Improving and Automating BFV Parameters Selection: An Average-Case Approach

Beatrice Biasioli[1], Chiara Marcolla[1], Marco Calderini[2], and Johannes Mono[3]

[1] Technology Innovation Institute, Abu Dhabi, United Arab Emirates
[2] Universitá degli studi di Trento, Italy
[3] Ruhr University Bochum, Bochum, Germany

Abstract. The Brakerski/Fan-Vercauteren (BFV) scheme is a state-of-the-art scheme in Fully Homomorphic Encryption based on the Ring Learning with Errors (RLWE) problem. Thus, ciphertexts contain an error that increases with each homomorphic operation and has to stay below a certain threshold for correctness. This can be achieved by setting the ciphertext modulus big enough. On the other hand, a larger ciphertext modulus decreases the level of security and computational efficiency, making parameters selection challenging. Our work aims at improving the bound on the ciphertext modulus, minimizing it.

Our main contributions are the following. Primarily, we perform the first average-case analysis of the error growth for the BFV scheme, significantly improving its estimation. For a circuit with multiplicative depth of only 3, our bounds are up to 18.6 bits tighter than previous analyses and within 1.1 bits of the experimentally observed values. Secondly, we give a general way to bound the ciphertext modulus for correct decryption that allows closed formulas.

Finally, we use our theoretical advances and propose the first parameter generation tool for the BFV scheme. Here we add support for arbitrary but use-case-specific circuits, as well as the ability to generate easy-to-use code snippets, making our theoretical work accessible to both researchers and practitioners.

Keywords: Fully Homomorphic Encryption, BFV, Parameter Generation, Average-Case Noise Analysis, OpenFHE

1 Introduction

Data privacy concerns are increasing significantly in the context of future-generation networking, such as Internet of Things, cloud services, edge computing, artificial intelligence applications, and artificial intelligence applications. Homomorphic encryption enables privacy-preserving data processing [22], namely data manipulation in the encrypted domain without decryption. More specifically, fully homomorphic encryption (FHE) schemes define ciphertext operations corresponding to operations on the underlying plaintext as additions or multiplications.

The first Fully Homomorphic Encryption (FHE) scheme was introduced in 2009 by Gentry in [16]. In his PhD thesis, Gentry provided a method for constructing a general FHE scheme from a scheme with limited but sufficient homomorphic evaluation capacity. Since then, novel constructions on FHE have been proposed following his idea, BGV [6], BFV [5,15], TFHE [9], and CKKS [8] some of the most representative.

The security of most of the FHE schemes is based on the presumed intractability of the decision Learning with Errors (LWE) problem, [25], and its ring variant (RLWE), [21]. Informally, they consist of distinguishing equations perturbed by small noise from random tuples. The problem arising from this construction is the noise growth. Indeed, in order to guarantee correct decryption, the error, also called noise, has to be small. However, the noise grows progressively as operations are performed, particularly when homomorphic multiplications are involved. One approach to accommodating more operations is increasing the ciphertext modulus q. However, a larger modulus also decreases the security level of the underlying scheme. To maintain an equivalent level of security, we must require a larger polynomial degree n at the cost of efficiency. This delicate balance between security (achieved with a small ciphertext modulus) and error margin (associated with a large ciphertext modulus) illustrates the difficulty of finding an optimal set of parameters for a specific FHE scheme. Addressing this challenge is crucial to achieving widespread adoption of FHE.

Related works. Several efforts have been made by the FHE community in facing this challenge. For instance, Bergerat *et al.* [4] proposed a framework for efficiently selecting parameters in TFHE-like schemes. Mono *et al.* [23] developed an interactive parameters generator for the leveled BGV scheme that supports arbitrary circuit models. Moreover, for all FHE schemes, the Homomorphic Encryption Standard [2] provides lookup tables that allow to fix the polynomial degree n and determine the maximum ciphertext modulus q required to achieve a desired security level λ, employing the Lattice Estimator[4]. On the other hand, there is no uniform way for all the schemes to find the minimal q that guarantees correct decryption. In particular, the average-case analysis of the error growth is exploited for the TFHE [9], CKKS [11] and BGV [24,13] schemes. However, our research has revealed that the same methods applied to BFV yield an underestimation of the bounds. Indeed, the state-of-the-art for the BFV scheme employs either the infinity [20] or the canonical norm [19,10,12], getting overly conservative bounds.

Our contribution. This paper improves the current state of BFV parameters selection by providing 1) the first estimation of the noise in average-case, 2) a consequent way to bound the ciphertext modulus for correctness and 3) a tool to automate the parameters generation based on our theoretical findings.

More in detail, we present a novel approach for the BFV scheme based on average-case noise analysis. Our method differs from the previously proposed

[4] The Lattice Estimator (`https://github.com/malb/lattice-estimator`) is a software tool to determine the security level of LWE instances against the known attacks.

for the BGV and CKKS schemes [11,24], since here the error coefficients are not independent among each other, making it impossible to apply the Central Limit Theorem. As a result, our analysis is more intricate, particularly for homomorphic multiplication, where we have to introduce a function to "correct" the variance product. To demonstrate the effectiveness of our method, we compare our bounds with the state-of-the-art noise analysis based on the canonical norm. For a circuit with multiplicative depth only 3, our bounds are up to 18.6 bits tighter and only within 1.1 bits lower than the experimentally observed values.

From the noise analysis, we provide closed formulas to compute a bound on the ciphertext modulus that guarantees correct decryption. While this can be done in general for any kind of circuit, we focus on the most common ones.

Finally, we develop an interactive parameters generator, which makes use of our theoretical results and the security formula proposed in [23]. This tool provides flexibility, allowing users to choose the desired security level, the degree of the arithmetic function to be evaluated homomorphically, and the error and secret distributions, among other parameters.

The structure of the paper is the following:

- To facilitate understanding of the paper, we present the notation and mathematical background required in Section 2.
- In Section 3, we comprehensively analyze and compute invariant noise after any operation in the BFV scheme.
- The core of the paper is Section 4, where we introduce our average-case approach.
- In Section 5, we exploit the novel error analysis to provide a general way to compute a bound on the ciphertext modulus, focusing on practical-used circuits. Additionally, we introduce our parameter generator to facilitate the selection of optimal parameters for the BFV scheme.
- In Section 6, we compare our average-case approach with prior bounds of BFV noise growth.
- Finally, Section 7 draws some conclusions and open problems.

2 Preliminaries

In this section, we first define the general notations that we will use in the remainder of the work, then we provide the mathematical background for the secret and error distributions, as well as their analysis.

2.1 Notation

Let $f(x)$ be a monic irreducible polynomial of degree n, in particular, we take $f(x) = x^n + 1$ with n a power of 2. We denote by $\mathcal{R} = \mathbb{Z}[x]/\langle f(x)\rangle$ and with $\mathcal{K} = \mathbb{Q}[x]/\langle f(x)\rangle$. Let $a \in \mathcal{K}$, we denote by $a|_i$ the coefficient of x^i. Note that, for $a, b \in \mathcal{K}$ we have that

$$(ab)|_i = \sum_{j=0}^{n-1} \xi(i,j)\, a|_j\, b|_{i-j}, \tag{1}$$

where $i-j$ is computed mod n and $\xi(i,j)$ is defined as 1 if $i-j \in [0,n)$ and -1 otherwise. For a positive integer p, $\mathbb{Z}_p$ denotes the set of integers in $(-p/2, p/2]$ and by $\mathcal{R}_p$ the set of polynomials in $\mathcal{R}$ with coefficients in $\mathbb{Z}_p$. Let $z \in \mathbb{Z}$, we write $[z]_p \in \mathbb{Z}_p$ for the centered representative of z mod p. For polynomials in $\mathcal{R}$, it denotes the element in $\mathcal{R}_p$ where $[\cdot]_p$ is applied coefficient-wise. Let $x \in \mathbb{Q}$, $\lfloor x \rceil$ be the rounding to the nearest integer. The same holds coefficient-wise for polynomials in $\mathcal{K}$.

The integer $t > 1$ denotes the plaintext modulus and with $\mathcal{R}_t$ the plaintext space. We further require $t \equiv 1 \pmod{2n}$. Analogously, we denote the ciphertext modulus by $q = \prod_{i=1}^{k} r_i$, and the ciphertext space follows as $\mathcal{R}_q$. $r_i > 1$ are pairwise coprime of approximately the same size, coprime with t and such that $r_i \equiv 1 \mod 2n$. Finally, for the BGV-like circuit case explained in Section 5.2, we need $L = M + 1$ sub-moduli p_j defined analogously to q, where M is the multiplicative depth of the circuit. For any ℓ, we denote by $q_\ell = \prod_{j=1}^{\ell} p_j = \prod_{i=1}^{k_\ell} r_i$, the initial ciphertext is $q = q_L$, or q_{ms} to distinguish it.

2.2 Secret and Error Distributions

Let χ be a probabilistic distribution and $a \in \mathcal{R}$, we write $a \leftarrow \chi$ when sampling each coefficient of a independently from χ. We use the following distributions.

- $\mathcal{DG}(0, \sigma^2)$, the discrete Gaussian distribution centered in 0 with standard deviation σ.
- $\mathcal{U}_p$, the uniform distribution over $\mathbb{Z}_p$, where p is a positive integer.
- $\mathcal{U}_I$, the uniform distribution over a real interval $I \subset \mathbb{R}$.
- $\mathcal{ZO}(\rho)$, a distribution over the ternary set $\{0, \pm 1\}$ with probability $\rho/2$ for ± 1 and probability $1-\rho$ for 0 with $\rho \in [0,1]$.

Finally, the distributions $\mathcal{HWT}(h)$ chooses a vector uniformly at random from $\{0, \pm 1\}^n$ with exactly h nonzero entries, where $h \leq n$ positive integer. Let χ_s, χ_u be secret key distributions and χ_e an error distribution from the Learning with Errors over Rings (RLWE) problem. Tipically, we have $\chi_e = \mathcal{DG}(0, \sigma^2)$, with $\sigma = 3.19$ and $\chi_s = \chi_u = \mathcal{U}_3$ [2]. Other common options for χ_s are $\mathcal{ZO}(0.5)$, $\mathcal{DG}(0, (3.19)^2)$ and $\mathcal{HWT}(64)$. A variable with any of the above distributions or from the uniform over a centred interval is symmetric, thus with mean 0, and has variance as follows.

- If $X \leftarrow \mathcal{DG}(0, \sigma^2)$ then $\mathsf{Var}(X) = \sigma^2$.
- If $X \leftarrow \mathcal{U}_p$ then $\mathsf{Var}(X) = (p^2-1)/12$. In particular,
  - If $X \leftarrow \mathcal{U}_q$ then $\mathsf{Var}(X) \approx q^2/12$.
  - If $X \leftarrow \mathcal{U}_3$ then $\mathsf{Var}(X) = 2/3$.
- If $X \leftarrow \mathcal{U}_{(-1/2,1/2]}$ then $\mathsf{Var}(X) = 1/12$.
- If $X \leftarrow \mathcal{ZO}(0.5)$ then $\mathsf{Var}(X) = 1/2$.
- If $X \leftarrow \mathcal{HWT}(64)$ then $\mathsf{Var}(X) = 64/n$.

Coverage probability for Gaussian-distributed variables. Let X be a random variable (r.v.) from a Gaussian distribution centred in 0 of variance V, then

$$\begin{aligned}\mathbb{P}\Big(|X| \leq x\Big) &= \mathbb{P}\Big(X \leq x\Big) - \mathbb{P}\Big(X \leq -x\Big) = \\ &= \frac{1}{2}\Big(1 + \operatorname{erf}\Big(\frac{x}{\sqrt{2V}}\Big)\Big) - \frac{1}{2}\Big(1 + \operatorname{erf}\Big(\frac{-x}{\sqrt{2V}}\Big)\Big) = \operatorname{erf}\Big(\frac{x}{\sqrt{2V}}\Big).\end{aligned} \tag{2}$$

Suppose now that we want to study the infinity norm of a vector. If its entries are independent, then $\mathbb{P}\Big(||\mathbf{X}||_\infty \leq x\Big) = \mathbb{P}\Big(|X| \leq x\Big)^n$. In general, we can give an upper bound on the complementary probability:

$$\mathbb{P}\Big(||\mathbf{X}||_\infty > x\Big) \leq n\mathbb{P}\Big(|X| > x\Big) = n\Big(1 - \operatorname{erf}\Big(\frac{x}{\sqrt{2V}}\Big)\Big). \tag{3}$$

Canonical embedding and norm. We recall the results of [10,19,12]. The *canonical embedding* of $a \in \mathcal{R}$ is the vector obtained by evaluating a in the primitive $2n$-th roots of unity. The *canonical embedding norm* of a is defined as the infinity norm of the canonical embedding.

Let us consider a random polynomial $a \in R$ where each coefficient is sampled independently from a zero-mean distribution, then $||a||^{can} \leq D\sqrt{nV_a}$ with high probability [10].

We now want to estimate the probability that the canonical norm of a random polynomial exceeds a certain value x.

Let us consider the case where the coefficients in a, $a|_0, ..., a|_{n-1}$, are i.i.d. with 0 mean and variance V_a, and suppose $\mathbb{E}(|a|_i|^{2+\delta}) < \infty$ for all i and for some fixed $\delta > 0$ (this last condition it is not restrictive in our case). As shown in [14], using the Lyapunov Central Limit Theorem, it is possible to prove that for any root of unity $\zeta = \cos(\alpha) + i\sin(\alpha)$, the r.v. $a(\zeta)$ is a complex r.v. which can be approximated by a complex Gaussian r.v.. That is, $a(\zeta)$ is approximated by a bivariate Normal distributed r.v. (X, Y). Moreover, X and Y are Normal distributed with variance $V_X = V_a(\sum_{j=0}^{n-1}\cos^2(j\alpha))$ and $V_Y = V_a(\sum_{j=0}^{n-1}\sin^2(j\alpha)) = nV_a - V_X$, respectively.

Let C be the diagonal matrix with the standard deviation of X and Y over the diagonal. We have that $(X, Y)^t = C \cdot (Z, Z')^t$ with Z and Z' i.i.d. standard Gaussian r.v.'s. Therefore,

$$\mathbb{P}\left(|a(\zeta)| < x\right) = \mathbb{P}\left(||(X, Y)||_2 < x\right) \geq \mathbb{P}\left(||C||_2||(Z, Z')||_2 < x\right).$$

Let M be the maximum between V_X and V_Y (note that $\frac{n}{2}V_a \leq M \leq nV_a$). The 2-norm of the matrix C is $\sqrt{M}$. Thus,

$$\mathbb{P}\left(||C||_2||(Z, Z')||_2 < x\right) = \mathbb{P}\left(||(Z, Z')||_2^2 < \frac{x^2}{M}\right).$$

Since Z, Z' are independent standard Gaussian r.v., $||(Z, Z')||_2^2$ is Chi-squared distributed and

$$\mathbb{P}\left(||(Z, Z')||_2^2 < \frac{x^2}{M}\right) = 1 - e^{-\frac{x^2}{2M}} \geq 1 - e^{-\frac{x^2}{nV_a}},$$

implying $\mathbb{P}\left(|a(\zeta_m)| > x\right) \leq e^{-\frac{x^2}{nV_a}}$. Therefore,

$$\mathbb{P}\left(||a||^{can} > x\right) \leq n e^{-\frac{x^2}{nV_a}}. \tag{4}$$

Probability operators. Let X, Y, Z be real random variables and c a constant. The expected value enjoys the following properties:

- it is linear: $\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$ and $\mathbb{E}[cX] = c\mathbb{E}[X]$;
- if X is sampled from a symmetric distribution, i.e. $\mathbb{P}(X = x) = \mathbb{P}(X = -x)$ for any $x \in \mathbb{R}$, then $\mathbb{E}[X] = 0$;
- if X and Y are independent, then $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$;
- in general, $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y] + \mathsf{Cov}(X, Y)$.

The covariance is consequently defined as $\mathsf{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$ and is such that

- if X and Y are independent, then $\mathsf{Cov}(X, Y) = 0$;
- it is bilinear.

Some characteristics of the variance are

- $\mathsf{Var}(X) \geq 0$;
- $\mathsf{Var}(X+Y) = \mathsf{Var}(X) + \mathsf{Var}(Y) + 2\mathsf{Cov}(X, Y)$ and, more in general, $V(\sum_i X_i) = \sum_i V(X_i) + \sum_{i_1 \neq i_2} \mathsf{Cov}(X_{i_1}, X_{i_2})$;
- if X and Y are independent, then $\mathsf{Var}(X+Y) = \mathsf{Var}(X) + \mathsf{Var}(Y)$;
- $\mathsf{Var}(cX) = c^2 \mathsf{Var}(X)$;
- if X and Y are independent and $\mathbb{E}[X] = \mathbb{E}[Y] = 0$, then $\mathsf{Var}(XY) = \mathsf{Var}(X)\mathsf{Var}(Y)$.

3 The BFV Scheme

The following describes the BFV scheme [5,15], a cutting-edge FHE scheme whose security relies on the hardness of the ring learning with errors (RLWE) problem. We consider the latest enhancements proposed in [20]. In particular, the authors revised the encryption algorithm replacing the term $\Delta m = \lfloor \frac{q}{t} \rfloor m$ with $\lfloor \frac{q}{t} m \rceil$, which eliminates the noise gap with respect to the BGV scheme.

KeyGen(λ, L)

Define parameters and distributions accordingly to λ and L. Sample $s \leftarrow \chi_s$, $a \leftarrow \mathcal{U}_q$ and $e \leftarrow \chi_e$. Output $\mathsf{sk} = s$ and $\mathsf{pk} = (b, a) = ([-as + e]_q, a)$.

Enc(m, pk)

Receive the plaintext $m \in \mathcal{R}_t$ and $\mathsf{pk} = (b, a)$. Sample $u \leftarrow \chi_u$ and $e_0, e_1 \leftarrow \chi_e$. Output $\mathfrak{c} = (\mathbf{c}, q, \nu_{\mathsf{clean}})$ with $\mathbf{c} = (c_0, c_1) = \left(\left[\lfloor \frac{q}{t} m \rceil + ub + e_0 \right]_q, [ua + e_1]_q \right)$.

Dec($\mathfrak{c}$, sk)

Receive the extended ciphertext $\mathfrak{c}$ for $\mathsf{sk} = s$. Output $\left[\left\lfloor \frac{t}{q_\ell}[c_0 + c_1 s]_{q_\ell} \right\rceil\right]_t$.

Let $\mathfrak{c} = (\mathbf{c}, q_\ell, \nu)$ be the *extended ciphertext*, where $\mathbf{c}$ is a ciphertext, q_ℓ denotes the ciphertext modulus and ν the *invariant noise*. The invariant noise [19] is the minimal $\nu \in \mathcal{K}$ such that

$$\frac{t}{q_\ell}[c_0 + c_1 s]_{q_\ell} = m + \nu + kt$$

for some $k \in \mathcal{R}$. Therefore, $\left[\left\lfloor \frac{t}{q}[c_0 + c_1 s]_q \right\rceil\right]_t = [\lfloor m + \nu + kt \rceil]_t = [m + \lfloor \nu \rceil]_t$. Hence, the decryption works properly as long as ν is small enough. In particular, it is correct when the coefficients of ν belong to the interval $(-\frac{1}{2}, \frac{1}{2}]$. After the encryption operation, the invariant noise is

$$\nu_{\mathsf{clean}} = \frac{t}{q}(\varepsilon + eu + e_0 + e_1 s) \tag{5}$$

where $\varepsilon = \left\lfloor \frac{q}{t} m \right\rceil - \frac{q}{t} m = -\frac{[qm]_t}{t}$, [20]. Indeed,

$$\begin{aligned} \frac{t}{q}[c_0 + c_1 s]_q =& \frac{t}{q}\left[\left\lfloor \frac{q}{t} m \right\rceil + ub + e_0 + (ua + e_1)s\right]_q \\ =& \frac{t}{q}\left(\frac{q}{t} m + \varepsilon + ue + e_0 + e_1 s\right) + kt = m + \nu_{\mathsf{clean}} + kt. \end{aligned}$$

Addition & Constant Multiplication.

Add($\mathfrak{c}, \mathfrak{c}'$)

Receive extended ciphertexts $\mathfrak{c} = (\mathbf{c}, q_\ell, \nu)$ and $\mathfrak{c}' = (\mathbf{c}', q_\ell, \nu')$.
Output $(\mathbf{c}_{\mathsf{add}}, q_\ell, \nu_{\mathsf{add}})$ with $\mathbf{c}_{\mathsf{add}} = ([c_0 + c_0']_{q_\ell}, [c_1 + c_1']_{q_\ell})$.

MulConst($\alpha, \mathfrak{c}$)

Receive constant polynomial $\alpha \in \mathcal{R}_t$ and extended ciphertext $\mathfrak{c} = (\mathbf{c}, q_\ell, \nu)$.
Output $(\mathbf{c}_{\mathsf{const}}, q_\ell, \nu_{\mathsf{const}})$ with $\mathbf{c}_{\mathsf{const}} = ([\alpha c_0]_{q_\ell}, [\alpha c_1]_{q_\ell})$.

By the definition of invariant noise, for some for some $u, k \in \mathcal{R}$, we have

$$\begin{aligned} \frac{t}{q_\ell}[c_0 + c_1 s + c_0' + c_1' s]_{q_\ell} =& \frac{t}{q_\ell}([c_0 + c_1 s]_{q_\ell} + [c_0' + c_1' s]_{q_\ell} - uq_\ell) \\ =& [m + m']_t + \nu + \nu' + kt \implies \nu_{\mathsf{add}} = \nu + \nu \end{aligned} \tag{6}$$

$$\begin{aligned} \frac{t}{q_\ell}[\alpha c_0 + \alpha c_1 ss]_{q_\ell} =& \frac{t}{q_\ell}\left(\alpha[c_0 + c_1 s]_{q_\ell} - uq_\ell\right) = [\alpha m]_t + \alpha\nu + kt \\ & \implies \nu_{\mathsf{const}} = \alpha\nu \end{aligned} \tag{7}$$

Multiplication & Modulus switching. In this section, we are going to see the multiplication algorithm presented in [20], which, before multiplying two ciphertexts, applies to one of them a modulus switch. This is done in order to make the Residue Number System (RNS) representation more efficient. The modulus switch technique was first introduced for the BGV scheme in [7] to reduce the error associated with a ciphertext. In the BFV scheme, this error reduction is made implicitly, so the purpose of the modulus switch is only to shift to a different ciphertext modulus.

ModSwitch$(\mathfrak{c}, q'_\ell)$

Receive the extended ciphertext $\mathfrak{c} = (\mathbf{c}, q_\ell, \nu)$ and the target modulo q'_ℓ. Output $\mathfrak{c}' = (\mathbf{c}', q'_\ell, \nu + \nu_{\mathsf{ms}}(q'_\ell))$ with $\mathbf{c}' = \left(\left[\left\lfloor \frac{q'_\ell}{q_\ell} c_0 \right\rceil\right]_{q'_\ell}, \left[\left\lfloor \frac{q'_\ell}{q_\ell} c_1 \right\rceil\right]_{q'_\ell} \right)$.

The noise added by the modulo switch operation is

$$\nu_{\mathsf{ms}}(q'_\ell) = \frac{t}{q'_\ell}(\varepsilon_0 + \varepsilon_1 s), \text{ with } \varepsilon_i = -\frac{[q'_\ell c_i]_{q_\ell}}{q_\ell}. \tag{8}$$

Indeed, since $\frac{t}{q'_\ell}[c'_0 + c'_1 s]_{q'_\ell} = \frac{t}{q'_\ell}\left[\lfloor \frac{q'_\ell}{q_\ell} c_0 \rceil + \lfloor \frac{q'_\ell}{q_\ell} c_1 \rceil s\right]_{q'_\ell}$, we have

$$\begin{aligned} \frac{t}{q'_\ell}[c'_0 + c'_1 s]_{q'_\ell} &= \frac{t}{q'_\ell}\left[\frac{q'_\ell}{q_\ell} c_0 + \varepsilon_0 + \frac{q'_\ell}{q_\ell} c_1 s + \varepsilon_1 s\right]_{q'_\ell} \\ &= \frac{t}{q_\ell}[c_0 + c_1 s]_{q_\ell} + \frac{t}{q'_\ell}(\varepsilon_0 + \varepsilon_1 s) + ht = m + \nu + \frac{t}{q'_\ell}(\varepsilon_0 + \varepsilon_1 s) + k't. \end{aligned}$$

The multiplication algorithm takes as input two extended ciphertexts $\mathfrak{c}$ and $\mathfrak{c}'$, where one of the ciphertexts, say $\mathfrak{c}'$, is the result of a modulo switch to q'_ℓ. The new modulus q'_ℓ is required to be of approximately the same size of q_ℓ, to satisfy $q'_\ell \equiv 1 \pmod{2n}$ and $(t, q'_\ell) = (q_\ell, q'_\ell) = 1$.

Ten$(\mathfrak{c}, \mathfrak{c}')$

Receive the extended ciphertexts $\mathfrak{c} = (\mathbf{c}, q_\ell, \nu)$ and $\mathfrak{c}' = (\mathbf{c}', q'_\ell, \nu')$. Output $\mathfrak{d} = (\mathbf{d}, q_\ell, \nu_{\mathsf{mul}}(q_\ell))$ with

$$\mathbf{d} = (d_0, d_1, d_2) = \left(\left[\left\lfloor \frac{t}{q'_\ell} c_0 c'_0 \right\rceil\right]_{q_\ell}, \left[\left\lfloor \frac{t}{q'_\ell} (c_0 c'_1 + c_1 c'_0) \right\rceil\right]_{q_\ell}, \left[\left\lfloor \frac{t}{q'_\ell} c_1 c'_1 \right\rceil\right]_{q_\ell} \right).$$

The multiplication output is a polynomial $\mathcal{R}_q^3$ that can be decrypted in the following way: $\left\lfloor \frac{t}{q_\ell}[d_0 + d_1 s + d_2 s^2]_{q_\ell} \right\rceil$. Let $\frac{t}{q_\ell}(c_0 + c_1 s) = m + \nu + ht$ and $\frac{t}{q'_\ell}(c'_0 + c'_1 s) = m' + \nu' + h't$, as per definition of invariant noise. Thus,

$$
\begin{aligned}
&\frac{t}{q_\ell}\left[\left\lfloor\frac{t}{q'_\ell}c_0c'_0\right\rceil + \left\lfloor\frac{t}{q'_\ell}(c_0c'_1 + c'_0c_1)\right\rceil s + \left\lfloor\frac{t}{q'_\ell}c_1c'_1\right\rceil s^2\right]_{q_\ell}\\
=&\frac{t}{q_\ell}\left[\frac{t}{q'_\ell}c_0c'_0 + \varepsilon_0 + \frac{t}{q'_\ell}(c_0c'_1 + c'_0c_1)s + \varepsilon_1 s + \frac{t}{q'_\ell}c_1c'_1s^2 + \varepsilon_2 s^2\right]_{q_\ell}\\
=&\frac{t}{q_\ell}(c_0 + c_1s)\cdot\frac{t}{q'_\ell}(c'_0 + c'_1s) + \frac{t}{q_\ell}(\varepsilon_0 + \varepsilon_1 s + \varepsilon_2 s^2) + h''t\\
=&[mm']_t + \nu(m' + h't) + \nu'(m + ht) + \nu\nu' + \frac{t}{q_\ell}(\varepsilon_0 + \varepsilon_1 s + \varepsilon_2 s^2) + kt\\
=&[mm']_t + \nu\left(\frac{t}{q'_\ell}(c'_0 + c'_1s) - \nu'\right) + \nu'\left(\frac{t}{q_\ell}(c_0 + c_1s) - \nu\right) + \nu\nu' +\\
&\quad + \frac{t}{q_\ell}(\varepsilon_0 + \varepsilon_1 s + \varepsilon_2 s^2) + kt =\\
=&[mm']_t + \nu_{\mathsf{mul}}(q_\ell) + kt,
\end{aligned}
$$

where the noise after the multiplication is

$$
\nu_{\mathsf{mul}}(q_\ell) = -\nu\nu' + \nu\frac{t}{q'_\ell}(c'_0 + c'_1s) + \nu'\frac{t}{q_\ell}(c_0 + c_1s) + \frac{t}{q_\ell}(\varepsilon_0 + \varepsilon_1 s + \varepsilon_2 s^2). \quad (9)
$$

Finally, the multiplication output needs to be transformed back to a ciphertext in $\mathcal{R}_q^2$. This is done by encrypting its last term d_2 via key switching (see Section 3.1), also called relinearization.

3.1 Key Switching

The key switch is used for (i) reducing the degree of a ciphertext polynomial, usually the multiplication output, or (ii) changing the key after a rotation. In the multiplication case, the term $d_2 \cdot s^2$ is converted into a polynomial $c_0^{\mathsf{ks}} + c_1^{\mathsf{ks}} \cdot s$ and the two components are added, obtaining the equivalent $\mathbf{c}' = (d_0 + c_0^{\mathsf{ks}}, d_1 + c_1^{\mathsf{ks}})$. In the rotation, where we need to go back to the original key s from $\mathrm{rot}(s)$, we convert the ciphertext term $c_1 \cdot \mathrm{rot}(s)$ into $c_0^{\mathsf{ks}} + c_1^{\mathsf{ks}} \cdot s$. In the following, we will only analyze the first case.

The idea is to encrypt the extra term s^2 under the secret key. However, in doing so, the resulting error would be too significant. Hence several variants exist to reduce its growth. This work considers the three main ones: Brakerski Vaikuntanathan (BV), Gentry Halevi Smart (GHS), and Hybrid. For the sake of simplicity, we present directly the variants compatible with the RNS representation [3,18,20]. The RNS method makes the scheme implementation substantially faster and allows parallelization. It does not add an error itself, but usually it is employed the FastBaseExtension function, which can be imprecise, to extend d_2 from the base q_ℓ to $q_\ell P$ (for further information, see [20]).

Brakerski-Vaikuntanathan The strategy is exploiting the Chinese Remainder Theorem (CRT) to decompose d_2 in the k_ℓ moduli $r_i \approx \sqrt[k]{q}$.

KeySwitchGen$^{\mathsf{BV}}$ (s, s^2)

Sample $a_i \leftarrow \mathcal{U}_q$, $e_i \leftarrow \chi_e$ and set $(b_i, a_i) = \left(\left[\left[\left(\frac{q}{r_i}\right)^{-1}\right]_{r_i} \frac{q}{r_i} s^2 - a_i s + e_i\right]_q, a_i\right)$ for $i = 1, \ldots, k$. Output $\mathsf{ks}^{\mathsf{BV}} = \{(b_i, a_i)\}$.

KeySwitch$^{\mathsf{BV}}$$(\mathbf{ks}^{\mathsf{BV}}, \mathfrak{c})$

Receive $\mathfrak{d} = (\mathbf{d}, q_\ell, \nu)$ with $\mathbf{d} = (d_0, d_1, d_2)$ and $\mathsf{ks}^{\mathsf{BV}} = \{(b_i, a_i)\}$. Output $\mathfrak{c} = (\mathbf{c}, q_\ell, \nu + \nu_{\mathsf{ks}}^{\mathsf{BV}}(q_\ell))$ where $\mathbf{c} = \left(\left[d_0 + \sum_{i=1}^{k_\ell} [d_2]_{r_i} b_i\right]_{q_\ell}, \left[d_1 + \sum_{i=1}^{k_\ell} [d_2]_{r_i} a_i\right]_{q_\ell}\right)$.

Observing that $\left[\sum_{i=1}^{k_\ell} [d_2]_{r_i} (b_i + a_i s)\right]_{q_\ell}$ is equal to

$$\left[\sum_{i=1}^{k_\ell} [d_2]_{r_i} \left(\left[\left(\frac{q_\ell}{r_i}\right)^{-1}\right]_{r_i} \frac{q_\ell}{r_i} s^2 + e_i\right)\right]_{q_\ell} = \left[d_2 s^2 + \sum_{i=1}^{k_\ell} [d_2]_{r_i} e_i\right]_{q_\ell},$$

we have $\frac{t}{q_\ell}[c_0 + c_1 s]_{q_\ell}$ is equal to

$$\frac{t}{q_\ell}\left[d_0 + d_1 s + d_2 s^2 + \sum_{i=1}^{k_\ell} [d_2]_{r_i} e_i\right]_{q_\ell} = m + \nu + \frac{t}{q_\ell} \sum_{i=1}^{k_\ell} [d_2]_{r_i} e_i + kt.$$

Thus, the error after the BV key switching is $\nu + \nu_{\mathsf{ks}}^{\mathsf{BV}}(q_\ell)$ where

$$\nu_{\mathsf{ks}}^{\mathsf{BV}}(q_\ell) = \frac{t}{q_\ell} \sum_{i=1}^{k_\ell} [d_2]_{r_i} e_i. \tag{10}$$

Gentry-Halevi-Smart An alternative is encrypting Ps^2 instead of s^2 with P a large number, usually of approximately the same size as q. In this way, the error quantity added is divided by P.

KeySwitchGen$^{\mathrm{GHS}}$$(s, s^2)$

Sample $a' \leftarrow \mathcal{U}_{qP}$, $e' \leftarrow \chi_e$ and output the key switching key

$$\mathsf{ks}^{\mathsf{GHS}} = (b', a') = ([Ps^2 - a's + e']_{qP}, a').$$

KeySwitch$^{\mathrm{GHS}}$$(\mathbf{ks}, \mathfrak{c})$

Receive extended ciphertext $\mathfrak{d} = (\mathbf{d}, q_\ell, \nu)$ and key switching key $\mathsf{ks}^{\mathsf{GHS}}$. Output $\mathfrak{c} = (\mathbf{c}, q_\ell, \nu + \nu_{\mathsf{ks}}^{\mathsf{GHS}}((q_\ell))$ with $\mathbf{c} = \left(\left[d_0 + \left\lfloor \frac{d_2 b'}{P} \right\rceil\right]_{q_\ell}, \left[d_1 + \left\lfloor \frac{d_2 a'}{P} \right\rceil\right]_{q_\ell}\right)$.

To compute the invariant noise, we have to perform the following operation

$$\frac{t}{q_\ell}[c_0 + c_1 s]_{q_\ell} = \frac{t}{q_\ell}\left[d_0 + d_1 s + \left\lfloor \frac{d_2 b'}{P} \right\rceil + \left\lfloor \frac{d_2 a'}{P} \right\rceil s\right]_{q_\ell}$$

$$= m + \nu + \frac{t}{q_\ell}\left(\frac{d_2 e'}{P} + \varepsilon_0 + \varepsilon_1 s\right) + kt.$$

Thus, the noise after the GHS key switching is $\nu + \nu_{\mathsf{ks}}^{\mathsf{GHS}}(q_\ell)$ where

$$\nu_{\mathsf{ks}}^{\mathsf{GHS}}(q_\ell) = \frac{t}{q_\ell}\left(\frac{d_2 e'}{P} + \varepsilon_0 + \varepsilon_1 s\right). \tag{11}$$

GHS-RNS In practice, d_2 in base $q_\ell P$ is computed with the `FastBaseExtension` technique [20], for better efficiency, which gives an approximate result $d_2 + uq_\ell$:

$$\sum_{i=1}^{k_\ell}\left[[d_2]_{r_i}\left[\left(\frac{q_\ell}{r_i}\right)^{-1}\right]_{r_i}\right]_{r_i}\frac{q_\ell}{r_i} = d_2 + uq_\ell, \quad u = \left\lfloor\sum_{i=1}^{k_\ell}\left[[d_2]_{r_i}\left[\left(\frac{q_\ell}{r_i}\right)^{-1}\right]_{r_i}\right]_{r_i}\frac{1}{r_i}\right\rceil.$$

Therefore, the added error becomes

$$\nu_{\mathsf{ks}}^{\mathsf{GHS-RNS}}(q_\ell) = \frac{t}{q_\ell}\left(\frac{(d_2 + uq_\ell)e'}{P} + \varepsilon_0 + \varepsilon_1 s\right). \tag{12}$$

Hybrid The Hybrid variant offers a trade-off between efficiency and security from the two previous variants. Indeed, the downside of the first one is the inefficiency due to a large number of multiplications to be performed. In contrast, the issue with the second one is that its security relies on the RLWE assumption with a larger factor $q_\ell P$, instead of q_ℓ. This larger factor means that to achieve the same level of security, the modulus q_ℓ must be smaller, which limits the depth of the circuit that can be evaluated homomorphically. In the Hybrid relinearization, the modulus is split in a smaller number of elements ω by gathering the r_i in chunks $\tilde{r_i}$, and the division is done considering $P \approx \sqrt[\omega]{q}$. For further information see [20,17].

KeySwitchGen$^{\mathsf{Hyb}}(s, s^2)$

Sample $a_i \leftarrow \mathcal{U}_{qP}$, $e_i \leftarrow \chi_e$ and output $\mathsf{ks}^{\mathsf{Hyb}} = \{(b_i, a_i)\}_{i=1,\ldots,\omega}$ with

$$(b_i, a_i) = \left(\left[P\left[\left(\frac{q}{\tilde{r_i}}\right)^{-1}\right]_{\tilde{r_i}}\frac{q}{\tilde{r_i}}s^2 - a_i s + e_i\right]_{qP}, a_i\right).$$

KeySwitch$^{\mathsf{Hyb}}(\mathbf{ks}^{\mathsf{Hyb}}, \mathfrak{c})$

Receive extended ciphertext $\mathfrak{d} = (\mathbf{d}, q_\ell, \nu)$ and key switching key $\mathsf{ks}^{\mathsf{Hyb}}$.
Output $\mathfrak{c} = (\mathbf{c}, q_\ell, \nu + \nu_{\mathsf{ks}}^{\mathsf{Hyb}}(q_\ell))$ with

$$\mathbf{c} = \left(\left[d_0 + \left\lfloor\frac{\sum_{i=1}^{\omega}[d_2]_{\tilde{r_i}} b_i}{P}\right\rceil\right]_{q_\ell}, \left[d_1 + \left\lfloor\frac{\sum_{i=1}^{\omega}[d_2]_{\tilde{r_i}} a_i}{P}\right\rceil\right]_{q_\ell}\right).$$

Since $[b_i + a_i s]_{q_\ell P} = \left[P\left[\left(\frac{q}{\tilde{r_i}}\right)^{-1}\right]_{\tilde{r_i}}\frac{q}{\tilde{r_i}}s^2 + e_i\right]_{q_\ell P}$, we have

$$\frac{t}{q_\ell}[c_0 + c_1 s]_{q_\ell} = \frac{t}{q_\ell}\left[d_0 + d_1 s + \frac{\sum_{i=1}^{\omega}[d_2]_{\tilde{r_i}}(b_i + a_i s)}{P} + \varepsilon_0 + \varepsilon_1 s\right]_{q_\ell}$$

$$= \frac{t}{q_\ell}\left[d_0 + d_1 s + d_2 s^2 + \frac{\sum_{i=1}^{\omega}[d_2]_{\tilde{r_i}} e_i}{P} + \varepsilon_0 + \varepsilon_1 s\right]_{q_\ell}$$
$$= m + \nu + \frac{t}{q_\ell}\left(\frac{\sum_{i=1}^{\omega}[d_2]_{\tilde{r_i}} e_i}{P} + \varepsilon_0 + \varepsilon_1 s\right) + kt.$$

Thus, the noise after the Hybrid key switching is $\nu + \nu_{\mathsf{ks}}^{\mathsf{Hyb}}(q_\ell)$, where

$$\nu_{\mathsf{ks}}^{\mathsf{Hyb}}(q_\ell) = \frac{t}{q_\ell}\left(\frac{\sum_{i=1}^{\omega}[d_2]_{\tilde{r_i}} e_i}{P} + \varepsilon_0 + \varepsilon_1 s\right). \tag{13}$$

Hyb-RNS Here, the FastBaseExtension is eventually applied to the terms $[d_2]_{\tilde{r_i}}$,

$$\sum_{r_j|\tilde{r_i}}\left[[d_2]_{r_j}\left[(\tfrac{\tilde{r_i}}{r_j})^{-1}\right]_{r_j}\right]_{r_j}\frac{\tilde{r_i}}{r_j} = [d_2]_{\tilde{r_i}} + u_i\tilde{r_i},\ u_i = \left\lfloor\sum_{r_j|\tilde{r_i}}\left[[d_2]_{r_j}\left[(\tfrac{\tilde{r_i}}{r_j})^{-1}\right]_{r_j}\right]_{r_j}\frac{1}{r_j}\right\rceil.$$

Therefore, the error added becomes

$$\nu_{\mathsf{ks}}^{\mathsf{Hyb-RNS}}(q_\ell) = \frac{t}{q_\ell}\left(\frac{\sum_{i=1}^{\omega}([d_2]_{\tilde{r_i}} + u_i\tilde{r_i})e_i}{P} + \varepsilon_0 + \varepsilon_1 s\right). \tag{14}$$

4 Average-Case Noise Analysis for BFV

The purpose of this section is to investigate the error behaviour during homomorphic operations among independently computed ciphertexts. The goal is to find a small ciphertext modulus ensuring correct decryption. More specifically, it has to make the error coefficients lie in $(-\frac{1}{2}, \frac{1}{2}]$ with overwhelming probability.

We observed that the distributions of the noise coefficients are well-approximated by identical distributed Gaussian centred in 0, but not independent. Therefore, we can bound the maximum error coefficient in absolute value with high probability just by limiting their variance V. In particular, by in Equation (3), setting $V \leq \frac{1}{8D^2}$, i.e. $D \leq \frac{1}{2\sqrt{2V}}$, the probability of failure for the decryption is

$$\mathbb{P}\Big(||\nu||_\infty > \frac{1}{2}\Big) \leq n\Big(1 - \mathrm{erf}(\frac{1}{2\sqrt{2V}})\Big) \leq n(1 - \mathrm{erf}(D)),$$

Usually $D = 6$. So, for example, for $n = 2^{13}$, we have $n(1 - \mathrm{erf}(D)) \approx 2^{-42}$.

In the following, we denote with ν the invariant noise of any ciphertext and with $\nu|_i$ the i-th coefficient of ν. Moreover, we indicate with a_ι the ι-th element of ν when written as a polynomial in s, i.e. $\nu = \sum_\iota a_\iota s^\iota$. Note that the element a_ι is a polynomial in $\mathcal{K}$ itself, then $a_\iota|_i$ is its i-th coefficient. Finally, the ciphertexts we are considering are computed independently. In other words, each time we perform addition and multiplication operations, we use ciphertexts that either encrypt two different messages or are the results of different circuits, and there are no shared messages between them.

4.1 Distribution Analysis

Our study of coefficients distribution has been performed computationally. We used the OpenFHE library [1] to compute 10000 error samples, then analysed their coefficients with the Python **fitter** package[5]. We obtained that their distributions can be well-approximated by Gaussians with confidence level 95%, indeed the resulting p-value is ≥ 0.05.

In Figure 1, we show the outcome for circuits of multiplicative depth 0, 1 and 2, in particular of the first coefficient. As parameters, we took $t = 65537$, $n = 2^{13}$, q as computed by the library to have at least 128 bit security, $\chi_s = \chi_u = \mathcal{U}_3$, and $\chi_e = \mathcal{DG}(0, \sigma^2)$ with $\sigma = 3.19$. We used the Hybrid key switching and `HPSPOVERQ` multiplication.

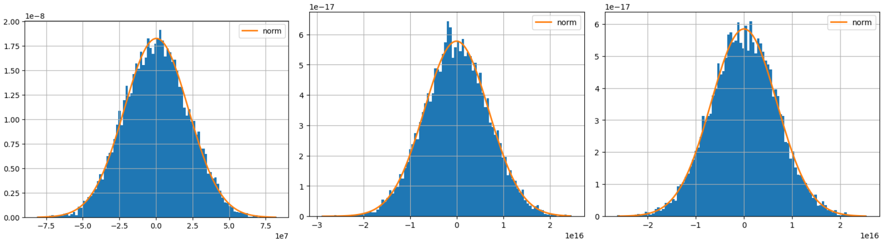


Fig. 1: (i) $\mathrm{ks}_{\mathrm{pval}}$ 0.588918; (ii) $\mathrm{ks}_{\mathrm{pval}}$ 0.596218; and (iii) $\mathrm{ks}_{\mathrm{pval}}$ 0.744975.

4.2 Mean Analysis

We will prove that the error coefficients always have mean 0.

Lemma 1. *Let $\nu = \sum_\iota a_\iota s^\iota$ be any invariant noise, a_ι the ι-th element of ν as a polynomial in s, and $a_\iota|_i$ its i-th coefficient. Then, $\mathbb{E}[a_\iota|_i] = 0, \forall \iota, i \in \mathbb{N}_{>0}$.*

See Appendix A for the proof of the lemma.

Proposition 1. *Let ν be any invariant noise and $\nu|_i$ the i-th coefficient, then the average value of its coefficients is 0, i.e. $\mathbb{E}[\nu|_i] = 0$.*

Proof. Let us write the invariant noise as a polynomial in as, $\nu = \sum_\iota a_\iota s^\iota$, as in Lemma 1. Then, by Equation (1),

$$\nu|_i = \sum_\iota (a_\iota s^\iota)|_i = \sum_\iota \sum_{j=0}^{n-1} \xi(i,j) a_\iota|_j s^\iota|_{i-j}.$$

Hence, by the linearity of the expected value and Lemma 1, we have that $\mathbb{E}[\nu|_i] = \sum_\iota \sum_{j=0}^{n-1} \xi(i,j) \mathbb{E}[a_\iota|_j] s^\iota|_{i-j} = 0$. Note that the secret key s is seen as a fixed vector. □

[5] https://fitter.readthedocs.io/en/latest/

4.3 Variance Analysis

In this section, we show how the variance of the error coefficients changes as homomorphic operations are performed. To do this, we need the following results.

Lemma 2. *Let $\nu = \sum_\iota a_\iota s^\iota$ be an invariant noise written as a polynomial in s, and $a_{\iota_1}|_{i_1}$, $a_{\iota_2}|_{i_2}$ any two coefficients i_1, i_2 of elements of ν . It follows that $\mathsf{Cov}(a_{\iota_1}|_{i_1}, a_{\iota_2}|_{i_2}) = 0$, if either $\iota_1 \neq \iota_2$ or $i_1 \neq i_2$.*

See Appendix A for the proof of the lemma.

Proposition 2. *Let $\nu = \sum_\iota a_\iota s^\iota$ be any invariant noise, $\nu|_i$ its i-th coefficients and $a_\iota|_i$ the i-th coefficients of the ι-th element of ν written as a polynomial in s. Then, the variance of the noise coefficients is*

$$\mathsf{Var}(\nu|_i) = \sum_\iota \sum_{j=0}^{n-1} \mathsf{Var}(a_\iota|_j) s^\iota|_{i-j}^2. \tag{15}$$

Proof. Let $\nu = \sum_\iota a_\iota s^\iota$ be the noise invariant, then, by Equation (1), the variance of its i-th coefficient is $\mathsf{Var}(\nu|_i) = \mathsf{Var}\Big(\sum_\iota \sum_{j=0}^{n-1} \xi(i,j) a_\iota|_j s^\iota|_{i-j}\Big)$. By the properties of the variance, it splits into

$$\sum_\iota \sum_{j=0}^{n-1} \mathsf{Var}(a_\iota|_j) s^\iota|_{i-j}^2 + \sum_{\substack{\iota_1 \neq \iota_2 \text{ or} \\ j_1 \neq j_2}} \xi(i,j_1)\xi(i,j_2)\mathsf{Cov}(a_{\iota_1}|_{j_1}, a_{\iota_2}|_{j_2}) s^{\iota_1}|_{i-j_1}^2 s^{\iota_2}|_{i-j_2}^2,$$

where the second term is null, thanks to Lemma 2, proving the thesis. □

We can finally state our results on the variance computation for operations, dedicating a special section to the multiplication.

Proposition 3 (Encryption). *Let $\mathbf{c}$ be a fresh ciphertext and let ν_{clean} be the invariant noise of $\mathbf{c}$. Thus, the variance of the error coefficients of $\mathbf{c}$ is*

$$V_{\mathsf{clean}} = \mathsf{Var}(\nu_{\mathsf{clean}}|_i) \approx \frac{B_{\mathsf{clean}}}{q^2}, \tag{16}$$

where $B_{\mathsf{clean}} = t^2\left(\frac{1}{12} + nV_eV_u + V_e + nV_eV_s\right)$.

Proof. By Equation (5), the fresh error ν_{clean} can be written as $\nu_{\mathsf{clean}} = a_0 + a_1 s$ with $a_0 = \frac{t}{q}(\varepsilon + eu + e_0)$, $a_1 = \frac{t}{q}e_1$. Let V_e, V_s, V_u be the variances of elements from the distributions χ_e, χ_s, χ_u, respectively. Then, we have that $\mathsf{Var}(a_0|_i) = \frac{t^2}{q^2}\left(\frac{1}{12} + nV_eV_u + V_e\right)$ and $\mathsf{Var}(a_1|_i) = \frac{t^2}{q^2}V_e$. It follows, by Equation (15), that

$$\mathsf{Var}(\nu_{\mathsf{clean}}|_i) = \mathsf{Var}(a_0|_i) + \sum_{j=0}^{n-1} \mathsf{Var}(a_1|_j) s|_{i-j \bmod n}^2,$$

where s is seen as a fixed vector. Since the elements $s|_i$ are sampled from a distribution with zero mean and variance V_s the element $s|_i^2$ has expected value V_s, and from the Law of Large Numbers (LLN) we can approximate $\sum_i s|_i^2 \approx nV_s$. Therefore, $\mathsf{Var}(\nu_{\mathsf{clean}}|_i) \approx \frac{t^2}{q^2}\left(\frac{1}{12}+nV_eV_u+V_e+nV_eV_s\right)$, where $\mathsf{Var}(\varepsilon|_i) = \frac{1}{12}$ comes from the fact that $\varepsilon = -\frac{[qm]_t}{t}$ and $[qm]_t$ can be consider a random element from the uniform distribution $\mathcal{U}_t$. □

Proposition 4 (Addition & Constant Multiplication). *Let $\mathbf{c}, \mathbf{c}'$ be two independently-computed ciphertexts and $\alpha \in \mathcal{R}_t$ a constant. Let ν, ν' be the errors of $\mathbf{c}$ and $\mathbf{c}'$, respectively. Then, the variance of the error coefficients after*

- *an addition of two ciphertexts* $\mathrm{Add}(\mathfrak{c}, \mathfrak{c}')$ *is*

$$\mathsf{Var}((\nu+\nu')|_i) = \mathsf{Var}(\nu|_i) + \mathsf{Var}(\nu'|_i). \tag{17}$$

- *a constant multiplication* $\mathrm{MulConst}(\alpha, \mathfrak{c})$ *is*

$$B_{\mathsf{const}}\mathsf{Var}(\nu|_i) \quad \textit{where} \quad B_{\mathsf{const}} = \frac{(t^2-1)n}{12}. \tag{18}$$

Proof.

- $\mathrm{Add}(\mathfrak{c}, \mathfrak{c}')$. Since $\mathbf{c}, \mathbf{c}'$ are two independently computed ciphertexts, then ν, ν' are independent themselves. Then, from Equation (6),

$$\mathsf{Var}((\nu+\nu')|_i) = \mathsf{Var}(\nu|_i) + \mathsf{Var}(\nu'|_i).$$

- $\mathrm{MulConst}(\alpha, \mathfrak{c})$. The coefficients of α behave as sampled independently at random from a uniform distribution over $\mathcal{U}_t$, thus we have that $\mathbb{E}[\alpha|_i] = 0$ and $\mathsf{Var}(\alpha|_i) \approx (t^2-1)/12$. It follows, by Equation (7) and the independence of α and ν, that $\mathsf{Var}(\nu_{\mathsf{const}}|_i) = \mathsf{Var}((\alpha\nu)|_i) = \sum_j \mathsf{Var}(\alpha|_j\nu|_{i-j})$, namely,

$$\sum_j \mathsf{Var}(\alpha|_j\nu|_{i-j}) = \sum_j \mathsf{Var}(\alpha|_j)\mathsf{Var}(\nu|_{i-j}) = \frac{(t^2-1)n}{12}\mathsf{Var}(\nu|_{i-j}).$$

□

Similar argument can be applied to the modulo and key switch operations.

Proposition 5 (Modulo switching). *Let $\mathfrak{c} = (\mathbf{c}, q_\ell, \nu)$ be an extended ciphertext. The variance of the error added by the modulo switch from q_ℓ to the target modulo q'_ℓ is*

$$\mathsf{Var}(\nu|_i) + V_{\mathsf{ms}}(q'_\ell) = \mathsf{Var}(\nu|_i) + \frac{B_{\mathsf{ms}}}{q'^2_\ell} \quad \textit{where} \quad B_{\mathsf{ms}} = \frac{t^2}{12}(1+nV_s). \tag{19}$$

Proof. The error added by the modulo switch from q_ℓ to q'_ℓ is independent from the starting error ν. Thus, by Equation (8), the total variance becomes $\mathsf{Var}(\nu|_i) + V_{\mathsf{ms}}(q'_\ell) = \mathsf{Var}(\nu|_i) + (\frac{t^2}{12}(1+nV_s))/q'^2_\ell$. □

Proposition 6 (Key switching). *Let $\mathfrak{d} = (\mathbf{d}, q_\ell, \nu)$ be an extended ciphertext, with $\mathbf{d} = (d_0, d_1, d_2) \in \mathcal{R}_{q_\ell}^3$. The variance of the error after the key switching is*

$$\mathsf{Var}(\nu|_i) + V_{\mathsf{ks}}(q_\ell), \tag{20}$$

where $V_{\mathsf{ks}}(q_\ell)$ depends on the chosen key-switching variants. Specifically, we have $V_{\mathsf{ks}}(q_\ell) = B_{\mathsf{ks}}/q_\ell^2$, where

$$B_{\mathsf{ks}} \leq \begin{cases} \frac{t^2}{12} k_\ell t^2 \sqrt[k]{q^2} n V_e & \text{for } BV \\ \frac{t^2}{12} \left(nV_e + 1 + nV_s\right) & \text{for } GHS \\ \frac{t^2}{12} \left(n(k+2)V_e + 1 + nV_s\right) & \text{for } GHS\text{-}RNS \\ \frac{t^2}{12} \left(\omega n V_e + 1 + nV_s\right) & \text{for } Hybrid \\ \frac{t^2}{12} \left((k+2\omega)nV_e + 1 + nV_s\right) & \text{for } Hybrid\text{-}RNS \end{cases} \tag{21}$$

Where ω, k and k_ℓ are defined as in Section 3.1.

Proof. Analogously of previous cases, the error added during the key switch procedure is independent from the starting error ν. Thus, the final variance is $\mathsf{Var}(\nu|_i) + V_{\mathsf{ks}}(q_\ell)$, where $V_{\mathsf{ks}}(q_\ell)$ depends on the following key-switching variants.

- *BV key switching.* Since $r_i \approx \sqrt[k]{q}$, by Equation (10) we have $V_{\mathsf{ks}}^{\mathsf{BV}}(q_\ell) \approx \frac{t^2}{q_\ell^2} \sum_{i=1}^{k_\ell} n \frac{r_i^2}{12} V_e \approx \frac{k_\ell t^2 \sqrt[k]{q^2} n V_e}{12 q_\ell^2}$.
- *GHS key switching.* From Equation (11) and $P \approx q$, we have $V_{\mathsf{ks}}^{\mathsf{GHS}}(q_\ell) \approx \frac{t^2}{q_\ell^2} \left(\frac{n q_\ell^2 V_e}{12 q^2} + \frac{1}{12} + \frac{nV_s}{12} \right) \leq \frac{t^2}{12 q_\ell^2} \left(nV_e + 1 + nV_s\right)$.
- *GHS RNS.* Analogously, from Equation (12), we obtain $V_{\mathsf{ks}}^{\mathsf{GHS-RNS}}(q_\ell) \approx \frac{t^2}{q_\ell^2} \left(\frac{n(k_\ell+2) q_\ell^2 V_e}{12 q^2} + \frac{1}{12} + \frac{nV_s}{12} \right) \leq \frac{t^2}{12 q_\ell^2} \left(n(k+2)V_e + 1 + nV_s\right)$.
- *Hybrid key switching.* Since $\tilde{r_i} \approx \sqrt[\omega]{q_\ell}$ and $P \approx \sqrt[\omega]{q}$, by Equation (13), we have $V_{\mathsf{ks}}^{\mathsf{Hyb}}(q_\ell) \approx \frac{t^2}{q_\ell^2} \left(\frac{\omega n \sqrt[\omega]{q_\ell^2} V_e}{12 \sqrt[\omega]{q^2}} + \frac{1}{12} + \frac{nV_s}{12} \right) \leq \frac{t^2}{12 q_\ell^2} \left(\omega n V_e + 1 + nV_s\right)$.
- *Hybrid RNS.* Finally, by Equation (14), we obtain $V_{\mathsf{ks}}^{\mathsf{Hyb-RNS}}(q_\ell)$ which is approximate $\frac{t^2}{q_\ell^2} \left(\frac{\omega n \sqrt[\omega]{q_\ell^2} \left(\frac{k_\ell}{\omega} + 2\right) V_e}{12 \sqrt[\omega]{q^2}} + \frac{1}{12} + \frac{nV_s}{12} \right) \leq \frac{t^2}{12 q_\ell^2} \left((k + 2\omega) n V_e + 1 + nV_s\right)$.

□

We want to point out that for all the relinearization variants except the BV one, B_{ks} is independent of q (and q_ℓ).

4.4 On the Estimation of the Variance in the Multiplication

In this section, we analyze the variance $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$ obtained from the multiplication of two independently-computed ciphertexts. The basic idea is to approximate its value with a simple formula involving the initial variances $\mathsf{Var}(\nu|_i)$, $\mathsf{Var}(\nu|_i)$. We do this as follows:

1. Firstly, we notice that $\mathsf{Var}((\nu\nu')|_i) \neq n\mathsf{Var}(\nu|_i)\mathsf{Var}(\nu'|_i)$. In particular, they differ in the powers of the secret key s. Indeed, where $\mathsf{Var}((\nu\nu')|_i$ contains $\sum_{j=0}^{n-1} s^{\iota_1+\iota_2}|_{i-j}^2$, $n\mathsf{Var}(\nu|_i)\mathsf{Var}(\nu'|_i)$ only has $\sum_{j_1=0}^{n-1} s^{\iota_1}|_{i-j_1}^2 \sum_{j_2=0}^{n-1} s^{\iota_2}|_{i-j_2}^2$. Hence, we will look for a function F such that $\sum_{j=0}^{n-1} s^{\iota_1+\iota_2}|_{i-j}^2 \approx F \cdot \sum_{j_1=0}^{n-1} s^{\iota_1}|_{i-j_1}^2 \sum_{j_2=0}^{n-1} s^{\iota_2}|_{i-j_2}^2$.
2. We analyze the relation between $\sum_{i=0}^{n-1} s^{\iota}|_i^2$ and $\sum_{i=0}^{n-1} s|_i^2 \sum_{i=0}^{n-1} s^{\iota-1}|_i^2$. For this special case, we find a function $f(\iota) = -1/e^{a\iota-b} + c$ that permits the approximation: $\sum_{i=0}^{n-1} s^{\iota}|_i^2 \approx f(\iota) \sum_{i=0}^{n-1} s|_i^2 \sum_{i=0}^{n-1} s^{\iota-1}|_i^2$.
3. In Lemma 3, we prove some properties of $f(\iota)$ and, in Theorem 1, we use these results to explicit F and give a bound on $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$.
4. Finally, we give a further simplification of $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$, especially useful in computing the closed formulas for the ciphertext modulus (Section 5).

Let us consider $\mathbf{c}, \mathbf{c}'$ two independently-computed ciphertexts and let their noises be respectively $\nu = \sum_{\iota_1=0}^{T_1} a_{\iota_1} s^{\iota_1}$ and $\nu' = \sum_{\iota_2=0}^{T_2} a'_{\iota_2} s^{\iota_2}$. By Equation (9), the error after the multiplication of $\mathbf{c}$ and $\mathbf{c}'$ is

$$\begin{aligned}\nu_{\mathsf{mul}}(q_\ell) = &-\textstyle\sum_{\iota_1}\sum_{\iota_2} a_{\iota_1} a'_{\iota_2} s^{\iota_1+\iota_2} + \sum_{\iota_1} a_{\iota_1}\left(\frac{t}{q'_\ell} c'_0 s^{\iota_1} + \frac{t}{q'_\ell} c'_1 s^{\iota_1+1}\right) + \\ &+ \textstyle\sum_{\iota_2} a_{\iota'_2}\left(\frac{t}{q_\ell} c_0 s^{\iota_2} + \frac{t}{q_\ell} c_1 s^{\iota_2+1}\right) + \frac{t}{q_\ell}(\varepsilon_0 + \varepsilon_1 s + \varepsilon_2 s^2).\end{aligned}$$

Then, thanks to Equation (15), the variance of its coefficients is

$$\begin{aligned}\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i) = n &\sum_{\iota_1}\sum_{\iota_2} \mathsf{Var}(a_{\iota_1}|_i)\mathsf{Var}(a'_{\iota_2}|_i) \sum_{j=0}^{n-1} s^{\iota_1+\iota_2}|_{i-j}^2 + \\ + n &\sum_{\iota_1} \mathsf{Var}(a_{\iota_1}|_i) \frac{t^2}{12} \sum_{j=0}^{n-1} \left(s^{\iota_1}|_{i-j}^2 + s^{\iota_1+1}|_{i-j}^2\right) + \\ + n &\sum_{\iota_2} \mathsf{Var}(a'_{\iota_2}|_i) \frac{t^2}{12} \sum_{j=0}^{n-1} \left(s^{\iota_2}|_{i-j}^2 + s^{\iota_2+1}|_{i-j}^2\right) + \\ + &\frac{t^2}{12q_\ell^2}\left(1 + \sum_{j=0}^{n-1} s|_{i-j}^2 + \sum_{j=0}^{n-1} s^2|_{i-j}^2\right).\end{aligned} \tag{22}$$

As mentioned before, the following part aims at deriving a simplified formula to approximate $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$ from $\mathsf{Var}(\nu|_i), \mathsf{Var}(\nu'|_i)$. However, considering

$$n\mathsf{Var}(\nu|_i)\mathsf{Var}(\nu'|_i) + n\mathsf{Var}(\nu|_i)\frac{t^2}{12}(1 + nV_s) + n\mathsf{Var}(\nu'|_i)\frac{t^2}{12}(1 + nV_s) + \ldots$$

from Equation (9), we only get

$$\begin{aligned}&n\textstyle\sum_{\iota_1}\sum_{\iota_2} \mathsf{Var}(a_{\iota_1}|_i)\mathsf{Var}(a'_{\iota_2}|_i) \sum_{j_1=0}^{n-1} s^{\iota_1}|_{i-j_1}^2 \sum_{j_2=0}^{n-1} s^{\iota_2}|_{i-j_2}^2 + \\ &+ n\textstyle\sum_{\iota_1} \mathsf{Var}(a_{\iota_1}|_i) \frac{t^2}{12} \sum_{j=0}^{n-1} s^{\iota_1}|_{i-j}^2 \left(1 + \sum_{j_1=0}^{n-1} s|_{i-j_1}^2\right) + \ldots\end{aligned}$$

For this reason, we need a function F that approximates $\sum_{j=0}^{n-1} s^{\iota_1+\iota_2}|_i^2$ as $F \cdot \sum_{j_1=0}^{n-1} s^{\iota_1}|_{i_1}^2 \sum_{j_2=0}^{n-1} s^{\iota_2}|_{i_2}^2$ for any ι_1, ι_2, i.e.

$$F \approx \frac{\sum_{i=0}^{n-1} s^{\iota_1+\iota_2}|_i^2}{\sum_{i_1=0}^{n-1} s^{\iota_1}|_{i_1}^2 \sum_{i_2=0}^{n-1} s^{\iota_2}|_{i_2}^2}.$$

We start by focusing on the particular case with $\iota_1 = 1$,

$$\frac{\sum_{i=0}^{n-1} s^{\iota}|_i^2}{\sum_{i_1=0}^{n-1} s|_{i_1}^2 \sum_{i_2=0}^{n-1} s^{\iota-1}|_{i_2}^2}, \tag{23}$$

analyzing its average value computationally for $\iota \geq 2$.

Heuristic 1 *For $\iota \geq 2$, Equation (23) is well-approximated by the function*

$$f(\iota) = -\frac{1}{e^{a\iota-b}} + c \approx \frac{\sum_{i=0}^{n-1} s^{\iota}|_i^2}{\sum_{i_1=0}^{n-1} s|_{i_1}^2 \sum_{i_2=0}^{n-1} s^{\iota-1}|_{i_2}^2}, \tag{24}$$

where a, b, c only depend on the distribution χ_s and the ring dimension n. We computed their values with Python function curve_fit[6]*:*

n	a	b	c
2^{12}	0.2417	2.3399	8.1603
2^{13}	0.2240	2.4181	8.8510
2^{14}	0.2058	2.4844	9.5691
2^{15}	0.1906	2.5489	10.2903

(a) $\chi_s = \mathcal{U}_3,\ \mathcal{ZO}(1/2),\ \mathcal{DG}(0, \sigma^2)$

n	a	b	c
2^{12}	0.2412	2.3087	7.9456
2^{13}	0.2191	2.3718	8.6115
2^{14}	0.2020	2.4355	9.2662
2^{15}	0.1871	2.4990	9.9499

(b) $\chi_s = \mathcal{HWT}(64)$

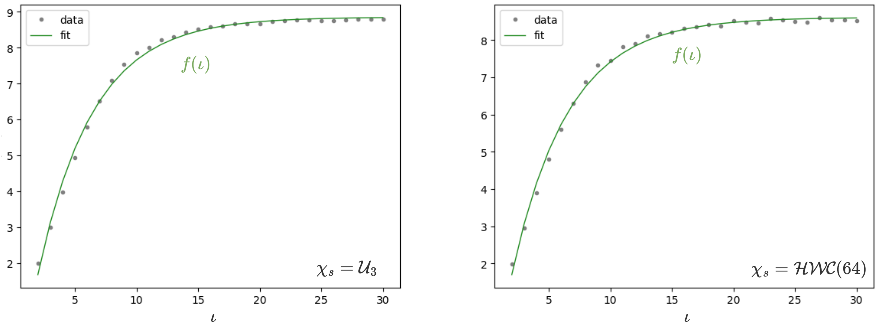


Fig. 3: Examples of $f(\iota)$ fitting the points from Equation (23) with $n = 2^{13}$.

To find F and, more in general, to provide an estimation of $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$, we need the following result on some properties of $f(\iota)$ (proof in Appendix B).

[6] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html

Lemma 3. *Let* $g(\iota) = \prod_{i=0}^{\iota} f(i)$ *with* $f(i)$ *as in Equation* (24) *and* $f(0) = f(1) = 1$. *Then*

(1) $\sum_{i=0}^{n-1} s^{\iota}|_i^2 \approx (nV_s)^{\iota} g(\iota)$.
(2) Let $\iota_j \in [0, T_j]$ *where* $T_j \in \mathbb{N}$, *for* $j = 1, 2$. *Then*

$$\frac{g(\iota_1 + \iota_2)}{g(\iota_1)g(\iota_2)} \leq \frac{g(T_1+T_2)}{g(T_1)g(T_2)}.$$

(3) Let $T_1, T_2 \in \mathbb{N}$, *then* $g(T_1+T_2)/g(T_1+1)g(T_2+1) \leq K_n$, *where* K_n *is a finite constant. In particular,* $K_{2^{12}} = 22$, $K_{2^{13}} = 39$, $K_{2^{14}} = 72$ *and* $K_{2^{15}} = 136$.

Thanks to Lemma 3, and under Heuristic 1, we are able to prove the following

Theorem 1. *Let* $\nu = \sum_{\iota_1=0}^{T_1} a_{\iota_1} s^{\iota_1}, \nu' = \sum_{\iota_2=0}^{T_2} a'_{\iota_2} s^{\iota_2}$ *be the noises of two independently-computed ciphertexts* $\mathbf{c}$ *and* $\mathbf{c}'$, *respectively. Then the variance of the error coefficients after the multiplication of* $\mathbf{c}$ *and* $\mathbf{c}'$ *is bounded by*

$$\begin{aligned} \mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i) \leq{}& n\mathsf{Var}(\nu|_i)\mathsf{Var}(\nu'|_i)\frac{g(T_1+T_2)}{g(T_1)g(T_2)} + n\mathsf{Var}(\nu|_i)\frac{t^2}{12}\big(1 + nV_s f(T_1+1)\big) \\ &+ n\mathsf{Var}(\nu'|_i)\frac{t^2}{12}\big(1 + nV_s f(T_2+1)\big) + \frac{t^2}{12q_\ell^2}\big(1 + nV_s + (nV_s)^2 f(2)\big), \end{aligned} \tag{25}$$

with $g(\iota) = \prod_{i=0}^{\iota} f(i)$, f *as in Equation* (24) *and* $f(0) = f(1) = 1$.

Proof. By point (1) of Lemma 3, we have that the correction function F is

$$F \approx \frac{\sum_{i=0}^{n-1} s^{\iota_1+\iota_2}|_i^2}{\sum_{i_1=0}^{n-1} s^{\iota_1}|_{i_1}^2 \sum_{i_2=0}^{n-1} s^{\iota_2}|_{i_2}^2} \approx \frac{g(\iota_1 + \iota_2)}{g(\iota_1)g(\iota_2)}.$$

In particular, from Equation (24), $\sum_{i=0}^{n-1} s^{\iota+1}|_i^2 \approx f(\iota) \sum_{i_1=0}^{n-1} s|_{i_1}^2 \sum_{i_2=0}^{n-1} s^{\iota}|_{i_2}^2$. Thus, Equation (22) can be written as

$$\begin{aligned} \mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i) \approx{}& n \sum_{\iota_1} \sum_{\iota_2} \mathsf{Var}(a_{\iota_1}|_i)\mathsf{Var}(a'_{\iota_2}|_i) \sum_{j_1=0}^{n-1} s^{\iota_1}|_{i-j_1}^2 \sum_{j_2=0}^{n-1} s^{\iota_2}|_{i-j_2}^2 \frac{g(\iota_1 + \iota_2)}{g(\iota_1)g(\iota_2)} \\ &+ n \sum_{\iota_1} \mathsf{Var}(a_{\iota_1}|_i)\frac{t^2}{12} \sum_{j=0}^{n-1} s^{\iota_1}|_{i-j}^2 \Big(1 + \sum_{j_1=0}^{n-1} s|_{i-j_1}^2 f(\iota_1 + 1)\Big) \\ &+ n \sum_{\iota_2} \mathsf{Var}(a'_{\iota_2}|_i)\frac{t^2}{12} \sum_{j=0}^{n-1} s^{\iota_2}|_{i-j}^2 \Big(1 + \sum_{j_1=0}^{n-1} s|_{i-j_1}^2 f(\iota_2 + 1)\Big) \\ &+ \frac{t^2}{12q_\ell^2}\Big(1 + nV_s + (nV_s)^2 f(2)\Big). \end{aligned}$$

To conclude, by Lemma 3 (2) and the monotonicity of f, we obtain

$$\begin{aligned}\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i) \leq n \sum_{\iota_1}\sum_{\iota_2} \mathsf{Var}(a_{\iota_1}|_i)\mathsf{Var}(a'_{\iota_2}|_i) \sum_{j_1=0}^{n-1} s^{\iota_1}|^2_{i-j_1} \sum_{j_2=0}^{n-1} s^{\iota_2}|^2_{i-j_2} \frac{g(T_1+T_2)}{g(T_1)g(T_2)} +\\ + n\sum_{\iota_1} \mathsf{Var}(a_{\iota_1}|_i)\frac{t^2}{12}\sum_{j=0}^{n-1} s^{\iota_1}|^2_{i-j}\Big(1+\sum_{j_1=0}^{n-1} s|^2_{i-j_1} f(T_1+1)\Big)+\\ + n\sum_{\iota_2} \mathsf{Var}(a'_{\iota_2}|_i)\frac{t^2}{12}\sum_{j=0}^{n-1} s^{\iota_2}|^2_{i-j}\Big(1+\sum_{j_1=0}^{n-1} s|^2_{i-j_1} f(T_2+1)\Big)+\\ + \frac{t^2}{12q_\ell^2}\Big(1+nV_s+(nV_s)^2 f(2)\Big),\end{aligned}$$

i.e., by Equation (15),

$$\begin{aligned}\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i) \leq n\mathsf{Var}(\nu|_i)\mathsf{Var}(\nu'|_i)\frac{g(T_1+T_2)}{g(T_1)g(T_2)} + n\mathsf{Var}(\nu|_i)\frac{t^2}{12}\Big(1+nV_s f(T_1+1)\Big)\\ + n\mathsf{Var}(\nu'|_i)\frac{t^2}{12}\Big(1+nV_s f(T_2+1)\Big) + \frac{t^2}{12q_\ell^2}\Big(1+nV_s+(nV_s)^2 f(2)\Big).\end{aligned}$$

□

Finally, we further simplify $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$, which is not explicitly dependent on the modulus q_ℓ anymore. To do this, we show that the first and last terms of (25) are negligible compared to the others.

Theorem 2 (Multiplication). *Let* $\nu = \sum_{\iota_1=0}^{T_1} a_{\iota_1} s^{\iota_1}, \nu' = \sum_{\iota_2=0}^{T_2} a'_{\iota_2} s^{\iota_2}$ *be the noises of two independently computed ciphertext* $\mathbf{c}$ *and* $\mathbf{c}'$*, respectively. Then the variance of the error coefficients after the multiplication of* $\mathbf{c}$ *and* $\mathbf{c}'$ *is well-approximated by*

$$\mathsf{Var}(\nu_{\mathsf{mul}}|_i) \approx \frac{t^2n^2V_s}{12}\Big(\mathsf{Var}(\nu|_i)f(T_1+1)+\mathsf{Var}(\nu'|_i)f(T_2+1)\Big), \tag{26}$$

with $g(\iota)=\prod_{i=0}^{\iota} f(i)$ *with* $f(i)$ *as in Equation* (24) *and* $f(0)=f(1)=1$.

Proof. We start from the first term in (25). To guarantee correct decryption, we impose the bound we computed on $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$ in (25) $\leq \frac{1}{8D^2}$. Since all the addends are positive quantities, this implies

$$\mathsf{Var}(\nu|_i)\frac{t^2n^2V_s}{12}f(T_1+1) \leq \frac{1}{8D^2},$$

i.e. $\mathsf{Var}(\nu|_i) \leq \frac{3}{2D^2t^2n^2V_sf(T_1+1)}$. Then, by (3) in Lemma 3,

$$n\mathsf{Var}(\nu|_i)\mathsf{Var}(\nu'|_i)\frac{g(T_1+T_2)}{g(T_1)g(T_2)} \leq n\frac{3}{2D^2t^2n^2V_sf(T_1+1)}\mathsf{Var}(\nu'|_i)\frac{g(T_1+T_2)}{g(T_1)g(T_2)}$$

$$\leq \frac{18K_n}{D^2 t^4 n^3 V_s^2} \frac{t^2 n^2 V_s}{12} f(T_2+1)\mathsf{Var}(\nu'|_i) \ll \frac{t^2 n^2 V_s}{12} f(T_2+1)\mathsf{Var}(\nu'|_i).$$

To prove that the last term is negligible, we consider two cases: either if we are in the modulus q or if a modulo switch to another modulus q_ℓ has been performed. In the first event, we know that $\mathsf{Var}(\nu|_i) \geq B_{\mathsf{clean}}/q^2$, since all the homomorphic operations performed increase the variance of the error coefficients. Therefore, by Equation (16), we get

$$\begin{aligned}\frac{t^2 n}{12}\mathsf{Var}(\nu|_i)\big(1+nV_s f(T_1+1)\big) &\geq \frac{t^2 n}{12}\frac{B_{\mathsf{clean}}}{q^2}\big(1+nV_s f(T_1+1)\big)\\ &\geq \frac{t^2}{12q^2} t^2 n^2 V_e V_s \big(1+nV_s f(T_1+1)\big) \gg \frac{t^2}{12q^2}\big(1+nV_s+n^2V_s^2 f(2)\big).\end{aligned}$$

The argument for the second event is analogous from $\mathsf{Var}(\nu|_i) \geq B_{\mathsf{ms}}/q_\ell^2$ and Equation (19). Hence, we can set

$$\begin{aligned}\mathsf{Var}(\nu_{\mathsf{mul}}|_i) &\approx \tfrac{t^2 n}{12}\Big(\mathsf{Var}(\nu|_i)\big(1+nV_s f(T_1+1)\big)+\mathsf{Var}(\nu'|_i)\big(1+nV_s f(T_2+1)\big)\Big)\\ &\approx \tfrac{t^2 n^2 V_s}{12}\big(\mathsf{Var}(\nu|_i)f(T_1+1)+\mathsf{Var}(\nu'|_i)f(T_2+1)\big).\end{aligned}$$

□

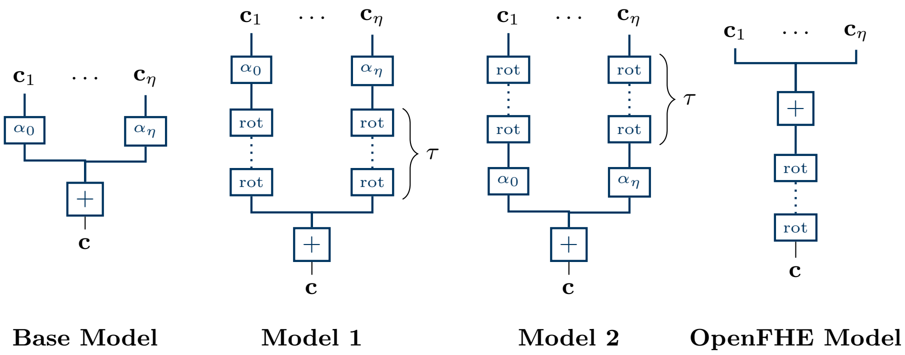
5 Modeling the Homomorphic Circuit

In this section, we exploit our theoretical work (Section 4) to improve the parameter generation for the BFV scheme, providing closed formulas to compute the ciphertext modulus q and, eventually, its sub-moduli p_j. These formulas are employed in our tool, which provides automated parameter selection for non-FHE experts (Section 5.3). In our analysis, we extend the previous work on BGV, focusing on the circuit models newly proposed by Mono *et al.* [23]. It is important to note that the arguments presented in the following can be easily tailored to suit any sequence of homomorphic operations, including non-repetitive ones.

Each circuit performs a list of operations on η ciphertexts c_i in parallel, as illustrated in Figure 4. The resulting ciphertexts are homomorphically multiplied with other ones computed analogously. This sequence is repeated M times.

Base model This is a simplified version of the other models, performing constant multiplications on the ciphertexts and summing them afterwards, before the homomorphic multiplication. It is mainly used to make the analysis easier, and it is equal to Model 1 and 2 with $\tau = 0$.

Model 1 & 2 Models 1 and 2 extend the Base Model performing τ rotations either after or before the constant multiplications, respectively.



Base Model **Model 1** **Model 2** **OpenFHE Model**

Fig. 4: Sequences of operations in the different models.

OpenFHE Model For comparison with previous work, we also define the model used in the OpenFHE library [20,1]. Here the first operation to be performed is a homomorphic multiplication, then η additions and τ rotations are carried out.

In the following, we consider the input ciphertexts in a circuit to encrypt different messages, therefore independent of each other. Moreover, we focus our analysis on Model 2, as it has the worst error growth. The same techniques can, however, be applied to all models as well, and we provide the results of our study in Table 1. In Section 5.2, we will study two other types of circuits, common in practice, making use of the modulus switching technique.

5.1 Closed Formulas for BFV Parameters

Consider a circuit with multiplicative depth $M = L - 1$. We analyse the noise growth in it through the variance, as seen in section Section 4. Let V_ℓ denote the variance of the error coefficients after the ℓ-th level. In particular, $V_0 = V_{\mathsf{clean}}$ is the variance just after the encryption, and V_ℓ is the variance after the ℓ-th multiplication. Since the variance increases with each operation, we only need to ensure that the final error coefficients (with variance V_{L-1}) satisfy the condition in Section 4 for correct decryption throughout the circuit, i.e. $V_{L-1} \leq 1/8D^2$.

We now examine the ℓ-th level of Model 2, in order to compute V_{L-1} recursively. Given the variance $V_{\ell-1}$ of each ciphertext in the circuit input, the evolution of the model can be described as follows:

- We first apply τ rotations, obtaining by Equation (20) $V_{\ell-1} + \tau V_{\mathsf{ks}}$.
  Note that when the modulus is not explicitly specified in the formulas, it is assumed to be q.
- Secondly, we have a constant multiplication. Thus, the variance is multiplied by $B_{\mathsf{const}} = \frac{(t^2-1)n}{12}$ (18), becoming $(V_{\ell-1} + \tau V_{\mathsf{ks}})B_{\mathsf{const}}$.
  If constant multiplications are not required, we set $B_{\mathsf{const}} = 1$.
- We add η ciphertexts, getting by Equation (17),

$$\eta(V_{\ell-1} + \tau V_{\mathsf{ks}})B_{\mathsf{const}}.$$

- During homomorphic multiplication, a modulo switch is applied from q to $q' \approx q$ on one of the ciphertexts. This operation, adds to the variance a quantity $V_{\mathsf{ms}}(q') \approx V_{\mathsf{ms}}(q)$, Equation (19), leading to a total variance of

  $$\eta(V_{\ell-1} + \tau V_{\mathsf{ks}})B_{\mathsf{const}} + V_{\mathsf{ms}}.$$

  Finally, after performing multiplication (with re-linearization) of two ciphertexts, we have, thanks to Equations (20) and (26),

  $$\begin{aligned} V_\ell &\approx \frac{t^2n^2V_s}{12}\Big(2\eta(V_{\ell-1} + \tau V_{\mathsf{ks}})B_{\mathsf{const}} + V_{\mathsf{ms}}\Big)f(\ell+1) + V_{\mathsf{ks}} \\ &\approx \frac{t^2n^2V_s}{12}\Big(2\eta(V_{\ell-1} + \tau V_{\mathsf{ks}})B_{\mathsf{const}} + V_{\mathsf{ms}}\Big)f(\ell+1). \end{aligned} \tag{27}$$

  since V_{ks} is negligible.

Since $V_\ell = B_\ell/q^2$ with B_ℓ is independent of q, we can rewrite Equation (27) as

$$V_\ell = \frac{B_\ell}{q^2} \approx \frac{(AB_{\ell-1} + C)f(\ell+1)}{q^2} \tag{28}$$

where $A = \frac{\eta t^2n^2V_s}{6}B_{\mathsf{const}}$ and $C = \frac{t^2n^2V_s}{12}(2\eta\tau B_{\mathsf{ks}}B_{\mathsf{const}} + B_{\mathsf{ms}})$. From Equation (28), we can recursively compute the final variance

$$\begin{aligned} V_{L-1} = \frac{B_{L-1}}{q^2} &\approx \frac{(AB_{L-2} + C)f(L)}{q^2} \approx \frac{A(AB_{L-3} + C)f(L-1)f(L)}{q^2} \\ &\approx \cdots \approx \frac{A^{L-2}(AB_{\mathsf{clean}} + C)g(L)}{q^2}, \end{aligned}$$

and use it to determine a bound on the ciphertext modulus. Indeed, since $V_{L-1} \leq 1/8D^2$, we have

$$q^2 \geq 8D^2A^{L-2}(AB_{\mathsf{clean}} + C)g(L). \tag{29}$$

Note that the bound on the modulus q is computed in the same way for all the models, except for the OpenFHE one, where the multiplication is done at the beginning of the circuit. In this case, we approximate $V_\ell = \frac{AB_{\ell-1}f(\ell+1)+C}{q^2}$, hence

$$q^2 \geq 8D^2A^{L-2}(AB_{\mathsf{clean}} + C/f(2))g(L). \tag{30}$$

In Table 1, we list the resulting A and C depending on the models.

5.2 Other Circuits Exploiting the Modulo Switch

In this section, we study two different kinds of circuits in which the modulo switch to smaller moduli is employed: the BGV-like one, see [7], and that proposed by Kim *et al.* in [20]. This technique was first introduced in the BGV scheme to reduce the error associated with a ciphertext. In the BFV scheme, this operation does not decrease the error, and the computations in smaller moduli have larger errors; however, it can still be useful for efficiency. In the next paragraphs, we briefly analyse the circuits, considering Model 2, Figure 4, and propose a set of parameters limiting the error growth difference with the previous circuit.

Model	A	C
Base Model	$\frac{\eta t^2 n^2 V_s}{6} B_{\mathsf{const}}$	$\frac{t^2 n^2 V_s}{12} B_{\mathsf{ms}}$
Model 1	$\frac{\eta t^2 n^2 V_s}{6} B_{\mathsf{const}}$	$\frac{t^2 n^2 V_s}{12}(2\eta\tau B_{\mathsf{ks}} + B_{\mathsf{ms}})$
Model 2	$\frac{\eta t^2 n^2 V_s}{6} B_{\mathsf{const}}$	$\frac{t^2 n^2 V_s}{12}(2\eta\tau B_{\mathsf{ks}} B_{\mathsf{const}} + B_{\mathsf{ms}})$
OpenFHE Model	$\frac{\eta t^2 n^2 V_s}{6}$	$(\eta+\tau)B_{\mathsf{ks}}$

Table 1: A, C to compute q with Equation (29) or (30) for the OpenFHE one.

BGV-like circuit In this type of circuit, the modulo switch is performed before every round of operations. Using the same argument as in section 5.1, we compute the noise variance starting from $V_0^{\mathsf{ms}} = V_{\mathsf{clean}}$ and only need to ensure that the final one, V_{L-1}^{ms}, is bounded. The analysis differs for the presence of many moduli, at the ℓ-th level we switch from $q_{L-\ell+1}$ to $q_{L-\ell}$, yielding

$$V_{\ell-1}^{\mathsf{ms}} + V_{\mathsf{ms}}(q_{L-\ell}) = V_{\ell-1}^{\mathsf{ms}} + \frac{B_{\mathsf{ms}}}{q_{L-\ell}^2}$$

with B_{ms} as in Equation (19) and the errors of the next operations are divided by $q_{L-\ell}^2$ as well. Thus, similarly to Equation (27), we have

$$V_\ell^{\mathsf{ms}} \approx \frac{t^2 n^2 V_s}{12}\left(2\eta\left(V_{\ell-1}^{\mathsf{ms}} + \frac{B_{\mathsf{ms}} + \tau B_{\mathsf{ks}}}{q_{L-\ell}^2}\right)B_{\mathsf{const}} + \frac{B_{\mathsf{ms}}}{q_{L-\ell}^2}\right) f(\ell+1).$$

Therefore

$$V_\ell^{\mathsf{ms}} \approx \left(A_{\mathsf{ms}} V_{\ell-1} + \frac{C_{\mathsf{ms}}}{q_{L-\ell}^2}\right) f(\ell+1), \tag{31}$$

where $A_{\mathsf{ms}} = \frac{\eta t^2 n^2 V_s}{6} B_{\mathsf{const}}$ and $C_{\mathsf{ms}} = \frac{t^2 n^2 V_s}{12}\big(2\eta\tau B_{\mathsf{ks}} B_{\mathsf{const}} + (2\eta B_{\mathsf{const}} + 1)B_{\mathsf{ms}}\big)$. Note that $A_{\mathsf{ms}} = A$ and $C_{\mathsf{ms}} > C$, where A, C are as in Table 1 for Model 2. Thanks to Equation (31), we can recursively compute the variance V_{L-1}^{ms} as

$$\begin{aligned} V_{L-1}^{\mathsf{ms}} &\approx A V_{L-2}^{\mathsf{ms}} f(L) + \tfrac{C_{\mathsf{ms}}}{q_1^2} f(L) \approx \\ &\approx A^2 V_{L-3}^{\mathsf{ms}} f(L-1) f(L) + \tfrac{A C_{\mathsf{ms}}}{q_2^2} f(L-1) f(L) + \tfrac{C_{\mathsf{ms}}}{q_1^2} f(L) \approx \cdots \approx \\ &\approx A^{L-1} V_0^{\mathsf{ms}} f(2) \cdots f(L) + \textstyle\sum_{i=1}^{L-1} \tfrac{A^{i-1} C_{\mathsf{ms}}}{q_i^2} f(L-i+1) \cdots f(L), \end{aligned}$$

therefore,

$$\frac{A^{L-1} B_{\mathsf{clean}}}{q_L^2} g(L) + \sum_{i=1}^{L-1} \frac{A^{i-1} C_{\mathsf{ms}}}{q_i^2} \frac{g(L)}{g(L-i)} \leq \frac{1}{8D^2}. \tag{32}$$

Observe that, since $C_{\mathsf{ms}} > C$ and $q_\ell \leq q_L$, $V_{L-1}^{\mathsf{ms}} > V_{L-1}$. This implies that the ciphertext modulus obtained with the modulus switch technique, $q_{\mathsf{ms}} = q_L$, is bigger than the one obtained in Equation (29), q. However, we can select specific sub-moduli for them to be close, improving efficiency.

Fact 1 *An optimal choice for the p_j's, maximizing the efficiency while keeping the ciphertext modulus close to the one gotten without modulus-switching, is* *obtained when the addends in Equation (32) are approximately of the same size, namely when*

$$p_1^2 \approx 8D^2LC_{\mathsf{ms}}f(L), \quad p_\ell^2 \approx Af(L{-}\ell{-}1), \quad p_L^2 \approx \frac{AB_{\mathsf{clean}}}{C_{\mathsf{ms}}}.$$

Then $q_{\mathsf{ms}}^2 \approx 8D^2LA^{L-1}B_{\mathsf{clean}}g(L)$*, which means that* q_{ms} *is approximately* $\sqrt{L}$ *times the ciphertext modulus* q *in Equation (29).*

Proof. We begin our proof by contradiction, assuming that there exists at least one index i in Equation (32) such that

$$\frac{A^{i-1}C_{\mathsf{ms}}}{q_i^2}\frac{g(L)}{g(L{-}i)} \gg \frac{A^{L-1}B_{\mathsf{clean}}}{q_L^2}g(L), \tag{33}$$

Then, called $N \geq 1$ the number such indices, we get from Equation (32)

$$V_{L-1}^{\mathsf{ms}} \approx \frac{NA^{i-1}C_{\mathsf{ms}}}{q_i^2}\frac{g(L)}{g(L{-}i)} \leq \frac{1}{8D^2}$$

and, consequently, $q_i^2 \geq 8D^2NA^{i-1}C_{\mathsf{ms}}\frac{g(L)}{g(L-i)}$. From Equation (33), it also follows $\frac{q_L^2}{q_i^2} \gg \frac{A^{L-i}B_{\mathsf{clean}}}{C_{\mathsf{ms}}}g(L{-}i)$, which implies $q_{\mathsf{ms}}^2 \gg 8D^2NA^{L-1}B_{\mathsf{clean}}g(L)$, much larger than the bound for q given by (29).

Thus, we now suppose that, for any index i, we have

$$\frac{A^{i-1}C_{\mathsf{ms}}}{q_i^2}\frac{g(L)}{g(L{-}i)} \leq \frac{A^{L-1}B_{\mathsf{clean}}}{q_L^2}g(L). \tag{34}$$

So that

$$V_{L-1}^{\mathsf{ms}} \leq \frac{LA^{L-1}B_{\mathsf{clean}}}{q_L^2}g(L), \tag{35}$$

namely, $q_{\mathsf{ms}}^2 \geq 8D^2LA^{L-1}B_{\mathsf{clean}}g(L)$. From Equation (34) we get

$$p_L^2 \leq \frac{AB_{\mathsf{clean}}}{C_{\mathsf{ms}}},\ p_{L-1}^2p_L^2 \leq \frac{A^2B_{\mathsf{clean}}}{C_{\mathsf{ms}}}g(2),\ \ \ldots,\ p_2^2\cdots p_L^2 \leq \frac{A^{L-1}B_{\mathsf{clean}}}{C_{\mathsf{ms}}}g(L{-}1).$$

Moreover, from Equation (35), we take $p_1^2\cdots p_L^2 \approx 8D^2LA^{L-1}B_{\mathsf{clean}}g(L)$. For maximal efficiency, we choose p_1 to be as small as possible by setting $p_2^2\cdots p_L^2$ the largest, i.e. satisfying $p_2^2\cdots p_L^2 \approx A^{L-1}B_{\mathsf{clean}}g(L{-}1)/C_{\mathsf{ms}}$. This yields $p_1^2 \approx 8D^2LC_{\mathsf{ms}}f(L)$. We can apply the same argument iteratively to $p_2, \ldots, p_{L-1}$, obtaining the values of the thesis, i.e. $p_\ell^2 \approx Af(L{-}\ell{-}1)$, for $\ell = 2, \ldots, L-1$. Finally, from these values and $p_1^2\cdots p_L^2 \approx 8D^2LA^{L-1}B_{\mathsf{clean}}g(L)$, we get $p_L^2 \approx AB_{\mathsf{clean}}/C_{\mathsf{ms}}$. □

KPZ-leveled circuit In [20], the authors proposed a different approach, switching to a smaller modulus q_{lev} only during multiplication, which is the most expensive operation. Therefore, we obtain

$$V_\ell \approx 2\left[\eta\left(V_{\ell-1}+\frac{\tau B_{\mathsf{ks}}}{q^2}\right)B_{\mathsf{const}}+\frac{B_{\mathsf{ms}}}{q_{lev}^2}\right]\frac{t^2n^2V_s}{12}f(\ell+1)+\frac{B_{\mathsf{ms}}}{q^2}+\frac{B_{\mathsf{ks}}}{q^2},$$

which, written as $V_\ell \approx AV_{\ell-1}f(\ell+1)+\frac{C_1f(\ell+1)+C_2}{q^2}+\frac{Ef(\ell+1)}{q_{lev}^2}$, yields

$$V_{L-1}\approx A^{L-2}g(L)\left[\frac{AB_{\mathsf{clean}}+C_1+C_2/f(2)}{q^2}+\frac{E}{q_{lev}^2}\right]\leq\frac{1}{8D^2},$$

with $A=\frac{\eta t^2n^2V_s}{6}B_{\mathsf{const}}$, $C_1=\eta\tau\frac{t^2n^2V_s}{6}B_{\mathsf{const}}B_{\mathsf{ks}}$, $C_2=B_{\mathsf{ms}}+B_{\mathsf{ks}}$, $E=\frac{t^2n^2V_s}{6}B_{\mathsf{ms}}$. To have a level of security similar to the previous one, we can take q_{lev} such that $\frac{AB_{\mathsf{clean}}+C_1+C_2/f(2)}{q^2}\approx\frac{E}{q_{lev}^2}$, i.e.

$$q_{lev}^2\approx\frac{E}{AB_{\mathsf{clean}}+C_1+C_2/f(2)}q^2.$$

Then the bound on q become approximately

$$q^2\geq 16D^2A^{L-2}g(L)(AB_{\mathsf{clean}}+C_1+C_2/f(2)).$$

5.3 A Parameter Generator for BFV

To make our work more valuable and approachable for practical purposes, we provide automated parameter generation implemented in Python and publicly available on GitHub [7]. We integrated our theoretical work for the BFV scheme in the tool of Mono *et al.* [23], combining the correctness analysis developed in the previous sections with the formula for security in their paper.

The generator interacts with the user by presenting a list of mandatory and optional inputs, generating code snippets containing the obtained parameters. As a result, it offers high versatility and comprehensiveness, supporting multiple state-of-the-art libraries and all the circuits in Figure 4. Moreover, its implementation is easily adaptable to any sequence of operations.

To support arbitrary circuit models, we adapt Mono *et al.* approach for the key switching noise estimation to our average-case analysis: we use fixed values for β and ω, per default $\beta = 2^{10}$ and $\omega = 3$. If applicable, we set the key switching modulus P to be roughly equal to the ciphertext modulus q in the GHS variant and to the submoduli $\tilde{r}_i$ that split it in the Hybrid one, and scale it by a constant K, per default $K = 100$. Now, we can use this estimate for the extension modulus to compute the noise bound programmatically. Note that we slightly overestimate the error this way but the error growth from the key switching is rather small compared to other operations, thus using this estimate results in valid parameter sets. This generalizes our theoretical work to arbitrary, use-case-specific circuit models with an easy-to-use interface.

[7] https://github.com/Crypto-TII/fhegen

Model	`'Base'`, `'Model1'`, `'Model2'`, `'OpenFHE'`
t or $\log t$	any integer ≥ 2
λ or m	any integer ≥ 40 or ≥ 4, respectively
M, η	any integer > 0
τ	any integer ≥ 0
Library	`'None'`, `'OpenFHE'`, `'PALISADE'`, `'SEAL'`
Full Batching	full batching with t, `'True'` or `'False'`
Secret Distribution	`'Ternary'`, `'Error'`
Key Switching	`'Hybrid'`, `'BV'`, `'GHS'`
β	any integer ≥ 2
ω	any integer ≥ 1

Table 2: Required and optional inputs to the parameter generator

6 Comparison with Previous Works

In this section, we demonstrate the efficacy of our average-case approach by comparing it to the state-of-the-art works [19,10,12,20] and the practical errors arising from OpenFHE [1].

In particular, we conduct this analysis for the basic homomorphic operations: Encryption of a fresh ciphertext and Addition and Multiplication between 2 fresh independent computed ciphertexts (Table 5). Moreover, in Table 6 we focus on circuits: the Base Model circuits (Figure 4) with $\eta = 8$ and of depth 2 and 3, respectively. Obviously, we can apply our analysis to any circuit evaluated on independent ciphertexts.

To ensure clarity, we summarize the main results needed for the comparison. The bounds with the canonical norm are computed following the latest work by Costache *et al.* [12], and Iliashenko [19], taking into account the modifications we made to the encryption and multiplication algorithms based on the work of Kim *et al.* [20]. Moreover, we recall our formulas from Sections 4 and 5.

Canonical norm. In contrast to our approach, the latest works establishing theoretical bounds on the BFV noise growth propose a worst-case analysis employing either the infinity norm [20] or the canonical norm [19,10,12]. The canonical norm is known to result in better parameters.

In Table 3 we summarize how the error behaves when the homomorphic operations are performed considering the error bounds using the canonical norm.

In [10], the authors used the bound $||a||^{can} \leq D\sqrt{nV_a}$ for polynomials $a \in \mathcal{R}$, assuming independence among the coefficients and V_a being the variance of the coefficients of a. With the same hypothesis, we can bound the canonical norm of the invariant noise ν with $||\nu||^{can} \leq D\sqrt{nV}$, whose probability is greater or equal to $1 - ne^{-D^2}$, by Equation (4). In line with the previous works, we set $D = 6$ which guarantees the bound with probability at least $1 - 2^{-36}$. It's worth noting that, in a practical scenario is better to choose $D = 8$ since the probability of failure is limited to 2^{-77} (for n smaller than 2^{15}).

Homomorphic operation	Error bounds with canonical norm
`Enc`	$\lVert\nu_{\mathsf{clean}}\rVert^{\mathrm{can}} \leq D\frac{t}{q}\sqrt{n\left(\frac{1}{12}+nV_eV_u+V_e+nV_eV_s\right)}$
`Mod Switch`(q')	$\lVert\nu+\nu_{\mathsf{ms}}(q')\rVert^{\mathrm{can}} \leq \lVert\nu\rVert^{\mathrm{can}} + \frac{D\sqrt{nB_{\mathsf{ms}}}}{q'}$
`Key switch`(q)	$\lVert\nu+\nu_{\mathsf{ks}}\rVert^{\mathrm{can}} \leq \lVert\nu\rVert^{\mathrm{can}} + D\sqrt{nV_{\mathsf{ks}}}$
`Add`$(\mathbf{c},\mathbf{c}')$	$\lVert\nu+\nu'\rVert^{\mathrm{can}} \leq \lVert\nu\rVert^{\mathrm{can}} + \lVert\nu'\rVert^{\mathrm{can}}$
`Const`$(\mathbf{c})$	$\lVert\alpha\nu\rVert^{\mathrm{can}} \leq D\sqrt{n\frac{(t^2-1)}{12}}\lVert\nu\rVert^{\mathrm{can}}$
`Mult`$(\mathbf{c},\mathbf{c}')$	$\lVert\nu_{\mathrm{mul}}\rVert^{\mathrm{can}} \leq \left(2\lVert\nu\rVert^{\mathrm{can}} + D\sqrt{nV_{\mathsf{ms}}(q)}\right)Dt\sqrt{\frac{n}{12}(1+nV_s)}$

Table 3: Canonical norm depending on the homomorphic operations.

Applying the same argument of Section 5.1, we get that the following bound on the final error of a Base Model circuit: $\lVert\nu_{L-1}\rVert^{\mathrm{can}} \leq A^{L-2}\left(AD\sqrt{nB_{\mathsf{clean}}}+C\right)/q$, with $A = D\eta t\sqrt{\frac{n}{3}(1+nV_s)}$ and $C = \frac{D^2t^2n}{12}(1+nV_s)$. Since the norm has to satisfy $\lVert\nu_{L-1}\rVert^{\mathrm{can}} \leq 1/2$, it follows that

$$q \geq 2A^{L-2}\left(AD\sqrt{nB_{\mathsf{clean}}}+C\right). \tag{36}$$

Average-case bounds. In the average-case approach, we set $\lVert\nu\rVert_\infty \leq D\sqrt{2V}$ with V variance of each coefficient of ν. Thanks to Equation (3), the bound holds with probability at least $1-n\left(1-\mathrm{erf}(D)\right)$, which for $D=6$ is at least $1-2^{-40}$.

Summarizing the results of Section 4, let ν,ν' be the invariant noises associated with the ciphertexts $\mathbf{c}$ and $\mathbf{c}'$, results of independent circuits of depth $\ell-1$. Let V be the variance of their coefficients, in Table 4 we recall how it changes depending on the homomorphic operations.

Homomorphic operation	Variance
`Enc`	$\frac{t^2}{q^2}\left(\frac{1}{12}+nV_eV_u+V_e+nV_eV_s\right)$
`Mod Switch`(q')	$V+\frac{t^2(1+nV_s)}{12q'^2}$
`Key switch`(q)	$V+V_{\mathsf{ks}}(q)$
`Add`$(\mathbf{c},\mathbf{c}')$	$2V$
`Const`$(\mathbf{c})$	$\frac{(t^2-1)n}{12}V$
`Mult`$(\mathbf{c},\mathbf{c}')$	$\frac{t^2n^2V_s}{12}(2V+V_{\mathsf{ms}})f(\ell+1)$

Table 4: Variance depending on the homomorphic operations.

In Tables 5 and 6, we compare the error analysis. For readability, we do not show the bounds themselves, but their *noise budget* [26]:

$$-\log_2(2\cdot\lVert\nu\rVert) = \log_2\left(\tfrac{1}{2}\right) - \log_2(\lVert\nu\rVert).$$

Roughly speaking, it measures in bits the distance between the input and $\frac{1}{2}$, limit for correct decryption.

The tag “can” denotes the state-of-the-art analysis carried out with the canonical norm, “our” presents the results obtained with the average-case approach presented in this paper, “exp” shows the observed values from OpenFHE [1] library with 10.000 polynomial samples (Table 5) and 100 (Table 6). We additionally display the average of the absolute error values under “mean”, in Table 6 we also present our estimation of it as $\sqrt{V}$ with the tag “our”.

For parameters, we use $t = 65537$, $n = 2^{12}, \dots, 2^{15}$ and q set by the library to have at least 128 bit security. We use Hybrid key switching and HPSPOVERQ multiplication and set $D = 6$, $\chi_s = \chi_u = \mathcal{U}_3$, and $\chi_e = \mathcal{DG}(0, \sigma^2)$, with $\sigma = 3.19$.

In Table 5, we display the results after only the encryption, an encryption followed by an addition or an encryption followed by a multiplication.

	Encryption				Addition				Multiplication			
	maximum value			mean	maximum value			mean	maximum value			mean
n	can	our	exp	exp	can	our	exp	exp	can	our	exp	exp
2^{12}	26.5	32.0	32.7	35.4	86.0	91.5	92.1	94.9	57.0	65.1	65.9	68.7
2^{13}	25.5	31.5	32.2	34.9	85.0	91.0	91.6	94.4	55.0	63.6	64.3	66.2
2^{14}	24.5	31.0	31.5	34.4	84.0	90.5	91.1	93.9	53.0	62.1	62.8	65.7
2^{15}	23.5	30.5	31.0	33.9	83.0	90.0	90.5	93.4	51.0	60.6	61.2	64.2

Table 5: Encryption, addition and multiplication of fresh ciphertexts.

In Table 6, we consider the Base Model circuit (Figure 4) of depth 2 and 3, taking $\eta = 8$.

	2 multiplications					3 multiplications				
	maximum value			mean value		maximum value			mean value	
n	can	our	exp	our	exp	can	our	exp	our	exp
2^{12}	21.5	35.0	35.9	38.1	38.6	-	-	-	-	-
2^{13}	18.5	32.5	33.6	35.6	36.1	45.0	62.5	63.6	65.6	66.3
2^{14}	15.5	30.0	30.9	33.1	33.6	41.0	59.1	60.1	62.2	62.7
2^{15}	12.5	27.6	28.4	30.7	31.1	37.0	55.6	56.4	58.7	59.2

Table 6: Comparison in the Base Model of depth 2 and 3 with $\alpha = 1$ and $\eta = 8$.

Tables 5 and 6 suggest that our approach is a promising method for analyzing noise in the BFV scheme. It provides more accurate results, very close to the experimentally observed ones, and it substantially improves upon previous works, especially as the multiplicative depth of the circuit grows.

Our last comparison is on the ciphertext modulus q. In Table 7, we present the obtained bounds for $\log_2(q)$ following from the two theoretical approaches

(Equation (36) and Equation (29)) when $M = 3$ and $\eta = 8$. We set $D = 8$ to have a failure probability smaller than 2^{-80}, which is usually required in a practical scenario.

n	2^{12}	2^{13}	2^{14}	2^{15}
can	75.0	79.0	83.0	87.0
our	56.7	60.2	63.7	67.2

Table 7: Comparison of $\log_2(q)$ in the Base Model circuit of depth 3 and $\eta = 8$.

Here we can see the impact that a better noise analysis has on the scheme's efficiency and security.

Finally, in Figure 5, we graphically compare our parameter generation with the OpenFHE one, based on theoretical work with the infinity norm [20]. We compare our generated bounds with the size of the ciphertext modulus generated for $\lambda = 128$.

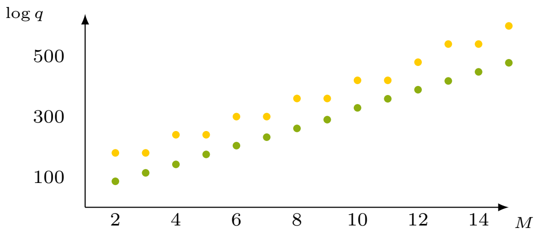


Fig. 5: Comparison of modulus sizes across multiplicative depths M with $\lambda = 128$ and $t = 2^{16} + 1$ for OpenFHE ● and our ● parameter generation.

7 Conclusion

To conclude, our average-case noise analysis outperforms the state-of-the-art methods for BFV, as shown in the examples in Section 6. Moreover, our approach provides very precise estimations for any multiplicative depth, in the examples they deviate by no more than 1.1 bits from the values observed in experiments. This is reflected in significantly smaller bounds on the ciphertext modulus, resulting in significantly improved performance. In addition, the finding of simple closed formulas for correctness facilitates the task of parameters selection. Furthermore, the development of the first automated parameter generation tool for BFV makes the scheme accessible to a wider range of users, still ensuring security, correctness, and high efficiency.

Future work. It is worth noting that this approach is expected to be adaptable to BGV and CKKS schemes. In particular, we are currently in the process of developing and implementing this approach for the BGV scheme. Finally, our study focuses on circuits that rely on independently computed ciphertexts. We believe it is feasible to extend this study to cover the general case.

References

1. Al Badawi, A., Bates, J., Bergamaschi, F., Cousins, D.B., Erabelli, S., Genise, N., Halevi, S., Hunt, H., Kim, A., Lee, Y., et al.: OpenFHE: Open-Source Fully Homomorphic Encryption Library. In: Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography. pp. 53–63 (2022)
2. Albrecht, M.R., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., Lokam, S., Micciancio, D., Moody, D., Morrison, T., Sahai, A., Vaikuntanathan, V.: Homomorphic encryption security standard. Tech. rep., HomomorphicEncryption.org, Toronto, Canada (2018)
3. Bajard, J.C., Eynard, J., Hasan, M.A., Zucca, V.: A full RNS variant of FV like somewhat homomorphic encryption schemes. In: International Conference on Selected Areas in Cryptography. pp. 423–442 (2016)
4. Bergerat, L., Boudi, A., Bourgerie, Q., Chillotti, I., Ligier, D., Orfila, J.B., Tap, S.: Parameter Optimization and Larger Precision for (T)FHE. Journal of Cryptology **36**(3), 28 (2023)
5. Brakerski, Z.: Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In: Advances in Cryptology – CRYPTO 2012. pp. 868–886 (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. pp. 505–524 (2011)
8. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Advances in Cryptology – ASIACRYPT 2017. pp. 409–437 (2017)
9. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: Advances in Cryptology – ASIACRYPT 2016. pp. 3–33 (2016)
10. Costache, A., Smart, N.P.: Which ring based somewhat homomorphic encryption scheme is best? In: Topics in Cryptology – CT-RSA 2016. pp. 325–340 (2016)
11. Costache, A., Curtis, B.R., Hales, E., Murphy, S., Ogilvie, T., Player, R.: On the precision loss in approximate homomorphic encryption. ePrint Archive (2022)
12. Costache, A., Laine, K., Player, R.: Evaluating the effectiveness of heuristic worst-case noise analysis in FHE. In: Computer Security – ESORICS 2020. pp. 546–565 (2020)
13. Costache, A., Nürnberger, L., Player, R.: Optimisations and Tradeoffs for HElib. In: Topics in Cryptology – CT-RSA 2023. pp. 29–53 (2023)
14. Di Giusto, A., Marcolla, C.: Breaking the power-of-two barrier: noise estimation for BGV in NTT-friendly rings. ePrint Archive, Paper 2023/783 (2023)
15. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. ePrint Archive (2012)
16. Gentry, C.: A fully homomorphic encryption scheme. Stanford university (2009)
17. Gentry, C., Halevi, S., Smart, N.P.: Homomorphic Evaluation of the AES Circuit. In: Advances in Cryptology – CRYPTO 2012. pp. 850–867 (2012)
18. Halevi, S., Polyakov, Y., Shoup, V.: An improved RNS variant of the BFV homomorphic encryption scheme. In: Topics in Cryptology – CT-RSA 2019. pp. 83–105 (2019)

19. Iliashenko, I.: Optimisations of fully homomorphic encryption. PhD thesis (2019)
20. Kim, A., Polyakov, Y., Zucca, V.: Revisiting homomorphic encryption schemes for finite fields. In: Advances in Cryptology–ASIACRYPT 2021. pp. 608–639 (2021)
21. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Advances in Cryptology – EUROCRYPT 2010. pp. 1–23 (2010)
22. 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)
23. Mono, J., Marcolla, C., Land, G., Güneysu, T., Aaraj, N.: Finding and evaluating parameters for bgv. In: International Conference on Cryptology in Africa – AFRICACRYPT 2023. pp. 370–394. Springer (2023)
24. Murphy, S., Player, R.: A central limit approach for ring-lwe noise analysis. ePrint Archive (2019)
25. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM) **56**(6), 1–40 (2009)
26. Microsoft SEAL (release 3.4). https://github.com/Microsoft/SEAL (Oct 2019)

A Proof of Lemma 1 and Lemma 2

Fresh ciphertext By Equation (5), we can write $\nu_{\mathsf{clean}} = a_0 + a_1 s$ with $a_0 = \frac{t}{q}(\varepsilon + eu + e_0)$ and $a_1 = \frac{t}{q}e_1$. Since the coefficients of all the polynomials in ν_{clean} are sampled independently from symmetric distributions,

$$\mathbb{E}[a_0|_i] = \frac{t}{q}(\mathbb{E}[\varepsilon|_i] + \sum_{j=0}^{n-1} \xi(i,j)\mathbb{E}[e|_j]\mathbb{E}[u|_{i-j}] + \mathbb{E}[e_0|_i]) = 0$$

and $\mathbb{E}[a_1|_i] = \frac{t}{q}\mathbb{E}[e_1|_i] = 0$, by the properties of the expectation operator. Moreover, we have that $\mathsf{Cov}(a_1|_{i_1}, a_1|_{i_2})$, $\mathsf{Cov}(a_0|_{i_1}, a_1|_{i_2}) = 0$, since the covariance is bilinear and $\mathsf{Cov}(X,Y) = 0$ if X, Y are independent. We also get $\mathsf{Cov}(a_0|_{i_1}, a_0|_{i_2}) = 0$, noting that $\mathsf{Cov}((eu)|_{i_1}, (eu)|_{i_2}) = 0$, as $\mathsf{Cov}(XY, Z) = 0$ when X is independent of Y, Z and its mean is 0.

Addition Let ν, ν' be the errors of two independently-computed ciphertexts, then $\nu = \sum_\iota a_\iota s^\iota$, $\nu' = \sum_{\iota'} a'_{\iota'} s^{\iota'}$ with a_ι, $a'_{\iota'}$ independent for any ι, ι'. It follows that $\nu_{\mathsf{add}} = \sum_\iota (a_\iota + a'_\iota)s^\iota$, where $\mathbb{E}[(a_\iota + a'_\iota)|_i] = \mathbb{E}[a_\iota|_i] + \mathbb{E}[a'_\iota|_i] = 0$ and $\mathsf{Cov}((a_{\iota_1} + a'_{\iota_1})|_{i_1}, (a_{\iota_2} + a'_{\iota_2})|_{i_2}) = 0$ if $\iota_1 \neq \iota_2$ or $i_1 \neq i_2$, indeed by the bilinearity of the covariance it splits in

$$\mathsf{Cov}(a_{\iota_1}|_{i_1}, a_{\iota_2}|_{i_2}) + \mathsf{Cov}(a_{\iota_1}|_{i_1}, a'_{\iota_2}|_{i_2}) + \mathsf{Cov}(a'_{\iota_1}|_{i_1}, a_{\iota_2}|_{i_2}) + \mathsf{Cov}(a'_{\iota_1}|_{i_1}, a'_{\iota_2}|_{i_2}),$$

where the variables in each pairs are either uncorrelated by induction hypotesis or independent because they come from different ciphertexts.

Modulo switch & Key switch The proof is analogous to the addition case, as the quantity added is independent of the error ν.

Constant multiplication Let $\nu = \sum_\iota a_\iota s^\iota$ be any invariant noise and α a polynomial with coefficients sampled randomly from $\mathcal{U}_t$, then $\alpha\nu = \sum_\iota (\alpha a_\iota) s^\iota$, since α is constant in s. Moreover, $\mathbb{E}\big[(\alpha a_\iota)|_i\big] = 0$ by the properties of the expected value and since the coefficients of α and a_ι are independent and with mean 0. Finally, $\mathsf{Cov}\big((\alpha a_{\iota_1})|_{i_1}, (\alpha a_{\iota_2})|_{i_2}\big) = 0$, because for each $\mathsf{Cov}\big(\alpha|_{j_1} a_{\iota_1}|_{i_1-j_1}, \alpha|_{j_2} a_{\iota_2}|_{i_2-j_2}\big)$ we have the case $\mathsf{Cov}\big(XY, ZW\big)$ with X, Z independent of Y, W and either X and Z uncorrelated with mean 0, or Y and W uncorrelated with mean 0. Then,

$$\mathsf{Cov}\big(XY, ZW\big) = \mathbb{E}[X]\mathbb{E}[Z]\big(\mathbb{E}[YW] - \mathbb{E}[Y]\mathbb{E}[W]\big) = 0,$$

or analogously.

Multiplication Let $\nu = \sum_j a_j s^j$, $\nu' = \sum_k a'_k s^k$ be the errors of two independently-computed ciphertexts, then $\nu\nu' = \sum_\iota \sum_{j+k=\iota} a_j a'_k s^\iota$. Note that the ι-th element of $\nu\nu'$, as a polynomial in s, is $\sum_{j+k=\iota} a_j a'_k$ where a_j, a'_k are independent for any j, k. It follows that

$$\mathbb{E}\big[\big(\sum_{j+k=\iota} a_j a'_k\big)|_i\big] = \sum_{j+k=\iota} \sum_{l=0}^{n-1} \xi(i,l)\mathbb{E}[a_j|_l]\mathbb{E}[a'_k|_{i-l}] = 0.$$

Furthermore, by bilinearity of the covariance,

$$\mathsf{Cov}\left(\Big(\sum_{j_1+k_1=\iota_1} a_{j_1} a'_{k_1}\Big)|_{i_1}, \Big(\sum_{j_2+k_2=\iota_2} a_{j_2} a'_{k_2}\Big)|_{i_2}\right)$$

is a linear combination of elements $\mathsf{Cov}(a_{j_1}|_{l_1} a'_{k_1}|_{i_1-l_1}, a_{j_2}|_{l_2} a'_{k_2}|_{i_2-l_2})$. For $\iota_1 \neq \iota_2$ or $i_1 \neq i_2$, all these terms are null, hence the thesis, since we fall in the same case as in constant multiplication.
Analogously, this holds for $\nu \frac{t}{q'_\ell}(c'_0 + c'_1 s)$, $\nu' \frac{t}{q_\ell}(c_0 + c_1 s)$. Finally, we have that the covariance of different summands in ν_{mul} is 0, hence the conditions hold also for $\nu_{\mathsf{mul}} = -\nu\nu' + \nu \frac{t}{q'_\ell}(c'_0 + c'_1 s) + \nu' \frac{t}{q_\ell}(c_0 + c_1 s) + \frac{t}{q}(\varepsilon_0 + \varepsilon_1 s + \varepsilon_2 s^2)$. □

B Proof of Lemma 3

(1) The proof is done by induction on ι.
- Since the elements $s|_i$ are sampled from a distribution with zero mean and variance V_s the element $s|_i^2$ has expected value V_s, and from the LLN we have $\sum_{i=0}^{n-1} s|_i^2 \approx nV_s = nV_s g(1)$.
- By induction hypothesis, $\sum_{i=0}^{n-1} s^{\iota-1}|_i^2 \approx (nV_s)^{\iota-1} g(\iota-1)$, then from Equation (24), we have

$$\sum_{i=0}^{n-1} s^\iota|_i^2 \approx \sum_{i_1=0}^{n-1} s|_{i_1}^2 \sum_{i_2=0}^{n-1} s^{\iota-1}|_{i_2}^2 f(\iota) \approx (nV_s)^\iota g(\iota).$$

(2) Let us fix ι_1 and consider ι_2, ι_2' with $\iota_2 \leq \iota_2'$. Since f is an increasing function, we have $f(\iota_2 + i) \leq f(\iota_2' + i)$, then

$$\frac{g(\iota_1+\iota_2)}{g(\iota_2)} = f(\iota_2+1)\cdots f(\iota_1+\iota_2) \leq f(\iota_2'+1)\cdots f(\iota_1+\iota_2') = \frac{g(\iota_1+\iota_2')}{g(\iota_2')}.$$

It follows, in particular, $\frac{g(\iota_1+\iota_2)}{g(\iota_1)g(\iota_2)} \leq \frac{g(\iota_1+T_2)}{g(\iota_1)g(T_2)}$. We get the thesis analogously.

(3) Let us assume $T_1 \leq T_2$, wlog. We set $T = T_1 + T_2$ and $\tau = \lfloor T/2 \rfloor$ then

$$\frac{g(T)}{g(T_1+1)g(T_2+1)} \leq \frac{g(T)}{g(\lfloor T/2\rfloor+1)g(\lceil T/2\rceil+1)} = \frac{1}{f(\tau+1)}\prod_{\iota=2}^{\tau}\frac{f(T-\tau+\iota)}{f(\iota)}.$$

Let $c_\iota = f(\iota) = c - e^{b-a\iota}$ and $\varepsilon_\iota = (1-e^{a(\tau-T)})e^{b-a\iota}$, then

$$\frac{g(T)}{g(T_1+1)g(T_2+1)} \leq \frac{1}{c_{\tau+1}}\prod_{\iota=2}^{\tau}\frac{c_\iota+\varepsilon_\iota}{c_\iota}.$$

Since $(c_\iota+\varepsilon_\iota)/c_\iota \geq 1$,

$$\prod_{\iota=2}^{\tau}\frac{(c_\iota+\varepsilon_\iota)}{c_\iota} \leq \exp\left(\sum_{\iota=2}^{\tau}\frac{\varepsilon_\iota}{c_\iota}\right).$$

Now, noting that $\frac{\varepsilon_\iota}{c_\iota} \leq \frac{1}{e^a}\frac{\varepsilon_{\iota-1}}{c_{\iota-1}}$, we get $\frac{\varepsilon_i}{c_i} \leq \frac{1}{e^a}\frac{\varepsilon_{i-1}}{c_{i-1}}$, then

$$\sum_{\iota=2}^{\tau}\frac{\varepsilon_\iota}{c_\iota} \leq \frac{\varepsilon_2}{c_2}\sum_{\iota=2}^{\tau}\left(\frac{1}{e^a}\right)^\iota \leq \frac{e^b}{(e^{2a}c-e^b)(e^{2a}-e^a)}.$$

It follows that $\frac{g(T_1+T_2)}{g(T_1+1)g(T_2+1)}$ is finite.
Since the bound we just computed is not tight, we estimate a better one evaluating $\frac{f(\lceil T/2\rceil+2)\cdots f(T)}{f(2)\cdots f(\lfloor T/2\rfloor+1)}$ for up to $T = 2^{20}$. The obtained values are the following:

n	2^{12}	2^{13}	2^{14}	2^{15}
K_n	22	39	72	136

□