Fully Homomorphic Encryption on large integers

P. Chartier, M. Koskas, M. Lemou and F. Méhats

Ravel Technologies
75 rue de Richelieu, 75020 Paris

June 12, 2024

Abstract

At the core of fully homomorphic encryption lies a procedure to refresh the ciphertexts whose noise component has grown too big. The efficiency of the so-called *bootstrap* is of paramount importance as it is usually regarded as the main bottleneck towards a real-life deployment of fully homomorphic crypto-systems. In two of the fastest implementations so far [4, 6], the space of messages is limited to binary integers. If the message space is extended to the discretized torus $\mathbb{T}_{p_i}$ or equivalently to $\mathbb{Z}_{p_i}$ with values of p_i large as compared to the dimension of the polynomial ring in which the operations are realised, the bootstrap delivers incorrect results with far too high probability. As a consequence, the use of a residue numeral system to address large integers modulo $p = p_1 \times \ldots \times p_\kappa$ would be of limited interest in practical situations without the following remedy : rather than increasing the polynomial degree and thus the computational cost, we introduce here a novel and simple technique (hereafter referred to as "collapsing") which, by grouping the components of the mask, attenuates both rounding errors and computational costs, and greatly helps to sharpen the correctness of the bootstrap. We then rigorously estimate the probability of success as well as the output error, determine practical parameters to reach a given correctness threshold and present implementation results.

Keywords: homomorphic encryption, bootstrap, large integers, rounding errors, collapse, probability estimates.

1 Introduction

This paper is concerned with a protocol of homomorphic encryption of large integers, which combines the encoding of integers modulo

$$p = p_1 \times \ldots \times p_\kappa$$

where the p_i's are pairwise coprime, through the *Chinese Remainder Theorem* (CRT) and a FHEW/TFHE-like encryption protocol [4, 6, 10] of a set of messages in the $\mathbb{Z}_{p_i}$'s. The extension – in the present context of *fully homomorphic encryption (FHE)* – of the message space from binary integers to the discretized torus with p_i messages (or equivalently $\mathbb{Z}_{p_i}$) has been introduced in [2, 3, 11] by the authors and in [9] by M. Joye. However, large values of the p_i's, which are necessary to attain 32-bits or 64-bits integers, may compromise the correctness of the bootstrapping [5, 6] operation. The main focus of this work is thus put on the analysis of a new bootstrapping procedure, which is a modification of the

FHEW/TFHE-bootstrap introduced in [5, 6], and allows for instance to handle 32-bits integers with a good level of security within the usual setting of parameters for polynomial rings of dimension 2^{10} without additional computational cost.

For the sake of clarity, let us stress that nothing prevents us in principle from considering values $p_i \geq 2$ in the usual FHEW/TFHE-bootstrap: nevertheless, the rounding process associated with the modulo switch introduces large biases that can be corrected only at the price of using ring ciphertexts with cyclotomic polynomials of large index[1]. This in turn renders the whole procedure computationally more costly, a particular undesirable consequence as it is already viewed as the main bottleneck of FHE. In order to allow for large values of p_i at no extra cost, we thus introduce a new technique of *collapsing*, that allows to reduce very significantly the rounding errors. Let us emphasize that, as in [8], it also allows for non-binary secret keys and that it decreases the computational cost *per se*.

We now illustrate its main idea in the TFHE setting [5]: during the first step of the bootstrapping procedure, the quantity (the so-called *phase*)

$$b - \sum_{i=1}^{n} s_i a_i = \frac{k}{p_i} + e \bmod 1 \tag{1.1}$$

where $k \in \mathbb{Z}_{p_i}$, $e \in \mathbb{R}/\mathbb{Z}$, $(a_1, \ldots, a_n) \in (\mathbb{R}/\mathbb{Z})^n$ and $(s_1, \ldots, s_n) \in \mathbb{S}^n \subset \mathbb{Z}^n$ are respectively the message, the noise, the mask and the secret key of an incoming ciphertext $c = (a, b)$, should ideally be rounded to its closest value in $\mathbb{Z}_M$, that is to say

$$\left\lfloor M \left(b - \sum_{i=1}^{n} s_i a_i \right) \right\rceil \bmod M \tag{1.2}$$

where M is the index of the cyclotomic polynomial ring $\mathbb{Q}[X]/\Phi_M(X)$ in which all further steps are carried out. However, the components of s being secret, one has to approximate the sum (1.2) and this is usually done by substituting to (1.2) the following expression

$$\lfloor Mb \rceil - \sum_{i=1}^{n} \lfloor s_i M a_i \rceil = \lfloor Mb \rceil - \sum_{i=1}^{n} \sum_{j \in \mathbb{S}} \lfloor M j a_i \rceil \, \delta_{j,s_i} \tag{1.3}$$

(where δ_{j,s_i} is the Kronecker symbol of j and s_i) and resorting[2] to encrypted values of the δ_{j,s_i}'s. In doing so, a large rounding bias is introduced, which can compromise the correctness of the whole bootstrapping procedure. Now, if instead of the brute-force approximation (1.3) we gather the components of the sum (1.2) in sub-sums of $m \geq 2$ elements[3], then we obtain the more accurate approximation

$$\lfloor Mb \rceil - \sum_{i=1}^{n/m} \underbrace{\sum_{\tilde{\jmath} \in \mathbb{S}^m} \left\lfloor M \left(\tilde{\jmath}_1 a_{(i-1)m+1} + \ldots + \tilde{\jmath}_m a_{im} \right) \right\rceil \delta_{\tilde{\jmath}, \tilde{s}_i}}_{\alpha_i} \tag{1.4}$$

[1] We shall consider here polynomial rings of the form $\mathbb{Q}[X]/\Phi_M(X)$ for values of M which are powers of two. Other choices are possible without fundamentally affecting the technique described here.

[2] Note that if $\mathbb{S} = \{0, 1\}$ or $\mathbb{S} = \{-1, 0, 1\}$, then the sum takes the more common form $\lfloor Mb \rceil - \sum_{i=1}^{n} \lfloor M a_i \rceil \, s_i$.

[3] For the sake of convenience, we assume here that m divides n. Otherwise, formula (1.4) can be straightforwardly adapted by adding a last block of reduced length.

m	1	2	3	4	5	6	7	8	9	10
Average	3.95	3.46	3.02	2.71	2.50	2.26	2.12	1.96	1.86	1.75
Variance	24.69	19.13	14.60	11.74	10.04	8.27	7.24	6.20	5.62	5.00

Table 1: Average and variance of the absolute rounding error (absolute value of difference between quantities (1.2) and (1.4)) for ciphertexts $(a, b) \in (\mathbb{R}/\mathbb{Z})^{601}$ with $k = 0$.

with $\tilde{s}_i = (s_{(i-1)m+1}, \ldots, s_{im})$ for $i = 1, \ldots, n/m$, where we used the generalized Kronecker symbol

$$\delta_{\tilde{\jmath},\tilde{s}} = 1 \text{ iff } \tilde{\jmath} = \tilde{s} \in \mathbb{S}^m$$

and where, from now on, encrypted values of the $\delta_{\tilde{\jmath},\tilde{s}_i}$'s are required. Our variant of the bootstrapping procedure of [6, 4] thus consists in *homomorphically* computing (see Section 3 for more details and a slightly improved version)

$$\mathrm{Coeff}_0 \left(X^{\lfloor Mb \rceil} \cdot \left(X^{-\alpha_{n/m}} \ldots (X^{-\alpha_2} \cdot (X^{-\alpha_1} \cdot v(X))) \ldots \right) \bmod \Phi_M(X) \right) \tag{1.5}$$

where

$$X^{-\alpha_i} = \sum_{\tilde{\jmath} \in \mathbb{S}^m} \delta_{\tilde{\jmath},\tilde{s}_i} X^{-\left\lfloor M\left(\tilde{\jmath}_1 a_{(i-1)m+1} + \ldots + \tilde{\jmath}_m a_{im}\right)\right\rceil \bmod M}, \quad i = 1, \ldots, n/m. \tag{1.6}$$

Let us mention that this procedure was patented by the authors and emphasize that, while sharing some similarities with [8], it significantly differs from the aforementioned publication insofar as *rounding* are operated by blocks of m values. In short, whereas in [8] the a_i-values are first rounded and then grouped, they are here first grouped and then rounded: the main objective of our collapsing technique is not primarily to accelerate the bootstrapping (the incidental speed-up obtained is indeed also obtained in [8]) but rather to diminish the rounding errors and offer much improved *correctness* probabilities (see Table 3 for a lower bound of the probability of a wrong bootstrap when only rounding errors are taken into account).

It is clear that the larger m is, the more accurate (1.4), (1.6) become (see Table 1) and the more computationally costly the internal sums (1.6) grow[4] (see Table 2). Note that for $m = 1$, the two sums (1.3) and (1.4) coincide while for $m = n$, the two sums (1.2) and (1.3) coincide, i.e. the approximation (1.4) becomes exact... and computationally intractable. Besides computational aspects, it is worth emphasizing that the number of $\delta_{\tilde{\jmath},\tilde{s}_i}$'s in (1.6) that need to be encrypted and publicly released, increases exponentially fast with m (more precisely like $n/m \cdot |\mathbb{S}|^m$) : alongside the computational cost of the inner loops of (1.4), this factor plays an important role to determine m. However, we will see that for moderate values of m (say from 2 to 5) an efficient trade-off can be obtained. To this aim, we will analyse rigorously the probability of correctness of the new complete bootstrapping procedure and show that it is possible to ensure that incorrect bootstrap remain extremely rare for realistic parameters. Our new *collapsing technique* appears to be necessary in this context to attain large values of p_i and thus the homomorphic treatment of large integers.

[4] Note that they could be evaluated independently from one another on a multi-threaded computer.

m	1	2	3	4	5	6	7	8	9	10
Number of terms in (1.6)	2	4	8	16	32	64	128	256	512	1024
$\lfloor n/m \rceil$	600	300	200	150	120	100	90	75	68	60

Table 2: Number of terms in the sums (1.6) and number of modular products in (1.5) for ciphertexts $(a, b) \in (\mathbb{R}/\mathbb{Z})^{601}$.

m	1	2	3	4	5	6	7	8	9	10
$\mathbb{P}$(error)	2^{-8}	2^{-15}	2^{-22}	2^{-28}	2^{-34}	2^{-41}	2^{-47}	2^{-54}	2^{-60}	2^{-67}

Table 3: Probability of error of the bootstrap for noise-free encryptions (i.e. $e = 0$ in (1.1)) of messages in $\frac{1}{29}\mathbb{Z}_{29}$ (i.e. with $p_i = 29$ in (1.1)).

In Section 2, we define the encoding protocol for large numbers as well as the TFHE-encryption and introduce some standard definitions for the homomorphic operations that are required here. In Section 3, we give a brief description of the bootstrapping, first for cleartexts, and then for ciphertexts where the modifications that we propose come into play. In Section 4, we estimate the variance of the error of the bootstrap output. Eventually, Section 5 is devoted to the statement of Proposition 5.2 which gives accurate estimates of the probability of correctness of the new bootstrap. The comparative advantage of this new estimate is illustrated by means of several numerical experiments. As an example of application of our results, the size of the standard deviation σ used for encryption is assessed for different values of the ring-dimension of the security parameter with aid of the *lattice estimator*[5]. The corresponding correctness probabilities may be then read from the figures given here and the parameters calibrated accordingly. Results of a C++ implementation illustrate the evolution of the computational cost with respect to the parameter m.

2 The chinese remainder theorem and the encryption scheme

Our homomorphic scheme is based on homomorphic multi-modular arithmetic, where large integers are described by a Residue Number System (RNS), owing to the Chinese Remainder Theorem (CRT). For modular operations such as addition, subtraction and multiplication, the computations with residues are independent with each other, which provides parallel, carry-free homomorphic arithmetic. However, non-modular operations that require the determination of magnitudes (e.g., number comparison, overflow detection, and general division) do not have a parallel form in RNS and need specific adaptations in order to get homomorphic counterparts.

[5] `https://github.com/malb/lattice-estimator`

2.1 Notations

In this subsection, we introduce a series of useful notations and the parameters of our cryptographic system. The scalar product of two vectors a and b is denoted by $a \cdot b$. The space for secret keys will be denoted by $\mathbb{S} \subset \mathbb{Z}$ (for instance $\mathbb{S} = \{0, 1\}$ for binary keys or $\mathbb{S} = \{-1, 0, 1\}$ for ternary keys), its cardinal being $|\mathbb{S}|$.

For all $m \in \mathbb{N}^*$, the m-rounding (half up) of a real number $x \in \mathbb{R}$ is defined by

$$\lfloor x \rceil_m = \left\lfloor \frac{x}{m} \right\rceil m = \left\lfloor \frac{x}{m} + \frac{1}{2} \right\rfloor m,$$

which ensures that the associated residue $x \bmod m := x - \lfloor x \rceil_m$ belongs to the interval

$$I_m = [-\lfloor m/2 \rfloor, \lfloor (m-1)/2 \rfloor] .$$

Algebraic structures

Denoting $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z}$, we shall consider the ring $(\mathbb{Z}_m, +, \times)$. For all $x \in \mathbb{Z}_m$, its canonical representative will be $x \bmod m \in \mathbb{I}_m = I_m \cap \mathbb{Z}$ (throughout this paper, when no confusion is possible, we will identify a class of $\mathbb{Z}_m$ and its representative). When m_i is a divisor of m, one can also define $x \bmod m_i \in \mathbb{I}_{m_i}$ (which is independent of the chosen representative of the class x). We will also consider the torus $\mathbb{T} = \mathbb{R}/\mathbb{Z}$, which is not a ring but is endowed with a $\mathbb{Z}$-module structure, equipped with the addition $+$ and the external product $\cdot$ with integers. The discrete torus $\mathbb{T}_m$ is defined by $\mathbb{T}_m = \frac{1}{m}\mathbb{Z}_m$ and will be considered as a subset of $\mathbb{T}$.

For all $x \in \mathbb{R}$, we denote by $\pi_m(x)$ its projection on $\mathbb{T}_m$, which is the element of $\mathbb{T}$ whose representative is $\lfloor mx \rceil / m \bmod 1$. Noticing that the result does not depend on the choice of the representative, this definition naturally extends to $x \in \mathbb{T}$. Moreover, this projection will also be applied to polynomials, coefficient by coefficient. Considering now two integers $q > p \geq 2$, we assume, in order to ease the presentation, that p divides q, so that

$$\mathbb{T}_p = \{i/p, \quad 0 \leq i \leq p-1\} \subset \mathbb{T}_q$$

is composed of exactly p elements. Nevertheless, our discussion and results can be easily adapted to the general situation where $p \nmid q$.

For $N \in \mathbb{N}^*$ a power of 2 and $m \in \mathbb{N}^*$, we consider the $2N$-th cyclotomic polynomial $X^N + 1$ and the polynomial sets

$$\mathbb{T}_N[X] = \mathbb{T}[X]/(X^N + 1), \qquad \mathbb{T}_{N,m}[X] = \mathbb{T}_m[X]/(X^N + 1)$$

which are $\mathbb{Z}_N[X]$-modules with $\mathbb{Z}_N[X] := \mathbb{Z}[X]/(X^N + 1)$. The set of polynomials of $\mathbb{Z}_N[X]$ with coefficients in $\mathbb{S}$ will be denoted by $\mathbb{S}_N[X]$. For all $P = \sum_{i=0}^{N-1} P_i X^i \in \mathbb{Z}_N[X]$, we shall denote $\|P\|_2^2 = \sum_{i=0}^{N-1} P_i^2$.

Probability distributions

Our cryptographic system will rely on random distributions. If B is a finite set, $a \overset{\$}{\leftarrow} B$ indicates that a is sampled uniformly in B. For $\sigma \in \mathbb{R}_+^*$ and $x \in \mathbb{R}$, we denote by $\rho_\sigma(x) = \exp(-x^2/2\sigma^2)$ the centered Gaussian with standard deviation σ. If S is a subset of $\mathbb{R}$, then $\rho_\sigma(S)$ denotes $\sum_{x\in S}\rho_\sigma(x)$ if S is discrete, or $\int_S \rho_\sigma(x)dx$, if S is Lebesgue measurable.

Let M be a (continuous or discrete) closed additive subgroup of $\mathbb{R}$. We define on M a restricted centered Gaussian distribution $\mathcal{D}_{M,\sigma}$ of standard deviation σ over M with the density function $\mathcal{D}_{M,\sigma}(x) = \rho_\sigma(x)/\rho_\sigma(M)$. If L is a discrete subgroup of M, then the modular Gaussian distribution $\mathcal{D}_{M/L,\sigma}$ over M/L exists and is defined by the density $\mathcal{D}_{M/L,\sigma}(x) = \mathcal{D}_{M,\sigma}(x)(x+L)$.

In particular, this defines the modular Gaussian distribution $\mathcal{D}_{\mathbb{T}_q,\sigma}$ over $\mathbb{T}_q$. Sampling polynomials according to the corresponding distribution $\mathcal{D}_{\mathbb{T}_{N,q},\sigma}$ will be done coefficient by coefficient.

The RNS representation

Consider a set of κ pairwise relatively prime integers $p_1, p_2, \cdots, p_\kappa$, also referred to as the moduli, and denote their product by p. In the corresponding RNS, a modulo p integer $x \in \mathbb{Z}_p$ is represented unambiguously by the set of residues $x_i = x \bmod p_i \in \mathbb{I}_{p_i}$. The κ-tuple

$$(x_1, x_2, \cdots, x_\kappa) \in \mathbb{Z}_{p_1} \times \mathbb{Z}_{p_2} \times \cdots \times \mathbb{Z}_{p_\kappa}$$

will be called the RNS representation of x. Note that, by virtue of the CRT, conversion from residues $(x_1, x_2, \cdots, x_\kappa)$ back to the modulo p of the integer x can be done as follows:

$$x = \sum_{i=1}^{\kappa} v_i \widehat{p_i} x_i \bmod p,$$

where v_i is the inverse of $\widehat{p_i} = p/p_i$ in $\mathbb{Z}_{p_i}$ thanks to the Bezout relation

$$(u_i p_i + v_i \widehat{p_i}) \mod p = 1.$$

Since addition, subtraction and multiplication are modular operations, we have, for all $(x, y) \in \mathbb{Z}_p^2$, for $\circ \in \{+, -; \times\}$ and for $i \in \{1, \cdots, \kappa\}$,

$$(x \circ y) \bmod p_i = (x \bmod p_i) \circ (y \bmod p_i) \bmod p_i.$$

2.2 Encryption schemes in the torus

In this section, we recall several basic schemes of TFHE [4, 5] that will be used in our construction. We first define $(q, n, \sigma, \mathbb{S})$ such that the LWE problem on $\mathbb{Z}_q$, of size n, with keys uniformly sampled in $\mathbb{S}$ and with normal error distribution $\mathcal{N}(0, (\sigma q)^2)$ has a security parameter λ.

EncryptLWE$_s(\mu)$

The TLWE encryption of $\mu \in \mathbb{T}_q$ with the key $s \in \mathbb{S}^n$ is defined as

$$c \leftarrow \mathrm{TLWE}_s(\mu) = (a_1, \cdots, a_n, b) \in \mathbb{T}_q^{n+1}$$

where

$$\begin{cases} a = (a_1, \cdots, a_n) \overset{\$}{\leftarrow} \mathbb{T}_q^n \\ e \leftarrow \mathcal{D}_{\mathbb{T}_q,\sigma} \\ b = a \cdot s + \mu + e. \end{cases}$$

We shall use the notation $\mathrm{Err}(c) := e$.

DecryptLWE$_s(c, p_i)$

To decrypt $c = (a_1, \cdots, a_n, b) \in \mathbb{T}_q^{n+1}$ in the plaintext space $\mathbb{T}_{p_i}$ with the key $s \in \mathbb{S}^n$, return

$$\pi_{p_i}(b - a \cdot s) \in \mathbb{T}_{p_i}.$$

Similarly, let us introduce the RLWE encryption. We define $(q, N, \sigma, \mathbb{S}_N[X])$ such that the RLWE problem on $\mathbb{Z}_{N,q}(X)$, with keys uniformly sampled in $\mathbb{S}_N(X)$ and with normal error distribution $\mathcal{N}(0, (\sigma q)^2)$ has a security parameter λ.

EncryptRLWE$_s(\mu)$

The TRLWE encryption of $\mu \in \mathbb{T}_{N,q}[X]$ with the key $s \in (\mathbb{S}_N[X])^{\mathbf{k}}$ is defined as

$$c \leftarrow \mathrm{TRLWE}_s(\mu) = (a_1(X), \ldots, a_{\mathbf{k}}(X), b(X)) \in \mathbb{T}_{N,q}[X]^{\mathbf{k+1}}$$

where

$$\begin{cases} a_j(X) \overset{\$}{\leftarrow} \mathbb{T}_{N,q}[X], \quad j = 1, \ldots, k \\ e(X) \leftarrow \mathcal{D}_{\mathbb{T}_{N,q},\sigma} \\ b(X) = \sum_{j=1}^{\mathbf{k}} a_j(X) s_j(X) + \mu(X) + e(X). \end{cases}$$

We shall also use the notation $\mathrm{Err}(c) := e$ in this context of polynomial ciphertexts.

DecryptRLWE$_s(c, p_i)$

To decrypt $c = (a_1(X), \ldots, a_{\mathbf{k}}(X), b(X)) \in \mathbb{T}_{N,q}[X]^{\mathbf{k+1}}$ in the plaintext space $\mathbb{T}_{N,p_i}[x]$ with the key $s \in (\mathbb{S}_N[X])^{\mathbf{k}}$, return

$$\pi_{p_i}\left(b(X) - \sum_{j=1}^{\mathbf{k}} a_j(X) s_j(X)\right) \in \mathbb{T}_{N,p_i}[x].$$

EncryptRGSW$_s(m)$

Given positive integers B, ℓ, the TRGSW encryption of $m \in \mathbb{Z}_{N,q}[X]$ with the key $s \in (\mathbb{S}_N[X])^{\mathbf{k}}$ is defined as

$$C \leftarrow \mathrm{TRGSW}_s(m) = Z + mH_{B,\ell} \in \mathcal{M}_{(\mathbf{k+1})\ell, \mathbf{k+1}}(\mathbb{T}_{N,q}[X])$$

where the $(\mathbf{k}+1)\ell$ rows of Z are occurrences of $\mathrm{TRLWE}_s(0)$ and $H_{B,\ell}$ is the gadget matrix

$$H_{B,\ell} = I_{\mathbf{k}+1} \otimes g \quad \text{with} \quad g = (B^{-1}, \ldots, B^{-\ell})^T.$$

2.3 Encoding and encrypting integers

The core of our encryption is the following private-key LWE encryption/decryption scheme, with the bit-precision q.

Encode(m)

The encoding of a message $m \in \mathcal{M}$ is

$$\boldsymbol{\mu} = (m_1/p_1, \cdots, m_\kappa/p_\kappa) \in \mathbb{T}_{p_1} \times \cdots \mathbb{T}_{p_\kappa},$$

with $m_i = m \bmod p_i$ for $i = 1, \cdots, \kappa$.

Encrypt$_\mathbf{s}$(μ)

The encryption of $\boldsymbol{\mu} \in \mathbb{T}_{p_1} \times \cdots \mathbb{T}_{p_\kappa}$ is the vector $\mathbf{c} = (c_1, \cdots, c_\kappa) \in (\mathbb{T}_q^{n+1})^\kappa$, where $c_i = \mathrm{EncryptTLWE}_\mathbf{s}(\pi_q(\mu_i))$ for $i = 1, \cdots, \kappa$.

Decrypt$_\mathbf{s}$(c)

The decryption of $\mathbf{c} = (c_1, \cdots, c_\kappa) \in (\mathbb{T}_q^{n+1})^\kappa$ is the vector $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_\kappa) \in \mathbb{T}_{p_1} \times \cdots \mathbb{T}_{p_\kappa}$, where $\mu_i = \mathrm{DecryptTLWE}_\mathbf{s}(c_i, p_i)$ for $i = 1, \cdots, \kappa$.

Decode($\boldsymbol{\mu}$)

The vector $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_\kappa) \in \mathbb{T}_{p_1} \times \cdots \mathbb{T}_{p_\kappa}$ is decoded into

$$m = \left(p \sum_{i=1}^{\kappa} v_i \cdot \mu_i \right) \bmod p.$$

2.4 Homomorphic operations on ciphertexts

2.4.1 Homomorphic addition

We denote unambiguously by $\oplus$ the addition of both T(R)LWE-ciphertexts and TRGSW-ciphertexts. We recall that, if c_1 and c_2 are two T(R)LWE-ciphertexts, i.e.

$$c_1 = \mathrm{T(R)LWE}_s(\mu_1) \text{ and } c_2 = \mathrm{T(R)LWE}_s(\mu_2),$$

then

$$c_1 \oplus c_2 = \mathrm{T(R)LWE}_s(\mu_1 + \mu_2)$$

where the equality means that

$$\mathrm{Decrypt(R)LWE}_s(c_1 \oplus c_2, p_i) = \mu_1 + \mu_2.$$

Similarly, if C_1 and C_2 are two TRGSW-ciphertexts, i.e.

$$C_1 = \mathrm{TRGSW}_s(m_1) \text{ and } C_2 = \mathrm{TRGSW}_s(m_2)$$

then

$$C_1 \oplus C_2 = \mathrm{TRGSW}_s(m_1 + m_2)$$

where the equality means that both sides are (possibly different) encryptions of $m_1 + m_2$.

2.4.2 Homomorphic modular product

We recall that the $\mathbb{Z}_{N,q}[X]$-module $\mathbb{T}_{N,q}[X]$ is by definition endowed with a *modular* product $\cdot$ whose counterpart on TRGSW-ciphertexts is the co-called *external* product $\boxdot$. Besides, if

$$C = \mathrm{TRGSW}_s(m) \text{ and } c = \mathrm{TRLWE}_s(\mu),$$

then

$$C \boxdot c = \mathrm{TRLWE}_s(m \cdot \mu)$$

in the sense that

$$\mathrm{DecryptRLWE}_s(C \boxdot c) = m \cdot \mu.$$

The effective external product of C and c is obtained through the vector-matrix multiplication

$$C \boxdot c = \mathrm{dec}_{B,\ell}(c)\ C$$

where $\mathrm{dec}_{B,\ell}(c) = (\mathrm{dec}_{B,\ell}(a_1(X)), \ldots, \mathrm{dec}_{B,\ell}(a_{\mathbf{k}}(X)), \mathrm{dec}_{B,\ell}(b(X)))$

$$\mathrm{dec}_{B,\ell}(a_j(X)) = \left(\sum_{r=0}^{N-1} \mathrm{dec}_{B,\ell}(a_{j,r})_1 X^r, \ldots, \sum_{r=0}^{N-1} \mathrm{dec}_{B,\ell}(a_{j,r})_\ell X^r \right)$$

with for all $x \in \mathbb{T} \equiv [-\frac{1}{2}, \frac{1}{2}[$

$$x = \sum_{t=1}^{\ell} \mathrm{dec}_{B,\ell}(x)_t B^{-t} - \delta(x), \quad \mathrm{dec}_{B,\ell}(x)_t \in \{-B/2, \ldots, B/2\}, \quad |\delta(x)| \leq \frac{B^{-\ell}}{2}.$$

3 Bootstrapping in $\mathbb{T}_{p_i}$

In the same line as most FHE libraries, RHE protocol relies on hard lattice problems: in order to ensure the security of encryption, a noise is added in the ciphertext. During the course of arithmetic operations, this noise accumulates up to a level where it overwhelms the message itself and prevents a correct decryption. In order to avoid this deadlock, an *ad-hoc* procedure, introduced in 2009 by Gentry [7] and named *bootstrapping*, has been specially designed to reduce the noise to an acceptable level, thus permitting further computations. Generally speaking, Gentry's concept consists in a sequence of homomorphic operations that emulate the decryption procedure. However, in our context the rounding operation inherent to the decryption formula

$$\mu' = \frac{\lfloor p_i(b - a \cdot s) \rceil}{p_i} \bmod 1, \tag{3.1}$$

(which coincides with $\mu \in \mathbb{T}_{p_i}$ as long as $|e| < \frac{1}{2p_i}$) prevents a direct implementation of the Gentry's strategy starting from this formula.

In order to remedy this problem, Ducas and Micciancio [6], and later on in a faster version, Chillotti et al. [4, 5], have introduced a very efficient bootstrapping based on the use of polynomials. In the rest of this section, we present the extension of this procedure to the case of messages in the discrete torus $\mathbb{T}_{p_i}$. In order to keep the rounding errors as small as possible, we resort to a simple technique (referred to as *collapsing*) which we incorporate into the original procedure of [4].

3.1 The construction for cleartexts

Consider, for any $v(X) = \sum_{j=0}^{N-1} v_j X^j$ in the $\mathbb{Z}_{N,q}[X]$-modulus $\mathbb{T}_{N,q}[X]$, the function

$$\begin{array}{rccl} f_v : & \mathbb{Z} & \to & \mathbb{T}_q \\ & j & \mapsto & f_v(j) = (-1)^{\lceil \frac{j}{N} \rceil} v_{N\lceil \frac{j}{N}\rceil - j} \end{array}$$

It can be easily checked that $f_v(-j) = v_j$ and that $f_v(j)$ is simply the constant coefficient of the modular product $X^j \cdot v(X)$ for $0 \leq j \leq N-1$. Now, the underlying idea of the bootstrapping procedure of [6, 4] consists in the homomorphic implementation of the function

$$\mu \in \mathbb{T}_q \mapsto f_v(\lfloor 2N\mu \rceil) = \mathrm{coeff}_0 \left(X^{\lfloor 2N\mu \rceil} \cdot v(X) \right)$$

where coeff_0 selects the constant term of a polynomial and where the so-called *test-polynomial* v is chosen so as so ensure that $\mu \mapsto f_v(\lfloor 2N\mu \rceil)$ is almost the identity function on $\mathbb{T}_{p_i}$ in the sense that

$$\forall \mu \in \mathbb{T}_{p_i}, \quad f_v\left(\lfloor 2N(\mu + e) \rceil\right) = \mu \tag{3.2}$$

for small enough e.

Remark 3.1 *One could replace the right-hand side of (3.2) by $g(\mu)$ for any function g from $\mathbb{T}_{p_i}$ to itself, provided*

$$\forall (\mu, \nu) \in \mathbb{T}_{p_i}^2, \quad \min_{k \in \mathbb{Z}} \left| \mu - \nu + \frac{k}{2} \right| > 0,$$

a condition satisfied e.g. for odd p_i's.

Formula (3.2) for $e = 0$ is generally not enough *per se* to define the coefficients of v. In order to increase the tolerance with respect to errors (i.e. to enforce (3.2) for values of e as large as possible), we choose to define the v_j's as follows

$$\forall j = 0, \ldots, N-1, \qquad v_j = (-1)^{\lfloor \frac{jp}{N} \rceil} \frac{\lfloor \frac{jp}{N} \rceil}{2p} - \frac{1}{2}\left(1 - (-1)^{\lfloor \frac{jp}{N} \rceil}\right). \tag{3.3}$$

Note that Formula (3.3) differs significantly from the corresponding expression in p. 681 of [9] and softens the bootstrapability condition.

3.2 The construction for ciphertexts

We now translate the preceding procedure in terms of operations on the ciphertext (a, b), which encrypts $\pi_q(\mu)$ for some $\mu \in \mathbb{T}_{p_i}$ with a key s. Note that, in practice, it may have been obtained from a keyswitch operation from another key $\hat{s}$ to this key s. We emphasize that the collapsing technique described below is primarily aimed at reducing the rounding errors, as compared to the original formulation in [4]. As observed in [8], it may also have the side-effect to reduce the computational cost of the bootstrap. The components s_j of the secret key s are assumed to be taken randomly from the set[6] $\mathbb{S}$ of integers with cardinal K possibly greater or equal than 3. As aforementioned, we group these components in

[6] Note that nothing prevents us from considering s' and s in different key sets.

vectors of m elements (collapsing). For the sake of simplicity, we suppose here that m divides n and denote, on the one hand

$$\tilde{s}_k = \big(s_{m(k-1)+1}, s_{m(k-1)+2}, \ldots, s_{mk}\big) \in \mathbb{S}^m, \quad k = 1, \ldots, n/m, \tag{3.4}$$

and on the other hand

$$\tilde{a}_k = \big(a_{m(k-1)+1}, a_{m(k-1)+2}, \ldots, a_{mk}\big) \in \mathbb{T}_q^m, \quad k = 1, \ldots, n/m. \tag{3.5}$$

Using the notation $\cdot$ for the Euclidean scalar product on both $\mathbb{R}^n$ and $\mathbb{R}^m$, we then have

$$\mu + e = b - a \cdot s = b - \sum_{j=1}^{n/m} \tilde{a}_j \cdot \tilde{s}_j.$$

It is important to note at this stage that we round partial sums $\tilde{a}_k \cdot \tilde{s}_k$ and not individual products $a_k s_k$ as it is customary [6, 4, 5]. This leads to the (improved) approximation

$$\lfloor 2N(\mu + e)\rceil \approx -\sum_{k=1}^{n/m} \sum_{\tilde{\jmath} \in \mathbb{S}^m} \delta_{\tilde{\jmath}, \tilde{s}_k} \bar{a}_{k,\tilde{\jmath}} = -\sum_{k=1}^{n/m} \bar{a}_{k,\tilde{s}_k} =: \alpha, \tag{3.6}$$

where we denote $\delta_{\tilde{\imath},\tilde{\jmath}}$, for $(\tilde{\imath}, \tilde{\jmath}) \in \mathbb{S}^m \times \mathbb{S}^m$, the symbol with value 1 if $\tilde{\imath} = \tilde{\jmath}$ and 0 otherwise, and where

$$\bar{a}_{1,\tilde{\jmath}} = \lfloor 2N\tilde{a}_1 \cdot \tilde{\jmath} - 2Nb\rceil, \qquad \bar{a}_{k,\tilde{\jmath}} = \lfloor 2N\tilde{a}_k \cdot \tilde{\jmath}\rceil \text{ for } k = 2, \ldots, n/m \text{ and } \tilde{\jmath} \in \mathbb{S}^m.$$

Note that the sum in (3.6) is valid for all m dividing n, in particular for $m = 1$, where we recover the usual expression, as seen in [4] for instance, or for $m = n$, where the two sides of the equation then becomes rigorously equal. We finally observe that

$$X^\alpha = \prod_{k=1}^{n/m} X^{-\bar{a}_{k,\tilde{s}_k}} = \prod_{k=1}^{n/m} \sum_{\tilde{\jmath} \in \mathbb{S}^m} \delta_{\tilde{\jmath},\tilde{s}'_k} X^{-\bar{a}_{k,\tilde{\jmath}}} = \prod_{k=1}^{n/m} H_k(X) \tag{3.7}$$

with

$$H_k(X) = \sum_{\tilde{\jmath} \in \mathbb{S}^m} \delta_{\tilde{\jmath},\tilde{s}_k} X^{-\bar{a}_{k,\tilde{\jmath}}} \in \mathbb{Z}_N[X], \quad k = 1, \ldots, n/m, \tag{3.8}$$

so that $X^\alpha \cdot v(X)$ can be computed as the result of n/m successive modular products $\mathbb{Z}_N[X] \cdot \mathbb{T}_N[X]$ applied from the right to the left

$$X^\alpha \cdot v(X) = H_{n/m}(X) \ldots (H_2(X) \cdot (H_1(X) \cdot v(X))) \ldots) \tag{3.9}$$

The complete bootstrap procedure then involves two steps:

1. a blind rotate operation that computes a TRLWE$_{\hat{s}}$-encryption of $X^\alpha \cdot v(X)$ for an appropriate $s(X)$;
2. an extract operation which computes a TLWE$_{\hat{s}}$-encryption of the constant term of $X^\alpha \cdot v(X)$, which is the final output of the bootstrap[7].

The last step is completely standard and does not differ in our implementation from other works [6, 4, 5]. As a consequence, we describe only the first step, namely the blind rotate operation.

[7] The final TLWE key $\hat{s} \in \mathbb{S}^{\mathbf{kN}}$ is a vectorial version of the polynomial key $\hat{s}(X) \in \mathbb{S}_N[X]^{\mathbf{k}}$.

3.3 Blind rotation

Given $\mathrm{TRGSW}_{\hat{s}}$-encryptions of the $\delta_{\tilde{\jmath},\tilde{s}_k}$, it is straightforward to compute homomorphically the $\mathrm{TRGSW}_{\hat{s}}$-encryptions of the $H_k(X), k = 1, \dots, n/m$, according to Formula (3.8) and

$$\mathrm{TRGSW}_{\hat{s}}(H_k) = \bigoplus_{\tilde{\jmath}\in\mathbb{S}^m} \Big(X^{-\bar{a}_{k,\tilde{\jmath}}} \cdot \mathrm{TRGSW}_{\hat{s}}(\delta_{\tilde{\jmath},\tilde{s}_k}) \Big), \tag{3.10}$$

where the $\cdot$ symbol stands for the term-by-term homomorphic modular product. The $\mathrm{TRLWE}_{\hat{s}}$-encrypted value $X^{\alpha} \cdot v(X)$ can now be homomorphically computed in agreement with Formula (3.9) as

$$\mathrm{TRGSW}_{\hat{s}}(H_{n/m}) \boxdot (\dots(\mathrm{TRGSW}_{\hat{s}}(H_1) \boxdot \mathrm{TRLWE}_{\hat{s}}(v))\dots). \tag{3.11}$$

Note that the $\mathrm{TRLWE}_{\hat{s}}(v)$ is taken as the trivial noise-free zero-mask $(0, \dots, 0, v(X))$.

4 Error estimate of the bootstrap output

Prior to the statement of the main result of this section, we introduce two notations. For all polynomial P, we denote by $\mathrm{coeff}_i(P)$ its i-th coefficient, and for all integer B, we denote $\xi_B = (3 \cdot (-1)^B + 1)/2$ (i.e. $\xi_B = 2$ if B is even and $\xi_B = -1$ if B is odd). We are now in position to state the following

Proposition 4.1 *Assume that $s \in \mathbb{S}^n$ with $K = |\mathbb{S}|$, consider an integer $1 \le m \le n$ dividing n and denote by $BK = (BK_{i,\tilde{\jmath}})_{1\le i\le n/m,\, \tilde{\jmath}\in\mathbb{S}^m}$, the bootstrap key defined as*

$$BK_{i,\tilde{\jmath}} = TRGSW_{\hat{s}}(\delta_{\tilde{\jmath},\tilde{s}_i}), \qquad 1 \le i \le n/m, \quad \tilde{\jmath} \in \mathbb{S}^m$$

where all the errors have been sampled according to a discrete normal distribution with the same standard deviation σ_{BK} and where

$$\tilde{s}_i = \left(s_{m(i-1)+1}, s_{m(i-1)+2}, \dots, s_{mi}\right), \quad 1 \le i \le n/m, \quad \tilde{\jmath} \in \mathbb{S}^m.$$

Then (under independence assumptions precised in the proof) the following estimate for the error $\mathcal{E}_{Boot}$ of the bootstrap output holds:

$$Var(coeff_i(\mathcal{E}^{Boot})) = \frac{n}{m}\left(1 + \sum_{j=1}^{\mathbf{k}} \|\hat{s}_j\|_2^2\right)\frac{B^{-2\ell}}{12} + \frac{n}{m}(\mathbf{k}+1)\ell N \frac{B^2+\xi_B}{12} K^m \sigma_{BK}^2,$$

for $i = 0, \cdots, N-1$.

Remark 4.2 *If we regard the key $\hat{s}$ as a random variable, and not as a vector of fixed polynomials, the same proof leads to*

$$Var(coeff_i(\mathcal{E}^{Boot})) = \frac{n}{m}\left(1 + \sum_{j=1}^{\mathbf{k}} \mathbb{E}(\|\hat{s}_j\|_2^2)\right)\frac{B^{-2\ell}}{12} + \frac{n}{m}(\mathbf{k}+1)\ell N \frac{B^2+\xi_B}{12} K^m \sigma_{BK}^2.$$

Proof. It is easy to show that the error of an external product of a TRGSW$_{\hat{s}}$-ciphertext by a TRLWE$_{\hat{s}}$-ciphertext c satisfies the following equality

$$\mathrm{Err}(\mathrm{TRGSW}_{\hat{s}}(\nu) \boxdot c) = \nu \mathrm{Err}(c) - \nu \varphi_{\hat{s}}(\delta) + \varphi_{\hat{s}}\left(\mathrm{dec}_{B,\ell}(c) Z\right),$$

where the so-called *phase*-function $\varphi_{\hat{s}}$ is defined for $c = (a, b) \in \mathbb{T}_{N,q}[X]^{\mathbf{k}+1}$ as

$$\varphi_{\hat{s}}(c)(X) = b(X) - \sum_{j=1}^{\mathbf{k}} a_j(X) \hat{s}_j(X),$$

where

$$\delta = c - \mathrm{dec}_{B,\ell}(c) H_{B,\ell}, \qquad \|\delta\|_\infty \leq B^{-\ell}/2,$$

and where we have denoted

$$\mathrm{TRGSW}_{\hat{s}}(\nu) = Z + \nu H_B^\ell \in \mathcal{M}_{(\mathbf{k}+\mathbf{1})\ell, \mathbf{k}+\mathbf{1}}. \tag{4.1}$$

We also define the error associated to this TRGSW ciphertext as the vector

$$\mathrm{Err}(\mathrm{TRGSW}_{\hat{s}}(\nu)) = Z(-s(X), 1)^T.$$

Now, denote $c^{(0)} = \mathrm{TRLWE}_{\hat{s}}(v(X))$ and, for $k = 1, \ldots, n/m$, we set

$$c^{(k)} = \mathrm{TRGSW}_{\hat{s}}(H_k) \boxdot c^{(k-1)},$$

where

$$\mathrm{TRGSW}_{\hat{s}}(H_k) = Z^{(k)} + \nu_k H_{B,\ell} = \bigoplus_{\tilde{\jmath} \in \mathbb{S}^m} \left(X^{-\bar{a}_{k,\tilde{\jmath}}} \cdot BK_{k,\tilde{\jmath}} \right)$$

is a TRGSW$_{\hat{s}}$-encryption of $\nu_k := X^{-\bar{a}_{k,\bar{s}_k}}$. If we set

$$\delta^{(k)} = c^{(k)} - \mathrm{dec}_{B,\ell}(c^{(k)}) H_{B,\ell}, \qquad k = 1, \ldots, n/m,$$

we obtain the following recurrence relation

$$\begin{aligned}
\mathrm{Err}(c^{(k)}) &= \nu_k \mathrm{Err}(c^{(k-1)}) - \nu_k \varphi_{\hat{s}}\left(\delta^{(k-1)}\right) + \varphi_{\hat{s}}\left(\mathrm{dec}_{B,\ell}(c^{(k-1)}) Z^{(k-1)}\right) \\
&= \nu_k \mathrm{Err}(c^{(k-1)}) - \nu_k \left(\delta_{\mathbf{k}+1}^{(k-1)} - \sum_{j=1}^{\mathbf{k}} \delta_j^{(k-1)} \hat{s}_j \right) \\
&\quad + \sum_{r=1}^{(\mathbf{k}+1)\ell} \mathrm{dec}_{B,\ell}(c^{(k-1)})_r \sum_{\tilde{\jmath} \in \mathbb{S}^m} X^{-\bar{a}_{k,\tilde{\jmath}}} \mathrm{Err}(BK_{k-1,\tilde{\jmath}})_r
\end{aligned}$$

where $Z_r^{(k-1)}$ is the r-th line of the matrix $Z^{(k-1)}$ (it is a TRLWE-encryption of 0).

Our main assumption is now that, with a good approximation, the random variables $\mathrm{Err}(c^{(k-1)})$, $\delta^{(k-1)}$, $\mathrm{dec}_{B,\ell}(c^{(k-1)})$ and $\mathrm{Err}(BK_{k,\tilde{\jmath}})$ are pairwise independent. Note that each polynomial in these random variables has centered, pairwise independent coefficients, with a uniform variance. Of course, the assumption of independence seems to be strong;

nevertheless, numerical experiments have shown that possible correlations between these random variable do not significantly affect the result.

Furthermore, since q is very large, it is difficult distinguish a uniform sampling in $\mathbb{T}_q$ from a uniform sampling in $(-1/2, 1/2)$. Hence, with a good approximation again, one can infer from Lemma A.1 in the Appendix that, for all $0 \leq i \leq N-1$,

$$\mathrm{Var}(\mathrm{coeff}_i(\mathrm{dec}_{B,\ell}(c^{(k-1)})_r)) = \frac{B^2 + \xi_B}{12}, \qquad \mathrm{Var}(\mathrm{coeff}_i(\delta_j^{(k-1)})) = \frac{B^{-2\ell}}{12}.$$

Since $\mathrm{Var}(\mathrm{Err}(BK_{k-1,\tilde{\jmath}})_r) = \sigma_{BK}^2$, we get from Lemma A.2 in the Appendix and from our assumptions that

$$\mathrm{Var}(\mathrm{Err}(c^{(k)}) = \mathrm{Var}(\mathrm{Err}(c^{(k-1)}) + \frac{B^{-2\ell}}{12}\left(1 + \sum_{j=1}^{\mathbf{k}} \|\hat{s}_j\|_2^2\right) + (\mathbf{k}+1)\ell N \frac{B^2 + \xi_B}{12} K^m \sigma_{BK}^2.$$

Taking into account that $\mathrm{Err}(c^{(0)}) = 0$, a direct induction leads to the result.

□

5 Conditions of correctness of the bootstrap

In this paragraph, we state some deterministic conditions for the bootstrapping operation to be correct. We then estimate the probability that these conditions are satisfied in practice. For the sake of clarity of the presentation, we assimilate in the beginning of this section (until Section 5.2 where the situation encountered in practice $q = 2^{64}$ is considered) the discrete torus $\mathbb{T}_q$ with the continuous one $\mathbb{T}$. By doing so, we neglect the influence of errors produced by rounding elements of $\mathbb{T}$ to elements of $\mathbb{T}_q$, which can be bounded by $1/(2q)$ and are thus very small in practice. In order to give a precise definition of correctness, we thus assume that the ciphertext $c = (a, b) \in \mathbb{T}^{n+1}$ encrypts a value $\mu \in \mathbb{T}_{p_i}$, that is to say that

$$b - a \cdot s = \mu + e, \quad \mu \in \mathbb{T}_{p_i},$$

where we assume that the error e is a normally distributed random variable in $\mathbb{T}$ with standard deviation σ_e.

Definition 5.1 *Let n be the number of components of the secret key $s \in \mathbb{S}$, $n/m \in \mathbb{N}$ be the number of vectors $\tilde{s}_k$ of length m as defined in (3.4) and N be the degree of polynomials considered in* TRLWE *and* TRGSW *ciphertexts. Let us suppose that*

$$w_2(\mathbb{T}_{p_i}) := \min_{\mu \neq \nu \in \mathbb{T}_{p_i}^2} \min_{k \in \mathbb{Z}} \left|\mu - \nu - \frac{k}{2}\right| > \frac{1}{2N}. \tag{5.1}$$

The bootstrapping procedure is said to be correct with probability $\mathbb{P}_{corr}$ if and only if for every ciphertext $(a, b) \in \mathbb{T}^{n+1}$ with

$$b = a \cdot s + \mu + e$$

for some $\mu \in \mathbb{T}_{p_i}$, we have

$$\mathbb{P}\Big(-\sum_{k=1}^{n/m} \bar{a}_{k,\tilde{s}_k} \in I_\mu \Big) = \mathbb{P}_{corr} \tag{5.2}$$

where

$$I_\mu = \left[2N\left(\mu - \frac{1}{2} w_2(\mathbb{T}_{p_i})\right), 2N\left(\mu + \frac{1}{2} w_2(\mathbb{T}_{p_i})\right) \right[\cap \mathbb{Z} + 2N\mathbb{Z}.$$

Condition (5.1) ensures that it is possible to construct a suitable polynomial $v(X)$ for the bootstrapping procedure (in particular, the sets I_μ and $N + I_\mu$ for $\mu \in \mathbb{T}_{p_i}$ are non-empty and do not intersect with each other). Note that whenever

$$-\sum_{k=1}^{n/m} \bar{a}_{k,\tilde{s}_k} \in I_\mu$$

the constant coefficient of $X^\alpha \cdot v(X)$ is precisely μ and the bootstrapping procedure delivers the correct result. The condition is sufficient, though not strictly necessary for all μ's. For instance, if $p_i = 5$ and $\mu = \frac{1}{5}$, a necessary and sufficient condition is that

$$-\sum_{k=1}^{n/m} \bar{a}_{k,\tilde{s}_k} \in \left[2N\left(\mu - \frac{1}{2} w_2(\mathbb{T}_{p_i})\right), 2N\left(\mu + \frac{3}{2} w_2(\mathbb{T}_{p_i})\right) \right[\cap \mathbb{Z} + 2N\mathbb{Z}$$

and the corresponding interval is strictly larger than I_μ for large enough N. Our aim is now to give lower bounds of the probability $\mathbb{P}_{corr}$. To state our main result, we need to recall and introduce a few notations. Given the ciphertext (a, b) whose n first components a_i have been grouped to form the $\tilde{a}_k \in \mathbb{T}^m$ as described in Section 3.2, let $\bar{a}_{k,\tilde{\jmath}} = \lfloor 2N(\tilde{a}_k \cdot \tilde{\jmath}) \rceil$. For $\tilde{\jmath} \in \mathbb{S}^m$ and $k = 2, \ldots, n/m$, we denote

$$X_{1,\tilde{\jmath}} = \bar{a}_{1,\tilde{\jmath}} - 2N\tilde{a}_1 \cdot \tilde{\jmath} + 2Nb \quad \text{and} \quad X_{k,\tilde{\jmath}} = \bar{a}_{k,\tilde{\jmath}} - 2N\tilde{a}_k \cdot \tilde{\jmath}.$$

Proposition 5.2 *Under the assumptions of Definition 5.1, the bootstrapping of*

$$c = (a_1, \ldots, a_n, \sum_{k=1}^{n} s_k\, a_k + \mu + e), \quad \mu \in \mathbb{T}_{p_i},$$

where $a \overset{\$}{\leftarrow} \mathbb{T}^n$ and where e is a normal random variable on $\mathbb{T}$ with standard deviation σ_e, is correct with probability $\mathbb{P}_{corr}$ bounded from below by

$$\mathbb{P}_{corr} \geq \frac{2}{\pi} \int_0^\infty \sin\big((t(\lambda^+ - \lambda^-)\big) \cos\big((t(\lambda^+ + \lambda^-)\big) \left(\operatorname{sinc}\left(\frac{t}{4N}\right)\right)^L \frac{\exp\left(-\sigma_e^2 t^2/2\right)}{t}\, dt \tag{5.3}$$

where $\lambda^\pm = \frac{\lceil 2N\mu \pm N w_2(\mathbb{T}_{p_i}) \rceil}{2N} - \frac{1}{4N} - \mu$ and L is the number of non-zero variables among the $(X_{k,\tilde{s}_k})_{k=2,\ldots,n/m}$.

Proof. By definition, the bootstrapping of c is correct if and only if $\left(-\sum_{k=1}^{n/m} \bar{a}_{k,\tilde{s}_k}\right) \in I_\mu$, that is to say, if and only if there exists $r \in \mathbb{Z}$ such that

$$\begin{aligned} 2N(\mu - \frac{1}{2}w_2(\mathbb{T}_{p_i})) &\leq -\left\lfloor \sum_{k=2}^{n/m} X_{k,\tilde{s}_k} - 2N\mu - 2Ne \right\rceil + 2rN \\ &< 2N(\mu + \frac{1}{2}w_2(\mathbb{T}_{p_i})). \end{aligned} \tag{5.4}$$

As a matter of fact, one has

$$\begin{aligned} \bar{a}_{1,\tilde{s}_1} &= \lfloor 2N\tilde{a}_1 \cdot \tilde{s}_1 - 2Nb \rceil \\ &= \left\lfloor 2N\tilde{a}_1 \cdot \tilde{s}_1 - 2N\mu - 2Ne - 2N\sum_{k=1}^{n/m} \tilde{a}_k \cdot \tilde{s}_k \right\rceil \\ &= \left\lfloor -2N\mu - 2Ne - 2N\sum_{k=2}^{n/m} \tilde{a}_k \cdot \tilde{s}_k \right\rceil \\ &= \left\lfloor -2N\mu - 2Ne - \sum_{k=2}^{n/m} \bar{a}_{k,\tilde{s}_k} + \sum_{k=2}^{n/m} \bar{X}_{k,\tilde{s}_k} \right\rceil \end{aligned}$$

so that

$$\sum_{k=1}^{n/m} \bar{a}_{k,\tilde{s}_k} = \left\lfloor -2N\mu - 2Ne + \sum_{k=2}^{n/m} X_{k,\tilde{s}_k} \right\rceil.$$

Now, for any $\alpha, \beta, \gamma \in \mathbb{R}^3$ and $k \in \mathbb{Z}$, we have, on the one hand

$$\alpha \leq k < \beta \text{ iff } \lceil \alpha \rceil \leq k \leq \lceil \beta \rceil - 1,$$

and on the other hand

$$-\lfloor -\gamma \rceil = k \text{ iff } k - \frac{1}{2} < \gamma \leq k + \frac{1}{2},$$

so that

$$\alpha \leq -\lfloor -\gamma \rceil < \beta \text{ iff } \lceil \alpha \rceil - \frac{1}{2} < \gamma \leq \lceil \beta \rceil - \frac{1}{2}.$$

Hence, condition (5.4) is equivalent to the existence of $r \in \mathbb{Z}$ such that

$$\begin{aligned} \left\lceil 2N(\mu - \frac{1}{2}w_2(\mathbb{T}_{p_i})) \right\rceil - \frac{1}{2} &< -\sum_{k=2}^{n/m} X_{k,\tilde{\mathbf{s}}_k} + 2N\mu + 2Ne + 2rN \\ &\leq \left\lceil 2N(\mu - \frac{1}{2}w_2(\mathbb{T}_{p_i})) \right\rceil - \frac{1}{2}. \end{aligned}$$

We thus obtain the following necessary and sufficient condition

$$\exists r \in \mathbb{Z}, \quad \lambda^- < -\frac{1}{2N}\sum_{k=2}^{n/m} X_{k,\tilde{\mathbf{s}}_k} + e + r \leq \lambda^+$$

where $\lambda^{\pm} = \frac{\lceil 2N\mu \pm N w_2(\mathbb{T}_{p_i})\rceil}{2N} - \frac{1}{4N} - \mu$. We now regard the components $X_{k,\tilde{s}_k}$, for $k = 2, \dots, n/m$ with $\tilde{s}_k \neq 0$, as independent variables uniformly distributed in $[-\frac{1}{2}, \frac{1}{2}]$. We wish to find the probability that the bootstrap is correct, that is to say to compute

$$I := \mathbb{P}\left(\bigcup_{r \in \mathbb{Z}} \left\{ \lambda^- < -\frac{1}{2N} \sum_{k=2}^{n/m} X_{k,\tilde{s}_k} + e + r < \lambda^+ \right\} \right).$$

Denoting $(r_\ell)_{\ell=1,\dots,L}$ the indices of non-zero variables among the $(X_{k,\tilde{s}_k})_{k=2,\dots,n/m}$, $\lambda_r^{\pm} = \lambda^{\pm} - r$ and

$$\mathcal{X} = -\frac{1}{2N} \sum_{\ell=1}^{L} X_{r_\ell, \tilde{s}_{r_\ell}} + e,$$

we finally have that

$$I = \sum_{r \in \mathbb{Z}} \mathbb{P}\left(\lambda_r^- < \mathcal{X} < \lambda_r^+ \right). \tag{5.5}$$

Note indeed that the intervals $[\lambda_r^-, \lambda_r^+]$, $r \in \mathbb{Z}$ do not intersect as

$$|\lambda^+ - \lambda^-| \leq \max\left(w_2(\mathbb{T}_{p_i}), \frac{1}{N} - w_2(\mathbb{T}_{p_i}) \right) < 1.$$

Using the characteristic functions

$$\varphi(t) = \mathbb{E}(\exp(ite)) \quad \text{and} \quad \varphi_{\mathcal{X}}(t) = \mathbb{E}(\exp(it\mathcal{X}))$$

we have, owing to Gil-Pelaez theorem, that

$$\mathbb{P}(\mathcal{X} \leq x) = \frac{1}{2} - \frac{1}{\pi} \int_0^\infty \frac{\mathrm{Im}[e^{-itx} \varphi_{\mathcal{X}}(t)]}{t}\, dt,$$

where

$$\varphi_{\mathcal{X}}(t) = \left(\prod_{\ell=1}^{L} \mathbb{E}\left(\exp(-i \frac{t}{2N} X_{r_\ell, \tilde{s}_{r_\ell}}) \right) \right) \varphi(t) = \left(\prod_{\ell=1}^{L} \mathrm{sinc}(t/(4N)) \right) \varphi(t).$$

Then, taking into account the symmetry of e (and thus the fact that φ is real-valued and even)

$$\mathbb{P}(\mathcal{X} \leq x) = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \sin(tx) \left(\frac{\sin(t/(4N))}{t/(4N)} \right)^L \frac{\varphi(t)}{t}\, dt$$

so that (see Equation (5.5))

$$\begin{aligned} I &= \frac{2}{\pi} \sum_{r \in \mathbb{Z}} \int_0^\infty \sin\left((t(\lambda^+ - \lambda^-) \right) \cos\left((t(\lambda_r^+ + \lambda_r^-) \right) \left(\frac{\sin(t/(4N))}{t/(4N)} \right)^L \frac{\varphi(t)}{t}\, dt \\ &\geq \frac{2}{\pi} \int_0^\infty \sin\left((t(\lambda^+ - \lambda^-) \right) \cos\left((t(\lambda^+ + \lambda^-) \right) \left(\frac{\sin(t/(4N))}{t/(4N)} \right)^L \frac{\varphi(t)}{t}\, dt. \end{aligned}$$

□

Remark 5.3 *The term for $r = 0$ is the only one that matters in the sum. As a matter of fact, since*

$$e - \frac{L}{4N} \leq \mathcal{X} \leq e + \frac{L}{4N}$$

one has

$$\mathbb{P}\left(\lambda_r^- < \mathcal{X} < \lambda_r^+\right) \leq \mathbb{P}\left(\lambda_r^- - \frac{L}{4N} - r \leq e \leq \lambda_r^+ + \frac{L}{4N} - r\right)$$

Note that the intervals on the right do not intersect as soon as

$$\lambda^+ - \lambda^- + \frac{L}{2N} \leq 1$$

a condition satisfied if $w_2(\mathbb{T}_{p_i}) + \frac{L+1}{2N} \leq 1$ and which can be checked in all practical situations. We then have

$$\sum_{r\neq 0} \mathbb{P}\left(\lambda_r^- < \mathcal{X} < \lambda_r^+\right) \leq \mathbb{P}\left(|e| \geq 1 - \frac{w_2(\mathbb{T}_{p_i})}{2} - \frac{L+1}{4N}\right)$$

Assume now that e follows a Gaussian law with standard deviation σ_e, then

$$\sum_{r\neq 0} \mathbb{P}\left(\lambda_r^- < \mathcal{X} < \lambda_r^+\right) \leq 1 - \operatorname{erf}\left(\frac{\lambda}{\sqrt{2}\sigma_e}\right), \quad \textit{with} \quad \lambda = 1 - \frac{w_2(\mathbb{T}_{p_i})}{2} - \frac{L+1}{4N}$$

where the erf-function is the usual error function

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

Typical values for the parameters are (see section below) $N = 1024, L = 210, p_i = 7, \sigma_e = 0.0016$, which yield (knowing that here we have $w_2(\mathbb{T}_{p_i}) = 1/(2p_i)$)

$$\sum_{r\neq 0} \mathbb{P}\left(\lambda_r^- < \mathcal{X} < \lambda_r^+\right) \leq 1 - \operatorname{erf}(403) \approx 6.5103937 \times 10^{-70537}$$

which is an extremely small probability.

5.1 Comparison with the law of probability obtained by the central limit theorem

If, instead of the detailed computations undertaken in previous subsection, one regards the random variable $\mathcal{X}$ as a Gaussian variable with variance

$$\sigma_{\mathcal{X}}^2 = \sigma_e^2 + \frac{L}{48N^2},$$

as the central limit theorem predicts in the limit where n and L are large, then the law of probability of $\mathcal{X}$ is simply approximated by

$$\mathbb{P}(\mathcal{X} \leq \lambda) \approx \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{\lambda}{\sqrt{2}\sigma_{\mathcal{X}}}\right)\right).$$

As an indication of how this approximation by the central limit theorem is accurate, we plot on Figure 1 the curves corresponding to the two expressions obtained here and in previous subsection, respectively in reed and blue colors on the graph.

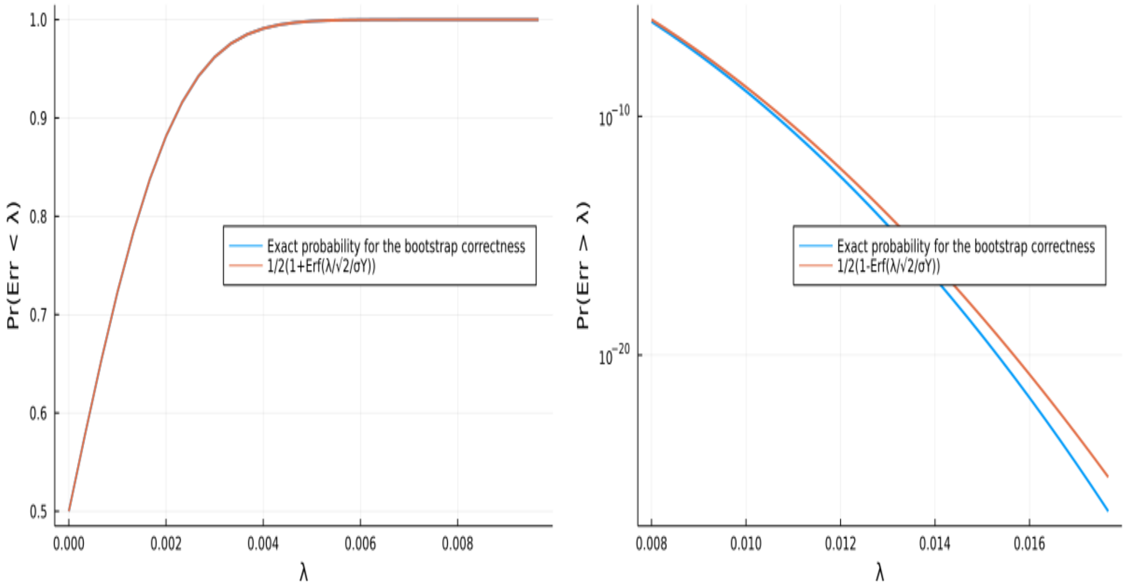

Figure 1: Probability of correctness given by characteristic function (blue) or by the central limit approximation (red).

It is clear that in most regimes, i.e. for probability of correctness that are not too stringent, the central limit approximation is enough. Below 10^{-12} one benefits from the better approximation obtained by characteristic functions. In next subsection, we will systematically use this improved approximation to calibrate our parameters, not only because it leads to more accurate results, but also because it is a *proven* lower bound, that the other approximation is not.

5.2 Calibration of parameters for the bootstrap

In this subsection, we use the lower bound obtained above to obtain various curves for the parameters p_i, m, N, σ_e. The number of non-zero values L is in average $\frac{n}{m}(1-K^{-m})$ for keys with $K = |\mathbb{S}|$ values and since the results are not very sensitive to the value of L in the neighbourhood of this value, we shall always consider

$$L = \frac{n}{m}(1-K^{-m}).$$

Only two instances of N are considered here, namely $N = 1024$ and $N = 2048$, and we shall take $n = 420$ and $m = 1,2,3,4,5,6,7$, corresponding to $L = 210, 158, 123, 98, 81, 69, 60$ and $K = 2$. To plot the curves of Figures 2 and 3, we have taken $p_i = 27$ and $\mu = 1/27$ for the previous list of values of m.

In order to design a secure system, it is known that σ_e should be greater than a certain threshold

$$\sigma_e \geq \sigma_{TH}$$

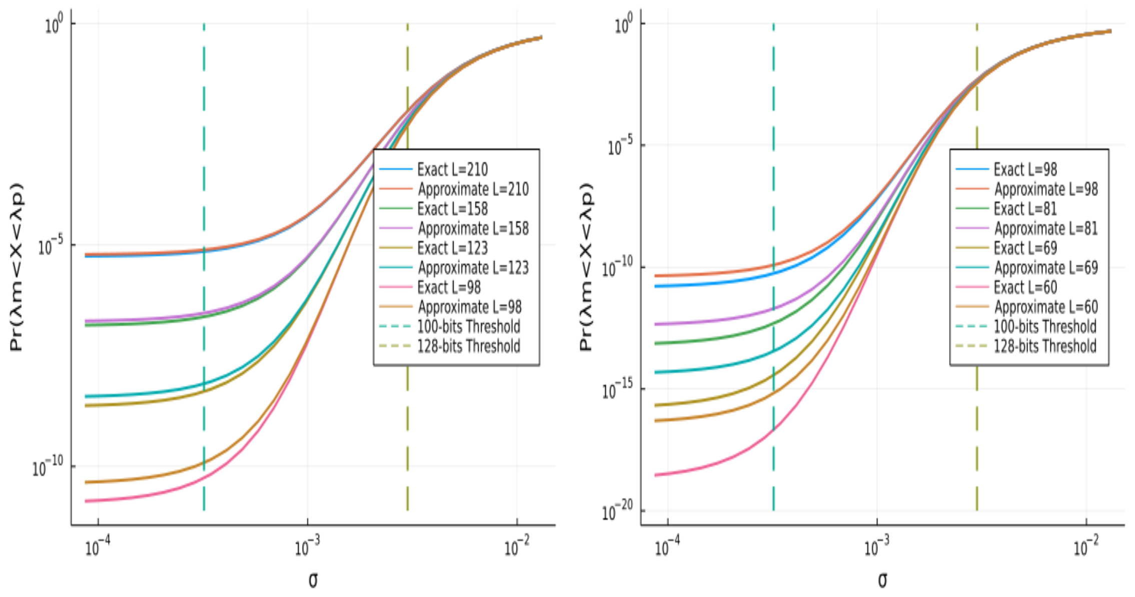

Figure 2: Probability of correctness given by characteristic function or by the central limit approximation in terms of σ_e. Values obtained with $N = 1024$ and m from 1 to 4 (left) and 4 to 7 (right).

where a practical value of σ_{TH} can be determined with the help of the lattice estimator designed by Martin R. Albrecht's team[8]. The values obtained by running this estimator with $n = 420$, $q = 2^{64}$ and $K = 2$ are reported below:

- for 100 bits of security: $\sigma_{TH} = 0.00032$;
- for 128 bits of security: $\sigma_{TH} = 0.003$.

The two thresholds appear as two vertical lines on all curves of Figures 2 and 3.

6 Implementation results and conclusion

It is clear that the larger m is, the smaller the rounding errors are and the higher the correctness probability becomes. With respect to accuracy, it is undoubtedly beneficial to increase m. However, the evolution of the computational cost is more contrasted, as the evaluation of the inner sums (3.10) becomes predominant for large m, as compared to the one of internal products (3.11). In order to appreciate the computational gain, we have implemented the complete bootstrapping procedure in `C++` and run the corresponding code for $1 \leq m \leq 10$. The value of the standard deviation of the gaussian noise entering in the TRLWE-encryption is evaluated through the lattice estimator [1] and set to $\sigma_{BK} = 2^{-41}$, a value which offers a definitely modest security but easily tractable of 80 bits for the TRLWE-encryption of the secret key $s \in \mathbb{S}_N[X]$ with $N = 1024$ used in (3.10). Similarly, for the purpose of this benchmark, we have chosen a secret key with 1024 components for

[8] https://lattice-estimator.readthedocs.io/en/latest/

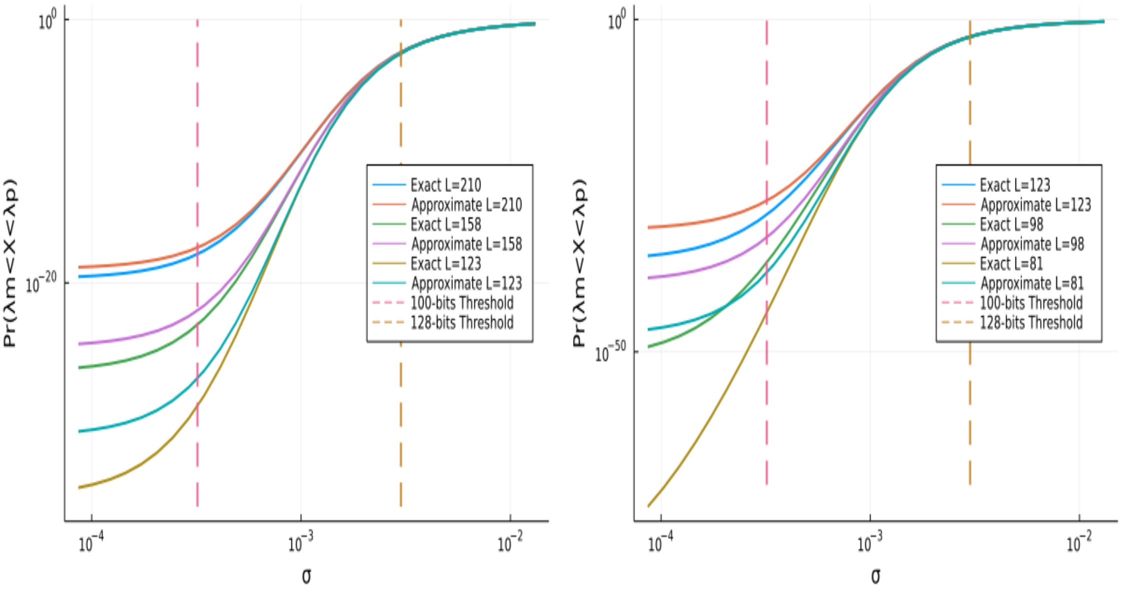

Figure 3: Probability of correctness given by characteristic function or by the central limit approximation in terms of σ_e. Values obtained with $N = 2048$ and m from 1 to 3 (left) and 4 to 6 (right).

the TLWE-encryption, given our (now standard) choice to switch the TLWE-ciphertext (a, b), prior to the bootstrap, to another one with a shorter key of size $n = 464$. Let us mention finally that the polynomial key is built so as to ensure that the TLWE-output of the bootstrap is encrypted with the same key as fresh TLWE-ciphertexts, thus avoiding another keyswitch.

We have reported in Table 4 the times required to compute the inner sums (3.10) as well as the times required for the blindrotate (3.11) and the complete bootstrap. All the results correspond to the following values of the parameters (see Proposition 4.1)

$$q = 2^{64}, \quad \max_i(p_i) = 27, \quad N = 1024, \quad n = 464, \quad B = 8192, \quad \ell = 2,$$

which have been chosen (together with ad-hoc parameters in the keyswitch) so as to ensure that at least 5000 additions can safely[9] be made between two bootstraps (multiplications are not considered in this estimate as their result is itself refreshed). It is apparent that a good trade-off is obtained for $m = 3$, a point where the computational time starts increasing again. This emphasizes that the computational gain remains relatively modest (note that the conclusion would be different on a multi-threaded machine as the sums (3.10) are completely independent and amenable to parallel computations). Nevertheless, as claimed in the abstract and introduction, the main interest of taking $m = 3, 4$ lies in the reduced rounding errors and the afferent possibility to consider larger values of p_i with the same correctness probability and a (marginally) smaller computational cost.

Finally, as it may objectively disputed that 80 bits of security is enough, we also include Figure 4 showing the relative cost (as compared to the one given in Table 4 for $m = 4$) for

[9] These values guarantee that the probability of an incorrect bootstrap remains below 10^{-9}.

m	1	2	3	4	5	6	7
CPU Time I	10	8	10	13	20	34	56
CPU Time II	10	5	3	2	2	2	1
Total CPU Time	20	13	13	15	22	36	57

Table 4: Second line: CPU times for the evaluation of the inner sums (3.10). Third line: CPU times for the evaluation of the external products (3.11). Fourth line: total CPU time. All times are given in milliseconds (ms).

80 bits, 100 bits and 128 bits of security. Note that these numbers are obtained under the same requirement that at least 5000 additions can be made between two bootstraps, that the correctness probability remains higher than $1-10^{-9}$ and with $N = 1024$ or $N = 2048$. Other parameters (n, m, B, ℓ) have been optimized for each point $(\max_i(p_i), N)$. The values in abscissa are those of $\max_i(p_i)$. Note that the curve in red has only one point as it is not possible to increase further than 3 the value of $\max_i(p_i)$ without noticeably compromising the correctness of the bootstrap.

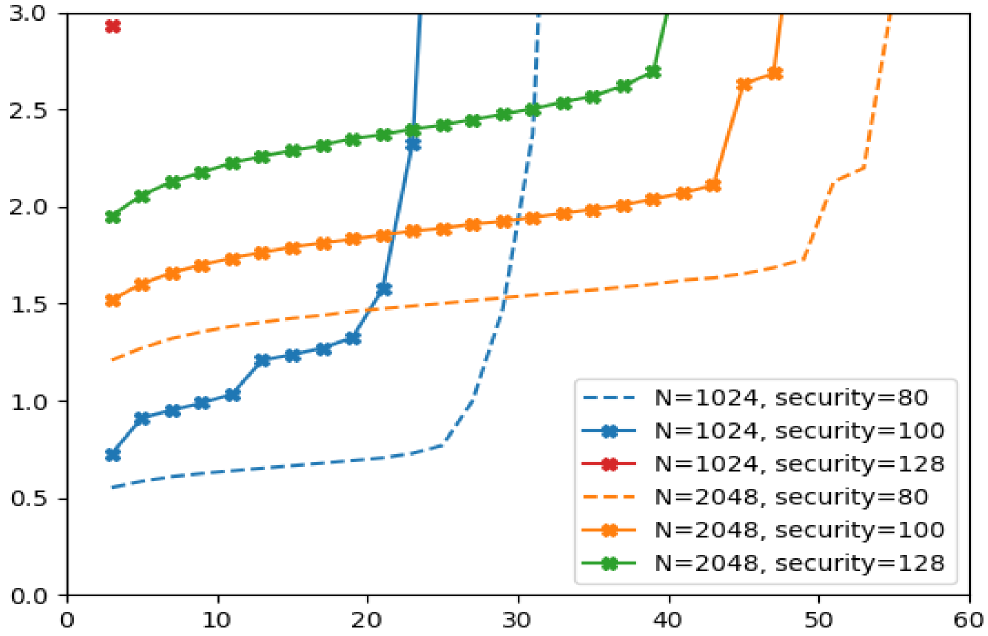

Figure 4: Relative computational cost of the bootstrap.

A Appendix

In this section, we state two elementary lemmas which are used in the proof of Proposition 4.1. We state the first one without explicit proof as it can be obtained by a simple induction.

Lemma A.1 *Let x be random variable uniformly distributed in the interval $(-1/2, 1/2)$ and let $(x_i)_{1\leq i\leq \ell}$ and δ_ℓ be the random variables obtained by decomposing x in the basis B according to the following algorithm:*

$$\widetilde{x}_1 = x, \qquad x_i = -\lfloor -B\widetilde{x}_i\rceil, \quad \widetilde{x}_{i+1} = B\widetilde{x}_i - x_i, \qquad \textit{for} \quad i = 1, \cdots \ell,$$

and

$$\delta_\ell = B^{-\ell}\widetilde{x}_{\ell+1}.$$

Then δ_ℓ is uniformly distributed in the interval $(-B^{-\ell}/2, B^{-\ell}/2)$ and the x_i's are discrete random variables with values in $\{-\lfloor B/2\rceil, -\lfloor B/2\rceil + 1, \cdots \lfloor B/2\rceil\}$ obeying the following law

$$\mathbb{P}(x_i = k) = \frac{1}{B}, \quad \textit{for} \quad -\lfloor B/2\rceil < k < \lfloor B/2\rceil,$$

$$\mathbb{P}(x_i = -\lfloor B/2\rceil) = \mathbb{P}(x_i = \lfloor B/2\rceil) = -\lfloor B/2\rceil/B + 1/2B + 1/2.$$

In particular

$$\mathit{Var}(x_i) = \frac{B^2 + \xi_B}{12} \quad \textit{with} \quad \xi_B = (3\cdot(-1)^B + 1)/2.$$

Lemma A.2 *Consider a random polynomial $P \in \mathbb{Z}_N[X]$ whose coefficients are pairwise independent, centered and have the same variance σ_P^2. Then, for all non-random polynomial $Q \in \mathbb{Z}_N[X]$ and all $i \in \{0, \ldots, N-1\}$, we have*

$$\mathbb{E}((PQ)_i) = 0, \qquad \mathit{Var}((PQ)_i) = \sigma_P^2 \, \|Q\|_2^2.$$

If Q is random and independent of P, we have

$$\mathbb{E}((PQ)_i) = 0, \qquad \mathit{Var}((PQ)_i) = \sigma_P^2 \, \mathbb{E}(\|Q\|_2^2).$$

Proof. The main argument of the proof stems from the periodicity of the coefficients of P and Q as elements of $\mathbb{Z}[X]/(X^N + 1)$. As a matter of fact, given $P(X) = \sum_{i=0}^{N-1} P_i X^i$ and $Q(X) = \sum_{i=0}^{N-1} Q_i X^i$, one has

$$(PQ)_i = \sum_{j=0}^{N-1} P_j Q_{i-j}.$$

As a consequence, if Q is non random and $\mathbb{E}(P_j) = 0$, we have

$$\mathbb{E}((PQ)_i) = \sum_{j=0}^{N-1} \mathbb{E}(P_j) Q_{i-j} = 0,$$

$$\begin{aligned}\mathbb{E}(((PQ)_i)^2) &= \mathbb{E}\left(\sum_{j,j'} P_j Q_{i-j} P_{j'} Q_{i-j'}\right)\\ &= \sum_j \mathbb{E}(P_j^2) Q_{i-j}^2 + \sum_{j\neq j'} \mathbb{E}(P_j)\mathbb{E}(P_{j'}) Q_{i-j} Q_{i-j'}\\ &= \sum_j \mathrm{Var}(P_i) Q_{i-j}^2 = \sigma_P^2 \|Q\|_2^2.\end{aligned}$$

If Q is random and is independent of P, we have again

$$\mathbb{E}((PQ)_i) = \sum_{j=0}^{N-1} \mathbb{E}(P_j)\mathbb{E}(Q_{i-j}) = 0,$$

$$\begin{aligned}\mathbb{E}(((PQ)_i)^2) &= \mathbb{E}\left(\sum_j \sum_{j'} P_j Q_{i-j} P_{j'} Q_{i-j'}\right)\\ &= \sum_j \mathbb{E}(P_j^2)\mathbb{E}(Q_{i-j}^2) + \sum_j \sum_{j'\neq j} \mathbb{E}(P_j)\mathbb{E}(P_{j'})\mathbb{E}(Q_{i-j} Q_{i-j'})\\ &= \sum_j \mathbb{E}(P_j^2)\mathbb{E}(Q_{i-j}^2) = \sigma_P^2\,\mathbb{E}(\|Q\|_2^2).\end{aligned}$$

□

Acknowledgments

We wish to acknowledge Etienne Jacob for the benchmark data of Table 4.

References

[1] Albrecht, M., Göpfert, F., Virdia, F., Wunderer, T., Revisiting the Expected Cost of Solving uSVP and Applications to LWE, Advances in Cryptology, ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, Dec. 3-7, 2017, Proceedings, Part I

[2] Chartier, P., Koskas, M., Lemou, M., Méhats, Method for homomorphically determining the sign of a message by dilation, associated methods and devices. Patent no WO2023242429 - 12/21/2023. Number and date of prority : FR2205957 - 17/06/2022.

[3] Chartier, P., Koskas, M., Lemou, M., Méhats, Homomorphic sign evaluation using Functional Bootstrapping with RNS representation of integers. Cryptology ePrint Archive.

[4] Chillotti, I., Gama, N., Georgieva, M., Izabachène, M. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In J. H. Cheon and T. Takagi, editors, *Advances in Cryptology. ASIACRYPT 2016, Part I*, volume 10031 of Lecture Notes in Computer Science, pages 3–33. Springer, 2016. doi:10.1007/978-3-662-53887-6_1

[5] Chillotti, I., Gama, N., Georgieva, M., Izabachène, M. TFHE: Fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(1):34–91, 2020. doi:10.1007/s00145-019-09319-x.

[6] Ducas, L., Micciancio, D. FHEW: Bootstrapping homomorphic encryption in less than a second. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology. EUROCRYPT 2015, Part I*, volume 9056 of Lecture Notes in Computer Science, pages 617–640. Springer, 2015. doi:10.1007/978-3-662-46800-5_24.

[7] Gentry, C. Fully homomorphic encryption using ideal lattices. In M. Mitzen- macher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178. ACM Press, 2009. doi:10.1145/1536414.1536440.

[8] Joye, M., Paillier, P. Blind rotation in fully homomorphic encryption with extended keys. In S. Dolev, J. Katz, and A. Meisels, editors, *Cyber Security Cryptography and Machine Learning (CSCML 2022)*, volume 13301 of Lecture Notes in Computer Science, pages 1–18. Springer, 2022. doi:10.1007/ 978-3-031-07689-3_1

[9] Joye, M. SoK: Fully Homomorphic Encryption over the [Discretized] Torus. IACR Transactions on Cryptographic Hardware and Embedded Systems, ISSN 2569-2925, Vol. 2022, No. 4, pp. 661–692. doi:10.46586/tches.v2022.i4.661-692

[10] Lee, Y., Micciancio, D., Kim, A., Choi, R., Deryabin, M., Eom, J. Efficient FHEW Bootstrapping with Small Evaluation Keys, and Applications to Threshold Homomorphic Encryption *EUROCRYPT 2023 Lecture Notes in Computer Science*, 2023, p. 227-256.

[11] Koskas, M., Chartier, P., Lemou, M., Méhats, Homomorphic encryption method and associated devices ans system. Patent no WO2022129979 - 06/23/2022. Number and date of priority : PCT/IB2020001147 - 12/18/2020.