Homomorphic sign evaluation with a RNS representation of integers

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

Ravel Technologies
75 rue de Richelieu, 75002 Paris

June 12, 2024

Abstract

In the context of *fully-homomorphic-encryption*, we consider the representation of large integers by their decomposition over a product of rings (through the *Chinese Remainder Theorem*) and introduce a new algorithm for the determination of the sign solely through the knowledge of ring-components. Our implementation with 128 bits of security delivers a correct result and a probability higher than $1 - 10^{-9}$ in less than 100 milliseconds for 32-bit integers on a laptop.

Keywords: fully homomorphic encryption, residue number system, sign, functional bootstrapping.

1 Introduction

On top of the two elementary arithmetic operations (addition and multiplication) included by design in all fully-homomorphic-encryption (FHE) systems, many real-world applications require comparisons[1]. As a consequence, algorithms aimed at computing the sign[2] of a message have been developed for the most prominent classes of FHE crypto-systems, that is to say FHEW/TFHE schemes for boolean circuits [22], Brakerski-Gentry-Vaikuntanathan (BGV), Brakerski/Fan-Vercauten (BFV) schemes for messages in finite fields [17, 25] and Cheon-Kim-Kim-Song (CKKS) scheme for real and complex messages [9, 19, 20]. We refer to [22] for an evaluation of the comparative merits of these various algorithms and for a description of what appears, up to our knowledge, as the most recent technique for the large-precision evaluation of the sign. However, none of the literature cited above is concerned with the sign evaluation of large-integers from its residues (encryptions thereof). It is precisely the objective of this work to introduce a method for determining the sign for a FHE crypto-system based on a *residue number system* (RNS). Using the Chinese Remainder Theorem (CRT) in order to build a FHE library is indeed a well-known theoretical alternative to the binary representation of large numbers (say 32-bits or 64-bits) and their treatment by circuits (see for instance papers on the TFHE [11] and FHEW [14] protocols). The advantage of the representation of numbers of $\mathbb{Z}/p\mathbb{Z}$

[1] This is in particular the case for training neural networks [2, 18] –or more generally statistical learning [10]– or requesting databases.

[2] The comparison of two messages a and b boils down to the determination of the sign of $a - b$.

by their moduli in a product of rings of $\mathbb{Z}/p_1\mathbb{Z} \times \cdots \times \mathbb{Z}/p_\kappa\mathbb{Z}$ lies in the fact that each ring can be handled separately as far as additions and multiplications are concerned. In the companion paper [5] by the same authors, a modification of the bootstrap procedure is introduced which aims at allowing (without extra computational cost) larger values of the pairwise coprime integers p_i's and thus values of $p = \prod_{i=1}^{\kappa} p_i$ up to 2^{64}. However, as aforementioned, one key aspect of the manipulation of large sets of data is the necessity to order and sort them: at the core of all FHE-library, should lie the possibility to determine the sign of a single number. Until now, this has prevented the use of the CRT in the context of FHE as the homomorphic determination of the sign has long been considered as a difficult question[3].

In this paper, we present a solution of this problem in the context of FHEW/TFHE encryption protocols. More precisely, we show how to compute with the help of homomorphic operations and several functional bootstrappings, an encrypted value of the sign of any element $\mu \in \mathbb{Z}/p\mathbb{Z}$ from the FHEW-encryptions c_i of its residues $\mu_i \in \mathbb{Z}/p_i\mathbb{Z}$ for $i = 1, \ldots, k$. To this aim, we first compute a series of scalings by $\bar{p}^r, r = 0, \ldots, r_{max}$, of the reconstructed ciphertext of the message μ

$$c^{[r]} = \sum_{i=1}^{k} (\bar{p}^r v_i \bmod p_i) c_i$$

where the v_i's are obtained in a standard way from the Bezout coefficients. A crucial observation is that the noise embedded in c does not grow with r (owing to the $\bmod p_i$'s) and consequently becomes smaller as compared to the message $\bar{p}^r\mu$ encrypted in $c^{[r]}$. We then show that among the consecutive magnifications of μ (again, in encrypted version), one allows to determine safely its sign. The idea is the following: given a noisy value $\mu + e$ with $\mu \in \mathbb{R}/p\mathbb{Z}$, its sign is ambiguous as soon as μ is close to 0 (or by action of the modulo, close to $\pm\frac{p}{2}$). In this case however, computing $\bar{p}\mu + e'$ with $|e'| \approx |e|$ alleviates the ambiguity as long as $\bar{p}\mu$ does not approach too closely from $\frac{p}{2}$ or is larger than $\frac{p}{2}$. If the sign can not be determined with sufficient confidence, i.e. if $\bar{p}\mu$ is still small, then one can repeat the operation. The result is then carried out through a cascade of linear combinations whose aim is to preserve the relevant information. This last trick is to a large extent similar to the one used in [3, 4]. We prove rigorously the correctness of the algorithm with very high probability for appropriate parameters and we explain how to choose them.

We finally conclude with some implementation results which demonstrate the efficiency of our algorithm: let us emphasize, for instance, that with a security level of 128 bits and a probability of an incorrect decryption smaller than 10^{-9}, we are able to homomorphically compare two 32-bit integers in just 100 milliseconds. Recalling that, to the best of our knowledge, no sign algorithm with an RNS encoding of cleartexts is available in the literature, a direct comparison is not possible. However, one of the leading FHE libraries[4] based on TFHE, reports comparison times (83.2 ms) very similar for 32-bit integers. It is worth noticing that within our implementation, addition and multiplication are distributed over the rings and thus a lot faster (our multiplication of two 32-bit integers

[3] Note that the use of a RNS is not *per se* a particular good choice if no other homomorphic computation than the sign is required.

[4] Namely THFE-rs by Zama, see https://docs.zama.ai/tfhe-rs/getting-started/benchmarks.

takes 20 ms as compared to the 211 ms announced by TFHE-rs)[5]. Alleviating the obstacle of the sign, even though its computation can not be distributed over the rings, thus allows to exploit in practice the full potential of RNS. The principal contribution of this paper can thus be summed up as follows

Main result: *Let p be a product of pairwise coprime integers p_i, $i = 1, \ldots, \kappa$. There exists an efficient algorithm, which, given FHEW-encryptions c_i of the residues $\mu_i \in \mathbb{Z}/p_i\mathbb{Z}$ for $i = 1, \ldots, \kappa$ of $\mu \in \mathbb{Z}/p\mathbb{Z}$, delivers a FHEW-encryption c of its sign, namely sign$(\mu) \in \mathbb{Z}/3\mathbb{Z}$, with very high probability of correct decryption and in less than* 100 *milliseconds for* 32*-bits integers.*

2 Background and setting of the problem

2.1 Notations and preliminaries on the Chinese remainder theorem

For all integer $p \geq 2$, the main representative of $\mu \in \mathbb{R}/p\mathbb{Z}$, denoted by $[\mu]_p$, will be taken in the interval $[-p/2, p/2[$, and the norm of μ is $|\mu| = |[\mu]_p|$. Throughout the paper, for all interval $I \subset \mathbb{R}$ of length smaller than p, for any $\mu \in \mathbb{R}/p\mathbb{Z}$, we shall say that $\mu \in I$ if there exists $k \in \mathbb{Z}$ such that $[\mu]_p - kp \in I$.
Consider an integer p of the form

$$p = \prod_{i=1}^{k} p_i$$

where the integers $p_i \geq 3$ are assumed to be odd and pairwise coprime, i.e.

$$\forall\, 1 \leq i < j \leq k, \quad p_i \wedge p_j = 1.$$

Any element μ in the set $\mathbb{Z}_p$ may be represented unambiguously (owing to the Chinese Remainder Theorem) by its coordinates

$$(\mu_1, \ldots, \mu_k) \in \mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_k}$$

with

$$\mu_i = \mu \bmod p_i, \quad i = 1, \ldots, k.$$

The Chinese Remainder Theorem states that the map

$$\begin{array}{rcl} \Phi : \mathbb{Z}_p & \to & \mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_k} \\ \mu & \mapsto & (\mu_1, \ldots, \mu_k) = (\mu \bmod p_1, \ldots, \mu \bmod p_k) \end{array}$$

is an isomorphism with inverse

$$\begin{array}{rcl} \Phi^{-1} : \mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_k} & \to & \mathbb{Z}_p \\ (\mu_1, \ldots, \mu_k) & \mapsto & \mu = \sum_{i=1}^{k} \widehat{p_i}^{-1} \widehat{p_i}\, \mu_i \bmod p \end{array}$$

where $\widehat{p_i} = p/p_i$ and where $\widehat{p_i}^{-1}$ denotes the inverse of $\widehat{p_i}$ in $\mathbb{Z}_{p_i}$, determined as a Bezout coefficient by Euclide's algorithm.

[5] The computation of an addition is even faster as it doesn't require any bootstrap: it takes just a few microseconds for 32-bits integers in the RNS representation, as compared to the 103 ms claimed for TFHE-rs.

2.2 LWE encryption and Functional Bootstrapping

In this section we recall the definition of LWE ciphertexts [26], and the properties of the functional bootstrapping procedure needed by our algorithm. The LWE cryptosystem is parametrized by a plaintext modulus p_i, a ciphertext modulus q and the secret dimension n. As in the BFV, FHEW and TFHE schemes, we shall encrypt any message in $\mathbb{Z}_{p_i}$ in the most significant digits of integers of $\mathbb{Z}_q$. The LWE encryption of a message $\mu_i \in \mathbb{Z}_{p_i}$ under (secret) key $s \in \mathbb{Z}^n$ is a vector $c = \mathrm{LWE}_s^{n,q,p_i}(\mu_i) = (a, b) \in \mathbb{Z}_q^{n+1}$ such that[6]

$$b = \langle a, s\rangle + \lfloor q\mu_i/p_i\rceil + e \bmod q$$

where $e \in \mathbb{Z}_q$ is the so-called noise, which is picked from a centered Gaussian distribution during secret-key encryption. For all ciphertext $c = (a, b) = \mathrm{LWE}_s^{n,q,p_i}(\mu_i)$, the so-called phase is the quantity

$$\varphi_s(c) := b - \langle a, s\rangle \in \mathbb{Z}_q$$

and we shall denote the error term associated to c by

$$\mathrm{Err}(c) = \varphi_s(c) - q\mu_i/p_i.$$

Introducing the *rounding error*

$$\delta_i := \lfloor q\mu_i/p_i\rceil - q\mu_i/p_i,$$

we have $\mathrm{Err}(c) = e + \delta_i \in \mathbb{Q}$ with $|\delta_i| \leq \frac{1}{2}$. The message μ_i is recovered by first computing the approximate decryption function

$$\varphi_s(c) = \lfloor q\mu_i/p_i\rceil + e = q\mu_i/p_i + \mathrm{Err}(c) \bmod q$$

and then rounding its main representative to the closest multiple of q/p_i. Decryption is correct if $|\mathrm{Err}(c)| < \frac{q}{2p_i}$. Now, if $p = \prod_{i=1}^{k} p_i$ is as in the previous section, the encryption of any (possibly large) integer $\mu \in \mathbb{Z}_p$ will be the set of encryptions $\mathrm{LWE}_s^{n,q,p_i}(\mu_i)$ of its components μ_i for $1 \leq i \leq k$.

Homomorphic arithmetic operations intrinsically increase the level of noise up to a point where the message can not decrypted. The *bootstrapping* procedure introduced by Gentry [16] and its generalisations to the evaluation of functions have been designed to re-encrypt a message with a lower noise without having to decrypt it beforehand. Ducas and Micciancio [14], and later on in a faster version, Chillotti et al. [11, 12], have introduced a very efficient bootstrapping based on the polynomial rings (see also [23, 21] for further improvements), whose details we shall not give here[7]. In the rest of this section, we nevertheless present its main properties for later use in the paper.

The FHEW/TFHE functional bootstrapping algorithm uses the polynomial ring

$$\mathcal{R}_{N,p'} = \mathbb{Z}_{p'}[X]/(X^N + 1)$$

where N is a power-of-two, so that $X^N + 1$ is the $2N$-th cyclotomic polynomial. The underlying idea of this method consists in the homomorphic implementation of a function

$$f_v \ : \ \mu \in \mathbb{Z}_{2N} \mapsto f_v(\mu) = \mathrm{coeff}_0\left(X^{\mu} v(X) \bmod (X^N + 1)\right) \in \mathbb{Z}_{p'} \tag{2.1}$$

[6] When $p_i = q$, the message $\mu_i \in \mathbb{Z}_q$ is not rescaled and the corresponding $\mathrm{LWE}_s^{n,q,q}(\mu_i)$ ciphertext will be denoted shortly as $\mathrm{LWE}_s^{n,q}(\mu_i)$.

[7] For a thorough description of the technique in the RNS context, we refer the reader to [5].

where coeff_0 selects the constant term of a polynomial and where $v \in \mathcal{R}_{N,p'}$ is the so-called *test-polynomial*, whose choice determines the characteristics of the functional bootstrapping. Note that this function f_v defined on $\mathbb{Z}_{2N}$ satisfies the negacyclic constraint

$$f_v(\mu + N) = -f_v(\mu). \tag{2.2}$$

Proposition 2.1 *Let c be a $\mathrm{LWE}_s^{n,q}$ ciphertext. For a given test-polynomial $v \in \mathcal{R}_{N,p'}$, there exists a homomorphic evaluation of the function f_v (a so-called "blind rotation") that provides a ciphertext*

$$c' = \mathrm{LWE}_s^{n,q',p'}\left(f_v(2N\varphi_s(c)/q + \delta(c))\right),$$

where the term $\delta(c)$ comes from specific rounding approximations on the ciphertext c after a rescaling. In the special case where $q|2N$, we have $\delta(c) = 0$. Moreover, the variance of the refreshed error associated to the ciphertext c' is constrained by security requirements only and does not depend on the error of the original ciphertext c.

Owing to this result, the key feature of the functional bootstrapping is that, if p_i is odd and small enough, then for any target function $F : \mathbb{Z}_{p_i} \mapsto \mathbb{Z}_{p'}$, it is possible to choose the test polynomial $v(X)$ such that

$$\forall \mu \in \mathbb{Z}_{p_i}, \quad f_v\left(\lfloor 2N\mu/p_i \rceil + \varepsilon\right) = F(\mu)$$

as soon as ε is small enough. This enables to obtain a $\mathrm{LWE}_s^{n,q',p'}(F(\mu))$ ciphertext from a $\mathrm{LWE}_s^{n,q,p_i}(\mu)$ ciphertext, with a refreshed error. In the special case where $p' = p_i$, $q' = q$ and F is the identity function, this operation is a bootstrapping in the usual sense.

2.3 Setting of the problem

We define the sign of an element $\mu \in \mathbb{Z}_p$ for odd p as the sign of its main representative

$$\mathrm{sign}(\mu) = \begin{cases} -1 & \text{if} \quad [\mu]_p \in \left\{-\frac{p-1}{2}, \ldots, -1\right\}, \\ 0 & \text{if} \quad [\mu]_p = 0, \\ +1 & \text{if} \quad [\mu]_p \in \left\{1, \ldots, \frac{p-1}{2}\right\}. \end{cases}$$

Our aim in this paper is the following

Objective: *Find the encrypted value of the sign of an element of $\mathbb{Z}_p$ from the encrypted values of its components. More precisely, given the k values*

$$c_i = \mathrm{LWE}_s^{n,q,p_i}(\mu_i) \in \mathbb{Z}_q^{n+1}, \quad i = 1, \ldots, k,$$

we aim at obtaining

$$\mathrm{LWE}_s^{n,q,3}\left(\mathrm{sign} \circ \Phi^{-1}(\mu_1, \ldots, \mu_k)\right),$$

where the sign $\in \{-1, 0, 1\}$ has been identified with an element of $\mathbb{Z}_3$.

Remark 2.2 *By linearity of LWE-encryption, the ciphertext* $c = \sum_{i=1}^{k} [\widehat{p_i}^{-1}]_{p_i} c_i$ *is an encrypted value of* $\mu = \Phi^{-1}(\mu_1, \dots, \mu_k)$*, i.e.* $c = \mathrm{LWE}_s^{n,q,p}(\mu)$ *with an error Err*(c)*. A probability estimate (see Appendix) of Err*(c) *shows that the decryption of* c *gives a wrong value with high probability (more than* 0.5*) for* $(p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8) = (7, 11, 13, 17, 19, 23, 25, 27)$ *with nominal assumptions on the errors Err*(c_i)*,* $i = 1, \dots, k$ *in the rings. This renders the determination of* μ *intractable as such and one should look for an algorithm to evaluate its sign without knowing it exactly.*

3 The sign algorithm for plaintexts

To introduce our method, let us examine a toy problem where we want to determine the sign of an integer $\mu \in \mathbb{Z}_p$, but instead of knowing its components μ_i, we only have at hand some noisy values $\widetilde{\mu_i} \in \mathbb{R}$ satisfying $\widetilde{\mu_i} = \mu_i + e_i$. We assume having an estimate on the error terms, more precisely $|e_i| \le \varepsilon/k$, for some $0 < \varepsilon \le 1/(2\bar{p}+2)$, and where $\bar{p} \ge 3$ is an odd rescaling parameter whose role will be made precise further on. Trying to reconstruct μ from the noisy values yields the approximate value

$$\widetilde{\mu}^{[0]} := \Phi^{-1}(\widetilde{\mu_1}, \dots, \widetilde{\mu_k}) = \sum_{i=1}^{k} [\widehat{p_i}^{-1}]_{p_i} \widehat{p_i}\, \widetilde{\mu_i} = \mu + e^{[0]} \bmod p,$$

with

$$e^{[0]} = \sum_{i=1}^{k} [\widehat{p_i}^{-1}]_{p_i} \widehat{p_i}\, e_i.$$

We have the estimate

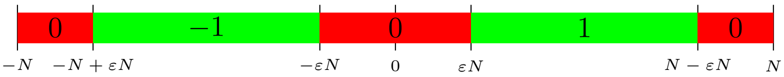
$$|\widetilde{\mu}^{[0]} - \mu| = |e^{[0]}| \le \frac{p}{2} \sum_{i=1}^{k} |e_i| \le \frac{\varepsilon}{2} p.$$

If $\frac{\varepsilon}{2} p \ge 1$, the signs of $\widetilde{\mu}^{[0]}$ and μ may be different and it is clear that knowing $\widetilde{\mu}^{[0]}$ may not be sufficient to determine the sign of μ.
The following function will be useful (the scaling by $2N$, unnatural here, prepares its use with ciphertexts in next section).

Definition 3.1 *Let* $0 \le \varepsilon \le 1$ *and* $N \ge 1$ *an integer. We introduce the function* g_ε *on* $\mathbb{R}/(2N\mathbb{Z})$ *by*

$$g_\varepsilon(\mu) = \begin{cases} +1 & \textit{if } \mu \in \left]\varepsilon N, N - \varepsilon N\right[, \\ -1 & \textit{if } \mu \in \left]-N + \varepsilon N, -\varepsilon N\right[, \\ 0 & \textit{otherwise.} \end{cases}$$

Note that g_ε is odd and satisfies the negacyclic constraint (2.2).



The function g_ε*.*

Proposition 3.2 *For $p \in \mathbb{N}^*$, $3 \le \bar{p} < p$ and odd integer and $0 < \varepsilon < \frac{1}{2(\bar{p}+1)}$, let us consider a noisy value of $\mu \in \mathbb{R}/p\mathbb{Z}$ of the form*

$$\widetilde{\mu}^{[0]} = \mu + e^{[0]} \ mod\, p \ with \ |e^{[0]}| \le \varepsilon\frac{p}{2} < 1. \tag{3.1}$$

The following statements hold

(i) if $g_\varepsilon(2N\widetilde{\mu}^{[0]}/p) = 1$, then $\mathrm{sign}(\mu) = 1$*;*

(ii) if $g_\varepsilon(2N\widetilde{\mu}^{[0]}/p) = -1$, then $\mathrm{sign}(\mu) = -1$*;*

(iii) if $g_\varepsilon(2N\widetilde{\mu}^{[0]}/p) = 0$, then $\bar{p}\mu$ and μ have the same sign.

Proof. We first assume that $g_\varepsilon(2N\widetilde{\mu}^{[0]}/p) = 1$. This means that

$$\widetilde{\mu}^{[0]} \in]\frac{\varepsilon}{2}p, \frac{p}{2} - \frac{\varepsilon}{2}p[$$

so that, from (3.1), we have

$$\mu \in]0, \frac{p}{2}[,$$

i.e. $\mathrm{sign}(\mu) = 1$. Similarly, if $g_\varepsilon(2N\widetilde{\mu}^{[0]}/p) = -1$, then we have $\mathrm{sign}(\mu) = -1$.
We now assume that $g_\varepsilon(2N\widetilde{\mu}^{[0]}/p) = 0$. Necessarily $\bar{p}\mu$ and μ have the same sign: as a matter of fact, either $\mu = 0$, or $\mu > 0$ or $\mu < 0$. If $\mu = 0$, then $\bar{p}\mu = 0$. If $\mu > 0$, then

$$\mu^{[0]} \in] - \varepsilon\frac{p}{2}, \varepsilon\frac{p}{2}] \cup [\frac{p}{2} - \varepsilon\frac{p}{2}, \frac{p}{2}[\cup [-\frac{p}{2}, -\frac{p}{2} + \varepsilon\frac{p}{2}[$$

and it stems from (3.1) and $0 < \mu < \frac{p}{2}$ (note that $\frac{p}{2} + \varepsilon\frac{p}{2} < 0$ owing to $0 < \varepsilon < 1$), that

$$\mu \in]0, \varepsilon p] \cup [\frac{p}{2} - \varepsilon p, \frac{p}{2}[$$

If μ lies in the first interval, we have

$$0 < \bar{p}\mu \le \varepsilon\bar{p}p \le \frac{\bar{p}}{2(\bar{p}+1)}p \le \frac{p}{2} - \varepsilon p, \tag{3.2}$$

while if μ lies in the second interval ($\mu \in [\frac{p}{2} - \varepsilon p, \frac{p}{2}[$), we have

$$\frac{p}{2} - \mu \in]0, \varepsilon p]$$

and similarly we get

$$0 < \bar{p}\,(\frac{p}{2} - \mu) \le \frac{p}{2} - \varepsilon p,$$

which is equivalent to

$$\varepsilon p \le \bar{p}\mu - \frac{\bar{p}-1}{2}p < \frac{p}{2}.$$

We recall that $\bar{p}$ is odd, so $\frac{\bar{p}-1}{2}$ is an integer, which yields $\bar{p}\mu \in [\varepsilon p, \frac{p}{2}[$. Consequently, in both cases, we have $\mathrm{sign}(\bar{p}\mu) = +1 = \mathrm{sign}(\mu)$. If $\mu < 0$, by considering $-\mu > 0$ we get that $\mathrm{sign}(\bar{p}\mu) = -1 = \mathrm{sign}(\mu)$. The claim is proved. □

We now consider the following approximation of $\bar{p}\mu$:

$$\widetilde{\mu}^{[1]} := \sum_{i=1}^{k} [\bar{p}\widehat{p_i}^{-1}]_{p_i}\widehat{p}_i\,\widetilde{\mu}_i = \bar{p}\mu + e^{[1]} \bmod p, \quad \text{with} \quad e^{[1]} = \sum_{i=1}^{k} [\bar{p}\widehat{p_i}^{-1}]_{p_i}\widehat{p}_i\, e_i.$$

Since we have the same estimate

$$|\widetilde{\mu}^{[1]} - \bar{p}\mu| = |e^{[1]}| \leq \frac{p}{2}\sum_{i=1}^{k} |e_i| \leq \frac{\varepsilon}{2}\, p, \tag{3.3}$$

the same reasoning as above leads to the fact that if $g_\varepsilon(2N\widetilde{\mu}^{[1]}/p) = +1$ (resp. $= -1$) then $\mathrm{sign}(\bar{p}\mu) = \mathrm{sign}(\mu) = +1$ (resp. $= -1$). In other words, in the case $g_\varepsilon(2N\widetilde{\mu}^{[0]}/p) = 0$, the quantity $g_\varepsilon(2N\widetilde{\mu}^{[1]}/p)$ is an estimator of the sign of μ with no false positive.
One can iterate on this method, considering all the rescalings

$$\widetilde{\mu}^{[r]} := \sum_{i=1}^{k} [\bar{p}^r\widehat{p_i}^{-1}]_{p_i}\widehat{p}_i\,\widetilde{\mu}_i = \bar{p}^r\mu + e^{[r]} \bmod p \,\text{with}\, e^{[r]} = \sum_{i=1}^{k} [\bar{p}^r\widehat{p_i}^{-1}]_{p_i}\widehat{p}_i\, e_i, \tag{3.4}$$

for $r \in \mathbb{N}$. By an induction argument, one can easily generalize the above proof and show that, if $g_\varepsilon(2N\widetilde{\mu}^{[0]}/p) = \ldots = g_\varepsilon(2N\widetilde{\mu}^{[r-1]}/p) = 0$, then all the terms $\bar{p}^r\mu$, $\bar{p}^{r-1}\mu$, ..., $\bar{p}\mu$ and μ have the same sign and, moreover, if $g_\varepsilon(2N\widetilde{\mu}^{[r]}/p) = +1$ (resp. $= -1$) then $\mathrm{sign}(\mu) = +1$ (resp. $= -1$).
In fact, the number of required rescalings can be bounded *a priori*. To see this point, we state a technical Lemma.

Lemma 3.3 *Let $3 \leq \bar{p} < p$ be odd integers, let $0 < \varepsilon \leq \frac{1}{2(\bar{p}+1)}$ and let $\mu \in \mathbb{R}/p\mathbb{Z}$. Consider the sequence $(\bar{p}^r\mu)_{r\geq 0}$. The following statements hold true.*

(i) If $\mu \in]0, \varepsilon p]$, then there exists $r^* \in \mathbb{N}^*$ such that for all $0 \leq r < r^*$, one has $\bar{p}^r\mu \in]0, \varepsilon p]$ and $\bar{p}^{r^*}\mu \in]\varepsilon p, \frac{p}{2} - \varepsilon p]$.

(ii) If $\mu \in [\frac{p}{2} - \varepsilon p, \frac{p}{2}[$, then there exists $r^* \in \mathbb{N}^*$ such that for all $0 \leq r < r^*$, one has $\bar{p}^r\mu \in [\frac{p}{2} - \varepsilon p, \frac{p}{2}[$ and $\bar{p}^{r^*}\mu \in [\varepsilon p, \frac{p}{2} - \varepsilon p[$.

(iii) If $\mu \in [-\varepsilon p, 0[$, then there exists $r^* \in \mathbb{N}^*$ such that for all $0 \leq r < r^*$, one has $\bar{p}^r\mu \in [-\varepsilon p, 0[$ and $\bar{p}^{r^*}\mu \in [-\frac{p}{2} + \varepsilon p, -\varepsilon p[$.

(iv) If $\mu \in]-\frac{p}{2}, -\frac{p}{2} + \varepsilon p[$, then there exists $r^* \in \mathbb{N}^*$ such that for all $0 \leq r < r^*$, one has $\bar{p}^r\mu \in]-\frac{p}{2}, -\frac{p}{2} + \varepsilon p[$ and $\bar{p}^{r^*}\mu \in]-\frac{p}{2} + \varepsilon p, -\varepsilon p]$.

Proof. Items *(iii)* and *(iv)* can be directly deduced from *(i)* and *(ii)* by $\mu \to \frac{p}{2} + \mu$. Note indeed that, $\bar{p}$ being odd, we have $\bar{p}^r\frac{p}{2} = \frac{p}{2} \bmod p$ for all $r \geq 0$.
Let us prove *(i)*. We thus assume that $\mu \in]0, \varepsilon p]$. Let $r^* \geq 1$ be the largest integer such that $\bar{p}^r\mu \in]0, \varepsilon p]$ for all $0 \leq r \leq r^* - 1$ (such an integer exists given that $\bar{p}^0\mu \in]0, \varepsilon p]$ and that the *real* sequence $\bar{p}^r\mu \to +\infty$ when $r \to +\infty$). By construction, we have $\bar{p}^{r^*-1}\mu \leq \varepsilon p < \bar{p}^{r^*}\mu \leq \frac{p}{2} - \varepsilon p$, which yields

$$r^* = 1 + \left\lfloor \log_{\bar{p}}\left(\frac{\varepsilon p}{\mu}\right)\right\rfloor.$$

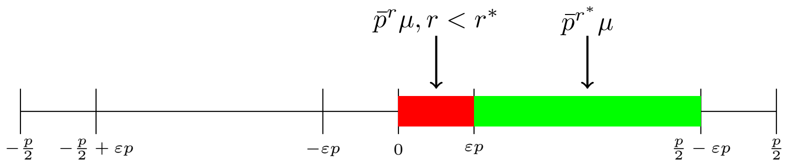


Figure 1: *Case (i) of Lemma 3.3.*

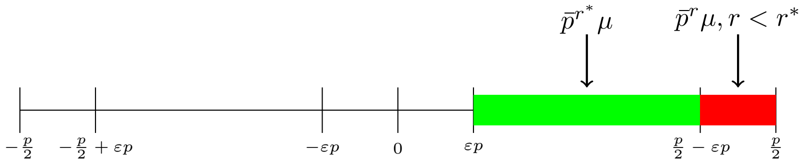


Figure 2: *Case (ii) of Lemma 3.3.*

Moreover, replacing μ by $\bar{p}^{r^*-1}\mu \in]0, \varepsilon p]$ in (3.2), we obtain $\bar{p}^{r^*}\mu \leq \frac{p}{2}-\varepsilon p$. We have proved *(i)*.
In order to prove *(ii)*, we now assume that $\mu \in [\frac{p}{2}-\varepsilon p, \frac{p}{2}[$. Then $\frac{p}{2}-\mu \in]0, \varepsilon p]$ so Item *(i)* can be applied to $\frac{p}{2}-\mu$. Setting $r^* = 1 + \left\lfloor \log_{\bar{p}} \left(\frac{\varepsilon p}{p/2-\mu} \right) \right\rfloor$, one has

$$\forall 0 \leq r \leq r^*-1, \qquad \bar{p}^r \left(\frac{p}{2} - \mu \right) \in]0, \varepsilon p] \quad \text{and} \quad \bar{p}^{r^*} \left(\frac{p}{2} - \mu \right) \in \left] \varepsilon p, \frac{p}{2} - \varepsilon p \right].$$

By substracting $p/2$, this yields

$$\forall 0 \leq r \leq r^*-1, \quad \bar{p}^r\mu - \frac{\bar{p}^r-1}{2} p \in \left[\frac{p}{2} - \varepsilon p, \frac{p}{2} \right[\text{ and } \bar{p}^{r^*}\mu - \frac{\bar{p}^{r^*}-1}{2} p \in \left[\varepsilon p, \frac{p}{2} - \varepsilon p \right[.$$

Since $\frac{\bar{p}^r-1}{2}$ is an integer for all $r>0$, the proof of *(ii)* is complete. □

Remark 3.4 *By considering the smallest and largest positive values in $\mathbb{Z}_p$, that is to say $\mu = 1$ and $\mu = \frac{p-1}{2}$, with $\varepsilon \leq \frac{1}{2(\bar{p}+1)}$, we can bound from above r^* by*

$$r^* \leq r_{max} = 1 + \left\lfloor \log_{\bar{p}} \left(\frac{p}{\bar{p}+1} \right) \right\rfloor.$$

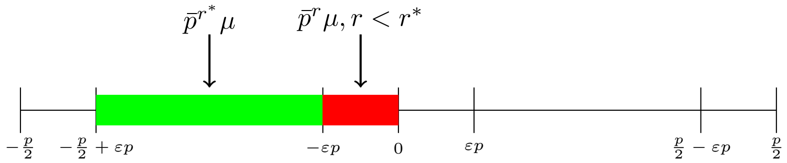


Figure 3: *Case (iii) of Lemma 3.3.*

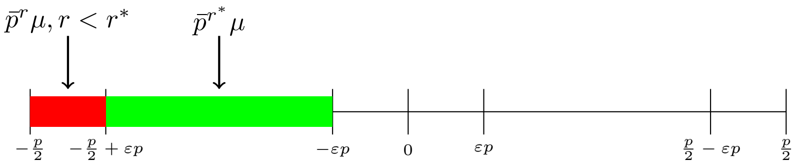


Figure 4: *Case (iv) of Lemma 3.3.*

In order to prepare the adaptation of this algorithm to ciphertexts, we summarize in the following proposition the result that we have proved.

Proposition 3.5 *Let $3 \leq \bar{p} < p$ be odd integers and let $0 < \varepsilon \leq \frac{1}{2(\bar{p}+1)}$. Let $\mu \in \mathbb{Z}_p$ and consider a sequence of real numbers $\mu^{[r]}$ defined for $r = 0, 1, \ldots, r_{max} = 1 + \left\lfloor \log_{\bar{p}}\left(\frac{p}{\bar{p}+1}\right)\right\rfloor$ and satisfying*

$$|\mu^{[r]} - 2N\bar{p}^r\mu/p|| \leq \varepsilon N. \tag{3.5}$$

Then, there exists $r^* \geq 0$ such that

1. *if $\mu > 0$ then $g_\varepsilon(\mu^{[r^*]}) = 1$ and for all $0 \leq r < r^*$, $g_\varepsilon(\mu^{[r]}) = 0$;*
2. *if $\mu < 0$ then $g_\varepsilon(\mu^{[r^*]}) = -1$ and for all $0 \leq r < r^*$, $g_\varepsilon(\mu^{[r]}) = 0$;*
3. *if $\mu = 0$ then $g_\varepsilon(\mu^{[r]}) = 0$ for all $r \geq 0$,*

where the function g_ε was introduced in Definition 3.1.

As a direct application of this proposition, one can directly determine the sign of μ by a lexicographic comparison of $(g_\varepsilon(2N\widetilde{\mu}^{[0]}/p), g_\varepsilon(2N\widetilde{\mu}^{[1]}/p), \ldots, g_\varepsilon(2N\widetilde{\mu}^{[r_{max}]})/p)$ with $(0, 0, \ldots, 0)$. Equivalently, we can state the

Corollary 3.6 *Let $3 \leq \bar{p} < p$ be odd integers, let $0 < \varepsilon \leq \frac{1}{2(\bar{p}+1)}$ and let $\mu \in \mathbb{Z}_p$. Denote $\mu_i = \mu$ mod p_i and let g_ε be the function given in Definition 3.1. If for $0 \leq r \leq r_{max} = 1 + \left\lfloor \log_{\bar{p}}\left(\frac{p}{\bar{p}+1}\right)\right\rfloor$, we define $\widetilde{\mu}^{[r]}$ by (3.4), where the noisy values $\widetilde{\mu}_i \in \mathbb{R}/p_i\mathbb{Z}$ satisfy $|\widetilde{\mu}_i - \mu_i| \leq \varepsilon/k$, then we have*

$$\operatorname{sign}(\mu) = \operatorname{sign}\left(\sum_{r=0}^{r_{max}} 2^{r_{max}-r}\, g_\varepsilon(2N\widetilde{\mu}^{[r]}/p)\right). \tag{3.6}$$

As a matter of fact, either all values $g_\varepsilon\left(2N\widetilde{\mu}^{[r]}/p)\right)$ remain null, and the sum accordingly, or the first non-vanishing value (either 1 or -1) dominates the sum (owing to the scaling factors $2^{r_{max}-r}$). We hereafter illustrate Corollary 3.6 with $\bar{p} = 13$ and $r_{max} = 8$ in the case where $\mu \in]0, \varepsilon p]$.

4 The homomorphic sign algorithm

With the notations introduced in Section 2, we consider a plaintext $\mu \in \mathbb{Z}_p$ encoded by its CRT components μ_i, $1 \leq i \leq k$, which are encrypted as $c_i = \mathrm{LWE}_s^{n,q,p_i}(\mu_i)$, with errors $\mathrm{Err}(c_i)$. Our aim is to obtain an encrypted value of $\operatorname{sign}(\mu)$. Three steps are necessary to adapt the above algorithm from plaintexts to ciphertexts.

r	$\tilde{\mu}^{[r]}$	Interval	$g_\varepsilon\left(2N\tilde{\mu}^{[r]}/p\right)$	S
0	$\approx \mu$	$]0, \varepsilon p]$	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$r^* - 1$	$\approx \bar{p}^{r^*-1}\mu$	$]0, \varepsilon p]$	0	0
r^*	$\approx \bar{p}^{r^*}\mu$	$]\varepsilon p, \frac{p}{2} - \varepsilon p]$	1	M
$r^* + 1$	$\approx \bar{p}^{r^*+1}\mu$	$[-\frac{p}{2}, \frac{p}{2}[$	$\in \{-1, 0, 1\}$	$\geq M - \frac{M}{2} = \frac{M}{2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_{max}	$\approx \bar{p}^{r_{max}}\mu$	$[-\frac{p}{2}, \frac{p}{2}[$	$\in \{-1, 0, 1\}$	$\geq M - \frac{M}{2} - \ldots - 1 = 1$

Table 1: Values of the inner sum (row S) of (3.6) with $M = 2^{r_{max} - r^*}$ and $\mu > 0$.

4.1 Rescaling ciphertexts

The first step consists in rescaling the ciphertexts c_i by factors $\bar{p}^r$. The following result is an adaptation of Proposition 3.5.

Proposition 4.1 *Let $3 \leq \bar{p} < p$ be odd integers and let $0 < \varepsilon \leq \frac{1}{2(\bar{p}+1)}$. Consider the sequence*

$$c^{[r]} = \sum_{i=1}^{k} [\bar{p}^r \widehat{p_i}^{-1}]_{p_i} c_i, \qquad r = 0, \ldots, r_{max} = 1 + \left\lfloor \log_{\bar{p}} \left(\frac{p}{\bar{p}+1} \right) \right\rfloor \tag{4.1}$$

of encrypted values $\mathrm{LWE}_s^{n,q,p}(\bar{p}^r \mu)$ and denote, for all $\mathrm{LWE}_s^{n,q}$ ciphertext c,

$$\widetilde{\varphi}_s(c) := 2N\varphi_s(c)/q + \delta(c) \in \mathbb{Z}_{2N}, \tag{4.2}$$

where $\delta(c)$ was defined in Proposition 2.1. Suppose that, for all r, we have the estimate

$$\left| 2NErr(c^{[r]})/q + \delta(c^{[r]}) \right| \leq \varepsilon N. \tag{4.3}$$

Then, there exists $r^* \in \{0, \ldots, r_{max}\}$ such that

1. *if $\mu > 0$ then $g_\varepsilon(\widetilde{\varphi}_s(c^{[r^*]})) = 1$ and for all $0 \leq r < r^*$, $g_\varepsilon(\widetilde{\varphi}_s(c^{[r]})) = 0$;*
2. *if $\mu < 0$ then $g_\varepsilon(\widetilde{\varphi}_s(c^{[r^*]})) = -1$ and for all $0 \leq r < r^*$, $g_\varepsilon(\widetilde{\varphi}_s(c^{[r]})) = 0$;*
3. *if $\mu = 0$ then $g_\varepsilon(\widetilde{\varphi}_s(c^{[r]})) = 0$ for all $r \geq 0$,*

where the function g_ε was defined in Definition 3.1.

Proof. This result if a direct application of Proposition 3.5, setting $\mu^{[r]} = \widetilde{\varphi}_s(c^{[r]})$. Indeed, (4.3) yields, for all r,

$$\left| \mu^{[r]} - 2N\bar{p}^r \mu/p \right| = \left| \widetilde{\varphi}_s(c^{[r]}) - 2N\bar{p}^r \mu/p \right| \leq \varepsilon N,$$

which enables to apply this proposition. □

Remark 4.2 *Piecewise constant functions may also be obtained through an elaboration of the same technique as for the sign. To this aim, it is sufficient to notice (i), that the Heaviside function $H(\mu)$ can be emulated through the same procedure by attributing the value* 0 *instead of* -1 *to all torus-elements in* $[-\frac{p}{2}+\frac{\varepsilon}{2}p, -\frac{\varepsilon}{2}p]$ *in the definition of* g_ε *and (ii), that all piecewise constant functions* f *on the discrete torus* $[-\frac{p}{4}, \frac{p}{4}]$ *are linear combinations of translated Heaviside functions* $f(x) = \sum_i \alpha_i H(x-\beta_i)$ *where the* α_i*'s are integers and the* β_i*'s are elements of* $[-\frac{p}{4}, \frac{p}{4}]$.

4.2 Emulating g_ε through bootstrapping

Having computed the rescaled ciphertexts $c^{[r]}$ for $0 \leq r \leq r_{max}$ by formula (4.1), the second step of the sign algorithm consists in a functional bootstrapping of each $c^{[r]}$ in order to compute an encrypted version of $g_\varepsilon(\widetilde{\varphi}_s(c^{[r]}))$. To this aim, we have to define a suitable test-polynomial $v(x)$.

More precisely, we aim in this subsection at constructing a test-polynomial $v^\kappa(X) \in \mathcal{R}_{N,2N}$ such that the associated function defined by (2.1) satisfies

$$\forall \mu \in \mathbb{Z}_{2N}, \qquad f_{v^\kappa}(\mu) = 2^\kappa g_\varepsilon(\mu), \tag{4.4}$$

where $0 \leq \kappa \leq \log N$ is a scaling factor so as to emulate the function g_ε, rescaled, in an encrypted form through a bootstrapping procedure (according to Proposition 2.1).

Let

$$\varepsilon = \frac{1}{2N} + \frac{\alpha}{N} \tag{4.5}$$

where α is an integer satisfying

$$0 \leq \alpha \leq \left\lfloor \frac{N}{2(\bar{p}+1)} - \frac{1}{2} \right\rfloor . \tag{4.6}$$

It is readily seen that the constraint

$$0 < \varepsilon \leq \frac{1}{2(\bar{p}+1)}$$

is fulfilled.

A key feature of functional bootstrapping based on blind rotation is that for any function F defined from $\mathbb{Z}_{2N}$ to $\mathbb{Z}$ and such that

$$\forall j \in \mathbb{Z}_{2N}, \quad F(j+N) = -F(j),$$

there exists a unique polynomial $v \in \mathbb{Z}[X]/(X^N+1)$ such that the function f_v defined by (2.1) satisfies

$$\forall j \in \mathbb{Z}_{2N}, \qquad f_v(j) = F(j).$$

Its coefficients v_j are given by $v_j = F(-j)$, $j = 0, \ldots, N-1$.

For a given α in (4.5), it is thus enough to define F on $\{0, \ldots, N-1\}$ as follows:

$$\begin{aligned} \forall\, 0 \leq j \leq \alpha, \qquad & F(j) = 0, \\ \forall\, \alpha+1 \leq j \leq N-\alpha-1, \qquad & F(j) = 2^\kappa, \\ \forall\, N-\alpha \leq j \leq N-1, \qquad & F(j) = 0, \end{aligned}$$

so that

$$v_j^\kappa := F(-j), \qquad j = 0, \dots, N-1,$$

that is to say

$$\begin{aligned} &v_0^\kappa = \dots = v_\alpha^\kappa = 0, \\ &v_{\alpha+1}^\kappa = \dots = v_{N-\alpha-1}^\kappa = -2^\kappa, \\ &v_{N-\alpha}^\kappa = \dots = v_{N-1}^\kappa = 0. \end{aligned} \tag{4.7}$$

For these specific choices of ε and v^κ, the equality (4.4) is satisfied.

4.3 Implementing the homomorphic lexicographic comparison

Arguing as for Corollary 3.6, it is clear from Proposition 4.1 that the sign of $\mu \in \mathbb{Z}_p$ can be obtained from the expression

$$\sum_{r=0}^{r_{max}} 2^{r_{max}-r} g_\varepsilon(\widetilde{\varphi}_s(c^{[r]})). \tag{4.8}$$

Assuming for a while that $2^{r_{max}} \leq N$ and using that the addition is homomorphic, an encryption of (4.8) is

$$\sum_{r=0}^{r_{max}} \mathrm{LWE}_s^{n,q,2N}\left(2^{r_{max}-r} g_\varepsilon(\widetilde{\varphi}_s(c^{[r]}))\right) = \sum_{r=0}^{r_{max}} \mathrm{LWE}_s^{n,q,2N}\left(f_{v^{r_{max}-r}}(\widetilde{\varphi}_s(c^{[r]}))\right). \tag{4.9}$$

According to Subsection 2.2, we can bootstrap directly $c^{[r]}$ onto the encryption of $f_{v^{r_{max}-r}}(\widetilde{\varphi}_s(c^{[r]}))$, by using the test-polynomial $v^{r_{max}-r}(X)$, and sum up homomorphically to obtained the desired ciphertext (4.9).
However, the noise in (4.9) is determined by the output noise of the boostrapping procedure. This may render the decryption of (4.9) incorrect, as soon as the noise is non zero (indeed, the smallest non zero value in (4.8) may be ±1). In order to overcome this difficulty, we first decompose the sum (4.8) into sub-sums of m terms as follows, where we have supposed, for the sake of simplicity, that $r_{max} + 1 = m^\ell$, where by definition $\ell = \log_m(r_{max} + 1)$. We suppose also that $2^{m+1} \leq N$. Let $\widetilde{\varepsilon}$ be defined by

$$\widetilde{\varepsilon} = \frac{1}{2N} + \frac{\widetilde{\alpha}}{N}, \qquad \text{with} \quad \widetilde{\alpha} = \frac{N}{2^{m+1}}, \tag{4.10}$$

and consider

$$g_{\widetilde{\varepsilon}}\left(\sum_{r_0=0}^{m-1} 2^{\log N - r_0 - 1} g_{\widetilde{\varepsilon}}\left(\sum_{r_1=0}^{m-1} 2^{\log N - r_1 - 1} g_{\widetilde{\varepsilon}}\left(\dots \sum_{r_{\ell-1}=0}^{m-1} 2^{\log N - r_{\ell-1} - 1} g_\varepsilon(\widetilde{\varphi}_s(c^{[j_\mathbf{r}]}))\right)\right)\right) \tag{4.11}$$

with

$$j_\mathbf{r} = \sum_{i=0}^{\ell-1} r_i m^{\ell-1-i}, \qquad \mathbf{r} = (r_0, \dots, r_{\ell-1}).$$

Now, the smallest non zero values in this new sum is larger than $2^{\log N - m} > 1$, which authorizes some noise in the encrypted form of (4.11). Note that the innermost loop involves g_ε, while all other loops resort to $g_{\widetilde{\varepsilon}}$. It is easy to check that, by Proposition 4.1 and as formula (4.8), this new expression also gives the sign of μ. Using once again the homomorphy of the addition and the bootstrapped version of g_ε, we obtain an encryption of the sign of μ from the sequence of ciphertexts $c^{[k]}$ through the expression

$$\widehat{F}\left(\sum_{r_0=0}^{m-1} \widetilde{F}_{r_0}\left(\sum_{r_1=0}^{m-1} \widetilde{F}_{r_1}\left(\dots \sum_{r_{\ell-1}=0}^{m-1} F_{r_{\ell-1}}\left(c^{[j_{\mathbf{r}}]}\right)\right)\right)\right)$$

where we have denoted, for any LWE ciphertext c,

$$F_j(c) := \mathrm{LWE}_s^{n,q,2N}\left(f_{v^{\log N-j-1}}(\widetilde{\varphi}_s(c))\right), \quad \widetilde{F}_j(c) := \mathrm{LWE}_s^{n,q,2N}\left(f_{\tilde{v}^{\log N-j-1}}(\widetilde{\varphi}_s(c))\right)$$

and

$$\widehat{F}(c) := \mathrm{LWE}_s^{n,q,3}\left(f_{\hat{v}}(\widetilde{\varphi}_s(c))\right)$$

with $\tilde{v}^\kappa$ and $\hat{v}$ obtained from the following adaptations of Formula (4.7):

$$\tilde{v}_0^\kappa = \dots = \tilde{v}_{\tilde{\alpha}}^\kappa = 0, \quad \tilde{v}_{\tilde{\alpha}+1}^\kappa = \dots = \tilde{v}_{N-\tilde{\alpha}-1}^\kappa = -2^\kappa, \quad \tilde{v}_{N-\tilde{\alpha}}^\kappa = \dots = \tilde{v}_{N-1}^\kappa = 0, \tag{4.12}$$

$$\hat{v}_0 = \dots = \hat{v}_{\tilde{\alpha}} = 0, \quad \hat{v}_{\tilde{\alpha}+1} = \dots = \hat{v}_{N-\tilde{\alpha}-1} = -1, \quad \hat{v}_{N-\tilde{\alpha}} = \dots = \hat{v}_{N-1} = 0. \tag{4.13}$$

The corresponding algorithm is illustrated in Figure 5, in a special case.

Algorithm 1 Homomorphic determination of the sign

```
For  $r = 0, \dots, m^\ell - 1$ do
     $c^{[r]} = \sum_{i=1}^{k} [\bar{p}^r \widehat{p_i}^{-1}]_{p_i} c_i$   with $c_i \in \mathbb{Z}_{2N}^{n+1}$
End
$S_0 = 0$
For  $r_0 = 0, \dots, m-1$ do
     $S_1 = 0$
     For  $r_1 = 0, \dots, m-1$ do
        ...
        $S_{\ell-2} = 0$
        For  $r_{\ell-2} = 0, \dots, m-1$ do
           $S_{\ell-1} = 0$
           For  $r_{\ell-1} = 0, \dots, m-1$ do
              $r = r_{\ell-1} + m r_{\ell-2} + \dots + m^{\ell-2} r_1 + m^{\ell-1} r_0$
              $S_{\ell-1} = S_{\ell-1} + F_{r_{\ell-1}}(c^{[r]})$
           End
           $S_{\ell-2} = S_{\ell-2} + \widetilde{F}_{r_{\ell-2}}(S_{\ell-1})$
        ...
        $S_1 = S_1 + \widetilde{F}_{r_1}(S_2)$
     End
     $S_0 = S_0 + \widetilde{F}_{r_0}(S_1)$
End
Return $\widehat{F}(S_0)$
```

Remark 4.3 *Note that the factors 2^j used here could be replaced by other choices. This one is optimal in the context of the sign but is not compatible with some other piecewise constant functions.*

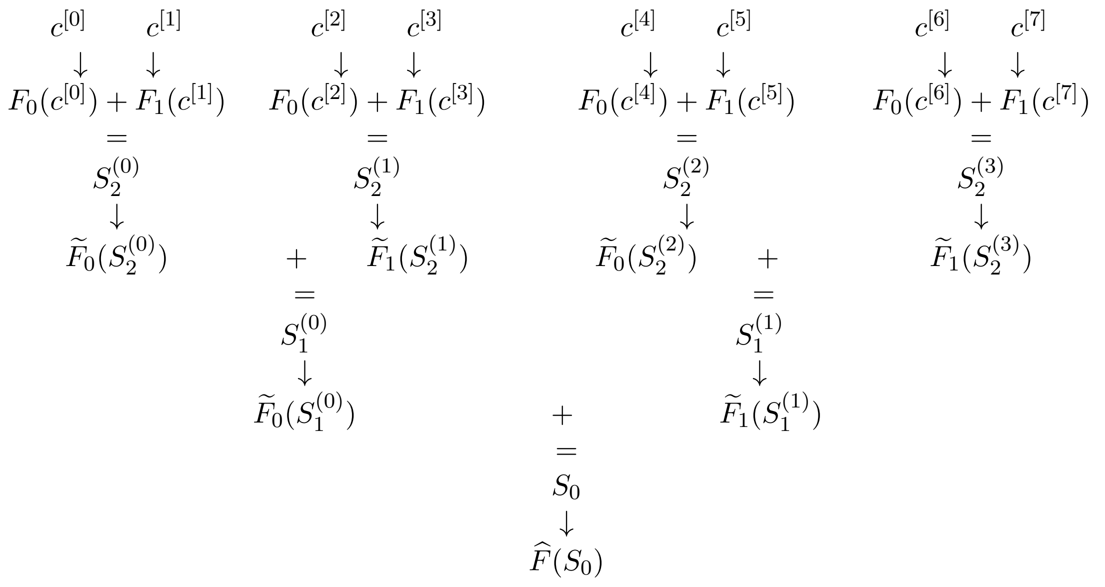

$c^{[0]}$
$c^{[1]}$
$c^{[2]}$
$c^{[3]}$
$c^{[4]}$
$c^{[5]}$
$c^{[6]}$
$c^{[7]}$
$F_0(c^{[0]}) + F_1(c^{[1]}) = S_2^{(0)}$
$F_0(c^{[2]}) + F_1(c^{[3]}) = S_2^{(1)}$
$F_0(c^{[4]}) + F_1(c^{[5]}) = S_2^{(2)}$
$F_0(c^{[6]}) + F_1(c^{[7]}) = S_2^{(3)}$
$\widetilde{F}_0(S_2^{(0)}) + \widetilde{F}_1(S_2^{(1)}) = S_1^{(0)}$
$\widetilde{F}_0(S_2^{(2)}) + \widetilde{F}_1(S_2^{(3)}) = S_1^{(1)}$
$\widetilde{F}_0(S_1^{(0)}) + \widetilde{F}_1(S_1^{(1)}) = S_0$
$\widehat{F}(S_0)$

Figure 5: Computation of the sign function for $m = 2$ and $\ell = 3$: each arrow represents a bootstrap.

The following proposition states the conditions under which our algorithm works.

Proposition 4.4 *Let $3 \leq \bar{p} < p$ be odd integers and let ε be given by* (4.5)*, where the integer α satisfies* (4.6)*. Let $\mu \in \mathbb{Z}_p$ and consider encryptions of its CRT components $c_i = \mathrm{LWE}_s^{n,q,p_i}(\mu_i)$. Consider the sequence $c^{[r]}$, for $r = 0, \ldots, r_{max} = 1 + \left\lfloor \log_{\bar{p}} \left(\frac{p}{\bar{p}+1} \right) \right\rfloor$ defined by* (4.1)*. Assume that* (4.3) *is satisfied for all r and that each LWE ciphertext S_i defined in Algorithm 1 as an argument of a function $\widetilde{F}_j$ or of $\widehat{F}$ satisfies the estimate*

$$|2N\mathit{Err}(S_i)/q + \delta(S_i)| \leq N/2^{m+1} - 1. \tag{4.14}$$

Then Algorithm 1 provides an $\mathrm{LWE}_s^{n,q,3}(\mathrm{sign}(\mu))$ ciphertext with an error bounded independently of the c_i's.

Proof. Thanks to Propositions 2.1 and 4.1, and by (4.4), we already know that the innermost loop, the only one that involves g_ε, is correct. Moreover, each S_i to be bootstrapped in the next steps with the function $g_{\widetilde{\varepsilon}}$ is an $\mathrm{LWE}_s^{n,q,2N}(\Sigma_i)$ ciphertext, where Σ_i is under the form

$$\Sigma_i = \sum_{j=0}^{m-1} 2^{\log N - j - 1} \xi_j \qquad \text{with} \quad \xi_j \in \{-1, 0, 1\}.$$

These values belong to the set

$$\frac{N}{2^m} \mathbb{Z}_{2^{m+1}-1} = \{0, \pm N/2^m, \pm 2N/2^m, \pm 3N/2^m, \ldots, \pm(2^m - 1)N/2^m\}.$$

Hence, owing to the formulae (4.12) or (4.13) of the test-polynomials $v(X)$ used in this bootstrap, only three cases have to be examined:

- if $\Sigma_i = 0$, then $\widetilde{F}_j(S_j)$ is a correct bootstrap if $|\widetilde{\varphi}_s(S_i)| \leq \widetilde{\alpha}$;

– if $\Sigma_i > 0$, then $\widetilde{F}_j(S_j)$ is a correct bootstrap if $\widetilde{\alpha} + 1 \leq \widetilde{\varphi}_s(S_i) \leq N - \widetilde{\alpha} - 1$;

– if $\Sigma_i < 0$, then $\widetilde{F}_j(S_j)$ is a correct bootstrap if $-N + \widetilde{\alpha} + 1 \leq \widetilde{\varphi}_s(S_i) \leq -\widetilde{\alpha} - 1$.

Since $\widetilde{\alpha} = N/2^{m+1}$, it can be observed that each of these three conditions is satisfied when (4.14) is fulfilled. □

4.4 Correctness of the associated sign function for a specific implementation of bootstrap

In this subsection, we show that our method is of practical interest by estimating its probability of success in a typical implementation. We shall consider the TFHE bootstrapping introduced in [11, 12], extended to messages in the discrete tori $\mathbb{T}_{p_i} = \frac{1}{p_i}\mathbb{Z}_{p_i}$. In order to make our Proposition 2.1 more precise, let us make a few assumptions. We refer to [12] (see e.g. Algorithm 1 in this paper) for the definition of the parameters B_g and ℓ_g involved in the gadget decomposition (and, also, we only consider the case where the associated $k_g = 1$). Moreover, the keys are binary and, for simplicity we only consider the case where no keyswitch is used in the bootstrap (although this restriction is far from optimal). If $\mathbb{B}$ denotes the set $\{0, 1\}$, the vectorial secret keys for LWE ciphertexts belong to $\mathbb{B}^N$ and the polynomial secret keys for RLWE ciphertexts belong to $\mathbb{B}[X]/(X^N + 1)$, where we recall that N is a power-of-two.

First, the deviation term $\delta(c)$ in Proposition 2.1 comes from rounding the mask $(a_i)_{i=1,\dots,N}$ of the $\mathrm{LWE}_s^{N,q}$ ciphertext $c = (a, b)$ at the beginning of the blind rotate process. More precisely, an instanciation of blind rotation leads to[8]

$$2N\varphi_s(c)/q + \delta(c) = \left\lfloor 2Nb/q - \frac{1}{2}\sum_{i=1}^{N}\delta_i \right\rceil - \sum_{i=1}^{N} \lfloor 2Na_i/q \rceil\, s_i,$$

where we have denoted $\delta_i = 2Na_i/q - \lfloor 2Na_i/q \rceil$. Since $b = \sum_i a_i s_i + \varphi_s(c)$, we compute

$$\delta(c) = \left\lfloor \sum_{i=1}^{N} (s_i - 1/2)\delta_i + 2N\varphi_s(c)/q \right\rceil - 2N\varphi_s(c)/q,$$

and denoting $\gamma = -\sum_{i=1}^{N}(s_i - 1/2)\delta_i - 2N\varphi_s(c)/q$, this yields

$$\delta(c) = \sum_{i=1}^{N} (s_i - 1/2)\delta_i + \gamma - \lfloor \gamma \rceil.$$

Since we have assumed that $s_i \in \{0, 1\}$ for all i, we finally get the estimate

$$|\delta(c)| \leq \frac{1}{2}\left(1 + \sum_{i=1}^{N} |2Na_i/q - \lfloor 2Na_i/q \rceil|\right). \tag{4.15}$$

Moreover, following [5], the variance of the refreshed error of the output $c' = \mathrm{LWE}_s^{N,q,p'}$ (we take $q' = q$) can be computed as[9]

$$\sigma_{out}^2 := \mathrm{Var}(\mathrm{Err}(c')) = N\left(1 + \frac{N}{2}\right) q^2 \frac{B_g^{-2\ell_g}}{12} + 2N^2 \ell_g \frac{B_g^2 + 2}{12} \sigma_{\mathrm{BK}}^2,$$

[8] The trick of substracting the term $\frac{1}{2}\sum_{i=1}^{N}\delta_i$ to $2Nb/q$ before rounding improves the total rounding error.

[9] We assume here that B_g is even.

where σ_{BK} is the standard deviation for the noise sampled to generate the RGSW bootstrap keys. Finally, we shall make the standard assumption [24, 13] that the Central Limit Theorem applies and that the output error $\mathrm{Err}(c')$ can be well approximated by a gaussian random variable.
A very conservative estimate of the correctness of the sign function can be obtained by computing the probability that all the bootstraps involved in Algorithm 1 are correct. Assuming that all the $e_i = \mathrm{Err}(c_i)$ associated to the c_i's involved in formula (4.1) are independent sub-gaussian random variables with parameters $\sigma(e_i)$, and that the terms in the sum in the right-hand side of (4.15) are uniformly distributed independent random variables and as such sub-gaussian with parameter $1/(2\sqrt{3})$, we may obtain the following upper bound of the probability of getting an incorrect sign by computing the probability that at least one condition on the errors in Proposition 4.4 is not satisfied. In other terms, we have

$$\mathbb{P}_{fail} \leq \mathbb{P}_{fr} + \mathbb{P}_{or} \tag{4.16}$$

where $\mathbb{P}_{fr}$ is the probability that one of the bootstrap of the innermost loop of Algorithm 1 fails and $\mathbb{P}_{or}$ is the probability that one of the bootstrap of the outer loops of Algorithm 1 fails.

Lemma 4.5 *Assuming that all the $e_i = Err(c_i), i = 1, \ldots, k$ are independent sub-gaussian random variables with parameters $\sigma(e_i)$, and that each rounding error term in* (4.15) *is sub-gaussian with parameter $1/(2\sqrt{3})$, we have*

$$\mathbb{P}_{fr} \leq 2 \sum_{r=0}^{r_{max}} \exp\left(-(\varepsilon - 1/(2N))^2/(8\sigma_r^2)\right) \tag{4.17}$$

where

$$\sigma_r^2 = \sum_{i=1}^{k} [\bar{p}^r \widehat{p_i}^{-1}]_{p_i}^2 \left(\sigma(e_i)/q\right)^2 + 1/(192N), \qquad r = 0, \ldots, r_{max}. \tag{4.18}$$

Moreover, assuming that the sums obtained in the outer loops of Algorithm 1 and used as input of further bootstraps are LWE ciphertext whoses errors are independent sub-gaussian variables with common parameter $\sqrt{m}\sigma_{out}$, we have

$$\mathbb{P}_{or} \leq \frac{2r_{\max}}{m-1} \exp\left(-(1/2^m - 1/N)^2/(32\widetilde{\sigma}_r^2)\right) \tag{4.19}$$

where

$$\widetilde{\sigma}_r^2 = m\sigma_{out}^2/q^2 + 1/(192N).$$

Proof. The first statement (4.17) follows from the upper-bound

$$\mathbb{P}_{fr} \leq \sum_{r=0}^{r_{max}} \mathbb{P}\left(|\mathrm{Err}(c^{[r]})/q + \delta(c^{[r]})/(2N)| \geq \varepsilon/2\right)$$

and from Markov's inequality. For the second statement, we have to estimate from above the probability $\mathbb{P}_{incbo}$ that one bootstrap $\widetilde{F}_{r_i}(S_i)$, $r_i \in \{0, \ldots, m-1\}$, of a sum S_i involved in the outer loops of Algorithm 1 is incorrect. Recall that each S_i is of the form $S_i = e + \sum_{j=0}^{m-1} 2^{\log N - j - 1}\xi_j$, where the ξ_j's take their values in $\{\pm 1, 0\}$ and e is a sub-gaussian variable with parameter $\sqrt{m}\sigma_{out}$. According to Proposition 4.4, we have

$$\mathbb{P}_{incbo} \leq \mathbb{P}\left(|e/q + \delta(S_i)/(2N)| \geq 1/2^{m+2}\right)$$

and using again Markov's inequality yields (4.19). Note that the number of such bootstraps is

$$m^{\ell-1} + m^{\ell-2} + \ldots + m^1 + m^0 = \frac{r_{\max}}{m-1}.$$

□

An example

For instance, consider the situation where

$$p = 7 \times 11 \times 13 \times 17 \times 19 \times 23 \times 25 \times 27 = 5019589575 > 2^{32}$$

and $\bar{p} = 13$. We take $m = 3$, $\ell = 2$ and compute $r_{max} = 8$. Maximizing (4.6), we obtain $\alpha = 36$. We compute successively the Bezout coefficients associated with $(p_i, p/p_i)$ for $i = 1, \dots, 8$ and the growth factors in (4.18). We have the following table:

p_j	7	11	13	17	19	23	25	27	$(\sum_j [\bar{p}^r \widehat{p_j}^{-1}]_{p_j})^2$
$[\widehat{p_j}^{-1}]_{p_j}$	2	3	-2	1	4	6	-3	5	104
$[13\widehat{p_j}^{-1}]_{p_j}$	-2	-5	0	-4	-5	9	11	11	393
$[13^2\widehat{p_j}^{-1}]_{p_j}$	2	1	0	-1	-8	2	-7	8	187
$[13^3\widehat{p_j}^{-1}]_{p_j}$	-2	2	0	4	-9	3	9	-4	211
$[13^4\widehat{p_j}^{-1}]_{p_j}$	2	4	0	1	-3	-7	-8	2	147
$[13^5\widehat{p_j}^{-1}]_{p_j}$	-2	-3	0	-4	-1	1	-4	-1	48
$[13^6\widehat{p_j}^{-1}]_{p_j}$	2	5	0	-1	6	-10	-2	-13	339
$[13^7\widehat{p_j}^{-1}]_{p_j}$	-2	-1	0	4	2	8	-1	-7	139
$[13^8\widehat{p_j}^{-1}]_{p_j}$	2	-2	0	1	7	-11	12	-10	423

As cryptographic parameters, let us take the following values, corresponding to a security[10] of $\lambda = 80$ bits.

N	q	σ_{BK}	B_g	ℓ_g
1024	2^{64}	1.3×10^7	2^{13}	2

We assume moreover that the ciphertexts c_i have been obtained by a bootstrap with the same parameters, i.e. for $i = 1, \dots, k$, we take $\sigma(e_i) = \sigma_{out}$. This set of parameters yields

$$\sigma_{out}^2/q^2 = 2.14 \times 10^{-11}, \quad \widetilde{\sigma}_r = 5.09 \times 10^{-6}, \quad \mathbb{P}_{fr} \leq 1.19 \times 10^{-12} \quad \mathbb{P}_{or} \leq 7.25 \times 10^{-41}$$

so, finally,

$$\mathbb{P}_{fail} \leq 1.2 \times 10^{-12}.$$

This proves the efficiency of our method. Note that the number of bootstraps involved in one homomorphic evaluation of the sign is $r_{max}+1+\frac{r_{max}}{m-1} = 13$ here, which is less than an homomorphic multiplication (which can be done with $2k = 16$ bootstraps).

5 Performance results

In the next tableaux, we give the computational times of the sign determination for integers of various sizes (from 8 bits to 64 bits). In order to emphasize the global interest of the RNS representation which is very beneficial for a multi-threaded execution, we also include the computational times of elementary operations $+$, $\times$. All the numbers under consideration are integers with various numbers of bits, from 16 to 64, and all computations are made on an average laptop. Note that for the sake of better readability the algorithm is used here with the same values $m = 3$ and $\ell = 2$ for all benchmarks. Better performances are obtained by optimizing the parameters in each situation (for a given security parameter, given size of integers and given error probability) and this work is ongoing. For instance, as stated in the abstract, our algorithm delivers a correct result with a probability error below 10^{-12} and a security of 128 bits in less than 100 milliseconds for 32-bit integers.

[10] According to the lattice estimator `https://github.com/malb/lattice-estimator`

Type	Security	$\mathbb{P}_{\text{fail}}$	$\times$	$=$	Sign	$+$
$U8$	80 bits	$1.e-5$	$7.96\,ms$	$14.74\,ms$	$40.02\,ms$	$5.92\,\mu s$
$U8$	80 bits	$1.e-9$	$8.59\,ms$	$15.69\,ms$	$43.19\,ms$	$5.89\,\mu s$
$U8$	80 bits	$1.e-12$	$8.93\,ms$	$16.11\,ms$	$44.63\,ms$	$5.91\,\mu s$
$U8$	100 bits	$1.e-5$	$18.18\,ms$	$33.78\,ms$	$92.10\,ms$	$5.93\,\mu s$
$U8$	100 bits	$1.e-9$	$19.85\,ms$	$36.56\,ms$	$98.23\,ms$	$5.92\,\mu s$
$U8$	100 bits	$1.e-12$	$16.40\,ms$	$26.34\,ms$	$71.16\,ms$	$8.97\,\mu s$
$U8$	128 bits	$1.e-5$	$26.56\,ms$	$48.85\,ms$	$132.7\,ms$	$14.02\,\mu s$
$U8$	128 bits	$1.e-9$	$28.62\,ms$	$50.70\,ms$	$137.35\,ms$	$14.05\,\mu s$
$U8$	128 bits	$1.e-12$	$27.98\,ms$	$51.29\,ms$	$138.33\,ms$	$13.98\,\mu s$
$U16$	80 bits	$1.e-5$	$17.91\,ms$	$15.50\,ms$	$42.38\,ms$	$11.91\,\mu s$
$U16$	80 bits	$1.e-9$	$18.37\,ms$	$16.45\,ms$	$46.01\,ms$	$11.79\,\mu s$
$U16$	80 bits	$1.e-12$	$20.25\,ms$	$17.82\,ms$	$50.11\,ms$	$11.95\,\mu s$
$U16$	100 bits	$1.e-5$	$38.78\,ms$	$34.62\,ms$	$94.76\,ms$	$11.96\,\mu s$
$U16$	100 bits	$1.e-9$	$52.55\,ms$	$46.54\,ms$	$126.62\,ms$	$11.93\,\mu s$
$U16$	100 bits	$1.e-12$	$47.52\,ms$	$40.75\,ms$	$108.58\,ms$	$14.89\,\mu s$
$U16$	128 bits	$1.e-5$	$35.52\,ms$	$54.46\,ms$	$151.95\,ms$	$20.93\,\mu s$
$U16$	128 bits	$1.e-9$	$60.09\,ms$	$52.47\,ms$	$145.40\,ms$	$28.18\,\mu s$
$U16$	128 bits	$1.e-12$	$61.62\,ms$	$53.56\,ms$	$145.91\,ms$	$28.16\,\mu s$

Figure 6: Computation times on milliseconds (except for the addition in microseconds)

Appendix

The sign of $\mu \in \mathbb{Z}_p$ can not be determined from the signs of its components $(\mu_1, \dots, \mu_k)$. This can be easily seen on the following example with $k = 2$, $p_1 = 3$ and $p_2 = 5$: both $2 \in \mathbb{Z}_{15}$ and $7 \in \mathbb{Z}_{15}$ have positive signs, while their components are respectively $(-1, 2) \in \mathbb{Z}_3 \times \mathbb{Z}_5$ and $(1, 2) \in \mathbb{Z}_3 \times \mathbb{Z}_5$ with signs $(-1, 1)$ and $(1, 1)$ respectively. This shows that the value of μ has to some extent to be computed through Φ^{-1} in order to evaluate its sign: denoting $c_i = (a_i, b_i)$ and $c = (a, b)$, we have

$$\begin{aligned} b - \langle a, s\rangle &= \sum_{i=1}^{k} [\widehat{p_i}^{-1}]_{p_i} b_i - \sum_{i=1}^{k} \left\langle [\widehat{p_i}^{-1}]_{p_i} a_i, s \right\rangle = \sum_{i=1}^{k} [\widehat{p_i}^{-1}]_{p_i} \left(b_i - \langle a_i, s\rangle\right) \\ &= \sum_{i=1}^{k} [\widehat{p_i}^{-1}]_{p_i} \left((q/p_i)\mu_i + \mathrm{Err}(c_i)\right) \mod q \\ &= (q/p) \sum_{i=1}^{k} \widehat{p_i}^{-1} \widehat{p_i}\, \mu_i + \sum_{i=1}^{k} [\widehat{p_i}^{-1}]_{p_i} \mathrm{Err}(c_i) \mod q \\ &= (q/p)\mu + \mathrm{Err}(c) \mod q, \end{aligned}$$

with

$$\mathrm{Err}(c) = \sum_{i=1}^{k} [\widehat{p_i}^{-1}]_{p_i} \mathrm{Err}(c_i).$$

In practice, the errors $\mathrm{Err}(c_i)$, for $i = 1, \dots, k$, are the sum of a fixed rational (the rounding error) and of a sub-gaussian random variable e_i with parameter $\sigma(e_i)$, that is to say such that

$$\mathbb{E}(e^{\lambda e_i}) \leq e^{\frac{1}{2}\sigma^2(e_i)\lambda^2}.$$

Type	Security	$\mathbb{P}_{\text{fail}}$	$\times$	$=$	Sign	$+$
$U32$	80 bits	$1.e-5$	$7.96\,ms$	$14.74\,ms$	$40.02\,ms$	$5.92\,\mu s$
$U32$	80 bits	$1.e-9$	$8.59\,ms$	$15.69\,ms$	$43.19\,ms$	$5.89\,\mu s$
$U32$	80 bits	$1.e-12$	$8.93\,ms$	$16.11\,ms$	$44.63\,ms$	$5.91\,\mu s$
$U32$	100 bits	$1.e-5$	$18.18\,ms$	$33.78\,ms$	$92.10\,ms$	$5.93\,\mu s$
$U32$	100 bits	$1.e-9$	$19.85\,ms$	$36.56\,ms$	$98.23\,ms$	$5.92\,\mu s$
$U32$	100 bits	$1.e-12$	$16.40\,ms$	$26.34\,ms$	$71.16\,ms$	$8.97\,\mu s$
$U32$	128 bits	$1.e-5$	$26.56\,ms$	$48.85\,ms$	$132.7\,ms$	$14.02\,\mu s$
$U32$	128 bits	$1.e-9$	$28.62\,ms$	$50.70\,ms$	$137.35\,ms$	$14.05\,\mu s$
$U32$	128 bits	$1.e-12$	$27.98\,ms$	$51.29\,ms$	$138.33\,ms$	$13.98\,\mu s$
$U64$	80 bits	$1.e-5$	$17.91\,ms$	$15.50\,ms$	$42.38\,ms$	$11.91\,\mu s$
$U64$	80 bits	$1.e-9$	$18.37\,ms$	$16.45\,ms$	$46.01\,ms$	$11.79\,\mu s$
$U64$	80 bits	$1.e-12$	$20.25\,ms$	$17.82\,ms$	$50.11\,ms$	$11.95\,\mu s$
$U64$	100 bits	$1.e-5$	$38.78\,ms$	$34.62\,ms$	$94.76\,ms$	$11.96\,\mu s$
$U64$	100 bits	$1.e-9$	$52.55\,ms$	$46.54\,ms$	$126.62\,ms$	$11.93\,\mu s$
$U64$	100 bits	$1.e-12$	$47.52\,ms$	$40.75\,ms$	$108.58\,ms$	$14.89\,\mu s$
$U64$	128 bits	$1.e-5$	$35.52\,ms$	$54.46\,ms$	$151.95\,ms$	$20.93\,\mu s$
$U64$	128 bits	$1.e-9$	$60.09\,ms$	$52.47\,ms$	$145.40\,ms$	$28.18\,\mu s$
$U64$	128 bits	$1.e-12$	$61.62\,ms$	$53.56\,ms$	$145.91\,ms$	$28.16\,\mu s$

Figure 7: Computation times in milliseconds (except for the addition in microseconds)

Neglecting rounding errors and using standard estimates, it is straightforward to show that the decryption of c in $\mathbb{Z}_p$ coincides with μ with probability

$$\mathrm{erf}\left(\frac{q}{2\sqrt{2}\sigma(e)p}\right)$$

with parameter $\sigma(e) = \sqrt{[\widehat{p_1}^{-1}]^2_{p_1}\sigma^2(e_1) + \ldots + ([\widehat{p_k}^{-1}]^2_{p_k}\sigma^2(e_k)}$. Now, assuming, for instance, that the parameters $\sigma(e_i) = \frac{q}{2\sqrt{2}p_i\theta}$ have all been adjusted so as to ensure a correct decryption in $\mathbb{Z}_{p_i}$ with a given probability $\mathrm{erf}(\theta)$, the probability that the decryption of μ fails can be bounded from below by

$$1 - \mathrm{erf}\left(\frac{q}{2\sqrt{2}\sigma p}\right) \geq \sqrt{\frac{e}{2\pi}}\ \exp\left(-\frac{2\theta^2 p_{max}^2}{kp^2}\right),$$

where $p_{max} = \max_{i=1,\ldots,k} p_i$. This bound shows that the decryption of c is incorrect with a probability higher than 0.5 for $(p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8) = (7, 11, 13, 17, 19, 23, 25, 27)$ and $1 - \mathrm{erf}(\theta) = 10^{-10}$ in the rings. This renders the determination of μ intractable as such and one should look for an algorithm to evaluate its sign without knowing it exactly.

References

[1] J. Alperin-Sheriff, and C. Peikert. *Faster Bootstrapping with Polynomial Error*, CRYPTO 2014, 297–314. Springer, 2014.

[2] F. Bourse, M. Minelli, M. Minihold, and Pascal Paillier, *Fast homomorphic evaluation of deep discretized neural networks*, Advances in Cryptology - CRYPTO, 483–512, Springer, 2018.

[3] F. Bourse, O. Sanders, and J. Traoré, *Improved secure integer comparison via homomorphic encryption*, Cryptographers' Track at the RSA Conference, 391–416, Springer, 2020.

[4] R. Carlton, A. Essex, and K. Kapulkin, *Threshold properties of prime power subgroups with application to secure integer comparisons*, Cryptographers' Track at the RSA Conference, 137–156. Springer, Heidelberg, 2018.

[5] P. Chartier, M. Koskas, M. Lemou, and F. Méhats, *Fully Homomorphic Encryption on large integers*, Cryptology ePrint Archive, 2024.

[6] P. Chartier, M. Koskas, M. Lemou, and F. 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.

[7] J. H. Cheon, M. Kim, and M. Kim, *Search-and-compute on encrypted data*, Financial Cryptography and Data Security, 142–159, Berlin, Heidelberg, Springer, 2015.

[8] J. H. Cheon, M. Kim, and M. Kim, *Optimized Search-and-Compute Circuits and Their Application to Query Evaluation on Encrypted Data*, IEEE Transactions on Information Forensics and Security, 11(1):188–199, 2016.

[9] J. H. Cheon, D. Kim, and D. Kim, *Efficient homomorphic comparison methods with optimal complexity*, Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, 221–256, 2020.

[10] H. Chen, R. Gilad-Bachrach, K. Han, Z. Huang, A. Jalali, K. Laine, and K. Lauter, *Logistic regression over encrypted data from fully homomorphic encryption*, Cryptology ePrint Archive, Report 2018/462, 2018.

[11] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, *Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds*, Advances in Cryptology – ASIACRYPT 2016, 3–33, Berlin, Heidelberg, Springer, 2016.

[12] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, *TFHE: Fast fully homomorphic encryption over the torus*, Journal of Cryptology, 33(1):34–91, 2020.

[13] A. Costache, B. R. Curtis, E. Hales, S. Murphy, T. Ogilvie, R. Player, *On the precision loss in approximate homomorphic encryption*, Cryptology ePrint Archive, Report 2022/162, 2022.

[14] L. Ducas and D. Micciancio, *FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second*, Advances in Cryptology – EUROCRYPT 2015, 617–640, Berlin, Heidelberg, Springer, 2015.

[15] N. Gama, M. Izabachene, P. Q. Nguyen, and X. Xie. *Structural lattice reduction: Generalized worst-case to average-case reductions and homomorphic cryptosystems*, EUROCRYPT 2016, 528–558. Springer, 2016.

[16] C. Gentry, *Fully homomorphic encryption using ideal lattices*, 41st Annual ACM Symposium on Theory of Computing, 169–178. ACM Press, 2009.

[17] I. Iliashenko, and V. Zucca, *Faster homomorphic comparison operations for BGV and BFV*, Proceedings on Privacy Enhancing Technologies, 2021.

[18] M. Izabachène, R. Sirdey, and M. Zuber, *Practical fully homomorphic encryption for fully masked neural networks*, Cryptology and Network Security - 18th International Conference, CANS, 24–36. Springer, 2019.

[19] E. Lee, J.-W. Lee, J.-S. No, and Y.-S. Kim, *Minimax approximation of sign function by composite polynomial for homomorphic comparison*, IEEE Transactions on Dependable and Secure Computing, 2021.

[20] E. Lee, J.-W. Lee, Y.-S. Kim, and J.-S. No, *Optimization of homomorphic compari- son algorithm on rns-ckks scheme*, Cryptology ePrint Archive, Report 2021/1215, 2021.

[21] Y. Lee, D. Micciancio, A. Kim, R. Choi, M. Deryabin, J. Eom, and D. Yoo, *Efficient FHEW Bootstrapping with Small Evaluation Keys, and Applications to Threshold Homomorphic Encryption*, Advances in Cryptology – EUROCRYPT 2023.

[22] Z. Liu, D. Micciancio, and Y. Polyakov, *Large-precision homomorphic sign evaluation using FHEW/TFHE bootstrapping*, IACR Cryptol. ePrint Arch., 2021/1337, 2021.

[23] D. Micciancio, and Y. Polyakov, *Bootstrapping in FHEW-like Cryptosystems*, Association for Computing Machinery, New York, NY, USA, 17–28, 2021.

[24] S. Murphy, R. Player, *A Central Limit Framework for Ring-LWE Decryption*, Cryptology ePrint Archive, Report 2019/452, 2019.

[25] H. Narumanchi, D. Goyal, N. Emmadi, and P. Gauravaram, *Performance analysis of sorting of fhe data: Integer-wise comparison vs bit-wise comparison*, 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), 902–908, 2017.

[26] O. Regev, *On lattices, learning with errors, random linear codes, and cryptography*, Journal of the ACM (JACM), 56(6):1–40, 2009.