	λ	838	493	341	259	208	173	149	131	117	105
$n = 2^{15}$	$\log q$	89	151	214	277	340	403	467	530	594	657
	$\log qC$	140	223	307	392	476	560	645	730	815	899
	λ	1080	601	413	310	248	206	176	153	136	123
$n = 2^{14} \cdot 3$	$\log q$	91	147	203	260	316	371	427	490	548	605
	$\log qC$	137	213	288	364	439	513	588	671	748	824
	λ	1850	1079	747	562	449	372	317	271	239	214

Table 9. Study of the estimates for Circuit 2.

8 Conclusions and Future Work

8.1 Conclusions

With this work, we showed how it can be more convenient to implement the BGV scheme over non-power-of-two cyclotomic rings in order to get better parameters for specific instantiations. In the process, we established many useful results.

Although it is a widely used fact, we could not find in the literature a satisfying proof for Theorem 2. Therefore, ours is the first formal demonstration of such a statement. The bounds on the variance of the product rings were essentially already established in [39] and [48] for the cases $m = 2^s$ and $m = 2^s 3^t$, respectively. However, we could not find any general results regarding the covariance matrices. While this matter is straightforward for the power-of-two case (see [39]), the same cannot be said for the case where the cyclotomic index is $m = 2^s 3^t$. We think Theorem 3 is a very interesting result because it shows how to compute the full covariance matrix in this case. Moreover, its proof is easily generalized to other cyclotomic rings, since it relies on CRT and probability theory. This result sheds some light on some properties that seem to characterize RLWE with respect to LWE: it makes no sense to perform a similar analysis in the LWE context because the algebraic structure is too simple, and the analogue of polynomial product is just multiplication in $\mathbb{Z}_q$.

Another topic we explored is the techniques used for noise estimation. We showed how to compute the worst-case canonical norm estimates in our non-power-of-two setting and obtained an improvement over state-of-the-art methods for constant multiplication (Equation (28)). The results of the estimations themselves are quite promising: in Section 7, we examine various sets of parameters and show a number of instances where it is recommendable to choose a non-power-of-two construction to achieve certain circuit and security targets (*corner cases*). This is mainly connected with the availability of efficient NTT algorithms, which are non-trivial to develop. However, at least for the case $m = 2^s \cdot 3^2$, we could find some solid ground for our idea to grow, yielding some concrete proposals for alternatives to power-of-two BGV. We point out that all the corner cases we find in Section 7 show a significant improvement with respect to the power-of-two they outperform. More precisely, the size of the modulus q is similar, and the NTT algorithms have comparable performance, but we have vectors whose length n is 25% shorter. This affects the quantity of memory needed to run the cryptosystem, and also makes it more agile. Indeed, the complexity of all the operations, including polynomial products that are the main bottleneck, depends on the degree of the cyclotomic ring $\mathcal{R}_q$. Hence reducing this degree affects the running time of all the algorithms, simply because we are using a *smaller* instantiation of BGV. From a computational standpoint, this enriches the landscape of possible solutions for somebody implementing FHE.

8.2 Future work

Although we showed how it is possible to obtain better parameters for BGV by also considering cyclotomic rings with index $m = 2^s \cdot 3^t$, if we look at the comparison tables in Section 7, we can see how there still are some big jumps in our estimates. For example, if we consider Table 9, we can see that to achieve 5 multiplications with $\lambda = 128$ we need to jump from $n = 2^{14}$ to $n = 2^{13} \cdot 3$, with λ increasing to 208 bits. This is again an overkill for an instantiation of BGV, meaning that if we could find a cyclotomic ring of degree $2^{14} < n < 2^{13} \cdot 3$ with efficient NTT then maybe we would also achieve more optimal parameters. A good direction for further work could be explored in cases where $m = 2^s \cdot 3^t$ with $t > 2$.

Another idea could be extending the estimates to cyclotomic rings with $m \neq 2^s$ or $2^s 3^t$; this would also involve generalizing Theorem 3 to new cases. The proof of Theorem 3 relies essentially on the Chinese Remainder Theorem and probability theory, and it seems that it can be extended to other quotient rings. This looks like a promising topic of self-standing interest in theoretical cryptography, also connected to understanding the extra layer of algebraic structure introduced by considering RLWE instead of LWE.

Another interesting line of research are the practical consequences of the existence of corner cases: as showed by [48], the interest for alternatives to power-of-two cyclotomic rings is growing in the lattice-based community. We showed that corner cases exist: the practical implications of this should be explored by optimizing and tailoring implementations to the new settings.

References

1. Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)*, 51(4):1–35, 2018.
2. Martin R Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption security standard. Technical report, Toronto, Canada, 2018.
3. Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of Learning with Errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.
4. Jean-Claude Bajard, Julien Eynard, Anwar Hasan, Paulo Martins, Leonel Sousa, and Vincent Zucca. Efficient reductions in cyclotomic rings - Application to Ring-LWE based FHE schemes. In *Selected Areas in Cryptography – SAC 2017*, pages 151–171, 2018.
5. Jean-Claude Bajard, Julien Eynard, M Anwar Hasan, and Vincent Zucca. A full RNS variant of FV like somewhat homomorphic encryption schemes. In *International Conference on Selected Areas in Cryptography*, pages 423–442. Springer, 2016.
6. Loris Bergerat, Anas Boudi, Quentin Bourgerie, Ilaria Chillotti, Damien Ligier, Jean-Baptiste Orfila, and Samuel Tap. Parameter Optimization and Larger Precision for (T)FHE. *Journal of Cryptology*, 36(3):28, 2023.
7. Daniel J Bernstein. Multidigit multiplication for mathematicians. *Advances in Applied Mathematics*, pages 1–19, 2001.
8. Beatrice Biasioli, Chiara Marcolla, Marco Calderini, and Johannes Mono. Improving and Automating BFV Parameters Selection: An Average-Case Approach. *Cryptology ePrint Archive, Paper 2023/600*, 2023.
9. Patrick Billingsley. *Probability and measure*. 2008.
10. Leo Bluestein. A linear filtering approach to the computation of discrete fourier transform. *IEEE Transactions on Audio and Electroacoustics*, 18(4):451–455, 1970.
11. Jean-Philippe Bossuat, Rosario Cammarota, Jung Hee Cheon, Ilaria Chillotti, Benjamin R Curtis, Wei Dai, Huijing Gong, Erin Hales, Duhyeong Kim, Bryan Kumara, et al. Security guidelines for implementing homomorphic encryption. *Cryptology ePrint Archive*, 2024.
12. Zvika Brakerski. Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In *Advances in Cryptology – CRYPTO 2012*, pages 868–886, 2012.
13. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) Fully Homomorphic Encryption without bootstrapping. *ACM Transactions on Computation Theory – TOCT 2014*, 6(3):1–36, 2014.
14. Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Advances in Cryptology–CRYPTO 2011*, pages 505–524, 2011.

15. Jung Hee Cheon, Anamaria Costache, Radames Cruz Moreno, Wei Dai, Nicolas Gama, Mariya Georgieva, Shai Halevi, Miran Kim, Sunwoong Kim, Kim Laine, et al. Introduction to homomorphic encryption and schemes. *Protecting Privacy through Homomorphic Encryption*, pages 3–28, 2021.
16. Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. A full RNS variant of approximate homomorphic encryption. In *International Conference on Selected Areas in Cryptography – SAC 2018*, pages 347–368, 2018.
17. Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology – ASIACRYPT 2017*, pages 409–437, 2017.
18. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *Advances in Cryptology – ASIACRYPT 2016*, pages 3–33, 2016.
19. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: Fast Fully Homomorphic Encryption over the Torus. *Journal of Cryptology*, 33(1):34–91, 2020.
20. Ilaria Chillotti, Damien Ligier, Jean-Baptiste Orfila, and Samuel Tap. Improved programmable bootstrapping with larger precision and efficient arithmetic circuits for tfhe. In *Advances in Cryptology–ASIACRYPT 2021*, pages 670–699. Springer, 2021.
21. James W Cooley and John W Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, 19(90):297–301, 1965.
22. Ana Costache and Nigel P Smart. Which ring based somewhat homomorphic encryption scheme is best? In *Topics in Cryptology – CT-RSA 2016: Cryptographers' Track at the RSA Conference*, pages 325–340, 2016.
23. Anamaria Costache, Benjamin R Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, and Rachel Player. On the precision loss in approximate homomorphic encryption. pages 325–345, 2023.
24. Anamaria Costache, Kim Laine, and Rachel Player. Evaluating the effectiveness of heuristic worst-case noise analysis in FHE. In *Computer Security – ESORICS 2020*, pages 546–565, 2020.
25. Anamaria Costache, Lea Nürnberger, and Rachel Player. Optimisations and Trade-offs for HElib. In *Topics in Cryptology – CT-RSA 2023: Cryptographers' Track at the RSA Conference*, pages 29–53, 2023.
26. David A Cox. *Galois theory*, volume 61. 2011.
27. Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Advances in Cryptology – CRYPTO 2012*, pages 643–662, 2012.
28. David Du Pont, Jonas Bertels, Furkan Turan, Michiel Van Beirendonck, and Ingrid Verbauwhede. Hardware acceleration of the prime-factor and rader ntt for bgv fully homomorphic encryption. *Cryptology ePrint Archive*, 2024.
29. Léo Ducas and Alain Durmus. Ring-lwe in polynomial rings. In *Public Key Cryptography – PKC 2012*, pages 34–51, 2012.
30. Léo Ducas and Daniele Micciancio. FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second. In *Advances in Cryptology – EUROCRYPT 2015*, pages 617–640, 2015.
31. Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *Cryptology ePrint Archive*, 2012.
32. Robin Geelen, Michiel Van Beirendonck, Hilder V. L. Pereira, Brian Huffman, Tynan McAuley, Ben Selfridge, Daniel Wagner, Georgios D. Dimou, Ingrid

Verbauwhede, Frederik Vercauteren, and David W. Archer. BASALISC: programmable hardware accelerator for BGV fully homomorphic encryption. *IACR Transactions on Cryptographic Hardware and Embedded Systems – TCHES 2023*, 2023(4):32–57, 2023.
33. C Gentry. A Fully Homomorphic Encryption scheme. Ph.D. thesis, Stanford University, 2009. 2009.
34. Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully Homomorphic Encryption with Polylog Overhead. In *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 465–482, 2012.
35. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In *Advances in Cryptology–CRYPTO 2012*, pages 850–867, 2012.
36. Shai Halevi and Victor Shoup. Design and implementation of HElib: a homomorphic encryption library. *Cryptology ePrint Archive*, 2020.
37. Kyoohyung Han and Dohyeong Ki. Better bootstrapping for approximate homomorphic encryption. In *Topics in Cryptology–CT-RSA 2020: The Cryptographers' Track at the RSA Conference 2020, San Francisco, CA, USA, February 24–28, 2020, Proceedings*, pages 364–390, 2020.
38. Ray S Hoyt. Probability functions for the modulus and angle of the normal complex variate. *The Bell System Technical Journal*, 26(2):318–359, 1947.
39. Ilia Iliashenko. Optimisations of fully homomorphic encryption. 2019.
40. Jean Jacod and Philip Protter. *Probability essentials*. 2004.
41. Andrey Kim, Antonis Papadimitriou, and Yuriy Polyakov. Approximate homomorphic encryption with reduced approximation error. In *Cryptographers' Track at the RSA Conference*, pages 120–144. Springer, 2022.
42. Andrey Kim, Yuriy Polyakov, and Vincent Zucca. Revisiting homomorphic encryption schemes for finite fields. In *Advances in Cryptology–ASIACRYPT 2021*, pages 608–639, 2021.
43. Elena Kirshanova, Chiara Marcolla, and Sergi Rovira. Guidance for efficient selection of secure parameters for Fully Homomorphic Encryption. In *International Conference on Cryptology in Africa*, pages 376–400. Springer, 2024.
44. Serge Lang. *Algebraic number theory*, volume 110. 2013.
45. Rudolf Lidl and Harald Niederreiter. *Finite fields*. Number 20. 1997.
46. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and Learning With Errors over rings. *Journal of the ACM (JACM)*, 60(6):1–35, 2013.
47. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for Ring-LWE cryptography. In *Advances in Cryptology – EUROCRYPT 2013*, pages 35–54, 2013.
48. Vadim Lyubashevsky and Gregor Seiler. NTTRU: Truly Fast NTRU Using NTT. *IACR Transactions on Cryptographic Hardware and Embedded Systems – TCHES 2019*, (3):180–201, 2019.
49. Chiara Marcolla, Victor Sucasas, Marc Manzano, Riccardo Bassoli, Frank HP Fitzek, and Najwa Aaraj. Survey on Fully Homomorphic Encryption, Theory, and Applications. *Proceedings of the IEEE*, 110(10):1572–1609, 2022.
50. Paulo Martins, Leonel Sousa, and Artur Mariano. A survey on fully homomorphic encryption: An engineering perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–33, 2017.
51. Johannes Mono and Tim Güneysu. Implementing and optimizing matrix triples with homomorphic encryption. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, pages 29–40, 2023.

52. Johannes Mono, Chiara Marcolla, Georg Land, Tim Güneysu, and Najwa Aaraj. Finding and evaluating parameters for BGV. In *International Conference on Cryptology in Africa – AFRICACRYPT 2023*, pages 370–394. Springer, 2023.
53. Sean Murphy and Rachel Player. A central limit approach for ring-LWE noise analysis. *IACR Communications in Cryptology*, 1(2), 2024.
54. Chris Peikert. A decade of lattice cryptography. *Foundations and trends in theoretical computer science*, 10(4):283–424, 2016.
55. Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 461–473, 2017.
56. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.