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. Therefore, choosing optimal parameters while balancing security, correctness, and computational efficiency, is challenging.

This work presents two main contributions to the parameter selection improvement in the BFV scheme. Primarily, we perform the first average case analysis to estimate the error growth. Our method significantly improves on previous works regarding the accuracy of bounds. 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 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 [23], 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 [17]. In his Ph.D. 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 [7], BFV [6, 16], TFHE [10], and CKKS [9] 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, [26], and its ring variant (RLWE), [22]. 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 a correct decryption, the error, also called noise, added has to be small. However, it increases as long as operations are carried on. In particular, it grows exponentially when homomorphic multiplications are computed. To increase the number of supported operations, we could increase the ciphertext modulus q. However, a larger modulus also decreases the security level of the underlying scheme, requiring a larger polynomial degree n at the cost of efficiency. This required trade-off between security (small ciphertext modulus) and error margin (large ciphertext modulus) illustrates the difficulty of finding an optimal set of parameters for a specific FHE scheme.

Related works. Several efforts have been made by the FHE community in facing this challenge. For instance, Bergerat *et al.* [5] proposed a framework for efficiently selecting parameters in TFHE-like schemes. Mono *et al.* [24] developed an interactive parameter generator for the leveled BGV scheme that supports arbitrary circuit models. Concerning security, the Homomorphic Encryption Standard [1] provides upper limits on the size of the ciphertext modulus for certain security levels λ and polynomial degrees n in the form of lookup tables, using the Lattice Estimator[4] [2].

Regarding the BFV selection of parameters, the state-of-the-art approach for establishing theoretical bounds on the error growth is based on using either the infinity norm [21] or the canonical norm [12, 14, 20]. The canonical norm is known to result in better parameters. However, both these methods often yield overly conservative bounds. An alternative approach is an average-case analysis, which provides a predicted error closer to the actual errors observed in experimental results. The first application of the average-case approach to FHE bounds was in the TFHE scheme [10]. Recent works have introduced similar techniques for CKKS [11] and BGV [13, 25].

Our contribution. This paper improves the current state of BFV parameters selection by providing 1) the first estimation of the noise in average-case and 2) 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 the successor of the LWE Estimator, which is a software tool to determine the security level of LWE instances since it shows the timeline and fundamentals of the main lattice attacks proposed until the present time.

for the BGV and CKKS schemes [11, 25], 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.

Finally, we develop an interactive parameters generator, which makes use of our theoretical results and the security formula proposed in [24]. 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 investigate the error behavior in four different circuits (proposed in [24]). We also consider two distinct circumstances in which the modulus switching is applied. As expected, the modulus switching technique increases the bounds. However, in Fact 1, we propose a set of parameters that achieve a similar ciphertext modulus while improving efficiency.
- Finally, in Section 6, we compare our average-case approach with prior bounds of BFV noise growth. Additionally, we introduce our parameter generator to facilitate the selection of optimal parameters for the BFV scheme.

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 $\mathcal{K} = \mathbb{Q}[x]/\langle f(x) \rangle$. Note that, for $a, b \in \mathcal{K}$ and defined $\xi(i,j)$ as 1 if $i - j \in [0, n)$ and -1 otherwise,

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

where $i - j$ is computed mod n.
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 $x \in \mathbb{Q}$, $\lfloor x \rceil$ is 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 $q = \prod_{i=1}^{k} r_i$, 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 $q_\ell = \prod_{j=1}^{\ell} p_j$, the initial ciphertext is $q_{\mathsf{ms}} = q_L$.

2.2 Secret and Error Distributions

Let χ_s and χ_u be secret key distributions and χ_e an error distribution from the Learning with Errors over Rings (RLWE) problem. Tipically, we have $\chi_s = \chi_u = \mathcal{U}_3$, the uniform distribution on $\mathbb{Z}_3$, and $\chi_e = \mathcal{DG}(0, \sigma^2)$, discrete Gaussian centered in 0 with standard deviation $\sigma = 3.19$ [1]. Note that in this article, we assume that the distributions are symmetric. In general, if χ is a probabilistic distribution and $a \in \mathcal{R}$ is a random polynomial, we write $a \leftarrow \chi$ when sampling each coefficient independently from χ.

Coverage probability for Gaussian-distributed variables. Let X be a random variable from a Gaussian distribution centered in 0 of variance V, then

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

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:

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

Canonical embedding and norm. We recall the results of [12, 14, 20]. 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 [14].

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 [15], using the Lyapunov Central Limit Theorem, it is possible to prove that for any root of unity $\zeta = \cos(\alpha) + i\sin(\alpha)$, the random variable $a(\zeta)$ is a complex random variable which can be approximated by a complex Gaussian random variable. 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 $(X, Y)^t = C(Z, Z')^t$ with Z and Z' i.i.d. standard Gaussian random variables. Therefore,

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

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 random variable, $||(Z, Z')||_2^2$ is Chi-squared distributed and

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

Therefore,

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

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 $\mathsf{Var}(XY) = \mathsf{Var}(X)\mathsf{Var}(Y)$.

We list the variances of the variable we will use in the rest of the work:

$$\begin{array}{lll} \text{If } X \leftarrow \mathcal{DG}(0,\sigma^2) & \text{then} & \mathsf{Var}(X) = \sigma^2 \\ \text{If } X \leftarrow \mathcal{U}_q & \text{then} & \mathsf{Var}(X) = (q^2-1)/12 \approx q^2/12 \\ \text{If } X \leftarrow \mathcal{U}_t & \text{then} & \mathsf{Var}(X) = (t^2-1)/12 \\ \text{If } X \leftarrow \mathcal{U}_3 & \text{then} & \mathsf{Var}(X) = (3^2-1)/12 = 2/3 \\ \text{If } X \leftarrow \mathcal{U}_{(-0.5,0.5)} & \text{then} & \mathsf{Var}(X) = 1/12 \end{array} \tag{5}$$

3 The BFV Scheme

The following describes the BFV scheme [16], 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 [21]. 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 $\mathbf{c} = (c_0, c_1) = \left(\left[\left\lfloor \frac{q}{t} m \right\rceil + ub + e_0\right]_q, [ua+e_1]_q\right)$.

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

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 [20] 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

$$\nu_{\mathsf{clean}} = \frac{t}{q}(\varepsilon + eu + e_0 + e_1 s) \tag{6}$$

where $\varepsilon = \left\lfloor \frac{q}{t} m \right\rceil - \frac{q}{t} m = -\frac{[qm]_t}{t}$, [21].

Proof.

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

Addition & Constant Multiplication.

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

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 $\mathbf{c}_{\mathsf{add}} = ([c_0 + c_0']_{q_\ell}, [c_1 + c_1']_{q_\ell})$.

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

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 $u, k \in \mathcal{R}$, we have

$$\begin{aligned} \frac{t}{q_\ell}[c_0 + c_1 s + c_0' + c_1' s]_{q_\ell} &= \frac{t}{q_\ell}([c_0 + c_1 s]_{q_\ell} + [c_0' + c_1' s]_{q_\ell} - uq_\ell) \\ &= [m + m']_t + \nu + \nu' + kt \implies \nu_{\mathsf{add}} = \nu + \nu \end{aligned} \tag{7}$$

$$\begin{aligned} \frac{t}{q_\ell}[\alpha c_0 + \alpha c_1 s s]_{q_\ell} &= \frac{t}{q_\ell}\left(\alpha[c_0 + c_1 s]_{q_\ell} - uq_\ell\right) = [\alpha m]_t + \alpha\nu + kt \\ &\implies \nu_{\mathsf{const}} = \alpha\nu, \end{aligned} \tag{8}$$

Multiplication & Modulus switching. In this section, we are going to see the multiplication algorithm presented in [21], 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 [8] 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.

$\mathrm{ModSwitch}(\mathfrak{c}, q_\ell')$

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

$$\nu_{\mathsf{ms}}(q'_\ell) = \frac{t}{q'_\ell}(\varepsilon_0 + \varepsilon_1 s), \text{ with } \varepsilon_i = -\frac{[q'_\ell c_i]_{q_\ell}}{q_\ell}. \tag{9}$$

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,

$$\begin{aligned}&\frac{t}{q_\ell}\left[\left\lfloor \frac{t}{q'_\ell} c_0 c'_0 \right\rceil + \left\lfloor \frac{t}{q'_\ell}(c_0 c'_1 + c'_0 c_1) \right\rceil s + \left\lfloor \frac{t}{q'_\ell} c_1 c'_1 \right\rceil s^2\right]_{q_\ell} \\ =&\frac{t}{q_\ell}\left[\frac{t}{q'_\ell} c_0 c'_0 + \varepsilon_0 + \frac{t}{q'_\ell}(c_0 c'_1 + c'_0 c_1)s + \varepsilon_1 s + \frac{t}{q'_\ell} c_1 c'_1 s^2 + \varepsilon_2 s^2\right]_{q_\ell} \\ =&\frac{t}{q_\ell}(c_0 + c_1 s) \cdot \frac{t}{q'_\ell}(c'_0 + c'_1 s) + \frac{t}{q_\ell}(\varepsilon_0 + \varepsilon_1 s + \varepsilon_2 s^2) + h''t \\ =&[mm']_t + \nu(m' + h't) + \nu'(m + ht) + \nu\nu' + \frac{t}{q_\ell}(\varepsilon_0 + \varepsilon_1 s + \varepsilon_2 s^2) + kt \\ =&[mm']_t + \nu_{\mathsf{mul}}(q_\ell) + kt,\end{aligned}$$

where the noise after the multiplication is

$$\nu_{\mathsf{mul}}(q_\ell) = -\nu\nu' + \nu\frac{t}{q'_\ell}(c'_0 + c'_1 s) + \nu'\frac{t}{q_\ell}(c_0 + c_1 s) + \frac{t}{q_\ell}(\varepsilon_0 + \varepsilon_1 s + \varepsilon_2 s^2). \tag{10}$$

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 [4, 19, 21]. 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 [21]).

Brakerski-Vaikuntanathan The strategy is exploiting the Chinese Remainder Theorem (CRT) to decompose d_2 in the 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 $i = 1, \dots, k$. 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

$$\begin{aligned} \frac{t}{q_\ell}[c_0 + c_1 s]_{q_\ell} &= \frac{t}{q_\ell}\left[d_0 + d_1 s + d_2 s^2 + \sum_{i=1}^{k_\ell} [d_2]_{r_i} e_i\right]_{q_\ell} \\ &= m + \nu + \frac{t}{q_\ell} \sum_{i=1}^{k_\ell} [d_2]_{r_i} e_i + kt. \end{aligned}$$

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

$$\nu_{\mathsf{ks}}^{\mathsf{BV}}(q_\ell) = \frac{t}{q_\ell} \sum_{i=1}^{k_\ell} [d_2]_{r_i} e_i. \tag{11}$$

Gentry-Halevi-Smart An alternative is encrypting Ps^2 instead of s^2 with P a large number, usually of approximately the same size of 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

$$\begin{aligned} \frac{t}{q_\ell}[c_0 + c_1 s]_{q_\ell} &= \frac{t}{q_\ell}\left[d_0 + d_1 s + \left\lfloor \frac{d_2 b'}{P} \right\rceil + \left\lfloor \frac{d_2 a'}{P} \right\rceil s\right]_{q_\ell} \\ &= \frac{t}{q_\ell}\left[d_0 + d_1 s + \frac{d_2(Ps^2 + e')}{P} + \varepsilon_0 + \varepsilon_1 s\right]_{q_\ell} \\ &= m + \nu + \frac{t}{q_\ell}\left(\frac{d_2 e'}{P} + \varepsilon_0 + \varepsilon_1 s\right) + kt. \end{aligned}$$

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

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

GHS-RNS In practice, d_2 in base $q_\ell P$ is usually approximated by the result of the FastBaseExtension as

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

Therefore, the added error becomes

$$\nu_{\mathsf{ks}}^{\mathsf{GHS-RNS}}(q_\ell) = \frac{t}{q_\ell}\left(\frac{(d_2 + uq_\ell)e'}{P} + \varepsilon_0 + \varepsilon_1 s\right). \tag{13}$$

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 [18, 21].

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

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

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

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

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

$$\begin{aligned}
\frac{t}{q_\ell} [c_0 + c_1 s]_{q_\ell} &= \frac{t}{q_\ell} \left[d_0 + d_1 s + \frac{\sum_{i=1}^{\omega} [d_2]_{\tilde{r_i}} (b_i + a_i s)}{P} + \varepsilon_0 + \varepsilon_1 s \right]_{q_\ell} \\
&= \frac{t}{q_\ell} \left[d_0 + d_1 s + d_2 s^2 + \frac{\sum_{i=1}^{\omega} [d_2]_{\tilde{r_i}} e_i}{P} + \varepsilon_0 + \varepsilon_1 s \right]_{q_\ell} \\
&= m + \nu + \frac{t}{q_\ell} \left(\frac{\sum_{i=1}^{\omega} [d_2]_{\tilde{r_i}} e_i}{P} + \varepsilon_0 + \varepsilon_1 s \right) + kt.
\end{aligned}$$

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

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

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

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

Therefore, the error added becomes

$$\nu_{\mathsf{ks}}^{\mathsf{Hyb-RNS}}(q_\ell) = \frac{t}{q_\ell}\left(\frac{\sum_{i=1}^{\omega}([d_2]_{\tilde{r_i}} + u_i\tilde{r_i})e_i}{P} + \varepsilon_0 + \varepsilon_1 s\right). \tag{15}$$

4 Average-Case Noise Analysis for BFV

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

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

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

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

4.1 Distribution

We have studied the distribution of the coefficients of the error vector computationally with Python **fitter** package[5], obtaining that they can be well-approximated by i.d. Gaussians. In Figure 3, we show the outcome for the first coefficient with $M = 0, 1, 2$.

4.2 Characterization of the Error

In the next paragraphs, we prove that the error coefficients have always mean 0, and we show how to compute the variance as the different operations are performed. To do so, we give a general characterization of the error as

$$\nu = \sum_{\iota} a_\iota s^\iota, \tag{16}$$

where the following conditions hold:

1. $\mathbb{E}[a_\iota|_i] = 0$ for any ι and i,

2. $\mathsf{Cov}(a_{\iota_1}|_{i_1}, a_{\iota_2}|_{i_2}) = 0$ if either $\iota_1 \neq \iota_2$ or $i_1 \neq i_2$.

See Appendix A for the proof.

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

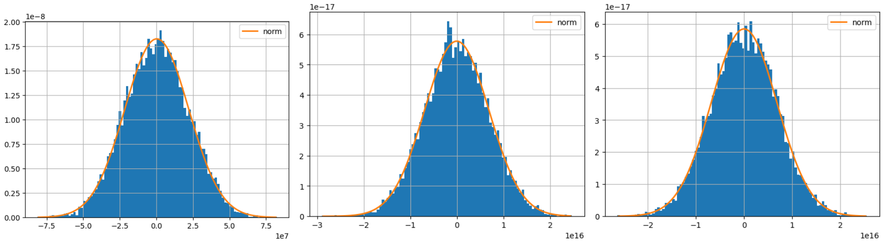

Fig. 1: Our analysis considers 10000 samples computed with the OpenFHE library [3] with Hybrid key switching, `HPSPOVERQ` multiplication, $\chi_s = \chi_u = \mathcal{U}_3$, and $\chi_e = \mathcal{DG}(0, 3.19^2)$. We set $t = 65537$, $n = 2^{13}$ and q as computed by the library. The results are: (i) $\mathrm{ks_{pval}}$ 0.588918; (ii) $\mathrm{ks_{pval}}$ 0.596218; and (iii) $\mathrm{ks_{pval}}$ 0.744975. Note that we can assume that the distribution is Gaussian with confidence level 95% as long as the p-value is ≥ 0.05.

Expected value. As a consequence of Condition 1, we obtain that the error coefficients have mean 0 all along the circuit, i.e.

$$\mathbb{E}[\nu|_i] = 0. \tag{17}$$

Proof. Since the error ν can be written as in (16), we have 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 (see Section 2.2) and Condition 1,

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

Note that the secret key s is seen as a fixed vector.

Variance. From Condition 2, we have

$$\mathsf{Var}(\nu|_i) = \sum_{\iota} \sum_{j=0}^{n-1} \mathsf{Var}(a_\iota|_j) s^\iota|_{i-j}^2. \tag{18}$$

Proof. Analogously to the previous proof and by the properties of the variance in section 2.2, we have

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

Thus, by Condition 2 in Equation (16), $\mathsf{Var}(\nu|_i) = \sum_\iota \sum_{j=0}^{n-1} \mathsf{Var}(a_\iota|_j) s^\iota|_{i-j}^2$.

4.3 Variance Analysis for BFV Operations

In the following, we study the variance behaviour for all the BFV operation.

Encryption. The variance of the error coefficients of a fresh ciphertext is

$$V_{\mathsf{clean}} \approx \frac{B_{\mathsf{clean}}}{q^2} \quad \text{with} \quad B_{\mathsf{clean}} = t^2 \left(\tfrac{1}{12} + nV_eV_u + V_e + nV_eV_s\right). \tag{19}$$

Proof. By Equation (6), 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, $\mathsf{Var}(a_0|_i) = \frac{t^2}{q^2}\left(\frac{1}{12} + nV_eV_u + V_e\right)$ and $\mathsf{Var}(a_1|_i) = \frac{t^2}{q^2}V_e$. It follows by Equation (18)

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

where s is seen as a fixed vector. Now, 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 Number (LLN) we can approximate $\sum_i s|_i^2 \approx nV_s$. Therefore,

$$\mathsf{Var}(\nu_{\mathsf{clean}}|_i) \approx \frac{t^2}{q^2}\left(\tfrac{1}{12} + nV_eV_u + V_e + nV_eV_s\right),$$

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 $\mathcal{U}_t$.

Addition. Let ν, ν' be the errors of two ciphertexts computed independently, so independent themselves. Then, from Equation (7),

$$\mathsf{Var}((\nu + \nu')|_i) = \mathsf{Var}(\nu|_i) + \mathsf{Var}(\nu'|_i). \tag{20}$$

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

Modulo switching. By Equation (9), the error added by the modulo switch from q_ℓ to q'_ℓ is independent from the starting error, so the total variance becomes

$$\mathsf{Var}(\nu|_i) + V_{\mathsf{ms}}(q'_\ell) = \mathsf{Var}(\nu|_i) + \frac{B_{\mathsf{ms}}}{q'^2_\ell} \quad \text{with} \quad B_{\mathsf{ms}} = \frac{t^2}{12}(1 + nV_s). \tag{21}$$

Key switching. Analogously, after the key switch, the variance becomes

$$\mathsf{Var}(\nu|_i) + V_{\mathsf{ks}}(q_\ell), \tag{22}$$

where $V_{\mathsf{ks}}(q_\ell)$ depends on the chosen key-switching variants. Specifically,

- *BV key switching.* By Equation (11), since $r_i \approx \sqrt[k]{q}$,

$$V_{\mathsf{ks}}^{\mathsf{BV}}(q_\ell) \approx \frac{t^2}{q_\ell^2} \sum_{i=1}^{k_\ell} n \frac{r_i^2}{12} V_e \approx \frac{k_\ell t^2 \sqrt[k]{q^2} n V_e}{12 q_\ell^2}. \tag{23}$$

- *GHS key switching.* From Equation (12) and $P \approx q$,

$$V_{\mathsf{ks}}^{\mathsf{GHS}}(q_\ell) \approx \frac{t^2}{q_\ell^2} \left(\frac{n q_\ell^2 V_e}{12 q^2} + \frac{1}{12} + \frac{n V_s}{12} \right) \leq \frac{t^2}{12 q_\ell^2} \left(n V_e + 1 + n V_s \right) \tag{24}$$

- *GHS RNS.* Analogously, from Equation (13),

$$\begin{aligned} V_{\mathsf{ks}}^{\mathsf{GHS}}(q_\ell) &\approx \frac{t^2}{q_\ell^2} \left(\frac{n(k_\ell + 2) q_\ell^2 V_e}{12 q^2} + \frac{1}{12} + \frac{n V_s}{12} \right) \\ &\leq \frac{t^2}{12 q_\ell^2} \left(n(k+2) V_e + 1 + n V_s \right) \end{aligned} \tag{25}$$

- *Hybrid key switching.* By Equation (14), $\tilde{r_i} \approx \sqrt[\omega]{q_\ell}$ and $P \approx \sqrt[\omega]{q}$, we have

$$V_{\mathsf{ks}}^{\mathsf{Hyb}}(q_\ell) \approx \frac{t^2}{q_\ell^2} \left(\frac{\omega n \sqrt[\omega]{q_\ell^2} V_e}{12 \sqrt[\omega]{q^2}} + \frac{1}{12} + \frac{n V_s}{12} \right) \leq \frac{t^2}{12 q_\ell^2} \left(\omega n V_e + 1 + n V_s \right) \tag{26}$$

- *Hybrid RNS.* In the same way, by Equation (15), we get

$$\begin{aligned} V_{\mathsf{ks}}^{\mathsf{Hyb}}(q_\ell) &\approx \frac{t^2}{q_\ell^2} \left(\frac{\omega n \sqrt[\omega]{q_\ell^2} (\frac{k_\ell}{\omega} + 2) V_e}{12 \sqrt[\omega]{q^2}} + \frac{1}{12} + \frac{n V_s}{12} \right) \\ &\leq \frac{t^2}{12 q_\ell^2} \left((k + 2\omega) n V_e + 1 + n V_s \right) \end{aligned} \tag{27}$$

Observe that, for all the relinearization variants except the BV one, we can write the variance as

$$\mathsf{Var}(\nu|_i) + \frac{B_{\mathsf{ks}}}{q_\ell^2},$$

for some constant B_{ks}. In Section 5, we will consider only this case.

Constant multiplication. The variance of the error coefficients after a constant multiplication is

$$B_{\mathsf{const}} \mathsf{Var}(\nu|_i) \quad \text{with} \quad B_{\mathsf{const}} = \frac{(t^2 - 1) n}{12}. \tag{28}$$

Proof. Since the coefficients of α behave as sampled independently at random from a uniform distribution over $\mathcal{U}_t$, then $\mathbb{E}[\alpha|_i] = 0$ and $\mathsf{Var}(\alpha|_i) \approx \frac{t^2-1}{12}$. It follows by Equation (8) and the independence of α and ν

$$\begin{aligned} \mathsf{Var}(\nu_{\mathsf{const}}|_i) = \mathsf{Var}((\alpha\nu)|_i) &= \sum_j \mathsf{Var}(\alpha|_j \nu|_{i-j}) \\ &= \sum_j \mathsf{Var}(\alpha|_j) \mathsf{Var}(\nu|_{i-j}) = \frac{(t^2-1)n}{12} \mathsf{Var}(\nu|_{i-j}). \end{aligned}$$

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 ciphertexts. Then by Equation (10),

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

Thanks to Equation (18), the variance of its coefficients is

$$\begin{aligned}\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i) = &\, n\sum_{\iota_1}\sum_{\iota_2} \mathsf{Var}(a_{\iota_1}|_i)\mathsf{Var}(a'_{\iota_2}|_i)\sum_{j=0}^{n-1} s^{\iota_1+\iota_2}|_{i-j}^2 + \\ &+ n\sum_{\iota_1}\mathsf{Var}(a_{\iota_1}|_i)\frac{t^2}{12}\sum_{j=0}^{n-1}\big(s^{\iota_1}|_{i-j}^2 + s^{\iota_1+1}|_{i-j}^2\big) + \\ &+ n\sum_{\iota_2}\mathsf{Var}(a'_{\iota_2}|_i)\frac{t^2}{12}\sum_{j=0}^{n-1}\big(s^{\iota_2}|_{i-j}^2 + s^{\iota_2+1}|_{i-j}^2\big) + \\ &+ \frac{t^2}{12q_\ell^2}\Big(1 + \sum_{j=0}^{n-1} s|_{i-j}^2 + \sum_{j=0}^{n-1} s^2|_{i-j}^2\Big).\end{aligned} \tag{29}$$

Note that keeping track of all the terms in Equation (29) can be rather complex, especially if many operations are performed. Thus, the following part aims at deriving a simplified formula to effectively approximate $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$, working directly on ν and ν'. We divide it in four steps: we start considering the easier scenario where the error coefficients are independent and compute their variance V_i. We find that $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$ differs from the target one, i.e. V_i, in the powers of s. We study a particular case of the ratio between the wanted and the obtained terms and find a function that approximates it; from this function, in Lemma 1, we give a general way to compute $\sum_{i=0}^{n-1} s^\iota|_i^2$. Lastly, in Lemma 2, we compute the terms in Equation (29) and in Theorem 1, we provide a formula for $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$. Moreover, with Proposition 1, we give a further simplification of $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$, independent of the modulus q_ℓ, which will be useful in Section 5.

First, we consider the scenario where the coefficients of a noise polynomial are independent of each other. In this case, from Equation (10), we obtain

$$\begin{aligned}&n\mathsf{Var}(\nu|_i)\mathsf{Var}(\nu'|_i) + n\mathsf{Var}(\nu|_i)\frac{t^2}{12}(1 + nV_s) + \\ &\qquad + n\mathsf{Var}(\nu'|_i)\frac{t^2}{12}(1 + nV_s) + \mathsf{Var}\Big(\frac{t}{q_\ell}(\varepsilon_0 + \varepsilon_1 s + \varepsilon_2 s^2)|_i\Big).\end{aligned}$$

However, the resulting variance

$$\begin{aligned}&n\sum_{\iota_1}\sum_{\iota_2}\mathsf{Var}(a_{\iota_1}|_i)\mathsf{Var}(a'_{\iota_2}|_i)\sum_{j_1=0}^{n-1}s^{\iota_1}|_{i-j_1}^2\sum_{j_2=0}^{n-1}s^{\iota_2}|_{i-j_2}^2+\\&+n\sum_{\iota_1}\mathsf{Var}(a_{\iota_1}|_i)\frac{t^2}{12}\sum_{j=0}^{n-1}s^{\iota_1}|_{i-j}^2\Big(1+\sum_{j_1=0}^{n-1}s|_{i-j_1}^2\Big)+\\&+n\sum_{\iota_2}\mathsf{Var}(a'_{\iota_2}|_i)\frac{t^2}{12}\sum_{j=0}^{n-1}s^{\iota_2}|_{i-j}^2\Big(1+\sum_{j_1=0}^{n-1}s|_{i-j_1}^2\Big)+\\&+\mathsf{Var}\Big(\frac{t}{q_\ell}(\varepsilon_0+\varepsilon_1 s+\varepsilon_2 s^2)|_i\Big),\end{aligned}$$

differs from $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$ in the powers of s. For instance, in the first line instead of $\sum_{j=0}^{n-1} s^{\iota_1+\iota_2}|_{i-j}^2$ we have $\sum_{j_1=0}^{n-1} s^{\iota_1}|_{i-j_1}^2 \sum_{j_2=0}^{n-1} s^{\iota_2}|_{i-j_2}^2$. As a consequence, our goal is to approximate the ratio between two expressions of the kind

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

We focus on the following particular case

$$\frac{\sum_{i=0}^{n-1} s^{\iota}|_i^2}{\sum_{i_1=0}^{n-1} s|_{i_1}^2 \sum_{i_2=0}^{n-1} s^{\iota-1}|_{i_2}^2}, \tag{30}$$

for $\iota \geq 2$, analyzing its average value computationally. As shown in Figure 2, this quantity is well-approximated by the function $f(\iota) = -\frac{1}{e^{a\iota-b}} + c$, i.e.

$$f(\iota) = -\frac{1}{e^{a\iota-b}} + c \approx \frac{\sum_{i=0}^{n-1} s^{\iota}|_i^2}{\sum_{i_1=0}^{n-1} s|_{i_1}^2 \sum_{i_2=0}^{n-1} s^{\iota-1}|_{i_2}^2}. \tag{31}$$

Moreover, a, b, c depend only on the ring dimension n. Their values, computed with Python function *curve_fit*[6] are listed in Table 1.

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

Table 1: Value for a, b, c setting $\chi_s = \mathcal{U}_3$.

With the following lemma, we give an approximation of $\sum_{i=0}^{n-1} s^{\iota}|_i^2$, exploiting the *correction function* f defined above.

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

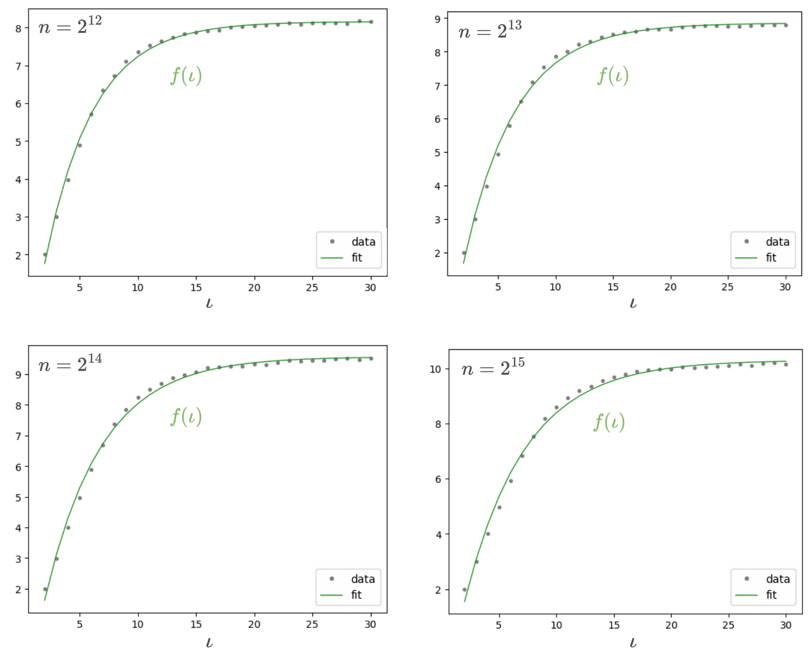

Fig. 2: The function $f(\iota)$ fitting the points from Equation (31).

Lemma 1. *Let f be defined as in Equation* (31)*, $f(0) = f(1) = 1$ and $g(\iota) = \prod_{i=0}^{\iota} f(i)$. Then, for $\iota \geq 1$, $\sum_{i=0}^{n-1} s^{\iota}|_i^2 \approx (nV_s)^{\iota} g(\iota)$.*

Proof. The proof is done by induction on ι.

- As explained in the proof of Equation (19), $\sum_{i=0}^{n-1} s|_i^2 \approx nV_s = nV_s g(1)$.
- By induction hypothesis, $\sum_{i=0}^{n-1} s^{\iota-1}|_i^2 \approx (nV_s)^{\iota-1} g(\iota-1)$, then from Equation (31), we have

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

It follows that we can approximate $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$ in Equation (29) with

$$\begin{aligned} & n \sum_{\iota_1} \sum_{\iota_2} \mathsf{Var}(a_{\iota_1}|_i) \mathsf{Var}(a'_{\iota_2}|_i) (nV_s)^{\iota_1+\iota_2} g(\iota_1+\iota_2) + \\ & + n \sum_{\iota_1} \mathsf{Var}(a_{\iota_1}|_i) \frac{t^2}{12} (nV_s)^{\iota_1} g(\iota_1) \, \Big(1 + nV_s f(\iota_1+1)\Big) + \\ & + n \sum_{\iota_2} \mathsf{Var}(a'_{\iota_2}|_i) \frac{t^2}{12} (nV_s)^{\iota_2} g(\iota_2) \, \Big(1 + nV_s f(\iota_2+1)\Big) + \\ & + \frac{t^2}{12 q_\ell^2} \Big(1 + nV_s + (nV_s)^2 f(2)\Big). \end{aligned}$$

Lemma 2. *Let f be defined as in Equation* (31)*, $f(0) = f(1) = 1$ and $g(\iota) = \prod_{i=0}^{\iota} f(i)$. Let $\iota_j = 0, \dots, T_j$, for $j = 1, 2$, then*

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

Proof. 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,

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

Analogously, we get

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

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 and let ν_{mul} be as in Equation* (10)*. Let $g(\iota) = \prod_{i=0}^{\iota} f(i)$, where f is defined as in Equation* (31)*, $f(0) = f(1) = 1$. Then*

$$\begin{aligned}\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i) \leq\, & n\mathsf{Var}(\nu|_i)\mathsf{Var}(\nu'|_i)\frac{g(T_1+T_2)}{g(T_1)g(T_2)} + \\ & + n\mathsf{Var}(\nu|_i)\frac{t^2}{12}\big(1 + nV_s f(T_1+1)\big) + n\mathsf{Var}(\nu'|_i)\frac{t^2}{12}\big(1 + nV_s f(T_2+1)\big) \\ & + \frac{t^2}{12q_\ell^2}\big(1 + nV_s + (nV_s)^2 f(2)\big).\end{aligned} \tag{32}$$

Proof. Applying Lemma 1 to Equation (29), it follows

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

Then, by Lemma 2, we get the following bound

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

i.e.

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

Finally, we show that the first and last terms of (32) are negligible compared to the others. In the proof, we use the following lemma (see Appendix A.1 for the proof).

Lemma 3. *Let T_1, T_2 be positive integers. Let f be defined as in Equation* (31)*, $f(0)=f(1)=1$ and $g(\iota)=\prod_{i=0}^{\iota} f(i)$. Then*

$$\frac{g(T_1+T_2)}{g(T_1+1)g(T_2+1)} \leq K_n < +\infty.$$

In particular,

n	2^{12}	2^{13}	2^{14}	2^{15}
K_n	22	38	70	133

Proposition 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 and let ν_{mul} as in Equation* (10)*. Let f be defined as in Equation* (31)*, $f(0)=f(1)=1$ and $g(\iota)=\prod_{i=0}^{\iota} f(i)$. Then the variance $\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i)$ is well-approximated as*

$$\mathsf{Var}(\nu_{\mathsf{mul}}|_i) \approx \frac{t^2n^2V_s}{12}\big(\mathsf{Var}(\nu|_i)f(T_1+1)+\mathsf{Var}(\nu'|_i)f(T_2+1)\big). \qquad (33)$$

Proof. We start from the first term and exploit a bound on the second term, given by correctness requirements, to prove that it is negligible compared to the third one. Indeed, to guarantee correct decryption, we impose

$$\mathsf{Var}(\nu_{\mathsf{mul}}(q_\ell)|_i) \leq \frac{1}{8D^2}.$$

Since all the addends in (32) are positive quantities, this also implies

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

i.e.

$$\mathsf{Var}(\nu|_i) \le \frac{3}{2D^2t^2n^2V_sf(T_1{+}1)}.$$

Then, by Lemma 3,

$$\begin{aligned} n\mathsf{Var}(\nu|_i)\mathsf{Var}(\nu'|_i)\frac{g(T_1+T_2)}{g(T_1)g(T_2)} &\le n\frac{3}{2D^2t^2n^2V_sf(T_1+1)}\mathsf{Var}(\nu'|_i)\frac{g(T_1+T_2)}{g(T_1)g(T_2)} \\ &\le \frac{18K_n}{D^2t^4n^3V_s^2}\frac{t^2n^2V_s}{12}f(T_2+1)\mathsf{Var}(\nu'|_i) \\ &\ll \frac{t^2n^2V_s}{12}f(T_2+1)\mathsf{Var}(\nu'|_i). \end{aligned}$$

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 $\mathsf{Var}(\nu|_i) \ge B_{\mathsf{clean}}/q^2$, since all the homomorphic operations performed increase the variance of the error coefficients. Therefore, by Equation (19), we get

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

The argument for the second event is analogous from $\mathsf{Var}(\nu|_i) \ge B_{\mathsf{ms}}/q_\ell^2$ and Equation (21). Hence, we can set

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

It is worth noting that if we consider two independently-computed ciphertexts $\mathbf{c}$ and $\mathbf{c}'$, output of a circuit with multiplicative depth $\ell-1$, their noises are as in Proposition 1 where $T_1 = T_2 = \ell$.

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 6.1). In our analysis, we extend the previous work on BGV [24] considering the circuit models newly proposed by Mono *et al.*.

Each circuit performs a list of operations on η ciphertexts c_i in parallel, as illustrated in Figure 3. The resulting ciphertexts are homomorphically multiplied with other ones computed analogously. This sequence is repeated M times.

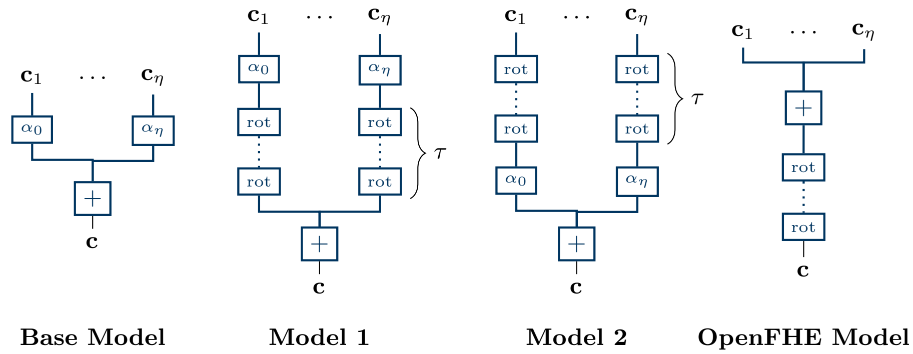

Fig. 3: Sequences of operations in the different models.

Base model This is a simplified version of the other models, performing constant multiplications on the ciphertexts and summing them afterward, 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.

OpenFHE Model For comparison with previous work, we also define the model used in the OpenFHE library [3, 21]. 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 2. In Section 5.2, we will study two other types of circuit, common in practice, making use 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. That is, we require

$$V_{L-1} \leq \frac{1}{8D^2}.$$

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 (22)

  $$V_{\ell-1} + \tau V_{\mathsf{ks}}.$$

  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 $B_{\mathsf{const}} = \frac{(t^2-1)n}{12}$ (28), 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 (20)

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

- 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 (21), 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 (22) and (33),

  $$\begin{aligned} V_\ell &\approx \frac{t^2 n^2 V_s}{12}\Big(2\eta(V_{\ell-1} + \tau V_{\mathsf{ks}})B_{\mathsf{const}} + V_{\mathsf{ms}}\Big) f(\ell+1) + V_{\mathsf{ks}} \\ &\approx \frac{t^2 n^2 V_s}{12}\Big(2\eta(V_{\ell-1} + \tau V_{\mathsf{ks}})B_{\mathsf{const}} + V_{\mathsf{ms}}\Big) f(\ell+1). \end{aligned} \tag{34}$$

  since V_{ks} is negligible.

Since $V_\ell = B_\ell/q^2$ with B_ℓ is independent of q, we can rewrite Equation (34) as

$$V_\ell = \frac{B_\ell}{q^2} \approx \frac{(AB_{\ell-1} + C)f(\ell+1)}{q^2} \tag{35}$$

where $A = \frac{\eta t^2 n^2 V_s}{6} B_{\mathsf{const}}$ and $C = \frac{t^2 n^2 V_s}{12}(2\eta\tau B_{\mathsf{ks}} B_{\mathsf{const}} + B_{\mathsf{ms}})$. From Equation (35), we can recursively compute the final variance

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

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

$$q^2 \geq 8D^2 A^{L-2}(AB_{\mathsf{clean}} + C)g(L). \tag{36}$$

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

$$q^2 \geq 8D^2 A^{L-2}(AB_{\mathsf{clean}} + C/f(2))g(L). \tag{37}$$

In Table 2, we list the resulting A and C depending on the models.

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 2: The constants A, C used in the Figure 3 models to compute q with Equation (36), or Equation (37) for the OpenFHE one.

5.2 Other Circuits Expoiting 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 [8], and that proposed by *Kim et al.* in [21]. 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 3, and propose a set of parameters that limits the difference of the error growth with the previous circuit.

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 (21) and the errors of the next operations are divided by $q_{L-\ell}^2$ as well. Thus, similarly to Equation (34), we have

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

Therefore

$$V_\ell^{\mathsf{ms}} \approx \left(A_{\mathsf{ms}} V_{\ell-1} + \frac{C_{\mathsf{ms}}}{q_{L-\ell}^2}\right) f(\ell+1), \tag{38}$$

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 2 for Model 2.

Thanks to Equation (38), we can recursively compute the variance V_{L-1}^{ms} as

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

therefore,

$$\frac{A^{L-1} B_{\mathsf{clean}}}{q_L^2} g(L) + \sum_{i=1}^{L-1} \frac{A^{i-1} C_{\mathsf{ms}}}{q_i^2} \frac{g(L)}{g(L-i)} \leq \frac{1}{8D^2}. \tag{39}$$

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 (36), 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* (39) *are approximately of the same size, namely when*

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

Then $q_{\mathsf{ms}}^2 \approx 8D^2 L A^{L-1} B_{\mathsf{clean}} g(L)$, which means that q_{ms} is approximately $\sqrt{L}$ times the ciphertext modulus q in Equation (36).

Proof. We begin our proof by contradiction, assuming that there exists at least one index i in Equation (39) such that

$$\frac{A^{i-1}C_{\mathsf{ms}}}{q_i^2}\frac{g(L)}{g(L-i)} \gg \frac{A^{L-1}B_{\mathsf{clean}}}{q_L^2}g(L), \tag{40}$$

Then, called $N \geq 1$ the number such indeces, we get from Equation (39)

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

and, consequently, $q_i^2 \geq 8D^2NA^{i-1}C_{\mathsf{ms}}\frac{g(L)}{g(L-i)}$. From Equation (40), 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 (36).

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

$$\frac{A^{i-1}C_{\mathsf{ms}}}{q_i^2}\frac{g(L)}{g(L-i)} \leq \frac{A^{L-1}B_{\mathsf{clean}}}{q_L^2}g(L). \tag{41}$$

So that

$$V_{L-1}^{\mathsf{ms}} \leq \frac{LA^{L-1}B_{\mathsf{clean}}}{q_L^2}g(L), \tag{42}$$

namely, $q_{\mathsf{ms}}^2 \geq 8D^2LA^{L-1}B_{\mathsf{clean}}g(L)$. From Equation (41) 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 (42), 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 [21], 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 $V_\ell \approx AV_{\ell-1}f(\ell+1) + \frac{C_1 f(\ell+1)+C_2}{q^2} + \frac{Ef(\ell+1)}{q_{lev}^2}$, 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 $A = \frac{\eta t^2 n^2 V_s}{6} B_{\mathsf{const}}$, $C_1 = \eta\tau \frac{t^2 n^2 V_s}{6} B_{\mathsf{const}} B_{\mathsf{ks}}$, $C_2 = B_{\mathsf{ms}} + B_{\mathsf{ks}}$, $E = \frac{t^2 n^2 V_s}{6} B_{\mathsf{ms}}$. 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

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

6 Results

In this section, we shortly describe our parameter generator, Section 6.1, and show the effectiveness of the average-case approach by comparing it to the state-of-the-art works [12, 14, 20, 21], Section 6.2. Additionally, we evaluate our theoretical method in practice with OpenFHE [3].

6.1 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.* [24]. The generator interactively prompts the user with a list of required and optional inputs, then outputs code snippets with the obtained parameters for multiple state-of-the-art libraries.

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 this 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.

Finally, in the following section, we use the formulas from section 4 for theoretical comparisons and the results of our generator for practical comparisons.

[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 3: Required and optional inputs to the parameter generator

6.2 Comparison with Previous Works

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 [20], taking into account the modifications we made to the encryption and multiplication algorithms based on the work of Kim *et al.* [21]. 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 [21] or the canonical norm [12, 14, 20]. The canonical norm is known to result in better parameters.

In Table 4 we summarize how the error behaves when the homomorphic operations are performed considering the error bounds using the canonical norm.

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

Table 4: Canonical norm depending on the homomorphic operations.

In [14], 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}).
Applying the same argument of Section 5.1, we get that the following bound on the final error of a Base Model circuit $||\nu_{L-1}||^{\text{can}} \leq \frac{A^{L-2}\left(AD\sqrt{nB_{\mathsf{clean}}}+C\right)}{q}$, 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 $||\nu_{L-1}||^{\text{can}} \leq 1/2$, it follows that

$$q \geq 2A^{L-2}\left(AD\sqrt{nB_{\mathsf{clean}}}+C\right). \tag{43}$$

Average-case bounds. In the average-case approach, we set $||\nu||_\infty \leq D\sqrt{2V}$ with V variance of each coefficient of ν. Thanks to Equation (3), the bound holds with probability at least $1 - n\,(1 - \text{erf}(D))$, 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 5 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 5: Variance depending on the homomorphic operations.

In Tables 6 and 7, we compare the error analysis. For readability, we do not show the bounds themselves, but their *noise budget*: $-\log_2(2 \cdot ||\nu||) = \log_2\left(\frac{1}{2}\right) - \log_2(||\nu||)$, [27]. 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 [3] library with 10.000 polynomial samples (Table 6) and 100 (Table 7). We additionally display the average of the absolute error values under "mean", in Table 7 we also present our estimation of it as $\sqrt{V}$ with the tag "our".

For parameters, we use $t = 65537$, $n = 2^{12}, \ldots, 2^{15}$ 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 6, 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 6: Comparison in encryption, addition and multiplication of fresh ciphertexts.

In Table 7, we consider the Base Model circuit (Figure 3) 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 7: Comparison in the Base Model of depth 2 and 3 with $\alpha = 1$ and $\eta = 8$.

Tables 6 and 7 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 8, we present the obtained bounds for $\log_2(q)$ following from the two theoretical approaches (Equation (43) and Equation (36)) 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 8: 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 efficiency and the security of the scheme.

Finally, in Figure 4, we graphically compare our parameter generation with the OpenFHE one, based on theoretical work with the infinity norm [21]. We compare our generated bounds with the size of the ciphertext modulus generated for $\lambda = 128$.

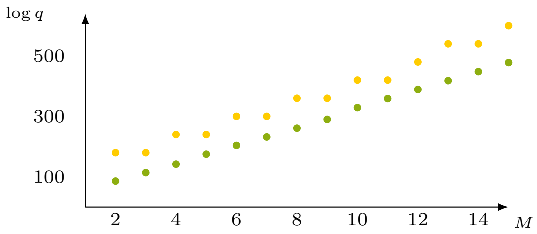

Fig. 4: Comparison of modulus sizes across multiplicative depths M with $\lambda = 128$ and $t = 2^{16} + 1$ for OpenFHE 🟡 and our 🟢 parameter generation.

7 Conclusion

The selection of optimal parameters for a specific Fully Homomorphic Encryption scheme can be challenging. Multiple approaches have been proposed in the analysis of the noise growth, such as the infinity norm or the canonical norm. Recently, works on other schemes have shown the effectiveness of the average-case approach for this task.

In this work, we provide an in-depth average-case noise analysis of the BFV scheme, outperforming all previous work on BFV parameter selection. Using a novel method due to the limitations of the CLT for BFV, we significantly improve the accuracy of error estimates for low and high multiplicative depths. For example, with just three multiplications, we tighten the previous bounds by at least 17 bits. More importantly, our new analysis results in bound very close to experimentally observed values differing by at most 3 bits. This suggests that our approach provides very reliable estimates for the BFV scheme reflected in significantly smaller bounds on the ciphertext modulus and thus much better performance.

In addition, we combine our analysis with the security formula proposed in [24], to develop the first parameter generator for multiple state-of-the-art BFV libraries. Flexible and easy-to-use, it aims at making our theoretical work accessible to the community.

Overall, our work advances the state-of-the-art parameter selection for the BFV scheme and provides a powerful tool that can assist in selecting efficient and reliable parameters for the BFV scheme, making the task significantly more efficient and accessible for researchers and practitioners.

Bibliography

[1] Martin R Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption security standard. Technical report, `HomomorphicEncryption.org`, Toronto, Canada, November 2018.

[2] Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

[3] Ahmad Al Badawi, Jack Bates, Flavio Bergamaschi, David Bruce Cousins, Saroja Erabelli, Nicholas Genise, Shai Halevi, Hamish Hunt, Andrey Kim, Yongwoo Lee, Zeyu Liu, Daniele Micciancio, Ian Quah, Yuriy Polyakov, Saraswathy R.V., Kurt Rohloff, Jonathan Saylor, Dmitriy Suponitsky, Matthew Triplett, Vinod Vaikuntanathan, and Vincent Zucca. Openfhe: Open-source fully homomorphic encryption library. Cryptology ePrint Archive, Paper 2022/915, 2022. `https://eprint.iacr.org/2022/915`.

[4] Jean-Claude Bajard, Julien Eynard, M Anwar Hasan, and Vincent Zucca. A full RNS variant of FV like somewhat homomorphic encryption schemes. In *International Conference on Selected Areas in Cryptography*, pages 423–442. Springer, 2016.

[5] Loris Bergerat, Anas Boudi, Quentin Bourgerie, Ilaria Chillotti, Damien Ligier, Jean-Baptiste Orfila, and Samuel Tap. Parameter Optimization & Larger Precision for (T) FHE. *Cryptology ePrint Archive*, 2022.

[6] Zvika Brakerski. Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, pages 868–886, Berlin, Heidelberg, 2012. Springer.

[7] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.

[8] Zvika Brakerski and Vinod Vaikuntanathan. Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, pages 505–524, Berlin, Heidelberg, 2011. Springer.

[9] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 409–437, Cham, 2017. Springer International Publishing.

[10] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *international conference on the theory and application of cryptology and information security*, pages 3–33. Springer, 2016.

[11] Anamaria Costache, Benjamin R Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, and Rachel Player. On the precision loss in approximate homomorphic encryption. *Cryptology ePrint Archive*, 2022.

[12] Anamaria Costache, Kim Laine, and Rachel Player. Evaluating the effectiveness of heuristic worst-case noise analysis in FHE. In *European Symposium on Research in Computer Security*, pages 546–565. Springer, 2020.

[13] Anamaria Costache, Lea Nürnberger, and Rachel Player. Optimisations and tradeoffs for helib. In *Topics in Cryptology–CT-RSA 2023: Cryptographers' Track at the RSA Conference 2023, San Francisco, CA, USA, April 24–27, 2023, Proceedings*, pages 29–53. Springer, 2023.

[14] Anamaria Costache and Nigel P Smart. Which ring based somewhat homomorphic encryption scheme is best? In *Cryptographers' Track at the RSA Conference*, pages 325–340. Springer, 2016.

[15] Andrea Di Giusto and Chiara Marcolla. Breaking the power-of-two barrier: noise estimation for BGV in NTT-friendly rings. *Cryptology ePrint Archive, Paper 2023/783*, 2023.

[16] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012.

[17] Craig Gentry. *A fully homomorphic encryption scheme*, volume 20. Stanford university Stanford, 2009.

[18] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic Evaluation of the AES Circuit. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, pages 850–867, Berlin, Heidelberg, 2012. Springer.

[19] Shai Halevi, Yuriy Polyakov, and Victor Shoup. An improved RNS variant of the BFV homomorphic encryption scheme. In *Cryptographers' Track at the RSA Conference*, pages 83–105. Springer, 2019.

[20] Ilia Iliashenko. Optimisations of fully homomorphic encryption. PhD thesis, 2019.

[21] Andrey Kim, Yuriy Polyakov, and Vincent Zucca. Revisiting homomorphic encryption schemes for finite fields, 2021.

[22] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 1–23, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[23] Chiara Marcolla, Victor Sucasas, Marc Manzano, Riccardo Bassoli, Frank HP Fitzek, and Najwa Aaraj. Survey on fully homomorphic encryption, theory, and applications. *Proceedings of the IEEE*, 110(10):1572–1609, 2022.

[24] Johannes Mono, Chiara Marcolla, Georg Land, Tim Güneysu, and Najwa Aaraj. Finding and Evaluating Parameters for BGV. *Africacrypt*, 2023.

[25] Sean Murphy and Rachel Player. A central limit approach for ring-lwe noise analysis.

[26] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 84–93, 2005.

[27] Microsoft SEAL (release 3.4). `https://github.com/Microsoft/SEAL`, October 2019. Microsoft Research, Redmond, WA.

A Characterization of the error

Proposition 2. *The noise invariant can always be written as*

$$\nu = \sum_{\iota} a_\iota s^\iota$$

and enjoys the following properties

1. $\mathbb{E}[a_\iota|_i] = 0$ *for any* ι,
2. $\mathsf{Cov}(a_{\iota_1}|_{i_1}, a_{\iota_2}|_{i_2}) = 0$ *if either* $\iota_1 \neq \iota_2$ *or* $i_1 \neq i_2$.

Proof. **Fresh ciphertext** Recall that the noise after the encryption is

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

by Equation (6). We can rewrite it as $a_0 + a_1 s$ with $a_0 = \frac{t}{q}(\varepsilon + eu + e_0)$ and $a_1 = \frac{t}{q}e_1$.

1. Since $e_1|_i \leftarrow \chi_e$ and χ_e is symmetric, we have $\mathbb{E}[e_1|_i] = 0$, then

$$\mathbb{E}[a_1|_i] = \frac{t}{q}\mathbb{E}[e_1|_i] = 0.$$

Analogously, $\mathbb{E}[\varepsilon|_i] = \mathbb{E}[e|_i] = \mathbb{E}[u|_i] = \mathbb{E}[e_0|_i] = 0$ and all the coefficients of these polynomial are independent among each other, so

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

2. By the independence of the coefficients of e_1 among each other and with the coefficients of the other polynomial and by the covariance bilinearity (see Section 2.2), we get $\mathsf{Cov}(a_1|_{i_1}, a_1|_{i_2}) = 0$ for $i_1 \neq i_2$ and $\mathsf{Cov}(a_0|_{i_1}, a_1|_{i_2}) = 0$ for any i_1, i_2. Again for independence of the coefficients and the bilinearity of the covariance, we have

$$\mathsf{Cov}(a_0|_{i_1}, a_0|_{i_2}) = \frac{t^2}{q^2}\mathsf{Cov}((eu)|_{i_1}, (eu)|_{i_2}).$$

Since $(eu)|_i = \sum_{j=0}^{n-1} \xi(i,j) e|_j u|_{i-j}$,

$$\mathsf{Cov}((eu)|_{i_1}, (eu)|_{i_2}) = \sum_{j_1, j_2 = 0}^{n-1} \xi(i_1, j_1)\xi(i_2, j_2)\mathsf{Cov}(e|_{j_1} u|_{i_1 - j_1}, e|_{j_2} u|_{i_2 - j_2})$$

$$= \sum_{j_1,j_2=0}^{n-1} \xi(i_1,j_1)\xi(i_2,j_2)\Big(\mathbb{E}[e|_{j_1}e|_{j_2}]\mathbb{E}[u|_{i_1-j_1}u|_{i_2-j_2}]+$$

$$-\mathbb{E}[e|_{j_1}]\mathbb{E}[e|_{j_2}]\mathbb{E}[u|_{i_1-j_1}]\mathbb{E}[u|_{i_2-j_2}]\Big) = 0,$$

as either $j_1 \neq j_2$, hence $\mathbb{E}[e|_{j_1}e|_{j_2}] = \mathbb{E}[e|_{j_1}]\mathbb{E}[e|_{j_2}]$, or $i_1 - j_1 \bmod n \neq i_2 - j_2 \bmod n$, so $\mathbb{E}[u|_{i_1-j_1}u|_{i_2-j_2}] = \mathbb{E}[u|_{i_1-j_1}]\mathbb{E}[u|_{i_2-j_2}]$.

Addition Let $\nu = \sum_\iota a_\iota s^\iota$, $\nu' = \sum_{\iota'} a'_{\iota'} s^{\iota'}$ be two errors as claimed in the proposition, then $\nu_{\mathsf{add}} = \sum_\iota (a_\iota + a'_\iota)s^\iota$ with

1. $\mathbb{E}[(a_\iota + a'_\iota)|_i] = \mathbb{E}[a_\iota|_i] + \mathbb{E}[a'_\iota|_i] = 0,$

2. If $\iota_1 \neq \iota_2$ or $i_1 \neq i_2$, $\mathsf{Cov}((a_{\iota_1} + a'_{\iota_1})|_{i_1}, (a_{\iota_2} + a'_{\iota_2})|_{i_2})$ is equal to

   $\mathsf{Cov}(a_{\iota_1}|_{i_1}, a_{\iota_2}|_{i_2}) + \mathsf{Cov}(a_{\iota_1}|_{i_1}, a'_{\iota_2}|_{i_2}) + \mathsf{Cov}(a'_{\iota_1}|_{i_1}, a_{\iota_2}|_{i_2}) + \mathsf{Cov}(a'_{\iota_1}|_{i_1}, a'_{\iota_2}|_{i_2}) = 0,$

   because a_{ι_1} and a'_{ι_2} are independent and the other pairs are uncorrelated.

Modulo switch & Key switch The proof is analogous to the addition one by independence of the added quantity with the error ν.

Constant multiplication Let $\nu = \sum_\iota a_\iota s^\iota$ be an error satisfying the properties above and α be a random element from $\mathcal{U}_t$, then $\alpha\nu = \sum_\iota \alpha a_\iota s^\iota$ and

1. Since the α and a_ι are independent with null expected value,

$$\mathbb{E}\big[(\alpha a_\iota)|_i\big] = \sum_{j=0}^{n-1} \xi(i,j)\mathbb{E}[\alpha|_j]\mathbb{E}[a_\iota|_{i-j}] = 0.$$

2. By bilinearity of the covariance, we have

$$\mathsf{Cov}\Big((\alpha a_{\iota_1})|_{i_1}, (\alpha a_{\iota_2})|_{i_2}\Big) = \sum_{j_1,j_2=0}^{n-1} \xi(i_1,j_1)\xi(i_2,j_2)\mathsf{Cov}\big(\alpha|_{j_1} a_{\iota_1}|_{i_1-j_1}, \alpha|_{j_2} a_{\iota_2}|_{i_2-j_2}\big),$$

   with $\mathsf{Cov}\big(\alpha|_{j_1} a_{\iota_1}|_{i_1-j_1}, \alpha|_{j_2} a_{\iota_2}|_{i_2-j_2}\big)$ equal to

$$\mathbb{E}[\alpha|_{j_1}\alpha|_{j_2}]\mathbb{E}[a_{\iota_1}|_{i_1-j_1}a_{\iota_2}|_{i_2-j_2}] - \mathbb{E}[\alpha|_{j_1}]\mathbb{E}[\alpha|_{j_2}]\mathbb{E}[a_{\iota_1}|_{i_1-j_1}]\mathbb{E}[a_{\iota_2}|_{i_2-j_2}].$$

   If $\iota_1 \neq \iota_2$ or $i_1 - j_1 \bmod n \neq i_2 - j_2 \bmod n$, then

$$\mathbb{E}[a_{\iota_1}|_{i_1-j_1}a_{\iota_2}|_{i_2-j_2}] = \mathbb{E}[a_{\iota_1}|_{i_1-j_1}]\mathbb{E}[a_{\iota_2}|_{i_2-j_2}]$$

   by the second property of the error ν.
   Otherwise, $j_1 \neq j_2$, by independence of the coefficients of α,

$$\mathbb{E}[\alpha|_{j_1}\alpha|_{j_2}] = \mathbb{E}[\alpha|_{j_1}]\mathbb{E}[\alpha|_{j_2}].$$

   It follows

$$\mathsf{Cov}\Big((\alpha a_{\iota_1})|_{i_1}, (\alpha a_{\iota_2})|_{i_2}\Big) = 0.$$

Multiplication Let ν, ν' as before, then $\nu\nu' = \sum_{\iota} \sum_{j+k=\iota} a_j a'_k s^{\iota}$.

1. Since a_j and a'_k are independent,
$$\mathbb{E}\big[\big(\sum_{j+k=\iota} a_j a'_k\big)|_i\big] = \sum_{j+k=\iota} \mathbb{E}[a_j]\mathbb{E}[a'_k] = 0.$$

2. For $\iota_1 \neq \iota_2$ or $i_1 \neq i_2$,
$$\mathsf{Cov}\big(\big(\sum_{j_1+k_1=\iota_1} a_{j_1} a'_{k_1}\big)|_{i_1}, \big(\sum_{j_2+k_2=\iota_2} a_{j_2} a'_{k_2}\big)|_{i_2}\big) =$$
$$= \sum_{j_1+k_1=\iota_1} \sum_{j_2+k_2=\iota_2} \sum_{l_1,l_2=0}^{n-1} \xi(i_1,l_1)\xi(i_2,l_2)\mathsf{Cov}(a_{j_1}|_{l_1} a'_{k_1}|_{i_1-l_1}, a_{j_2}|_{l_2} a'_{k_2}|_{i_2-l_2}),$$
where
$$\begin{aligned}\mathsf{Cov}(a_{j_1}|_{l_1} a'_{k_1}|_{i_1-l_1}, a_{j_2}|_{l_2} a'_{k_2}|_{i_2-l_2}) &= \mathbb{E}[a_{j_1}|_{l_1} a_{j_2}|_{l_2}]\mathbb{E}[a'_{k_1}|_{i_1-l_1} a'_{k_2}|_{i_2-l_2}]+\\ &- \mathbb{E}[a_{j_1}|_{l_1}]\mathbb{E}[a_{j_2}|_{l_2}]\mathbb{E}[a'_{k_1}|_{i_1-l_1}\mathbb{E}[a'_{k_2}|_{i_2-l_2}] = 0,\end{aligned}$$
indeed, if $\iota_1 = \iota_2$ then $j_1 \neq j_2$ or $k_1 \neq k_2$, while if $i_1 \neq i_2$ then $i_1 - l_1 \bmod n \neq i_2 - l_2 \bmod n$ or $l_1 \neq l_2$.

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 is 0, hence the conditions hold also for $\nu_{\mathsf{mul}} = -\nu\nu' + \nu \frac{t}{q'_\ell}(c'_0 + c'_1 s) + \nu' \frac{t}{q_\ell}(c_0 + c_1 s) + \frac{t}{q}(\varepsilon_0 + \varepsilon_1 s + \varepsilon_2 s^2)$.

A.1 Proof of Lemma 3

Proof. Let us assume $T_1 \leq T_2$. Observe that

$$\begin{aligned}
\frac{g(T_1+T_2)}{g(2)g(T_1+T_2)} &= \frac{1}{f(2)} \quad \text{if } T_1 = 1,\ T_2 = T_1 + T_2 - 1,\\
\frac{g(T_1+T_2)}{g(3)g(T_1+T_2-1)} &= \frac{f(T_1+T_2)}{f(2)f(3)} \quad \text{if } T_1 = 2,\ T_2 = T_1 + T_2 - 2,\\
\frac{g(T_1+T_2)}{g(4)g(T_1+T_2-2)} &= \frac{f(T_1+T_2)f(T_1+T_2-1)}{f(2)f(3)f(4)} \quad \text{if } T_1 = 3,\ T_2 = T_1 + T_2 - 3,\\
&\vdots\\
\frac{g(T_1+T_2)}{g(\lfloor\frac{T_1+T_2}{2}\rfloor+1)g(\lceil\frac{T_1+T_2}{2}\rceil+1)} &= \frac{f(\lceil\frac{T_1+T_2}{2}\rceil+2)\cdots f(T_1+T_2)}{f(2)\cdots f(\lfloor\frac{T_1+T_2}{2}\rfloor+1)} \quad \text{if } T_1 = \lfloor\tfrac{T_1+T_2}{2}\rfloor,
\end{aligned}$$

which are in increasing order. Hence

$$\frac{g(T_1+T_2)}{g(T_1+1)g(T_2+1)} \leq \frac{f(\lceil\frac{T_1+T_2}{2}\rceil+2)\cdots f(T_1+T_2)}{f(2)\cdots f(\lfloor\frac{T_1+T_2}{2}\rfloor+1)} =$$

$$\leq \frac{1}{f(\lfloor \frac{T_1+T_2}{2} \rfloor + 1)} \prod_{\iota=2}^{\lfloor \frac{T_1+T_2}{2} \rfloor} \frac{f(\lceil \frac{T_1+T_2}{2} \rceil + \iota)}{f(\iota)}$$

We set $T = T_1 + T_2$ and $\tau = \lfloor \frac{T}{2} \rfloor$. Moreover, we define $c_\iota = f(\iota) = c - \frac{1}{e^{a\iota - b}}$ and $\varepsilon_\iota = \left(1 - \frac{1}{e^{a(T-\tau)}}\right) \frac{1}{e^{a\iota - b}}$, then

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

Since $\frac{c_\iota + \varepsilon_\iota}{c_\iota} \geq 1$, we have that

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

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

$$\begin{aligned} \sum_{i=2}^{\tau} \frac{\varepsilon_i}{c_i} \leq & \frac{\varepsilon_2}{c_2} \sum_{i=2}^{\tau} \left(\frac{1}{e^a}\right)^i \leq \frac{\varepsilon_2}{c_2} \left(\frac{e^a}{e^a - 1} \cdot \frac{e^{(\tau+1)a} - 1}{e^{(\tau+1)a}} - \frac{e^a + 1}{e^a} \right) \\ \leq & \frac{\varepsilon_2}{c_2} \left(\frac{1}{e^{2a} - e^a} \right) \leq \frac{e^b}{(e^{2a}c - e^b)(e^{2a} - e^a)}. \end{aligned}$$

We conclude that $\frac{g(T_1+T_2)}{g(T_1+1)g(T_2+1)}$ is bounded by a finite constant and we estimated it computationally by evaluating

$$\frac{f(\lceil \frac{T_1+T_2}{2} \rceil + 2) \cdots f(T_1 + T_2)}{f(2) \cdots f(\lfloor \frac{T_1+T_2}{2} \rfloor + 1)},$$

for up to $T_1 + T_2 = 2^{20}$. The obtained values are listed in upper bounds K_n in Lemma 3.