

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

Andrea Di Giusto and Chiara Marcolla

Technology Innovation Institute, Abu Dhabi, United Arab Emirates

Abstract. 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 can sometimes also cause a jump of 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 $n = 2^s \cdot 3^{t-1}$. 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.

Keywords: Fully Homomorphic Encryption, BGV, non-power-of-two, parameter estimation

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.

After Gentry's breakthrough thesis [20], several FHE schemes were proposed. Among all FHE schemes, the most practical, efficient and widely adopted [30] are BGV [6], BFV [5, 18], TFHE [10, 11] which improve [17], and CKKS [8, 9]. In

this work, we focus on the Brakerski-Gentry-Vaikuntanathan (BGV) [6] scheme. BGV can be instantiated using either the integers or cyclotomic rings, yielding a scheme based on Learning with Errors (LWE) or its ring variant (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 random tuples. 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.

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

Powers-of-two are sparse, and, as mentioned before, this is sometimes 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. Promising results have been obtained by applying it to NTRU, as demonstrated in [29].

Our contribution. In this work, we investigate what happens if 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 conditions many different aspects of BGV constructions. The most important ones are the algorithms for the NTT and how modular reductions affect the computation of polynomial products. The first topic has been recently addressed in [29], showing how it is possible to find algorithms that are competitive with the radix-2 NTT. Regarding the second topic, we show 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 exploits the theoretical idea behind the NTT algorithm in [29], suggesting the potential for generalization to other scenarios.

To estimate the parameters, we perform a worst-case analysis using the canonical norm. This analysis for power-of-two BGV was carried out in previous works, such as [12, 14, 31]. Although this is not the most pioneering method, as demonstrated in several papers proposing the average-case analysis on different FHE schemes, such as [3, 10, 13], it is one of the soundest in the literature and

still gives good estimates. Moreover, the purpose of this work is not to improve noise estimation techniques as much as it is to analyze the new framework. In this context, using Ljapunov's Central Limit Theorem [4], 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).

Finally, we compare our results with the power-of-two setting and find that there are many cases where it is recommendable to use non-power-of-two BGV. This is a nice improvement 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 BGV and our techniques for the parameter estimation, including the tools for the non-power-of-two framework.
- In Section 4, we study how the basic operations interact to form the leveled circuits and how this impacts noise estimation.
- In Section 5, we present our results for non-power-of-two parameter estimation, and draw comparisons with the power-of-two instantiations.
- Finally, in Section 6, 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 $[a] = \{0, 1, \dots, a-1\}$.
- 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 $[-\lfloor \frac{q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor]$.
- 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., $\mathbf{a} = (a_0, \dots, a_{n-1})$ where each a_i lies in some ring, $[\mathbf{a}]_q$ indicates the vector $([a_0]_q, \dots, [a_{n-1}]_q)$. $\|\mathbf{a}\|$ is shorthand for the infinity norm of \mathbf{a} .
- Given a random vector $\mathbf{X} = (X_0, \dots, X_{n-1})$, $\mathbf{E}[\mathbf{X}] = (\mathbf{E}[X_0], \dots, \mathbf{E}[X_{n-1}])$ is its expected value and $\text{Var}(\mathbf{X})$ is the vector of variances; for a random vector \mathbf{Y} , $\text{CovM}(\mathbf{X}, \mathbf{Y}) = (\text{Cov}(X_i, Y_i))_{i,j=1, \dots, n}$ 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. 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 $\mathcal{R}_r = \mathcal{R}/(r\mathcal{R})$. With t and q , we denote the plaintext and the ciphertext moduli, respectively. The plaintext space is $\mathcal{R}_t = \mathbb{Z}[x]/(\Phi_m(x))$, while the ciphertext space is $\mathcal{R}_q = \mathbb{Z}[x]/(\Phi_m(x))$, where $\Phi_m(x)$ is the cyclotomic polynomial (see Section 2.2).

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

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

It has degree $n = \phi(m)$, i.e., the Euler's totient function. Let $m = \prod_{i=1}^l p_i^{\alpha_i}$ be a natural number, where p_i are distinct primes. Then the radical $\text{rad}(m)$ is the product of its prime factors, namely, $\text{rad}(m) = \prod_{i=1}^l p_i$.

Lemma 1 ([15]). *For any $m \in \mathbb{N}$ we have $\Phi_m(x) = \Phi_{\text{rad}(m)}(x^{m/\text{rad}(m)})$.*

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

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 ([28]). *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 ([27]). *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 $(a(\zeta^i) : i \in \mathbb{Z}_m^*)$. Ordering the roots appropriately we have $\sigma : K \rightarrow H$, where

$$H = \{(x_1, \dots, x_n) \in \mathbb{R}^{t_1} \times \mathbb{C}^{2t_2} : x_{s_1+i} = \overline{x_{s_1+s_2+i}} \text{ for } i = 1, \dots, s_2\} \subset \mathbb{C}^n \quad (1)$$

and this is a ring homomorphism; by identifying the conjugate couples, we have $H \cong \mathbb{R}^{s_1+s_2}$. The canonical norm $\|\cdot\|^{can}$ is the pull-back of the infinity norm via the canonical embedding: $\|\cdot\|^{can} = \|\cdot\|_{\infty}^{can}$. 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 more extensive background, we refer to [16].

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 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 it enjoys the following properties.

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

1. *for $r = \text{rad}(m)$ 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 $c_m \leq 1.1547$.

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 [25]. 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 ternary distribution \mathcal{T} , having variance $\sigma_{\mathcal{T}}^2 = 2/3$;
- for an odd $q \in \mathbb{N}$, the uniform centered discrete distribution \mathcal{U}_q over \mathbb{Z} or \mathbb{Z}_q with variance is $q^2/2$;
- The continuous Gaussian distribution on \mathbb{R} with variance σ^2 , denoted as $\mathcal{N}_r = \mathcal{N}(0, \sigma^2)$.

Note that in our work we use $\chi_s = \mathcal{T}$ and $\chi_e = \mathcal{DG}(0, \sigma^2)$ with $\sigma = 3.19$.

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, which will be used in an *approximate way* (see [32]) to justify some theoretical results needed for our estimates.

Theorem 1. (*Lyapunov CLT*) *Let $X_0, X_1, \dots, X_j, \dots$ be a sequence of independent random variables each with mean μ_i 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*

$$\lim_{n \rightarrow \infty} \frac{1}{s_n^{2+\delta}} \sum_{j=0}^{n-1} \mathbb{E}[|X_j - \mu_j|^{2+\delta}] = 0. \quad (4)$$

Then

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

Polynomial multiplication The main computational bottleneck in lattice-based cryptosystems is polynomial multiplication, which can be tackled in different ways [2]. In our work, we use the Number Theoretic Transform (NTT) proposed in [29], which is particularly well-suited for our specific requirements.

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

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

Using the fact that $\zeta^5 = \bar{\zeta} = 1 - \zeta$, the CRT isomorphism corresponding to this factorization can be computed with only $n/2$ additions more than a radix-2 NTT layer. After this step, we can proceed with some $s - 1$ radix-2 steps, obtaining 2^s rings of degree smaller than 3^{t-1} . We consider especially the case of $t = 2$, where at this point it is more convenient to compute the product of 2^s polynomials of degree at most 2 rather than using e.g. a radix-3 NTT layer. This is very similar to what done in [29], with the only difference that the exponent s will be bigger than 8 in our case. This way we obtain an algorithm for polynomial multiplication in non power-of-two cyclotomic rings that is competitive with the power-of-two instantiations.

We point out that to have efficient NTT algorithms we need the modulus q to be congruent to 1 modulo m .

3 The BGV Scheme and Noise Estimation

In this section, we present the BGV scheme [6] and our techniques to perform noise estimation in the setting of a non-power-of-two cyclotomic ring. We perform a worst-case analysis based on the canonical norm.

3.1 The BGV cryptosystem

BGV functionalities can be divided into two main categories: the basic encryption scheme, including key generation, encryption and decryption, and the homomorphic operations.

Basic encryption scheme The three basic algorithms of BGV are as follows:

- Key generation ($\text{KeyGen}(\lambda)$): 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 $\text{sk} = s$ and the public key $\text{pk} = (b, a) = [(-a \cdot s + te, a)]_{q_L}$;
- Encryption ($\text{Enc}_{\text{pk}}(m)$): given a plaintext $m \in \mathcal{R}_t$ and the public key $\text{pk} = (b, a)$, sample $u \leftarrow \chi_s$, $e_0, e_1 \leftarrow \chi_e$ and output $\mathbf{c} = (c_0, c_1, \nu)$ where the ciphertext is

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

and l and ν are quantities related to noise management, whose role we explain below. The triad \mathbf{c} is called *extended ciphertext*.

- Decryption ($\text{Dec}_{\text{sk}}(\mathbf{c})$): 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 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.

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

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

and this quantity increases through homomorphic operations [14]. 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 quantity $||\nu||$, called *noise*.

In this work, we consider 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

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

This representation allows using native data types for integers because the p_i s 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 $A = a_0 \dots a_k$, $B = b_0 \dots b_{k'}$ are the two product decompositions used for RNS. For this purpose, we need a Fast Base Extension (FBE) algorithm [19]. Namely, if $\mathbf{a} \in \mathcal{R}_{a_0} \times \dots \times \mathcal{R}_{a_k}$, then

$$\text{FBE}(\mathbf{a}, A, B) = \left[\sum_{j=0}^k \left[\mathbf{a} \left(\frac{A}{a_j} \right)^{-1} \right]_{a_j} \frac{A}{a_j} \right]_{b_j}. \quad (8)$$

Homomorphic operations We introduce the three homomorphic ring operations (addition, multiplication and constant multiplication) and two key sub-routines (key switching and modulus switching).

- The addition $\text{Add}(\mathbf{c}, \mathbf{c}')$ is defined as

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

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 $\text{Mul}(\mathbf{c}, \mathbf{c}')$ outputs

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

where $\mathbf{c}'' = (c''_0, c''_1, c''_2)$ represents the product of $c(s)$ and $c'(s)$. 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 $\mathbf{c}'' = (c''_0, c''_1, c''_2) \in \mathcal{R}_{q_l}^3$ back to a ciphertext $\bar{\mathbf{c}} = (\bar{c}_0, \bar{c}_1) \in \mathcal{R}_{q_l}^2$ (see Equation (16)). The critical quantity ν_{Mul} is well posed, since

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

We point out that in this operation, the noise growth is multiplicative, which is the worst case among basic operations.

- The constant multiplication $\text{ConstMul}(\alpha, \mathbf{c})$ defined as

$$\text{ConstMul}(\alpha, \mathbf{c}) = ((\alpha \cdot c_0, \alpha \cdot c_1), l, \alpha \cdot \nu) = (\alpha \cdot \mathbf{c}, l, \nu_{\text{ConstMul}}), \quad (11)$$

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

$$\text{ModSw}(\mathbf{c}, l') = \left(\mathbf{c}' = \left[\frac{q_{l'}}{q_l} (\mathbf{c} + \delta) \right]_{q_{l'}}, l', \nu_{\text{ModSw}} \right). \quad (12)$$

The polynomial δ is a correction term computed as

$$\delta = t[-t^{-1}\mathbf{c}]_{q_l/q_{l'}} = t[(t^{-1}c_0, t^{-1}c_1)]_{q_l/q_{l'}} , \quad (13)$$

and it is formulated only to affect the errors. It makes the ciphertext divisible by $q_l/q_{l'}$, allowing it to descend in the moduli ladder from q_l to $q_{l'}$. The formal proof of why this procedure reduces the noise is in [6, Lemma 5]. If we consider only one-step modulus switching, i.e., $k = 1$ and $l' = l - 1$, then $q_l/q_{l'} = 1/p_l$ and we have

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

Hence the critical quantity for modulus switching is

$$\nu_{\text{ModSw}} = \frac{\nu + \delta(s)}{p_l} . \quad (14)$$

The last procedure that we are going to analyse 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 rotation. For multiplication, we convert the ciphertext term $c_2'' \cdot s^2$ to a polynomial $c_0^{\text{ks}} + c_1^{\text{ks}} \cdot s$, and for a rotation, we convert the ciphertext term $c_1 \cdot \text{rot}(s)$ to a polynomial $c_0^{\text{ks}} + c_1^{\text{ks}} \cdot s$. In the following, we will only analyze multiplication. This procedure can be divided into two parts, a key generation (**KeySwGen**) that *somehow* encrypts s^2 under s itself 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

$$\text{KeySwGen}(s, s^2) = \mathbf{ks} = (\mathbf{ks}_0, \mathbf{ks}_1) = [(-a \cdot s + te + s^2, a)]_{q_l} . \quad (15)$$

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

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

and outputs

$$\text{KeySw}(\mathbf{ks}, \mathbf{c}) = \mathbf{c}' = (\mathbf{c}', l, \nu_{\text{KeySw}}) . \quad (16)$$

The critical quantity after this operation is $\nu_{\text{KeySw}} = \nu + tc_2 \cdot e$. Unfortunately, if we tried to compute $\|\nu_{\text{KeySw}}\|$, especially after a few homomorphic operations have been performed, it becomes evident that the noise growth introduced by the term $tc_2 \cdot e$ in the critical quantity is too big. To address this issue, several variations of the **KeySw** procedure have been developed, aiming to control the growth of noise introduced during computations effectively. We focus on the Hybrid variant presented in [21], called so because it is a mix of the BV [7] and the GHS [21] variants. From the former we need the following

decompositions: let $b \in \mathbb{N}$ be a basis, then for $k = \lceil \log_b q_l \rceil + 1$ and any $\alpha \in \mathcal{R}_{q_l}$, if we define

$$D_b(\alpha) = ([\alpha]_b, [[\alpha/b]]_b, [[\alpha/b^2]]_b, \dots, [[\alpha/b^{k-1}]]_b) \quad (17)$$

$$P_b(\alpha) = ([\alpha]_{q_l}, [b\alpha]_{q_l}, [b^2\alpha]_{q_l}, \dots, [b^{k-1}\alpha]_{q_l}) . \quad (18)$$

Then for any $\alpha, \beta \in \mathcal{R}_q$ we obtain $\langle D_b(\alpha), P_b(\beta) \rangle = \alpha \cdot \beta$ [26]. 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 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

$$\text{KeySwGen}^{\text{Hybrid}}(s, s^2) = \text{ks}^{\text{Hybrid}} = [(-\mathbf{a} \cdot s + t\mathbf{e} + CP_b(s^2), \mathbf{a})]_{Q_l} \quad (19)$$

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

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

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

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

Finally, the output of the Hybrid key switching is

$$\text{KeySw}^{\text{Hybrid}}(\text{ks}^{\text{Hybrid}}, \mathbf{c}) = (\mathbf{c}'', l, \nu_{\text{ks}}^{\text{Hybrid}}) \quad (20)$$

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

$$\nu_{\text{KeySw}}^{\text{Hybrid}} = \nu + \frac{t \langle D_b(c_2), \mathbf{e} \rangle + \delta(s)}{C} . \quad (21)$$

The Hybrid key switching achieves better efficiency than the BV and better noise management than GHS, and it is for this reason the preferred one when it comes to implementations [23].

3.2 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. We remark that by random polynomial we mean a polynomial whose coefficients are sampled independently from some distribution.

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. Similar bounds were already provided in previous works (e.g. [12, 24]). In our paper, we take an additional step to provide comprehensive proof to support our findings.

Theorem 2. Let $a(x) = \sum_{i=0}^n a_i x^i \in \mathcal{R}$ be a random polynomial whose coefficients are identically distributed, having mean zero, finite variance V_a and that there exists some $\delta > 0$ such that

$$E[|a_j|^{2+\delta}] < \gamma_1 \in \mathbb{R}_{>0}. \quad (22)$$

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

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

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

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

Now we show that $a(\zeta)$ has 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

$$\begin{aligned} \text{Var}(W_j) &= (\eta \cos(\alpha j) + \rho \sin(\alpha j))^2 \text{Var}(a_j) = \\ &= \eta^2 \cos^2(\alpha j) + \rho^2 \sin^2(\alpha j) + 2\eta\rho \cos(\alpha j) \sin(\alpha j) V_a. \end{aligned}$$

This implies

$$\begin{aligned} s_n^2 &= \sum_{j=0}^{n-1} \text{Var}(W_j) = \sum_{j=0}^{n-1} (\eta^2 \cos^2(\alpha j) + \rho^2 \sin^2(\alpha j) + 2\eta\rho \cos(\alpha j) \sin(\alpha j)) V_a = \\ &= \sum_{j=0}^{n-1} (\eta^2 \cos^2(\alpha j) + \rho^2 \sin^2(\alpha j)) V_a + 2\eta\rho V_a \sum_{j=0}^{n-1} \cos(\alpha j) \sin(\alpha j) = \\ &= \sum_{j=0}^{n-1} (\eta^2 \cos^2(\alpha j) + \rho^2 \sin^2(\alpha j)) V_a. \end{aligned}$$

The last equality is motivated by the fact that since ζ is a primitive root of unity, $\sum_{j=0}^{n-1} \cos(\alpha j) \sin(\alpha j) = 0$. Indeed, for any j such that $\cos(\alpha j) \sin(\alpha j) \neq 0$ in the sum above, we have that also $-\cos(\alpha j) \sin(\alpha j)$ appears in the sum since it is the term related to the complex conjugate of $\cos(\alpha j) + i \sin(\alpha j)$. These couples cancel each other out in the sum, and there are no other nonzero elements because $\{\zeta^j : j = 0, \dots, n-1\}$ is symmetric with respect to the real axis, and each element is used exactly once. Notice that, except for trivial cases, $(\eta^2 \cos^2(\alpha j) + \rho^2 \sin^2(\alpha j))$ is positive. Moreover, there exists $\gamma_2 \in \mathbb{R}_{>0}$ such that

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

and hence $s_n^2 > n\gamma_2 V_a$. Let $\delta > 0$ be such that $E[|a_j|^{2+\delta}] < \gamma_1$. 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

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

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

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

and hence we can state (always in an approximate way)

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

Then we can use Lyapunov's CLT to state that $\eta X + \rho Y$ is very well approximated by a Gaussian $\forall \eta, \rho \in \mathbb{R}$, hence X and Y are jointly Gaussian and the random vector Z is Gaussian.

A further step that can be taken is to prove that X and Y are uncorrelated; by Lemma 7, this also implies that they are independent. In fact, we have

$$\begin{aligned} \text{Cov}(X, Y) &= E\left[\left(\sum_{j=0}^{n-1} a_j \cos(\alpha j)\right)\left(\sum_{j=0}^{n-1} a_j \sin(\alpha j)\right)\right] \\ &= \sum_{j,k=0}^{n-1} \text{Cov}(a_j, a_k) \cos(\alpha j) \sin(\alpha k) = V_a \sum_{j=0}^{n-1} \cos(\alpha j) \sin(\alpha j) = 0. \end{aligned}$$

□

If we take a random polynomial $a \in \mathcal{R}_q$, the condition in Equation (22) is easily satisfied since the distributions of the coefficients are bounded. Hence we can use Theorem 2 to derive a bound on $\|a\|^{can}$ in the following way. For a complex

centered Gaussian random variable Z with variance V_Z , for any $B \in \mathbb{R}_{>0}$ we have $|Z| > B$ with probability $\exp(-B^2/V_Z^2)$. This result can be obtained by integrating the chi-squared distribution. In our case then $Z = a(z)$, $V_Z = nV_a$ and by setting $B = D\sqrt{V_Z}$ we get

$$P(|a(z)| \leq D\sqrt{V_Z}) = e^{-D^2} .$$

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

$$\|a\|^{can} < D\sqrt{nV_a} \tag{23}$$

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

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 [12, 31]) 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 [24], 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 [29, Section 3.2], the authors give a bound on the variance by making some considerations on the behaviour of the product, finding

$$V_c \leq \frac{3}{2}nV_aV_b . \tag{24}$$

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 generalisation of the result in [29], as we compute all the variances exactly and not only an upper bound, and gives deep hindsight on the behaviour of random polynomials. 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

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

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

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

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 [24], one formula is sufficient to express all the coefficients. For this reason, straightforward substitution does not enable us to compute $\text{Cov}(c_i, c_j)$ and so we need the following theorem.

Theorem 3. *Let $c(x) = a(x)b(x)$ be the product of two random polynomials 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 $n/2$:*

$$\text{CovM}(\mathbf{c}) = \begin{pmatrix} \text{Diag}(\alpha_0, \dots, \alpha_{n/2-1}) & \text{Diag}(\beta_0, \dots, \beta_{n/2-1}) \\ \text{Diag}(\beta_0, \dots, \beta_{n/2-1}) & \text{Diag}(\alpha_{n/2}, \dots, \alpha_{n-1}) \end{pmatrix} \quad (25)$$

where

$$\alpha_k = \begin{cases} (\frac{3}{2}n - (k+1)) V_a V_b & \text{if } 0 \leq k < n/2 \\ \frac{3}{2}n V_a V_b & \text{if } n/2 \leq k < n \end{cases}$$

$$\beta_k = (k+1-n) V_a V_b \quad 0 \leq k < n/2 .$$

Notice how the bound in Equation (24) follows immediately from the theorem: the variances of the coefficients are the values α_i in the matrix above.

Proof. The fundamental construction in this proof is the radix-6 NTT isomorphism (Section 2.2)

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

where $\zeta = 1/2 + \sqrt{3}/2i$ is a complex primitive 6th 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:

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

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 2.2). 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}^ℓ and \mathcal{R}^r is that their quotienting polynomial is of the form $x^\alpha + \text{constant}$, 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}^ℓ 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 $i = 0, \dots, n/2 - 1$. 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

$$\begin{aligned} \mathbb{E}[a_i^\ell] &= \mathbb{E}[a_i^r] = 0 \\ \text{Var}(a_i^\ell) &= \text{Var}(a_i^r) = \mathbb{E}[(a_i + \zeta a_{i+n/2})(\overline{a_i + \zeta a_{i+n/2}})] \\ &= \mathbb{E}[a_i^2 + \zeta a_i a_{i+n/2} + (1 - \zeta)a_i a_{i+n/2} + \zeta(1 - \zeta)a_{i+n/2}^2] \\ &= \mathbb{E}[a_i^2] + \mathbb{E}[a_i a_{i+n/2}] + \mathbb{E}[a_{i+n/2}^2] = 2V_a. \end{aligned} \quad (26)$$

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_i s, 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 $i \neq j$, $\text{Cov}(a_i^\ell, a_j^\ell) = \text{Cov}(a_i^\ell, a_j^r) = \text{Cov}(a_i^r, a_j^r) = 0$ and the only nonzero covariances are given by

$$\begin{aligned} \text{Cov}(a_i^\ell, a_i^r) &= \mathbb{E}[(a_i + \zeta a_{i+n/2})(\overline{a_i + (1 - \zeta)a_{i+n/2}})] \\ &= \mathbb{E}[(a_i + \zeta a_{i+n/2})^2] = (1 + \zeta^2)V_a = \zeta V_a. \end{aligned} \quad (27)$$

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

2. *Product in \mathcal{R}^ℓ 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^ℓ s, implying

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

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

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

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

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

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

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

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

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

Thus, we can substitute Equations (26) and (27) obtaining

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

Hence

$$\text{Var}(c_k^l) = \text{Var}(c_k^r) = \frac{n}{2} \cdot 2V_a \cdot 2V_b = 2nV_aV_b \quad (28)$$

and

$$\begin{aligned} \text{Cov}(c_k^l, c_k^r) &= (k+1)\zeta V_a \zeta V_b + \zeta^2 \left(\frac{n}{2} - (k+1) \right) \zeta V_a \zeta V_b \\ &= (\zeta^2 + \zeta)(k+1)V_aV_b - \zeta \frac{n}{2} V_aV_b. \end{aligned} \quad (29)$$

3. *The isomorphism Ψ^{-1} .* The inverse NTT butterfly operation in [29] is given by the following matrix-vector product: for any $k = 0, \dots, n/2 - 1$

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

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

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

Regarding the exact formulas for the nonzero terms in $\text{CovM}(\mathbf{c})$, we have different cases according to k . Notice that $(1-2\zeta)^{-1}(1/1-2\zeta)^{-1} = 1/3$; moreover, for any $z \in \mathbb{C}$ we have $z + \bar{z} = 2\Re(z)$, and by the properties of covariance $\text{Cov}(X, Y) = \overline{\text{Cov}(Y, X)}$.

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

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

Substituting Equations (26) and (27), we get

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

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

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

Since $\zeta^2 + \zeta = \sqrt{3}i$ has real part equal to 0 and thanks to Equations (26) and (27), we have:

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

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

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

and substituting Equations (28) and (29) we get

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

□

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 coherent with the bound on the variance of a random product in [24].

Theorem 4. *Let $c(x) = a(x)b(x)$ be the product of two random polynomials 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 diagonal, and in particular $\text{CovM}(\mathbf{c}) = \text{Diag}(nV_aV_b)$.*

3.3 Noise estimates for homomorphic operations

In this section, we develop the noise bounds for the operations described in Section 3.1 with the aid of the results in Section 3.2. The main properties we use are the following.

- 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 (23) 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 $\gamma \in \mathbb{Z}_q$

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

In the literature, this approach is also referred to as a *worst-case* canonical embedding analysis. A similar work for the power-of-two case is [31].

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 ,

$$\begin{aligned} \|\nu\|^{can} &\leq 4\sqrt{3nV_{m+t(e \cdot u + e_1 \cdot s + e_0)}} \\ &\leq 4\sqrt{3n \left(\frac{t^2}{12} + t^2 \left(\frac{3}{2}nV_e V_u + \frac{3}{2}nV_{e_1} V_s + V_{e_0} \right) \right)} \\ &\leq 4t\sqrt{3n \left(\frac{1}{12} + 3nV_e V_s + V_e \right)} = \mathbf{B}_{\text{clean}}. \end{aligned} \tag{30}$$

By this computation, we set $\mathbf{B}_{\text{clean}}$ 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

$$\|\nu_{\text{Add}}\|^{can} = \|\nu + \nu'\|^{can} \leq \|\nu\|^{can} + \|\nu'\|^{can} \tag{31}$$

and this actually applies to any sum of polynomials.

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

$$\|\nu_{\text{Mul}}\|^{can} \leq \|\nu\|^{can} \|\nu'\|^{can} \tag{32}$$

which is used to estimate the noise.

Finally, for ConstMul (Equation (11)), the critical quantity is again a polynomial product $\nu_{\text{ConstMul}} = \alpha\nu$. 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 (24). Moreover, we have $V_\alpha = \frac{t^2}{12}$ and $\|\nu\|^{can} \approx 6\sqrt{nV_\nu}$. So

$$\begin{aligned} \|\nu_{\text{ConstMul}}\|^{can} &\leq 6\sqrt{nV_{\alpha\nu}} \leq 6\sqrt{n\frac{3}{2}nV_\alpha V_\nu} \\ &\leq \sqrt{\frac{3}{2}n\frac{t^2}{12}}6\sqrt{nV_\nu} = t\sqrt{\frac{1}{8}n}\|\nu\|^{can}. \end{aligned} \quad (33)$$

This is an improvement on previous bounds, which used again the sub-multiplicativity of the canonical norm.

Modulus switching After the one-step modulus switching, the critical quantity is given by Equation (14) as $\nu_{\text{ModSw}} = \nu + \delta(s)/p_l$, where $\delta(s)$ is as in Equation (13). By using the triangular inequality, we get

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

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} . 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,

$$\|\nu_{\text{ModSw}}\|^{can} \leq \frac{\|\nu\|^{can}}{p_l} + \mathbf{B}_{\text{scale}} \quad \text{where } \mathbf{B}_{\text{scale}} = 6t\sqrt{n\left(\frac{1}{12} + \frac{1}{8}nV_s\right)}. \quad (34)$$

Notice that the term $\mathbf{B}_{\text{scale}}$ 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

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

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 $\|\nu_{\text{ModSw}}\|^{can} \leq \frac{q_\nu}{q_l} \|\nu\|^{can} + \sqrt{k} \mathbf{B}_{\text{scale}}$.

Key switching Performing computations similar to those in [31], it is possible to find the following bounds for the noise in the BV and GHS variants.

Specifically,

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

Instead, for the Hybrid variant, by Equation (21) we have

$$\begin{aligned} \|\nu_{\text{KeySw}}^{\text{Hybrid}}\|^{can} &\leq \left\| \nu + \frac{t\langle D_b(c_2), \mathbf{e} \rangle + \delta(s)}{C} \right\|^{can} \\ &\leq \|\nu\|^{can} + \frac{b\sqrt{\log_b q_l}}{C} \mathbf{B}_{\text{KeySw}} + \mathbf{B}_{\text{scale}} . \end{aligned} \quad (35)$$

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

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

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 .

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

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 $\tilde{q}_0, \dots, \tilde{q}_{h-1}$. This affects the second summand in the GHS estimate, as we have to account for the BV-style decomposition of c_2 : we have

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

and so

$$\begin{aligned} \|\nu_{\text{KeySw}}^{\text{Hybrid-RNS}}\|^{can} &\leq \|\nu\|^{can} + \frac{\sqrt{l+1}}{C} \sqrt{h} \max_{i \in [h]}(\tilde{q}_i) \mathbf{B}_{\text{KeySw}} + \sqrt{k} \mathbf{B}_{\text{scale}} \\ &\leq \|\nu\|^{can} + \sqrt{h(L+1)} \frac{\max_{i \in [h]}(\tilde{q}_i)}{C} \mathbf{B}_{\text{KeySw}} + \sqrt{k} \mathbf{B}_{\text{scale}} . \end{aligned} \quad (36)$$

4 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 (32)). 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 (30)) 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.

4.1 Building blocks

This work studies Model 1 [31, Section 3]: we assume to be working with η ciphertexts $\mathbf{c}_1, \dots, \mathbf{c}_\eta$ in parallel and

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

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

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

We now compute a bound $\mathbf{B}_{\text{block}}$ for the output noise of one such blocks. We analyze the noise growth by assuming that each of the η input ciphertexts $\mathbf{c}_i = (\mathbf{c}_i, l, \nu_i)$ has noise $\|\nu_i\|^{can} < \mathbf{B}$. Then, by Equation (33), after the step 1.,

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

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 3.3, we get that the noise in \mathbf{c}_i^{II} is bounded by

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

The values of γ_0 and γ_1 are given by either Equation (35) or Equation (36) if we are using the RNS representation. Namely, we have

$$(\gamma_0, \gamma_1) = \begin{cases} (b\sqrt{\log_b q_l}, 1) & \text{(Hybrid)} \\ (\sqrt{h(L+1)} \max_{i \in [h]}(\tilde{q}_i), \sqrt{k}) & \text{(Hybrid - RNS)} \end{cases}. \quad (37)$$

The next step in the block is the sum of the η ciphertexts \mathbf{c}_i^{II} ; by Equation (31),

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

Finally, in a building block, two ciphertexts computed as \mathbf{c}^{III} are multiplied together. Then Equation (32) implies

$$\mathbf{B}_{\text{block}} = \eta^2 (\varepsilon \mathbf{B} + \tau v)^2 \quad (38)$$

4.2 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 $\mathbf{B}_{\text{clean}}$ down to the base noise \mathbf{B} . This implies p_{L-1} can be smaller than the other p_i s.
- The *middle moduli* p_i , $i = L-2, \dots, 1$ are used to reduce the noise back to \mathbf{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 (38). After the homomorphic product of ciphertexts concludes the block, we perform two more operations: a key switching to relinearize the result of the product and a modulus switching to reduce the noise. In the Hybrid variant, it is possible to merge these two because in `KeySwHybrid` (Equation (20)), 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 (35) and (36) the condition on \mathbf{B} is

$$\frac{\eta^2 (\varepsilon \mathbf{B} + \tau v)^2}{p_l} + \frac{\gamma_0}{Cp_l} \mathbf{B}_{\text{KeySw}} + \gamma_1 \mathbf{B}_{\text{scale}} < \mathbf{B} . \quad (39)$$

where γ_i are as in (37). Expanding the square in this inequality, we get

$$\frac{\eta^2 (\varepsilon \mathbf{B} + \tau v)^2}{p_l} = \frac{\eta^2 \varepsilon^2}{p_l} \mathbf{B}^2 + \frac{2\eta^2 \varepsilon \tau}{p_l} v \mathbf{B} + \frac{\eta^2 \tau^2}{p_l} v^2 .$$

Following [21], to isolate the terms in \mathbf{B} we let

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

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_l s; moreover, we want this term to be as close as possible to $\mathbf{B}_{\text{scale}}$ (notice that for sure $R_l > \gamma_1 \mathbf{B}_{\text{scale}}$). We can modify C to achieve this goal: letting

$$C > K \gamma_0 \frac{B_{\text{KeySw}}}{\mathbf{B}_{\text{scale}}} \quad (40)$$

for some large $K \in \mathbb{N}$, e.g. $K = 100$, Equation (39) becomes the following inequality in \mathbf{B} :

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

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

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

which results in an estimate for the prime moduli:

$$p_1 \approx \dots \approx p_{L-2} \approx 4\eta^2 \varepsilon \gamma_1 (\tau + \varepsilon) \mathbf{B}_{\text{scale}}. \quad (41)$$

Setting p_l as Equation (41), for each l , we have the $\Delta = 0$. Thus, we recover \mathbf{B}

$$\begin{aligned} \mathbf{B} &\approx - \frac{\left(\frac{2\eta^2 \tau \varepsilon \gamma_1}{p_l} \mathbf{B}_{\text{scale}} - 1 \right)}{2 \frac{\eta^2 \varepsilon^2}{p_l}} = \frac{p_l}{2\eta^2 \varepsilon^2} - \frac{\tau \gamma_1}{\varepsilon} \mathbf{B}_{\text{scale}} \\ &\approx \frac{4\eta^2 \varepsilon \gamma_1 (\tau + \varepsilon) \mathbf{B}_{\text{scale}}}{2\eta^2 \varepsilon^2} - \frac{\tau \gamma_1}{\varepsilon} \mathbf{B}_{\text{scale}} \approx \gamma_1 \left(\frac{\tau}{\varepsilon} + 2 \right) \mathbf{B}_{\text{scale}}. \end{aligned} \quad (42)$$

To conclude our estimates, we bound the constant C in the key switching by looking at the explicit values of γ_0 in Equation (40). For $l = 1, \dots, L - 2$ 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

$$C \geq \begin{cases} Kb\sqrt{\log_b q_{L-2}} \frac{B_{\text{KeySw}}}{\mathbf{B}_{\text{scale}}} & \text{(Hybrid)} \\ K p_{L-2}^{L/h} \sqrt{h(L-1)} \frac{B_{\text{KeySw}}}{\mathbf{B}_{\text{scale}}} & \text{(Hybrid - RNS)} \end{cases} \quad (43)$$

where $K \approx 100$. According to [31], 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 B_{clean} . We want the noise after **ModSw** to be smaller than a threshold B . Following Equation (34) the inequality determining the top modulus p_{L-1} is $B_{\text{clean}}/p_{L-1} + B_{\text{scale}} < B$ and using the approximation in Equation (42) we get

$$p_{L-1} > \frac{B_{\text{clean}}}{((\frac{\tau}{\varepsilon} + 2)\gamma_1 - 1) B_{\text{scale}}} .$$

Bottom modulus At this level, the decryption condition is applied directly to the noise bound for the building block (Equation (38)), resulting in the bound $p_0 = q_0 > 2c_m\eta^2 (\varepsilon B + \tau v)^2$. Since the constant C is quite large, we have

$$v = \gamma_0 \frac{B_{\text{KeySw}}}{C} + \gamma_1 B_{\text{scale}} \approx \gamma_1 B_{\text{scale}} .$$

Moreover, thanks to Equation (42), we have

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

Finally, we get the following condition on p_0 :

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

4.3 Parameters specification

We briefly recall the conditions of the parameters.

- $m = 2^i 3^j$ 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* ([22]), 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 (43).
- $V_e = \sigma^2$, where $\sigma = 3.19$, and $V_s = 2/3$ are the variances of the errors and of the secret key.
- τ is the number of 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 1 and Table 2, we summarize all the results coming from previous sections.

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

Table 1. Intermediate noise bounds

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

Table 2. Sizes of the prime moduli

5 Our Results

5.1 Performance comparison

In this section, we draw a comparison between the power-of-two case and the new setting with cyclotomic index $m = 2^s 3^t$. The estimates for the former case are based on the formulas in [31] and follow the same blueprint of Section 4; 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 estimated using the Lattice Estimator by Albrecht et al. [1]. In Tables 3 to 6, we report both the sizes of the ciphertext modulus q and the modulus qC used in the Hybrid key switching (Section 3.1). 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 4.1) 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 again to this construction, thus we name it Circuit 1. We run the computation for $m = 2^{13}$ and $m = 2^{14}$. Namely, we work in 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 3. 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 3). Considering the power-of-two rings, the only option we have at this

	M	1	2	3	4	5	6	7	8	9	10
$n = 2^{12}$	$\log q$	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
$n = 2^{13}$	$\log q$	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

Table 3. Power-of-two estimates for Circuit 1.

point is to *jump* to $n = 2^{13}$ where a solution with $\lambda = 209$ is available (the green cells in Table 3). Moreover, for this n , it is also possible to use the circuits with $M = 4, 5$ (the blue cells in Table 3) and decrypt after the desired number of multiplications since they also feature $\lambda \geq 128$. Clearly, this approach is suboptimal since it requires significantly larger ciphertext moduli ($\log q$). Consequently, the computational cost of operations in \mathbb{Z}_q increases significantly. The main issue with all these constructions is that increasing the dimension does not come for free. Indeed, n is also the degree of the quotienting polynomial in the ring \mathcal{R}_q where the cryptosystem lives. Hence, by moving from $n = 2^{12}$ to $n = 2^{13}$ we are doubling the length of all the vectors involved. 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 $m = 2^{11} \cdot 3 = 6144$, using the formulas in Section 4 we get Table 4.

	M	1	2	3	4	5	6	7	8	9	10
$n = 2^{11} \cdot 3$	$\log q$	59	72	95	120	146	169	193	217	242	267
	$\log qC$	84	105	137	170	204	235	268	300	333	367
	λ	303	203	151	120	100	86	76	69	63	57

Table 4. 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 4), only this time we have $\lambda = 151$ instead of 209. This happens because the dimension of the lattice n is smaller, as $2^{11} \cdot 3 = 6144 < 2^{13} = 8192$. The length of vectors is reduced by 25%, while maintaining comparable efficiency for the NTT with the help of the technique explained in [29].

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 2.2, 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.

Circuit 1 We conclude the work started in Example 1 with a full comparison for Circuit 1: we recall this is one of the most basic constructions, with only an addition in each building block. We merge and extend Table 3 and Table 4 in Table 5, and get an extensive study involving all multiplicative levels form 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 $M = 3, 6, 7, 8, 9, 10$ the

	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
$n = 2^{10} \cdot 3$	$\log q$	47	69	92	115	138	161	185	209	232	246
	$\log qC$	71	101	132	163	194	225	257	289	320	342
	λ	145	100	77	63	54	45	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
$n = 2^{11} \cdot 3$	$\log q$	59	72	95	120	146	169	193	217	242	267
	$\log qC$	84	105	137	170	204	235	268	300	333	367
	λ	303	203	151	120	100	86	76	69	63	57
$n = 2^{13}$	$\log q$	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
$n = 2^{12} \cdot 3$	$\log q$	51	75	100	125	150	175	201	227	252	278
	$\log qC$	77	109	143	177	210	244	279	313	347	381
	λ	1033	669	481	371	300	252	215	187	167	150

Table 5. Study of the estimates for Circuit 1.

optimal estimate is achieved by a non power-of-two construction.

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 6. Again for each value of M we highlight in green the optimal instances with respect to the security threshold $\lambda = 128$.

	M	1	2	3	4	5	6	7	8	9	10
$n = 2^{10} \cdot 3$	$\log q$	72	119	168	216	265	313	362	411	460	510
	$\log qC$	113	177	242	307	372	436	502	567	633	699
	λ	143	85	61	48	45	45	45	44	44	44
$n = 2^{12}$	$\log q$	80	136	193	250	307	364	421	479	536	594
	$\log qC$	127	202	278	354	431	507	583	660	737	814
	λ	175	99	71	56	45	45	45	45	45	45
$n = 2^{11} \cdot 3$	$\log q$	75	124	175	225	276	327	377	428	480	531
	$\log qC$	117	184	252	319	387	455	522	590	660	728
	λ	298	169	116	90	74	64	56	49	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	371	202	137	105	85	73	64	57	51	47
$n = 2^{12} \cdot 3$	$\log q$	78	129	182	234	287	340	393	446	499	552
	$\log qC$	122	196	261	331	402	473	544	614	685	756
	λ	640	358	240	181	144	121	104	92	82	75
$n = 2^{14}$	$\log q$	86	146	207	268	329	390	451	513	575	636
	$\log qC$	135	216	298	379	461	542	624	707	789	870
	λ	802	434	289	215	171	142	122	106	95	86
$n = 2^{13} \cdot 3$	$\log q$	81	134	189	243	298	353	408	463	518	573
	$\log qC$	126	198	271	344	417	491	564	637	711	785
	λ	1380	775	518	386	304	250	212	184	162	146

Table 6. Study of the estimates for Circuit 2.

6 Conclusions and Future Work

6.1 Conclusions

With this work, we showed how it is 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, we believe ours is the first formal demonstration of such a statement. The bounds on the variance of the product rings were essentially already established in [24] and [29] 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, 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 in this sense because it shows how to compute the full covariance matrix in this case. Moreover, its proof seems easy to generalize to other cyclotomic rings. 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 (33)). The results of the estimations themselves are quite promising: in Section 5.1, 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. 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 5.1 show a significant improvement with respect to the power-of-two they outperform. In fact, 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 not only affects the quantity of memory we need but also makes the cryptosystem 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.

6.2 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 5.1, we can see how there still are some big jumps in our estimates. For example, if we consider Table 6, we can see that to achieve 7 multiplications with $\lambda = 128$ we need to jump from $n = 2^{14}$ to $n = 2^{13} \cdot 3$, with λ increasing to 212 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.

Regarding specifically parameter estimation for FHE schemes, the most promising lines of research are those developing new techniques to substitute the worst-case canonical norm approach. It seems that there is a discrepancy between the estimates based on this technique and experimental data ([12]), and an *average case* approach has been proposed to overcome this [3, 13, 32]. This topic is for sure very interesting, and any progress makes FHE easier to deploy in real-life applications.

Bibliography

- [1] Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of Learning with Errors. *Journal of Mathematical Cryptology* **9**(3), 169–203 (2015)
- [2] Bernstein, D.J.: Multidigit multiplication for mathematicians. *Advances in Applied Mathematics* pp. 1–19 (2001)
- [3] Biasioli, B., Marcolla, C., Calderini, M., Mono, J.: Improving and Automating BFV Parameters Selection: An Average-Case Approach. *Cryptology ePrint Archive*, Paper 2023/600 (2023)
- [4] Billingsley, P.: Probability and measure. John Wiley & Sons (2008)
- [5] Brakerski, Z.: Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In: Safavi-Naini, R., Canetti, R. (eds.) *Advances in Cryptology – CRYPTO 2012*, pp. 868–886, Springer, Berlin, Heidelberg (2012)
- [6] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) Fully Homomorphic Encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)* **6**(3), 1–36 (2014)
- [7] Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In: *Advances in Cryptology–CRYPTO 2011: 31st Annual Cryptology Conference*, Santa Barbara, CA, USA, August 14–18, 2011. *Proceedings* 31, pp. 505–524, Springer (2011)
- [8] Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: A full RNS variant of approximate homomorphic encryption. In: *International Conference on Selected Areas in Cryptography – SAC 2018*, pp. 347–368, Springer (2018)
- [9] Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic Encryption for Arithmetic of Approximate Numbers. In: *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security*, Hong Kong, China, December 3–7, 2017, *Proceedings, Part I* 23, pp. 409–437, Springer (2017)
- [10] Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: *international conference on the theory and application of cryptology and information security*, pp. 3–33, Springer (2016)
- [11] Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: TFHE: Fast Fully Homomorphic Encryption over the Torus. *Journal of Cryptology* **33**(1), 34–91 (2020)
- [12] Costache, A., Laine, K., Player, R.: Evaluating the effectiveness of heuristic worst-case noise analysis in FHE. In: *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security*, ESORICS 2020, Guildford, UK, September 14–18, 2020, *Proceedings, Part II* 25, pp. 546–565, Springer (2020)
- [13] Costache, A., Nürnberger, L., Player, R.: Optimisations and tradeoffs for helib. In: *Topics in Cryptology–CT-RSA 2023: Cryptographers’ Track at*

- the RSA Conference 2023, San Francisco, CA, USA, April 24–27, 2023, Proceedings, pp. 29–53, Springer (2023)
- [14] Costache, A., Smart, N.P.: Which ring based somewhat homomorphic encryption scheme is best? In: Cryptographers’ Track at the RSA Conference, pp. 325–340, Springer (2016)
- [15] Cox, D.A.: Galois theory, vol. 61. John Wiley & Sons (2011)
- [16] Damgård, I., Pastro, V., Smart, N., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Advances in Cryptology–CRYPTO 2012: 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2012. Proceedings, pp. 643–662, Springer (2012)
- [17] Ducas, L., Micciancio, D.: FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology – EUROCRYPT 2015, pp. 617–640, Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
- [18] Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive (2012)
- [19] Geelen, R., Van Beirendonck, M., Pereira, H.V., Huffman, B., McAuley, T., Selfridge, B., Wagner, D., Dimou, G., Verbauwhede, I., Vercauteren, F., et al.: Basalisc: Programmable asynchronous hardware accelerator for bgv fully homomorphic encryption. Cryptology ePrint Archive (2022)
- [20] Gentry, C.: A Fully Homomorphic Encryption scheme. Ph.D. thesis, Stanford University, 2009 (2009)
- [21] Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluation of the AES circuit. In: Advances in Cryptology–CRYPTO 2012: 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2012. Proceedings, pp. 850–867, Springer (2012)
- [22] Halevi, S., Shoup, V.: Design and implementation of HELib: a homomorphic encryption library. Cryptology ePrint Archive (2020)
- [23] Han, K., Ki, D.: 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, pp. 364–390, Springer (2020)
- [24] Iliashenko, I.: Optimisations of fully homomorphic encryption (2019)
- [25] Jacod, J., Protter, P.: Probability essentials. Springer Science & Business Media (2004)
- [26] Kim, A., Polyakov, Y., Zucca, V.: Revisiting homomorphic encryption schemes for finite fields. In: Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part III 27, pp. 608–639, Springer (2021)
- [27] Lang, S.: Algebraic number theory, vol. 110. Springer Science & Business Media (2013)
- [28] Lidl, R., Niederreiter, H.: Finite fields. 20, Cambridge University Press (1997)
- [29] Lyubashevsky, V., Seiler, G.: NTTRU: truly fast NTRU using NTT. Cryptology ePrint Archive (2019)

- [30] Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F.H., Aaraj, N.: Survey on Fully Homomorphic Encryption, Theory, and Applications. *Proceedings of the IEEE* **110**(10), 1572–1609 (2022)
- [31] Mono, J., Marcolla, C., Land, G., Güneysu, T., Aaraj, N.: Finding and evaluating parameters for BGV. *International Conference on Cryptology in Africa - AFRICACRYPT 2023* (2023)
- [32] Murphy, S., Player, R.: A Central Limit Framework for Ring-LWE Decryption. *Cryptology ePrint Archive, Paper 2019/452* (2019)