CHIMERA: Combining Ring-LWE-based Fully Homomorphic Encryption Schemes

Christina Boura, Nicolas Gama, Mariya Georgieva, Dimitar P. Jetchev

Research Article

Christina Boura, Nicolas Gama, Mariya Georgieva*, and Dimitar Jetchev

CHIMERA: Combining Ring-LWE-based Fully Homomorphic Encryption Schemes

https://doi.org/10.1515/jmc-2019-0026
Received Jul 12, 2019; accepted May 11, 2020

Abstract: This paper proposes a practical hybrid solution for combining and switching between three popular Ring-LWE-based FHE schemes: TFHE, B/FV and HEAAN. This is achieved by first mapping the different plaintext spaces to a common algebraic structure and then by applying efficient switching algorithms. This approach has many practical applications. First and foremost, it becomes an integral tool for the recent standardization initiatives of homomorphic schemes and common APIs. Then, it can be used in many real-life scenarios where operations of different nature and not achievable within a single FHE scheme have to be performed and where it is important to efficiently switch from one scheme to another. Finally, as a byproduct of our analysis we introduce the notion of a FHE module structure, that generalizes the notion of the external product, but can certainly be of independent interest in future research in FHE.

Keywords: fully homomorphic encryption, Ring-LWE, lattice based cryptography, floating point computation, TFHE, B/FV, HEAAN

2010 Mathematics Subject Classification: 94A60

1 Introduction

Homomorphic encryption enables computations on encrypted data without decrypting it. Shortly after the development of the first fully homomorphic encryption (FHE) scheme by Gentry [23], extensive research has been carried out on the design, implementation and cryptanalysis of various other FHE schemes.

Several constructions based on the Ring-LWE problem [26] are today among the most promising FHE candidates, each of them having particular advantages that depend on the type of the target homomorphic operations and arithmetic. More precisely, certain schemes are better for integer arithmetic whereas others have advantages for performing arithmetic with real numbers; some are more suitable for vector operations whereas others perform well in the case of sequential combinatorial operations on individual slots such as the evaluation of Boolean circuits, finite state machines or lookup tables. In addition, different constructions typically tolerate different amounts of noise and hence, homomorphic evaluation of circuits of different multiplicative depth.

It is thus of central importance to be able to use the optimal scheme for each type of operation in a particular computation. This motivates the need for building an efficient hybrid solution combining more than one scheme and switching between these schemes during the individual operations.

This paper proposes such a practical hybrid solution based on efficient switching algorithms for three different Ring-LWE schemes, where each scheme is best suited for certain types of operations: 1) TFHE [18, 19]

***Corresponding Author: Mariya Georgieva:** Inpher, Lausanne, Switzerland; EPFL, Lausanne, Switzerland
Christina Boura: Laboratoire de Mathématiques de Versailles, UVSQ, CNRS, Université Paris-Saclay, Versailles, France; Inria, Paris, France
Nicolas Gama: Laboratoire de Mathématiques de Versailles, UVSQ, CNRS, Université Paris-Saclay, Versailles, France; Inpher, Lausanne, Switzerland
Dimitar Jetchev: Inpher, Lausanne, Switzerland; EPFL, Lausanne, Switzerland

particularly suitable for combinatorial operations on individual slots and tolerating large noise and thus, large multiplicative depth. 2) B/FV [6, 13, 21] allowing to perform large vectorial arithmetic operations as long as the multiplicative depth of the evaluated circuit remains small; 3) HEAAN [15, 16] - a mixed encryption scheme shown to be very efficient for floating-point computations.

We achieve such a hybrid solution by first mapping the different plaintext spaces of the different schemes to a common algebraic structure using certain natural algebraic homomorphisms. Once such a uniformization of the plaintext spaces has been achieved, we describe our scheme switching algorithms. The main idea here is to replace the expensive bootstrapping algorithms with more efficient key-switching operations. Recall that if $\mathcal{S}_1$ and $\mathcal{S}_2$ are two homomorphic encryption schemes then the concept of *bootstrapping*, originally introduced by Gentry [23], is a homomorphic evaluation (under the scheme $\mathcal{S}_2$) of the decryption function for the scheme $\mathcal{S}_1$. Since all Ring-LWE-based FHE decryption functions evaluate an inner product followed by a rounding function and since the rounding function is a step function, instead of first applying the expensive bootstrapping and then evaluating a function f, one can replace the last rounding step in the bootstrapping by a more general step function g that approximates f and use a homomorphic lookup table evaluation. This is essentially the idea of the key-switching algorithms described in Section 2.2 that are later used to achieve a switch between the schemes in Sections 3 and 4. Note that the concept of functional key-switch is not new (it appeared already in the context of TFHE [18] and implicitly in the work of Ducas and Micciancio [20]). Yet, its application to scheme switching presented in this work is novel.

To better explain the algebraic aspects of our approach, we let $\mathcal{R}_m$ denote the quotient ring $(\mathbb{Z}/m\mathbb{Z})[X]/(X^N+1)$ for some integers m and N and we let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ be the torus. To introduce the algebraic structure of the plaintext spaces in TFHE, we distinguish two different underlying encryption schemes (an encryption scheme on a ring and a homomorphic one on a module over that ring):

- TRGSW: encryption scheme on the ring $\mathcal{R} := \mathbb{Z}[X]/(X^N + 1)$,
- TRLWE: HE scheme on the $\mathcal{R}$-module $\mathbb{T}_{\mathcal{R}} := (\mathbb{R}[X]/(X^N + 1))/\mathcal{R}$.

The plaintext space for the scheme TRGSW is the ring $\mathcal{R}$ whereas its ciphertext space is $(\mathbb{T}^N)^{2\ell}$ for some integer ℓ that depends on various precision and noise parameters (see Section 2 for details). Elseways, the plaintext space for the scheme TRLWE is the $\mathcal{R}$-module $\mathbb{T}_{\mathcal{R}}$ and the ciphertext space is $(\mathbb{T}^N)^2$ (see Section 2.2 for details as well as for the definition of the external product which is the homomorphic evaluation algorithm for the ring action on the module).

On the other hand, the plaintext space of B/FV is $\mathcal{R}_p$ for some integer p (a prime, a power of 2 or a small number $1 \mod 2N$, depending on the functionality that we want) and the ciphertext space is $\mathcal{R}_q^2$ for some larger integer q. Finally, HEAAN has for message space a ball of radius B in $\mathcal{R}_q$ with respect to a certain ℓ_∞-norm defined in Section 2.4 and its ciphertext space is $\mathcal{R}_q^2$.

It is not too hard to verify that in the three cases listed above, the ciphertext spaces share very similar algebraic structures. For B/FV and HEAAN this is trivial to do; for TFHE, it suffices to identify the ciphertext space $\mathcal{R}_q^2$ with a subgroup of the torus $(\mathbb{T}^N)^2$ as follows: using that $\mathcal{R}_q^2 \simeq ((\mathbb{Z}/q\mathbb{Z})^N)^2$, we simply identify $q^{-1}\mathbb{Z}/\mathbb{Z} \simeq \mathbb{Z}/q\mathbb{Z}$, the latter being the multiplication-by-q isomorphism of $\mathbb{Z}$-modules.

Yet, the plaintext spaces are *a priori* rather different. In order to apply any key-switching technique, one thus needs to uniformize them (i.e., map them to the same algebraic structure). In addition, this algebraic structure should carry a specific metric allowing us to quantify and measure the noise in the decryption function. We achieve this by using the plaintext space of TFHE, namely $\mathbb{R}[X]/(X^N + 1)$ modulo $\mathbb{Z}$ (a space we introduce in Section 2 and denote by $\mathbb{T}_{\mathcal{R}}$ throughout the paper). We then show how to map both B/FV and HEAAN plaintexts to $\mathbb{T}_{\mathcal{R}}$ in Sections 3 and 4.

Our hybrid solution has many practical applications. First and foremost, it becomes an integral tool for the recent standardization initiatives of homomorphic schemes and common APIs [8], especially since TFHE, B/FV (Seal) and HEAAN are part of this standardization process.

Then, imagine a scenario where operations on large datasets must be performed and a decision must be taken on the result. The first part would be easier via the B/FV scheme, whereas the decision function can be evaluated faster in TFHE. Therefore, one would like an efficient method to pass from B/FV to TFHE. Similarly, if more approximate computations have to be performed, one would benefit from a scheme switch-

ing between HEAAN and TFHE. Another example comes from the recent Idash'18 Track 2 [1] competition on designing homomorphic solutions for semi-parallel Genome Wide Association Studies (GWAS). One of the operations needed for the challenge required to compute homomorphically the product $G^{-1} \cdot S$, for a small 4×4 symmetric matrix G and a much larger 4×10000 matrix S. In this scenario, the bootstrapping of TFHE can be used to compute the 10 coefficients of G^{-1} (via either Gaussian elimination or Cholesky factorization), so these algorithms that include loops, inversions and have very high multiplicative depth, benefit from fast bootstrapping on individual data. Then, the matrix multiplication $G^{-1} \cdot S$ can be massively vectorized using the SIMD operations of HEAAN. Similarly, different machine learning algorithms use SIMD operations to produce an output vector and at the end, one needs to compute the maximum of its coordinates. Conversely, various financial systems need to perform small computations on an encrypted database with a potentially very large multiplicative depth over a long period of time and at the end of the period provide statistics on the current dataset. In this case, it is essential to operate in bootstrapped mode as in TFHE and then perform the low-depth statistical calculations in B/FV or HEAAN. In such a scenario, it is important to transform TFHE ciphertexts to B/FV or HEAAN ones.

As a byproduct of our analysis of the three above schemes from the perspective of TFHE, we propose the general definition of a *FHE module structure*, that is, an external product allowing to homomorphically evaluate the action of the ring R on the module M whenever M is equipped with an HE scheme and R is equipped with a FHE scheme. This concept already appears in a particular case in [18] (named as *external product*), but it can certainly be of independent interest in future research in FHE. For instance, it permits to express the relinearization in the internal products of HEAAN and B/FV in terms of the FHE module structure for TFHE and this without loss as it is the same algorithm.

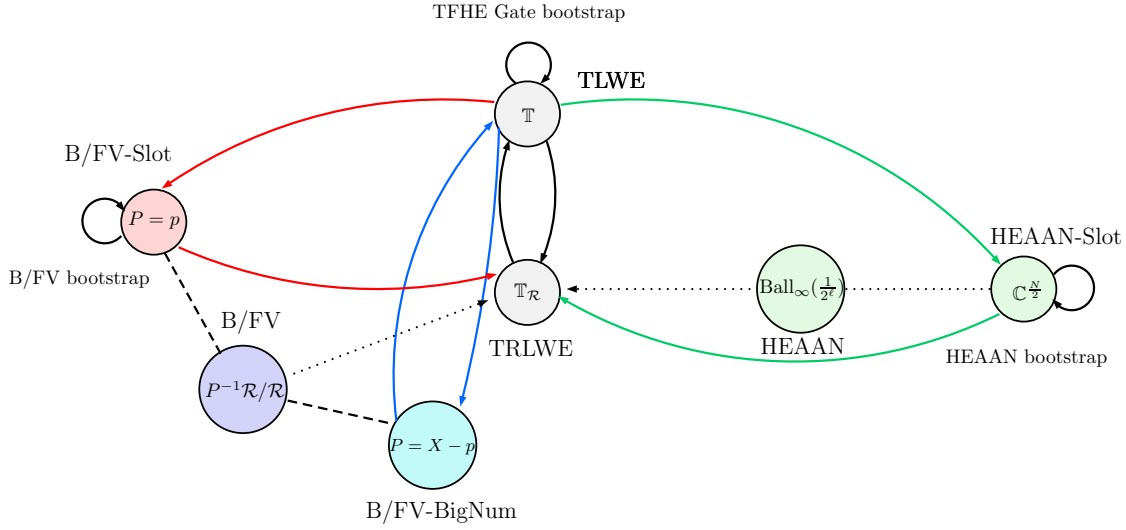

Figure 1: Bridges between Ring-LWE homomorphic schemes. Plain arrows represent bridges between the different schemes. Dotted arrows represent inclusion. Finally, the spaces $P = p$ and $P = X - p$ are instantiations of $P^{-1}\mathcal{R}/\mathcal{R}$.

2 Preliminaries

Let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ be the real torus and let $\mathcal{R} = \mathbb{Z}[X]/(X^N + 1)$ be the ring of polynomials with integer coefficients modulo $X^N + 1$. For a ring A (e.g., $A = \mathbb{Z}, \mathbb{R}, \mathbb{C}$), let $\mathcal{R}_A := \mathcal{R} \otimes_{\mathbb{Z}} A = A[X]/(X^N + 1)$ be the ring of polynomials modulo $X^N + 1$ with coefficients in A. In particular, $\mathcal{R} = \mathcal{R}_{\mathbb{Z}}$, so we interchangeably omit the index.

Let $\mathbb{T}_{\mathcal{R}} = \mathcal{R}_{\mathbb{R}}/\mathcal{R}_{\mathbb{Z}}$ (a.k.a $\mathbb{R}[X] \mod X^N + 1 \mod \mathbb{Z}$) which we view as an $\mathcal{R}$-module (it has no ring structure). We often refer to the left action of $\mathcal{R}$ on $\mathbb{T}_{\mathcal{R}}$ as an *external multiplication* with coefficients in $\mathcal{R}$. Moreover,

let $\mathcal{B}$ be the subset of all polynomials of $\mathcal{R}_{\mathbb{Z}}$ with coefficients in $\{0, 1\}$ (here, we identify $\mathcal{R}_{\mathbb{Z}} \simeq \mathbb{Z}^N$ as $\mathbb{Z}$-modules in the natural way). Finally, if $x \in \mathcal{R}$, let $\mathcal{R}_x = \mathcal{R}/x\mathcal{R}$ and let $\pi_x \colon \mathcal{R} \to \mathcal{R}_x$ be the natural surjection.

We use the ℓ_p-distance on the N-dimensional torus $\mathbb{T}^N$ and write $\|x - y\|_p$ for the distance between two elements $x, y \in \mathbb{T}$. Note that it satisfies $\forall m \in \mathbb{Z}$, $\|m \cdot \mathbf{x}\|_p \leq |m| \, \|\mathbf{x}\|_p$. For an integer element $a \in \mathbb{T}_{\mathcal{R}}$, consider any representative $a(X) = a_0 + \cdots + a_{N-1}X^{N-1} \in \mathbb{R}[X]$. We will (unambiguously) write $\|a\|_p$ for the norm of the image of $(a_0, \ldots, a_{N-1})$ in $\mathbb{T}^N$.

The notion of Lipschitz function always refers to the ℓ_∞-distance: a function $f \colon \mathbb{T}^m \to \mathbb{T}^n$ is said to be κ-lipschitz if $\|f(x) - f(y)\|_\infty \leq \kappa \, \|x - y\|_\infty$ for all inputs x, y, where $\| \cdot \|_\infty$ is the ℓ-infinity norm.

In this work, we revisit the three scale-invariant families of HE schemes (TFHE in Section 2, B/FV in Section 3 and HEAAN in Section 4), all of them based on the Ring-LWE problem introduced in [26].

More precisely, we present a slightly different interpretation of these schemes through the concept of FHE module structure that provides a more conceptual interpretation of the corresponding homomorphic evaluation of the action of a ring on a module over that ring. More importantly, in the subsequent sections, we realize each of the plaintext spaces of the schemes B/FV, TFHE and HEAAN as subsets of the $\mathcal{R}$-module $\mathbb{T}_{\mathcal{R}}$ which enables the scheme-switching algorithms.

2.1 The concept of FHE module structure

The definition below assumes that the decryption algorithms do not introduce any noise in the plaintext (i.e., the decryption is exactly the original message). Note that this allows for probabilistic encryption algorithms.

Definition 1 (**Noiseless FHE Module Structure**) By a *noiseless FHE module structure*, we mean a 7-tuple $(R, M, \mathsf{Enc}_R, \mathsf{Dec}_R, \mathsf{Enc}_M, \mathsf{Dec}_M, \boxdot)$ where

- R is a ring with an encryption scheme $(\mathsf{Enc}_R, \mathsf{Dec}_R)$ on R without decryption noise and with ciphertext space $\mathcal{C}_R$,
- M is an R-module with a homomorphic encryption scheme $(\mathsf{Enc}_M, \mathsf{Dec}_M)$ without decryption noise and with ciphertext space $\mathcal{C}_M$,
- $\boxdot \colon \mathcal{C}_R \times \mathcal{C}_M \to \mathcal{C}_M$ is an operation (external product) satisfying

$$\mathsf{Dec}_M(\mathsf{Enc}_R(r) \boxdot \mathsf{Enc}_M(m)) = r \cdot m, \qquad \forall \, r \in R, \; m \in M.$$

Although quite general, the above definition has several drawbacks as detailed through the following remarks.

Remark 1. While TFHE fits exactly in this definition (with the algorithms TRGSW and TRLWE), the scheme HEAAN falls outside of its scope, as its decryption is noisy (i.e., no rounding is used).

Remark 2. In practice, the above definition is not sufficient to model the properties of the FHE schemes with bootstrapping where the notion of noise in the decryption algorithm is of primary importance. The algebraic structures above are too general to enable a proper model for the noise distributions. Without further assumptions on M (e.g., metric properties, Gaussian distributions supported on M), this definition is incapable of:

1. Abstracting an adequate model for the correctness of the decryption.
2. Abstracting the security properties of these schemes.

To address these two major questions, the pure probabilistic approach seems insufficient. In the next section, we will restrict the above definition to $M = \mathbb{T}_{\mathcal{R}}$ and use the metric properties of the torus to define a noisy version that would address points 1. and 2. above.

2.2 TFHE

Here, we revisit the TFHE scheme using the definition of the FHE module structure. In practice, the definition of the FHE module structure does not directly apply to TFHE. One needs here a leveled version of this definition (i.e., after each operation, the standard deviation or the noise in the decryption increases until it is impossible to decrypt by rounding).

TFHE consists of three major encryption/decryption schemes, each represented by a different plaintext space. First, the scheme TLWE encrypts messages over the entire torus $\mathbb{T}$ and produces ciphertexts in $\mathbb{T}^{N+1}$. The other two schemes are:

- TRGSW encrypts elements of the ring $\mathcal{R}_{\mathbb{Z}}$ (integer polynomials) with bounded ℓ^∞-norms (of the corresponding vectors in $\mathbb{Z}^N$ under the natural identification $\mathcal{R} \simeq \mathbb{Z}^N$).
- TRLWE encrypts elements μ of the $\mathcal{R}_{\mathbb{Z}}$-module $\mathbb{T}_{\mathcal{R}}$ that can also be viewed as elements of $\mathbb{T}^N$ via the natural bijection $\mathbb{T}_{\mathcal{R}} \simeq \mathbb{T}^N$.

There is an external product $\boxdot_\alpha$ depending on a noise parameter α [18, Cor. 3.14] that we recall in detail in Theorem 1 below. This theorem yields a FHE module structure on the schemes TRGSW and TRLWE.

In TFHE, TLWE ciphertexts of a message $\mu \in \mathbb{T}$ have the form $(\mathtt{a}, \mathtt{b} = \langle \mathtt{s}, \mathtt{a} \rangle + \mu + e) \in \mathbb{T}^{N+1}$ where $\mathtt{s} \in \{0,1\}^N$ is the secret key, $\mathtt{a} \in \mathbb{T}^N$ is uniformly random and $e \in \mathbb{T}$ is sampled according to a noise distribution centered at zero. Similarly, for TRLWE, ciphertexts of $\mu \in \mathbb{T}_{\mathcal{R}}$ are of the form $(a, b = s \cdot a + \mu + e) \in \mathbb{T}_{\mathcal{R}}^2$ where $s \in \mathcal{B}$, $a \in \mathbb{T}_{\mathcal{R}}$ is uniformly random and $e \in \mathbb{T}_{\mathcal{R}}$.

The decryption in TLWE (resp. TRLWE) uses a secret κ-Lipschitz function (here, $\kappa > 0$ is small and we mean "with respect to the ℓ^∞-norm on the torus") $\varphi_{\mathbf{s}} \colon \mathbb{T}^N \times \mathbb{T} \to \mathbb{T}$ (resp. $\varphi_s \colon \mathbb{T}_{\mathcal{R}} \times \mathbb{T}_{\mathcal{R}} \to \mathbb{T}_{\mathcal{R}}$) called *phase* parametrized by a small (often binary) secret key $\mathtt{s} \in \{0,1\}^N$ (resp. $s \in \mathcal{B}$) and defined by $(\mathtt{a}, \mathtt{b}) \in \mathbb{T}^N \times \mathbb{T} \mapsto \mathtt{b} - \langle \mathtt{s}, \mathtt{a} \rangle$ (resp. $(a, b) \mapsto b - s \cdot a$). The fact that the phase is a κ-Lipschitz function for small $\kappa \leq N + 1$ makes the decryption tolerant to errors and allows working with approximated numbers.

Ciphertexts are either fresh (i.e., generated by directly encrypting a plaintext) or they are produced by a sequence of homomorphic operations. In both cases, one views the ciphertext as a random variable depending on the random coins used to generate a and e as well as all random coins used in all these homomorphic operations.

Since $\varphi_s(a, b) = b - s \cdot a = \mu + e$, the decryption μ and the noise parameter α are the mean and the standard deviation of the phase function $\varphi_s(a, b)$, respectively (here, the mean and standard deviation are computed over the random coins in the encryption algorithm).

The expectation, variance and standard deviation on the torus are well defined only for concentrated distributions (defined in [18, 2.1]) whose support is included in a ball of radius $1/4$ (up to negligible tails). This is the case of the error distribution of T(R)LWE. More information on the definition of expectation and standard deviation for concentrated distributions on the torus can be found in [18, 19]. The benefit of the definition of the message as the expectation of the phase is that it is valid with infinite precision on any (discrete or non-discrete) subset of $\mathbb{T}_{\mathcal{R}}$. Note that this definition is only useful for analysis (e.g., proving the correctness of the cryptosystem) and cannot be used for decryption since the expectation of the phase cannot be computed in practice from a single sample of the distribution.

Below, we describe the parameters and the algorithms that are used for TFHE with the TRLWE encryption scheme.

Parameters: A security parameter λ and a minimal noise parameter α. These parameters implicitly define a minimal key size N. For more details see the FHE standardization workshop security document [2].

KeyGen/Phase: A uniformly random binary key $s \in \mathcal{B}$. This key implicitly defines the secret *phase* function

$$\varphi_s \colon \mathbb{T}_{\mathcal{R}}^2 \to \mathbb{T}_{\mathcal{R}}, (a, b) \mapsto (b - s \cdot a).$$

Encrypt (μ, s, α): To encrypt a message $\mu \in \mathbb{T}_{\mathcal{R}}$, choose a uniformly random $a \in \mathbb{T}_{\mathcal{R}}$ and a small Gaussian error $e \leftarrow \mathcal{D}_{\mathbb{T}_{\mathcal{R}}, \alpha}$, and return

$$\mathbf{Encrypt}(\mu, s, \alpha) \underset{\text{def}}{=} (a, s \cdot a + \mu + e).$$

DecryptApprox(c, s): (approx) Return $\varphi_s(c)$ which is close to the actual message.

Decrypt(c, s, $\mathcal{M}$): (rounded) Round $\varphi_s(c)$ to the nearest point in $\mathcal{M}$. Here, $\mathcal{M} \subset \mathbb{T}_{\mathcal{R}}$ is subset of plaintext messages and the nearest point is with respect to the distance function on $\mathbb{T}_{\mathcal{R}} \simeq \mathbb{T}^N$.

Message$_s$(c): (probabilistic) This is the expectation of $\varphi_s(c)$ (with respect to the random coins used in the noise). It can be viewed as a perfect decryption algorithm with infinite precision.

Public linear combination over $\mathcal{R}_{\mathbb{Z}}$: return $\sum_{i=1}^{k} a_i \cdot c_i$, where $c_i \in \mathbb{T}^2_{\mathcal{R}}$ and $a_i \in \mathcal{R}_{\mathbb{Z}}$ for $i = 1, \ldots, k$.

External product: Given a TRGSW ciphertext A encrypting a message $\mu_A \in \mathcal{R}_{\mathbb{Z}}$ and a TRLWE ciphertext b of a message $\mu_b \in \mathbb{T}_{\mathcal{R}}$, compute $A \boxdot_\alpha b$ at precision $\alpha > 0$, which encrypts $\mu_A \cdot \mu_b \in \mathbb{T}_{\mathcal{R}}$ (see [18] and Theorem 1)

$$\boxdot_\alpha : \mathsf{TRGSW} \times \mathsf{TRLWE} \longrightarrow \mathsf{TRLWE}.$$

Apply a $\mathbb{Z}$-module morphism f: use a key-switch algorithm (Theorem 2 and described in Algorithm 2 in [18]).

SampleExtract$_i$(c) Given a TRLWE ciphertext c of a message $\mu \in \mathcal{R}_{\mathbb{Z}}$, extract from c the TLWE sample that encrypts the ith coefficient μ_i with at most the same noise variance or amplitude as c (see [18] Section 4.2), defined as:

$$\mathbf{SampleExtract}_i(c) \underset{\text{def}}{=} ((a_i, a_{i-1}, \ldots, a_{i-N+1}), b_i).$$

To ease the reading of this paper, we specify only one particular instantiation of TFHE[1], all bit-decompositions are binary. We present all theorems with decomposition in base 2 (for bootstrapping, key-switch, external product and bitdecomp) to minimize the number of parameters in the formulas. To perform an operation at precision $\alpha \ll 1$ (i.e. noise standard deviation α during the operation), we always use $\ell = -\log_2(\alpha)$ terms in every bit decomposition.

The primary definition of α is the noise's standard deviation during the current operation: α is thus not a static parameter of the framework. Then, the notions of noise rate, precision and approximate decomposition error, key-switch noise and bootstrapping noises are equal to α or proportional to it.

Any TLWE, TRLWE, TRGSW ciphertext, bootstrapping key or key-switching key given at a higher precision, can always be rounded and truncated to match the current precision α. Working with precision α implies a minimal size for the current binary key (as a simple rule of thumb, the minimal key size N that provides 128 bits of security is roughly the smallest power of two larger than $\max(256, 40|\log_2 \alpha|)$. The actual key size should be determined either from the standardization document [2] or from the LWE estimator by Albrecht et al. [3]. Whenever α varies (e.g. increases after each multiplication, or decreases after a bootstrapping), we always use the last key-switching and bootstrapping operation to switch to the smallest possible key that matches from the security estimates.

External product

There is a compatibility between the ciphertexts of TRGSW and TRLWE that can be expressed algebraically in the following manner: observe that the plaintext space for TRGSW is the ring $\mathcal{R}$ and the plaintext space for TRLWE is the $\mathcal{R}$-module $\mathbb{T}_{\mathcal{R}}$. In this case, one can define the following external product (a homomorphic action) (first introduced in [7] and formalized on the torus in [19]) of $\mathcal{R}$ on $\mathbb{T}_{\mathcal{R}}$:

Theorem 1. (*External product [18, Cor. 3.14]*) *Let c_r be a* TRGSW *ciphertext of a message $r \in \mathcal{R}_{\mathbb{Z}}$ and let c_m be a* TRLWE *ciphertext of a message $m \in \mathbb{T}_{\mathcal{R}}$ such that c_m is independent of the random coins used for c_r. There exists a homomorphic external product algorithm (described explicitly in [18, Sec. 3.3]) and denoted by $c_r \boxdot_\alpha c_m$*

1 With this choice, the parameters from [18] correspond to: $k = 1$, $\beta = 1$, $Bg = 2$, $\ell = -\log(\alpha)$, $t = \ell$, $n = N$, $VKS = VBK = \alpha^2$, which implies $\epsilon = \alpha/2$. As usual, a non binary decomposition is possible and gives small poly-logarithmic improvements.

such that $\mathit{Message}_s(c_r \boxdot_\alpha c_m) = r \cdot m$ *with noise standard deviation* $\alpha > 0$ *and*

$$\mathit{Var}(\mathit{Err}(c_r \boxdot_\alpha c_m)) \le 2\ell N \mathit{Var}(\mathit{Err}(c_r)) + \frac{1+N}{4}\|r\|_2^2\,\alpha^2 + \|r\|_2^2\,\mathit{Var}(\mathit{Err}(c_m)).$$

In general, this theorem will be used to multiply a TRLWE ciphertext c_m by a precomputed TRGSW ciphertext c_r (e.g., a ciphertext of the binary secret key in the case of bootstrapping): in this case, we can choose $\mathsf{Var}(\mathsf{Err}(c_r)) = \alpha^2$, and we have $\|r\|_2^2 \le N$ (or even $\|r\|_2^2 = 1$ if TRGSW ciphertexts encrypt only single bits, as in [19] and [18]). In this case, the working precision α equals the targeted output precision divided by N, so that the first two error terms in the theorem remain negligible.

Key-switching

In order to switch between the scalar and polynomial message spaces $\mathbb{T}$ and $\mathbb{T}_{\mathcal{R}}$, the authors of [18] generalized the notions of sample extraction and key-switching. On one side, a $\mathsf{PubKS}(f, \mathsf{KS}, \mathrm{c}_1, \ldots, \mathrm{c}_n)$ algorithm homomorphically evaluates linear morphisms f from any $\mathbb{Z}$-module $\mathbb{T}^n$ to $\mathbb{T}_N[X]$ using the functional key-switching key KS. It is possible to evaluate also a private linear morphism, but the algorithm is slower.

Theorem 2. (*Functional key-switch adapted from [18, Thm. 4.2]*) *Given r* TLWE *ciphertexts* $\mathrm{c}_i \in \mathsf{TLWE}_S(\mu_i)$, *a public* κ*-Lipschitz* $\mathbb{Z}$*-module homomorphism*

$$f\colon \mathbb{T}^r \to \mathbb{T}_{\mathcal{R}}$$

and $\mathsf{KS}_{i,j} \in \mathsf{T(R)LWE}_{K,\alpha}(\frac{S_i}{2^j})$ *with standard deviation* α *and* S_i *the i-th coefficient of the key S, Algorithm 2 described in [18] outputs a* T(R)LWE *sample* $\mathbf{c} \in \mathsf{T(R)LWE}_K(f(\mu_1, \ldots, \mu_r))$ *such that*

$$\mathsf{Var}(\mathsf{Err}(\mathbf{c})) \le \kappa^2 \max(\mathsf{Var}(\mathsf{Err}(\mathrm{c}_i))) + \alpha^2(\ell N^2 + \frac{N}{4}).$$

Non-linear functions

In TFHE, a negacyclic function $f\colon \mathbb{T} \to \mathbb{T}$ (i.e. $f(x+1/2) = -f(x)$) can be homomorphically applied to the phase of an individual TLWE ciphertext $\mathrm{c} \in \mathbb{T}^{n+1}$ with a n-bit key K where each individual coefficient is rounded to the nearest multiple of $1/2N$. More precisely, given the values $f(i/2N)$ for $i = 0, 1, \ldots, 2N-1$, [18, Alg.4] outputs a ciphertext $\mathrm{c}' \in \mathbb{T}^{N+1}$ of the plaintext $f(\lfloor\varphi_K(c)\rceil)$ encrypted with key $S \in \{0,1\}^N$.

Theorem 3. (*Functional bootstrap adapted from [18, Thm. 4.3]*) *Given a* TLWE *ciphertext* $\mathrm{c} \in \mathbb{T}^{n+1}$ *encrypted with an n-bit key* $K \in \{0,1\}^n$ *where each individual coefficient of* c *is rounded to the nearest multiple of* $1/2N$, *a negacyclic function* $f\colon \mathbb{T} \to \mathbb{T}$ *restricted to* $(2N)^{-1}\mathbb{Z}/\mathbb{Z} \subset \mathbb{T}$ *and a bootstrapping key* $BK = \mathsf{TRGSW}_S(K_i)$ *with standard deviation* α, *Algorithm 4 described in [18] outputs a* TLWE *sample* $\mathrm{c}' \in \mathbb{T}^{N+1}$ *encrypting* $f(\varphi_K(\mathrm{c}))$ *with key* $S \in \{0,1\}^N$ *such that:*

$$\mathsf{Var}(\mathsf{Err}(\mathrm{c}')) \le \alpha^2 n(2\ell N + N + \frac{5}{4} + \ell).$$

Until now, the most frequent non-linear function used in bootstrapping is the rounding function, since rounding the phase of a ciphertext is equivalent to decrypting it, and homomorphic decryption is the noise-reduction method proposed by Gentry in 2009 [23].

2.3 B/FV (Brakerski, Fan–Vercauteren, and Seal).

In this scheme, the message space is the finite ring $\mathcal{R}_p = (\mathbb{Z}/p\mathbb{Z})[X]/(X^N+1)$ for some integer p (typically a power of 2 or a prime number). A message $\mu \in \mathcal{R}_p$ is encrypted on a quotient ring $\mathcal{R}_q$ (for a larger modulus q) as a ciphertext $(a, b) \in \mathcal{R}_q^2$ where $a \in \mathcal{R}_q$ is chosen uniformly at random and b is sampled from $\mathcal{D}_{\mathcal{R}_q,\sigma,s\cdot a+\frac{p}{q}\mu}$. Here, $\mathcal{D}_{\mathcal{R}_q,\sigma,\mu}$ is the discrete Gaussian distribution over $\mathcal{R}_q$ centered at μ with standard deviation σ (discrete means that the values are integers only). In addition, $s \in \mathcal{B}$ is the secret key.

Homomorphic addition of two ciphertexts (a_1, b_1) and (a_2, b_2) is achieved by component-wise addition. The idea behind the homomorphic multiplication of two ciphertexts (a_1, b_1) and (a_2, b_2) is a technique referred to as *relinearization*: one first lifts $(a_i, b_i) \in \mathcal{R}_q^2$ to $(\widetilde{a}_i, \widetilde{b}_i) \in \mathcal{R}^2$ where each coefficient is lifted to $[-q/2, q/2)$ and then view of μ_i as being expressed as a linear polynomial on s (i.e., $\mu_i \sim \frac{p}{q}(b_i + s \cdot a_i)$). One then computes the quadratic polynomial corresponding to the product, namely $\frac{p}{q}(b_1 + s \cdot a_1) \cdot \frac{p}{q}(b_2 + s \cdot a_2)$, and uses the relinearization described in [21, p.7–9] to write this product as $\frac{p}{q}(b + s \cdot a)$ and determine the coefficients $(a, b) \in \mathcal{R}_q$.

The noise amplitude grows by a small factor $O(N)$ on average after each multiplication, so it is a common practice to perform a *modulus-rescaling* step, that divides and rounds each coefficient as well as the modulus q by the same scalar in order to bring the noise amplitude back to $O(1)$ so that the subsequent operations continue on smaller ciphertexts. For more details and formal definitions see [6, 21].

We will show in Section 3 how to embed the plaintext space of B/FV in $\mathbb{T}_{\mathcal{R}}$ and how to then use the TFHE module structure to evaluate the above B/FV homomorphic product in a natural way.

2.4 HEAAN

In this scheme, the message space is the subset of $\mathcal{R}_q$ containing all elements of norm $\leq B$ for some bound B, where the norm of an element $x \in \mathcal{R}_q$ is defined as $\|\widetilde{x}\|_\infty$. Here $\widetilde{x} \in \mathcal{R}_{\mathbb{R}}$ is the minimal lift of x, i.e., coefficients lifted to $[-q/2, q/2)$. The message is decrypted up to a constant number of least significant bits which are considered as noise.

A HEAAN ciphertext is also a Ring-LWE pair $(a, b) \in \mathcal{R}_q^2$ where $a \in \mathcal{R}_q$ is uniformly random and b is equal to $s \cdot a + \mu$ up to a Gaussian error of small standard deviation. This time, plaintexts and ciphertexts share the same space, so no rescaling factor p/q is used. Multiplication of two messages uses the same formula as in B/FV including relinearization: if both input messages are bounded by B with $O(1)$ noise, the product is a message bounded by B^2 with noise $O(B)$, so it is a common practice at this point to perform a modulus-rescaling step that divides everything by B to bring the noise back to $O(1)$ (see [16]). Unlike B/FV, this division in the modulus switching scales not only the ciphertext but also the plaintext by B. This can be fixed by adding a (public) tag to the ciphertext to track the number of divisions by B performed (see more details in Section 4).

2.5 $\mathbb{T}_{\mathcal{R}}$ as the common plaintext algebraic structure

As mentioned earlier, the three homomorphic schemes use nearly the same ciphertext space (up to rescaling by a factor q). In addition, all decryption methods make use of the phase function up to minor differences: $b - sa$ versus $b + sa$ in B/FV or $\sum_{i=0}^{m} s^i \cdot a_i$, $s \in \mathcal{R}$, $a_i \in \mathbb{T}_{\mathcal{R}}$ for various values of m in Seal [13]. Yet, the plaintext spaces are *à priori* different as they are based on different mathematical structures (groups, rings, intervals, random sets).

In order to exploit the advantages of each scheme, we need to be able to homomorphically switch between the different plaintext spaces, and most importantly, to give a meaningful semantic to these transformations.

The main idea that enables us to give a meaningful semantic to the scheme switching transformation is to interpret all the plaintext spaces as being embedded in the same module $\mathbb{T}_{\mathcal{R}}$ as shown in Figure 2 and to use the distance function on the torus to quantify the transformation error. In this setting, all schemes use the same ciphertext space $\mathbb{T}_{\mathcal{R}}^2$, the same key space $\mathcal{B}$ and the same phase function $\varphi_s(a, b) = b - s \cdot a$. Thus, the probabilistic characterization of the message and error as the expectation and the standard deviation of the TFHE phase, respectively, are automatically extended to all schemes.

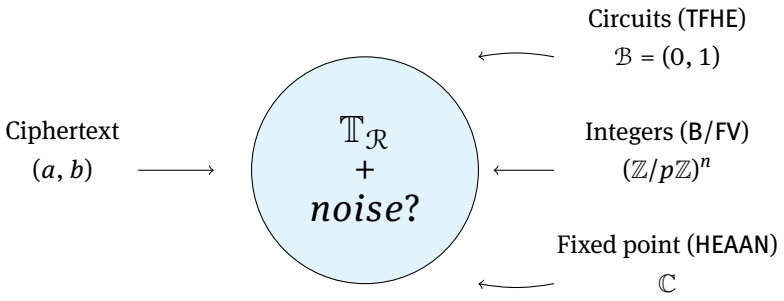

Figure 2: Representation of the plaintext space over the $\mathbb{T}_{\mathcal{R}}$

2.6 Real and complex slots for B/FV, TFHE and HEAAN

In this section we recall the representation of the plaintext spaces of the three schemes via the *user slots*. This is indeed the representation used in the current implementations.

B/FV

In B/FV, homomorphic operations are performed either on N SIMD slots modulo a medium-size integer p [13, 21], or more recently, on a single big-number slot modulo $p^N + 1$ [14]. In both cases, the set of these slots has a ring structure and is isomorphic to a quotient of $\mathcal{R}_{\mathbb{Z}}$ by a principal ideal (the *native plaintext* in [14]).

HEAAN

In HEAAN, homomorphic operations are performed on $N/2$ SIMD slots containing complex numbers in fixed-point representation with the same public exponent and the same precision. By interpolating the complex roots of $X^N + 1$, the native plaintext can be mapped to small polynomials of $\mathbb{T}_{\mathcal{R}}$ (i.e., a zero-centered ball of small fixed radius) since the complex DFT matrix is Hermitian. We achieve this in Section 4.

TFHE

Finally, in TFHE, the message space is an arbitrary subset of $\mathbb{T}_{\mathcal{R}}$, without any particular structure.

Preserving the user slots

In order to introduce the notion of slots (real or complex), we use the following two isomorphisms of $\mathbb{R}$-vector spaces:

$$\mathcal{R}_{\mathbb{R}} \simeq \mathbb{R}^N, \qquad f = a_0 + \cdots + a_{N-1}X^{N-1} \mapsto (a_0, \ldots, a_N), \tag{1}$$

and

$$\mathcal{R}_{\mathbb{R}} \simeq \mathbb{C}^{N/2}, \qquad f \mapsto (f(\zeta), f(\zeta^3), \ldots, f(\zeta^{N-1})). \tag{2}$$

Here $\zeta = e^{\pi i/N}$ is a primitive root of $X^N + 1$. Representation (1) corresponds to what is called the *coefficient packing* and representation (2) corresponds to what is called the *slot packing*.

The unified native plaintext spaces and the correspondence with the user-space slots are shown in Figure 3.

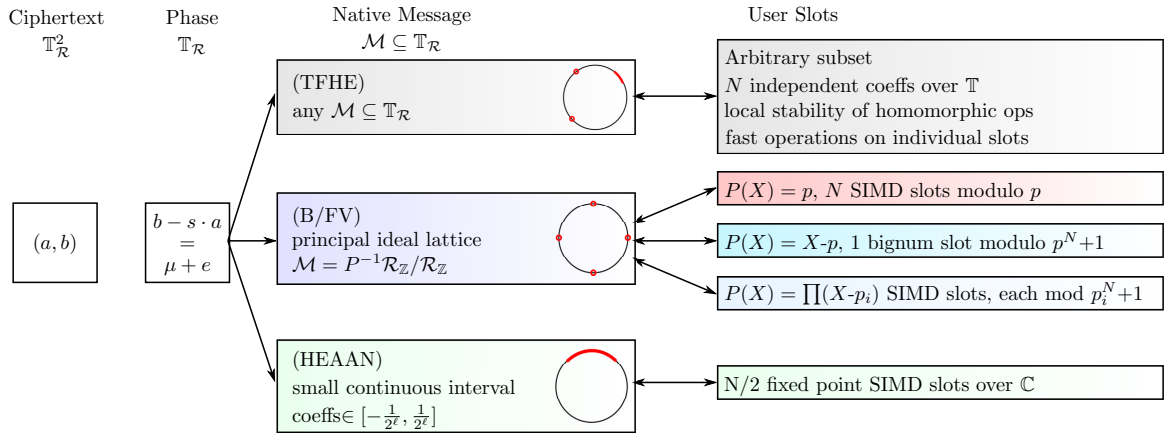

Figure 3: Unifying the plaintext space in RLWE-schemes. See the following sections for the definition of the notation.

3 A general abstraction of B/FV over the torus

The B/FV scheme is very efficient in evaluating arithmetic circuits (i.e. polynomials on the plaintext). Yet, huge slowdowns are observed when evaluating comparisons, the sign function or other non-linear decision diagrams that do not correspond to sparse low-degree polynomials.

Here, we propose an alternative approach that switches to a different scheme, and for instance, executes the non-arithmetic operation via TFHE's gate bootstrapping. We explain how to represent the plaintext spaces in order to enable this conversion.

Originally, the construction of B/FV uses $\mathcal{R}_p$ as the plaintext space, where $p \geq 2$ is a plaintext modulus. For more flexibility (see, e.g., the B/FV with big numbers below), one can use an arbitrary element $P \in \mathcal{R}$ and take $\mathcal{R}_{\mathbb{Z}}/P\mathcal{R}_{\mathbb{Z}}$ as the plaintext space.

Following the ideas of [14, 22], given an element $P \in \mathcal{R}_{\mathbb{Z}}$ that is invertible in $\mathcal{R}_{\mathbb{R}}$, let $\mathcal{L}(P)$ and $\mathcal{L}(P^{-1})$ be the real lattices generated by the rows of the matrices associated to P and P^{-1}, respectively. Recall that these are the matrices (with respect to the standard basis $\{1, X, \ldots, X^{N-1}\}$) corresponding to the linear transformation of the $\mathbb{R}$-vector space $\mathcal{R}_{\mathbb{R}}$ that is multiplication by P. The native plaintext space $\mathcal{M}$ is precisely the subgroup $P^{-1}\mathcal{R}_{\mathbb{Z}}/\mathcal{R}_{\mathbb{Z}} \subset \mathcal{R}_{\mathbb{R}}/\mathcal{R}_{\mathbb{Z}} = \mathbb{T}_{\mathcal{R}}$. We then have

$$P^{-1}\mathcal{R}_{\mathbb{Z}}/\mathcal{R}_{\mathbb{Z}} \simeq \mathcal{R}_P := \mathcal{R}_{\mathbb{Z}}/P\mathcal{R}_{\mathbb{Z}},$$

where the isomorphism is induced by $u \mapsto P \cdot u$ for $u \in P^{-1}\mathcal{R}_{\mathbb{Z}}/\mathcal{R}_{\mathbb{Z}}$. Thus, $\mathcal{M} \simeq \mathcal{R}_P$. Geometrically, $\mathcal{M}$ can be thought of as the vectors in the lattice $\mathcal{L}(P^{-1})$ whose coordinates are taken modulo $\mathbb{Z}$. Algebraically, $\mathcal{M}$ is the P-torsion of the $\mathbb{T}_{\mathcal{R}}$.

Via this isomorphism, the native plaintext space $\mathcal{M}$ is equipped with the following internal Mongomery-type product:

Definition 2 (Native plaintext product) Given an element $P \in \mathcal{R}_{\mathbb{Z}}$ as above, we define a product $\boxtimes_P \colon \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ by $\mu_1 \boxtimes_P \mu_2 := P \cdot \mu_1 \cdot \mu_2$. Here, the product $P \cdot \mu_1 \cdot \mu_2$ is defined in $\mathcal{R}_{\mathbb{R}}$ by choosing arbitrary lifts of μ_1, μ_2 to $P^{-1}\mathcal{R}_{\mathbb{Z}}$. Notice that this definition is independent of the choices of these lifts.

Multiplication in B/FV

We will now deploy our definition of an FHE module structure to recover the internal homomorphic product of B/FV in terms of the external product. We view the plaintext space $M := \mathcal{R}_P$ as an $R := \mathcal{R}_{\mathbb{Z}}$-module and let $s \in \mathcal{B} \subset \mathcal{R}_{\mathbb{Z}}$ be the key.

Given a TRLWE ciphertext $c = (a, b) \in \mathbb{T}^2_{\mathcal{R}}$, we denote by $(\widetilde{a}, \widetilde{b}) \in \mathcal{R}^2_{\mathbb{Z}}$ the *optimal lift* of c to $\mathcal{R}^2_{\mathbb{R}}$, that is the pair of the unique lifts of c for which all coefficients of $\widetilde{a}$ and $\widetilde{b}$ are in $[-1/2, 1/2)$. Let $(a_1, b_1), (a_2, b_2) \in \mathbb{T}^2_{\mathcal{R}}$

be two ciphertexts of plaintexts $\mu_1, \mu_2 \in \mathcal{M} \subset \mathbb{T}_{\mathcal{R}}$. We are searching for a ciphertext (a, b) that is a valid encryption of the product $\mu_1 \boxtimes_P \mu_2$. Since $\mu_i = b_i - s \cdot a_i$ for $i = 1, 2$, we can write

$$\mu_1 \boxtimes_P \mu_2 = P\left(\widetilde{b}_1\widetilde{b}_2 - s \cdot (\widetilde{a}_1\widetilde{b}_2 + \widetilde{a}_2\widetilde{b}_1) + s^2\widetilde{a}_1\widetilde{a}_2\right).$$

We would like to define the internal product $(a_1, b_1) \boxtimes_P (a_2, b_2)$ as a pair (a, b) with the property that

$$\mathsf{Dec}_M(a, b) = P\left(\widetilde{b}_1\widetilde{b}_2 - s \cdot (\widetilde{a}_1\widetilde{b}_2 + \widetilde{a}_2\widetilde{b}_1) + s^2\widetilde{a}_1\widetilde{a}_2\right)$$

The important point here is that we would like to find $a, b \in \mathbb{T}_{\mathcal{R}}$ such that

$$a - s \cdot b = P\left(\widetilde{b}_1\widetilde{b}_2 - s \cdot (\widetilde{a}_1\widetilde{b}_2 + \widetilde{a}_2\widetilde{b}_1) + s^2\widetilde{a}_1\widetilde{a}_2\right).$$

This would have been trivial without the s^2 term. To deal with the latter, we use an encryption $\mathsf{Enc}_R(s) =: \mathsf{RK}$ (the relinearization key) and the definition of an FHE module structure to deduce that

$$s^2\widetilde{a}_1\widetilde{a}_2 = \mathsf{Dec}_M(\mathsf{Enc}_R(s) \boxdot \mathsf{Enc}_M(s\widetilde{a}_1\widetilde{a}_2)).$$

To summarize, we get

$$(a, b) = (\widetilde{a_1}\widetilde{b_2} + \widetilde{a_2}\widetilde{b_1}, \widetilde{b}_1\widetilde{b}_2) + \mathsf{RK} \boxdot (\widetilde{a}_1\widetilde{a}_2, 0).$$

Alternatively, the term $s^2\widetilde{a}_1\widetilde{a}_2$ can also be computed as $\mathsf{RK}' \boxdot \mathsf{Enc}_M(\widetilde{a}_1\widetilde{a}_2)$ where $\mathsf{RK}' = \mathsf{Enc}_R(s^2)$.

Letting RK be a relinearization key as above, that is a TRGSW encryption of s with key s (denoted by $\mathsf{TRGSW}_s(s)$), this analysis in the noiseless setting motivates the following definition of a B/FV internal homomorphic product in the presence of noise:

Definition 3 (Internal homomorphic product) We define the internal homomorphic product between two ciphertexts $c_1, c_2 \in (\mathbb{T}_{\mathcal{R}})^2$ as $c_1 \boxtimes_{P,\alpha} c_2$ as follows:

$$c_1 \boxtimes_{P,\alpha} c_2 = (C_1, C_0) - RK \boxdot_\alpha (C_2, 0), \tag{3}$$

where $C_0 = P \cdot \widetilde{b}_1 \cdot \widetilde{b}_2$, $C_1 = P \cdot (\widetilde{a}_1 \cdot \widetilde{b}_2 + \widetilde{a}_2 \cdot \widetilde{b}_1)$ and $C_2 = P \cdot \widetilde{a}_1 \cdot \widetilde{a}_2$.

Note that this definition of the internal homomorphic B/FV product relies on a precomputed TRGSW external product, or homomorphic action, which is faster than the internal product. The relation between the two products is possible thanks to the common plaintext space for B/FV and TFHE. A formulation closer to the original B/FV relinearization presented in [6], [21] would rather take RK′ an encryption of s^2 and compute $(C_1, C_0) + \mathsf{RK}' \boxdot_\alpha (0, C_2)$.

Yet, this approach generates more noise than using directly the encryption of s since the propagation depends on the norm of the plaintext in TRGSW.

Proposition 1 (B/FV noise propagation) *Let* RK *be a relinearization key, let c_1, c_2 be two* TRLWE *ciphertexts of $\mu_1, \mu_2 \in \mathcal{M} = P^{-1}\mathcal{R}_{\mathbb{Z}}/\mathcal{R}_{\mathbb{Z}}$, respectively, with the same key $s \in \mathcal{B}$ as in* Definition *3. The internal homomorphic product $c_1 \boxtimes_{P,\alpha} c_2$ is then an encryption of $\mu_1 \boxtimes_P \mu_2$ and the noise variance satisfies*

$$Var(Err(c_1 \boxtimes_{P,\alpha} c_2)) \leq \frac{1+N+N^2}{2} \|P\|_2^2 \max(Var(Err(c_i)) + \left(2\ell N + \frac{N^2+N}{4}\right)\alpha^2. \tag{4}$$

Proof. Let $\mu_1, \mu_2, e_1, e_2 \in \mathcal{R}_{\mathbb{R}}$ be the smallest representatives of the message and error of c_1 and c_2, respectively. By definition, for each $i = 1, 2$, we have $b_i - s \cdot a_i = \mu_i + e_i + I_i$ where I_i is an integer and the variance of I_i is $\leq N$.

$$\begin{aligned}
\varphi_s(c_1 \boxtimes_{P,\alpha} c_2) &= C_0 + s \cdot C_1 + s^2 \cdot C_2 + \mathsf{Err}(RK \boxdot_\alpha (0, C_2)) \mod \mathbb{Z} \\
&= (b_1 - sa_1)(b_2 - sa_2)P + \mathsf{Err}(RK \boxdot_\alpha (0, C_2)) \mod \mathbb{Z} \\
&= (\mu_1 + e_1 + I_1)(\mu_2 + e_2 + I_2)P + \mathsf{Err}(RK \boxdot_\alpha (C_2, 0)) \mod \mathbb{Z} \\
&= \mu_1\mu_2P + e_1\mu_2P + e_2\mu_1P + e_1e_2P + e_1I_2P + e_2I_1P
\end{aligned}$$

$$
\begin{aligned}
&+ \mathsf{Err}(RK \boxdot_\alpha (C_2, 0)) \mod \mathbb{Z} \\
&= \mu_1 \boxtimes_P \mu_2 + e_1\mu_2 P + e_2\mu_1 P + e_1 e_2 P + e_1 I_2 P + e_2 I_1 P \\
&+ \mathsf{Err}(RK \boxdot_\alpha (C_2, 0)) \mod \mathbb{Z}
\end{aligned}
$$

Taking the expectation, since all multiples of e_i as well as $\mathsf{Err}(RK \boxdot (0, C_2))$ have a null expectation, the message of $c_1 \boxtimes_{P,\alpha} c_2$ is $\mu_1 \boxtimes_P \mu_2$. By bounding the variance of each error term, we prove Eq. (4). □ □

The working precision α of the input ciphertexts is set in a way that the term in α^2 remains negligible compared to the first one in (4). Thus, the noise standard deviation multiplicative overhead is bounded by $O(N||P||_2)$ in the average case.

3.1 Bridging B/FV slots with TFHE

In the classical description of B/FV or Seal, $P(X)$ is usually chosen as a constant integer p. In this case, the plaintext space $\mathcal{M}$ consists in all the multiples of $P^{-1} = p^{-1} \bmod (X^N + 1) \mod \mathbb{Z}$, which is a rescaling of the classical plaintext space description $\mathcal{R}_p$ (as shown in Figure 4). In particular, if $X^N + 1$ has N roots modulo p, $\mathcal{M}$ is isomorphic to N independent integer slots modulo p (else, there are less slots, in extension rings or fields). From a lattice perspective, $\mathcal{M}$ is viewed as the orthogonal lattice generated by $1/pI_N$ (I_N is the $N \times N$ identity matrices). The packing radius of $\mathcal{M}$ is $1/2p$ which is the maximal error tolerable by the phase. Rounding an element to $\mathcal{M}$ consists of rounding each coordinate independently to the nearest multiple of $1/p$.

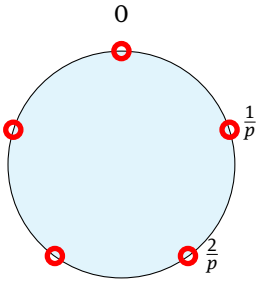

Figure 4: The B/FV plaintext space over the $\mathbb{T}_\mathcal{R}$

In the literature, the isomorphism used to obtain the slot representation is

$$p^{-1}\mathcal{R}_\mathbb{Z}/\mathcal{R}_\mathbb{Z} \simeq \mathcal{R}_p \simeq (\mathbb{Z}/p\mathbb{Z})^N, \qquad \mu \mapsto (\mu(r_0), \ldots, \mu(r_{N-1})). \tag{5}$$

Here, one assumes that the polynomial $X^N + 1 \bmod p$ factorizes into distinct linear factors $X - r_0, \ldots, X - r_{N-1}$ (i.e., it has N distinct roots $r_0, \ldots, r_{N-1} \bmod p$). This isomorphism allows to manipulate N independent slots in parallel. Typical values are $N = 2^{15}$ and $p = 2^{16} + 1$ (allowing a very small noise $\alpha \approx 2^{-886}$ according to [8], so a multiplicative depth of ≈ 28 without key-switch according to the propagation of Lemma 1).

If we identify a polynomial in $\mathcal{R}_p$ with its coefficient vector in $(\mathbb{Z}/p\mathbb{Z})^N$, the bijection between the coefficients and the slot then corresponds to the Vandermonde matrix of $(r_0, \ldots, r_{N-1}) \mod p$

$$\mathsf{VDM} = \begin{bmatrix} 1 & r_0^1 & \cdots & r_0^{N-1} \\ 1 & r_1^1 & \cdots & r_1^{N-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & r_{N-1}^1 & \cdots & r_{N-1}^{N-1} \end{bmatrix} \mod p. \tag{6}$$

3.1.1 B/FV → TFHE.

In this section we show how B/FV ciphertexts, interpreted as TRLWE ciphertexts with plaintext (slots in) $\mathcal{R}_p$, can be transformed into k independent TLWE ciphertexts.

Let $z = (z_0, \ldots, z_{N-1}) \in (\mathbb{Z}/p\mathbb{Z})^N$ be a B/FV plaintext and, as in the previous section, let $p^{-1}\mathcal{R}_\mathbb{Z}/\mathcal{R}_\mathbb{Z} \simeq (\mathbb{Z}/p\mathbb{Z})^N$ be the identification (5) and let $\mu \in p^{-1}\mathcal{R}_\mathbb{Z}/\mathcal{R}_\mathbb{Z} \subset \mathbb{T}_\mathcal{R}$ be the preimage of z. More generally, suppose that we start with a $\mathbb{Z}$-module homomorphism $f\colon (\mathbb{Z}/p\mathbb{Z})^N \to (\mathbb{Z}/p\mathbb{Z})^N$ with a matrix $F \in \mathcal{M}_N(\mathbb{Z})$, where $\mathcal{M}_N(\mathbb{Z})$ is the class of integer $N \times N$ matrices (this could arise in practice as either a projection, extraction or permutation of the slots) and get the output $f(z)$ as a polynomial $\sum_{i=0}^{N-1} \mu'_i X^i \in p^{-1}\mathcal{R}_\mathbb{Z}/\mathcal{R}_\mathbb{Z}$ where[2] $\mu' := (\mu'_0, \ldots, \mu'_{N-1}) = \frac{1}{p} f(z_0, \ldots, z_{N-1}) \mod \mathbb{Z}$.

Then, by definition, we have $\mu' = (F \cdot \mathsf{VDM})\mu \mod \mathbb{Z}$, where $F \cdot \mathsf{VDM}$ can be any integer representative of $F \cdot \mathsf{VDM} \mod p$. In particular, we can always take the representative with all coefficients in $[-p/2, p/2)$. This is a $(Np/2)$-Lipschitz $\mathbb{Z}$-module homomorphism, and can be evaluated via the functional key-switch of Theorem 2 (see Algorithm 1). Coefficients can then be homomorphically extracted as individual TLWE ciphertexts with the SampleExtract of TFHE.

Algorithm 1 Public Functional Switching B/FV to TFHE

Input: TRLWE ciphertext $c(X) = (a(X), b(X))$ encoding N slots $(z_0, \ldots, z_{N-1}) \mod p$ with key $K \in \mathcal{B}$, a public $\mathbb{Z}$-module homomorphism $f\colon (\mathbb{Z}/p\mathbb{Z})^N \to (\mathbb{Z}/p\mathbb{Z})^N$ with matrix $F \in \mathcal{M}_N(\mathbb{Z})$, and key switch key $KS_{i,j} \in \mathsf{TRLWE}_S(\frac{K_i}{2^j})$, where $S \in \mathcal{B}$.

Output: a TRLWE ciphertext $c' \in \mathbb{T}^2_\mathcal{R}$ encrypted with key $S \in \mathcal{B}$ whose message is $\sum_{i=0}^{N-1} \mu'_i X^i$ where $(\mu'_0, \ldots, \mu'_{N-1}) = \frac{1}{p} f(z_0, \ldots, z_{N-1})$

1: $F' = F \cdot VDM \mod p$ (all coefficients lifted in $[-p/2, p/2)$)
2: Compute $B = F'(b(X)) \in \mathbb{T}^N$
3: **for** $i \in 0 \to N-1$ **do do**
4: Compute $A_i = F'(X^i a(X)) = A \in \mathbb{T}^N$
5: Decompose $A_i = \sum_{j=0}^{l} A_{i,j} \cdot 2^{-j}$ with $A_{i,j} \in \mathbb{B}^N$
6: **end for**
7: **return** $(0, B) - \sum_{i=0}^{N-1} \sum_{j=0}^{l} A_{i,j} \cdot KS_{i,j}$

Proposition 2 (B/FV slots → TFHE) *Let $c = (a, b) \in \mathbb{T}^2_\mathcal{R}$ be a* TRLWE *ciphertext encrypting N slots $(z_0, \ldots, z_{N-1}) \mod p$ with key $K \in \mathcal{B}$, let $f\colon (\mathbb{Z}/p\mathbb{Z})^N \to (\mathbb{Z}/p\mathbb{Z})^N$ be a public $\mathbb{Z}$-module homomorphism with matrix $F \in \mathcal{M}_N(\mathbb{Z})$. Let $\mathsf{KS}_{i,j} \in \mathsf{TRLWE}_{S,\alpha}(K_i/2^j)$ be a key switching key $(0 \le i < N,\ 1 \le j < \log_2 \alpha)$ and K_i is the i-th coefficient of K. By applying the functional key-switch of* Theorem *2 using the integer transformation $F \cdot VDM \mod p$ whose coefficients are between $[-p/2, p/2)$ and we obtain a* TRLWE *ciphertext $c' \in \mathbb{T}_\mathcal{R}$ encrypted with key $S \in \mathcal{B}$ whose message is $\sum_{i=0}^{N-1} \mu'_i X^i$ where $(\mu'_0, \ldots, \mu'_{N-1}) = \frac{1}{p} f(z_0, \ldots, z_{N-1})$. The noise variance satisfies*

$$Var(Err(c')) \le \left(\frac{Np}{2}\right)^2 Var(Err(c)) + \alpha^2 \left(\ell N^2 + \frac{N}{4}\right).$$

[2] *A priori* $f(z) \in (\mathbb{Z}/p\mathbb{Z})^N$ is identified with an element of $\mathcal{R}_p$ which is then identified with $p^{-1}\mathcal{R}_\mathbb{Z}/\mathcal{R}_\mathbb{Z}$; the coefficient vector $\mu' = (\mu'_0, \ldots, \mu'_{N-1}) \in (p^{-1}\mathbb{Z}/\mathbb{Z})^N \subset \mathbb{T}^N$ yields the output.

3.1.2 TFHE → B/FV.

Conversely, we would like to transform k independent TLWE ciphertexts (encryptions of messages $(\mu_0, \dots, \mu_{k-1}) \in \mathbb{T}^k$) into a TRLWE ciphertext with slots in $\mathbb{Z}/p\mathbb{Z}$. Again, we will need to define a Lipschitz $\mathbb{Z}$-module homomorphism $g\colon \mathbb{T}^k \to (\mathbb{Z}/p\mathbb{Z})^N$. Unfortunately, since for all $x \in \mathbb{T}^k$ there exists $y \in \mathbb{T}$ such that $x = p \cdot y$, we have $g(x) = p \cdot g(y) = 0$ in $(\mathbb{Z}/p\mathbb{Z})^N$ and this implies that g is zero everywhere, which is of limited interest.

Therefore, we need to restrict the message space only to multiples of $1/p$ (this prevents division by p). Such a plaintext space restriction may imply that input TLWE ciphertexts must be bootstrapped before exporting them as B/FV slots using gate bootstrapping from Theorem 3. Note that B/FV manages with a noise that is smaller than in the TFHE. Therefore for this transformation, it is not enough to only switch between different ciphertexts types, but also to decrease the noise. Also the bootstrapping of the input permits to map the plaintext space to the space composed of exact multiples of $1/p$.

Then, let $g\colon (\mathbb{Z}/p\mathbb{Z})^k \to (\mathbb{Z}/p\mathbb{Z})^N$ be a $\mathbb{Z}$-linear map whose matrix G is in $\mathcal{M}_{N,k}(\mathbb{Z})$. To obtain a B/FV ciphertext whose slots are $g(p\mu_0 \mod p, \dots, p\mu_{k-1} \mod p)$, the actual transformation to apply is $\mathsf{VDM}^{-1} \cdot G \mod p$ (see Algorithm 2). Again, we can choose the representative with coefficients in $[-p/2, p/2)$ which is $(Np/2)$-Lipschitz.

Algorithm 2 Public Functional Switching TFHE to B/FV

Input: $k < N$ TLWE ciphertexts (a_i, b_i) encoding $\mu_i \in \mathcal{M}_{N,k}(\mathbb{Z})$ with key $S \in \mathbb{B}^n$, a public $(Np/2)$-Lipschitz $\mathbb{Z}$-module morphism $g : (\mathbb{Z}/p\mathbb{Z})^k \to (\mathbb{Z}/p\mathbb{Z})^N$, whose matrix $G \in \mathcal{M}_{N,k}(\mathbb{Z})$ and $KS_i = \mathsf{TRGSW}_\mathsf{K}(S_i)$ with $K \in \mathcal{B}$ and $1 \le i \le n$.

Output: a B/FV ciphertext $(a(X), b(X))$ encoding a plaintext with slots are $g(p\mu_0 \mod p, \dots, p\mu_{k-1} \mod p)$, with key $K \in \mathcal{B}$

1: $G' = VDM^{-1} \times G \mod p$ (all coefficients lifted in $[-p/2, p/2)$)
2: Compute $B = G'(b_0, \dots, b_k) \in \mathbb{T}_\mathcal{R}$
3: **for** $i \in 0 \to N-1$ **do do**
4: Compute $A[i] = G'(a_0[i], \dots, a_k[i]) \in \mathbb{T}_\mathcal{R}$
5: **end for**
6: **return** $(0, B) - \sum_{i=0}^{N-1} KS_i \boxdot (0, A[i])$

Bootstrapping in B/FV

If we want to decrease the noise of the output, different possibilities for the algorithm of bootstrapping exist in the literature [12, 21]. The first one is the naive bootstrapping, where we evaluate the rounding function. In this case for $p > 2N + 1$ prime ($p = 1 \mod N$), we need $O(\sqrt{p})$ internal products for the evaluation and we preserve the N slots. The second one is the bootstrapping proposed in [12], where $p = r^e$ is a power of a prime number, and we need only $(e-1)(r-1)$ multiplications, but the number of slots is reduced.

Non-linear functions in B/FV

In B/FV, an arbitrary function $f\colon \mathbb{Z}/p\mathbb{Z} \to \mathbb{Z}/p\mathbb{Z}$, for a prime p, can be interpolated by a polynomial of degree $\le p-1$ (by Lagrange interpolation) and can thus be evaluated as an arithmetic circuit of multiplicative depth $2\log_2 p$. The evaluation of the polynomial is performed simultaneously on the N slots. If the function f is viewed as a function $f\colon \mathbb{T} \to \mathbb{T}$, both the domain and the range are in this case rounded to exact multiples of $1/p$. Compared to TFHE, the output is constrained in the subset $p^{-1}\mathbb{Z}/\mathbb{Z}$ (e.g., only to multiples of $1/p$), but on the other hand, there is no negacyclic constraint on the input, so the graph of the non-linear function can be arbitrary everywhere. However, except in very special circumstances, the polynomial to evaluate is dense, so the number of homomorphic operations is $\Theta(p)$ which is impractical for large p's and thus, the method

does not apply to big-number slots. One notable exception to this rule is the bootstrapping in [12] modulo p^k, which proves that the rounding function is the composition of sparse polynomials.

3.2 Scheme-switching between B/FV-big-number and TFHE.

In the case of [14], using the NTRU trick [25], the plaintext space is $\mathcal{R}/(X-p)\mathcal{R} \simeq \mathbb{Z}/(1+p^N)\mathbb{Z}$. Usually, p is small, but p^N+1 is large, so a slot corresponding to $\mathbb{Z}/(1+p^N)\mathbb{Z}$ allows us to perform arithmetic operations on big numbers. If $P = X - p$ then the native plaintext space $\mathcal{M}$ is $P^{-1}\mathcal{R}_{\mathbb{Z}}/\mathcal{R}_{\mathbb{Z}}$ with

$$P^{-1} = -\frac{1}{p^N+1}\sum_{i=0}^{N-1} p^{N-1-i}X^i.$$

Interestingly, since the leading coefficient of the polynomial P^{-1} is $1/(p^N+1)$, the isomorphism $\mathcal{M} \simeq \frac{1}{p^N+1}\mathbb{Z}/\mathbb{Z}$ corresponds to extracting the coefficient in X^N-1 (i.e. the map $\mu = \sum_{i=0}^{N-1}\mu_i X^i \mapsto \mu_{N-1} \mod \mathbb{Z}$). On this native plaintext space, the naïve rounding algorithm computing $\mu = P^{-1}\cdot\lfloor\varphi_s(c)\cdot P\rceil$ can solve the BDD problem up to a distance $\approx 1/2p$ (which is the packing radius of the lattice $\mathcal{M}$; note that the vector $P \in \mathcal{R}_{\mathbb{Z}}$ is a short vector). Here, by $\varphi_s(c)$, we mean an arbitrary lift in $\mathcal{R}_{\mathbb{R}}$. This allows to operate on ciphertexts with very large noise $\approx 1/2\sqrt{N}p$.

3.2.1 B/FV-big-number → msb-TFHE.

Given a TRLWE ciphertext $c = (a, b) \in \mathbb{T}^2_{\mathcal{R}}$ encrypting $\mu \in \mathbb{T}^N$ with a key K, to obtain the TLWE encryption of the most significant bit of μ with key K, it is enough to extract $c_{N-1} = \mathsf{SampleExtract}_{N-1}(c)$.

3.2.2 TFHE → B/FV-big-number.

Conversely, to transform $k < N$ TLWE independent ciphertexts $c_0, \ldots, c_{k-1}$ encrypting $\mu_i = x_i/p \in p^{-1}\mathbb{Z}/\mathbb{Z}$ (for $x_i \in [0, p-1]$) with key $S \in \mathbb{B}^N$ into a TRLWE ciphertext encrypting the big-number $R = \sum_{i=0}^{k-1} x_i p^{N-1-i} \mod p^N+1$ with key $K \in \mathcal{B}$, we can return an encryption of $(\mu_0, \ldots, \mu_{k-1}) \to \sum_{i=0}^{k-1}\mu_i X^{N-1-i} \in p^{-1}\mathcal{R}_{\mathbb{R}}/\mathcal{R}_{\mathbb{R}}$. Indeed, this polynomial is very close to our target $P^{-1}R \mod \mathcal{R}_{\mathbb{Z}}$. To that end, we can just apply the public key-switch $c = \mathsf{PubKS}(id, \mathsf{KS}, (c_0, \ldots, c_{k-1}))$, where the key-switching key is composed by $\mathsf{KS}_i = \mathsf{TRGSW}_K(S_i)$ to pack the k ciphertexts as a single TRLWE ciphertext.

4 A general abstraction of HEAAN over the torus

For HEAAN, the homomorpic encryption scheme of approximate numbers, the idea is that instead of correcting the error during the decryption for the sake of increased noise, one keeps the approximation error and thus, reduces the noise. In this scheme only the significant digits are used and the phase is taken as a good approximation of the plaintext. HEAAN is a mixed encryption scheme dedicated to fixed-point arithmetic with public exponent. Similarly to scale invariant schemes, the noise is appended to the least significant bits of the phase, but unlike B/FV, the space of valid messages is a small bounded interval rather than evenly-distributed in the whole space. Also, the maximal amount of noise is interpreted in terms of the ratio between the diameter of the message space and the scale modulus q rather than the usual noise amplitude versus packing radius. As we keep performing homomorphic operations, the message space diameter increases until the majority of the space is covered at this point, the scale invariance property enables to extract the message as a classical LWE sample that can be processed, for instance, by TFHE. To fully enable this switch between schemes, it is necessary to relate the message spaces of HEAAN and TFHE.

Tags of ciphertexts

To do this, we revisit the representation of HEAAN ciphertexts by adding the following three tags/parameters that we define and clarify below:

- $L \in \mathbb{N}$: level exponent of the ciphertext - overall, it quantifies the maximum amount of homomorphic operations before performing a bootstrapping (the native plaintext $\|\mu\|_\infty \leq 2^{-L}$),
- $\rho \in \mathbb{N}$: bits of precision of the plaintext, that is, the number of bits in the mantissa. Since it is a global constant, we omit it in general.
- $\tau \in \mathbb{Z}$: slot exponent (order of magnitude of the complex values in each slot). Here, we use floating-point representation for which the exponent is public and fixed across all the coordinates of the vector and only the mantissas are secret. More precisely, a complex number z is represented as $z = m \cdot 2^\tau$ where τ is public and $m \in 2^{-\rho} \cdot (\mathbb{Z} + i\mathbb{Z})$ with $0 \leq |m| < 1$.

In Figure 5, we show the plaintext space (with the rounding error) using the tags defined before.

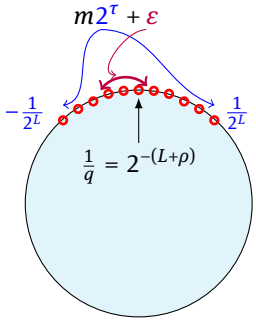

Figure 5: The HEAAN plaintext space over the $\mathbb{T}_\mathcal{R}$, where the error ε is an approximation error of the same order as $\alpha = 2^{-(L+\rho)}$.

More precisely, L is needed since HEAAN can be viewed as an instantiation of TRLWE whose native plaintext space is the subset of all polynomials $\mu \in \mathbb{T}_\mathcal{R}$ of small norm $\|\mu\|_\infty \leq 2^{-L}$. The integer $L > 0$ is the level exponent of the ciphertext, it is public and decreases with each multiplication. When the level is too low, the ciphertext must be bootstrapped to allow further processing.

The plaintext space is always described with a global and fixed number ρ of significant bits. One can define the *noise amplitude* $\alpha := 2^{-(L+\rho)}$. Finally, since the goal is to represent ρ-bit fixed-point values of any order of magnitude, each ciphertext carries a public integer exponent $\tau \in \mathbb{Z}$ which represents the order of magnitude of its slots.

We choose these three tags because they are helpful for the parameter selection of the cryptosystem (N and α) from the user point of view. For that, the first step is to fix ρ, the precision needed at the end of the algorithm, the second step is to determine the range of each variable (so the slot exponents τ): this can be done either from the domain of the evaluated function or experimentally by running the algorithm on a fake data. Then, using the noise propagation formula for elementary operations given in this section, we compute the smallest level exponent $L > 1$ for each variable by traversing the arithmetic graph of the algorithm. Finally, we use the security API document [8] (or the LWE estimator) to find the smallest key size N that supports a noise rate $\alpha = 2^{-(L+\rho)}$ to the desired security parameter.[3] The original HEAAN choice of tags is inherent and to determine the final system parameters, we have to solve a complex system of equations.

3 There is a very light dependency requirement $\rho \geq \log_2(N)$ in the noise propagation formula of the product, but in general, choosing $\rho \geq 15$ should be enough.

Complex-valued slots for plaintexts

Recall that N is always a power of 2. Given a message $\mu \in \mathcal{R}_{\mathbb{R}}$ where $\|\mu\|_\infty \leq 2^{-L}$, its complex slots $(z_1, \ldots, z_{N/2})$ are defined as the (rescaled) evaluation on the complex roots of $X^N + 1$, so $z_k = 2^{L+\tau}\mu(\zeta_k) \in \mathbb{C}$. We omit the values on the last $N/2$ roots as they are complex conjugates of the first ones since μ is real. If $\|\mu\|_\infty \approx 2^{-L}$, this indeed implies that slot values $|z_k| \approx \sqrt{N}2^\tau$ and that the slot precision is up to $2^{\tau-\rho}$.

In order to unify the message spaces, we redefine tagged HEAAN ciphertexts as a quadruple $(a, b, \tau, L) \in \mathbb{T}_{\mathcal{R}}^2 \times \mathbb{Z} \times \mathbb{N}$ where (a, b) is a TRLWE ciphertext. As usual, the phase of a ciphertext is $(b - s \cdot a) \in \mathbb{T}_{\mathcal{R}}$ and the message is the expectation of the phase. The slots of a ciphertext are the slots of its message $\mu \in \mathbb{T}_{\mathcal{R}}$ represented by a lift in $\mathcal{R}_{\mathbb{R}}$ of small real norm $\leq 2^{-L}$. From a matrix point of view, the transformation between the coefficients and the slots is the multiplication with $2^{\tau+L}$ times the complex DFT matrix of ζ_k. We have $(z_1, \ldots, z_{N/2}) = \mathsf{DFT} \cdot (\mu_0, \ldots, \mu_{N-1})$ and $(\mu_0, \ldots, \mu_{N-1}) = 2Re(\mathsf{IDFT} \cdot (z_1, \ldots, z_{N/2}))$.

$$\mathsf{DFT} = \begin{bmatrix} 1 & \zeta_1^1 & \cdots & \zeta_1^{N-1} \\ 1 & \zeta_2^1 & \cdots & \zeta_2^{N-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & \zeta_{N/2}^1 & \cdots & \zeta_{N/2}^{N-1} \end{bmatrix}, \mathsf{IDFT} = \frac{1}{N} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \bar{\zeta}_1^1 & \bar{\zeta}_2^1 & \cdots & \bar{\zeta}_{N/2}^1 \\ \vdots & \vdots & \cdots & \vdots \\ \bar{\zeta}_1^{N-1} & \bar{\zeta}_2^{N-1} & \cdots & \bar{\zeta}_{N/2}^{N-1} \end{bmatrix} \tag{7}$$

4.0.1 Tag propagation formulas.

We now describe the tag propagation formulas for the homomorphic operations over the slot representation of the ciphertext[4]:

Addition $\mathtt{HeaanAdd}((\mathbf{c}_1, \tau_1, L_1), (\mathbf{c}_2, \tau_2, L_2)) \to$

$$\begin{cases} \mathbf{c} = 2^{\tau_1+L_1-\tau-L}\mathbf{c}_1 + 2^{\tau_2+L_2-\tau-L}\mathbf{c}_2 \mod \mathbb{Z}, \\ \tau = \max(\tau_1, \tau_2) + 1, \\ L = \min(L_1 + \tau_1, L_2 + \tau_2) - \tau \end{cases}$$

Proof. We can check that this transformation changes the slot value into $2^{\tau+L}\mu(\zeta_k) = 2^{\tau_1+L_1}\mu_1(\zeta_k) + 2^{\tau_2+L_2}\mu_2(\zeta_k) = (z_1 + z_2)$, that proves the correctness of the sum of two slots. The fact that $\tau_i + L_i - \tau - L \geq 0$, for $i \in \{1, 2\}$ means that the sum is an integer combination of the ciphertexts. At the end, we verify that

$$\begin{aligned} \|\mu(X)\|_\infty &\leq 2^{\tau_1+L_1-\tau-L}\|\mu_1(X)\|_\infty + 2^{\tau_2+L_2-\tau-L}\|\mu_2(X)\|_\infty \leq \\ &\leq 2^{\tau_1+L_1-\tau-L-L_1} + 2^{\tau_2+L_2-\tau-L-L_2} \leq 2^{-L}. \end{aligned}$$

□

Decrease level $\mathtt{HeaanRS}_{L\to L'}((\mathbf{c}, \tau, L), L' < L) \to (2^{L-L'}\mathbf{c}' \mod \mathbb{Z}, \tau, L')$

Proof. We verify that the slot values are preserved and that $\|\mu(X)\|_\infty < 2^{L-L'}\|\mu'(X)\|_\infty \leq 2^{L-L'-L} = 2^{-L'}$. □

Binary Shift (multiply by 2^t) $\mathtt{HeaanBS}(t, (\mathbf{c}, \tau, L)) \to (\mathbf{c}, \tau + t, L)$

Proof. Slots are indeed transformed as $z' = 2^{\tau+t+L}\mu(\zeta_k) = 2^t z$ where t is the shift and $z = 2^{\tau+L}\mu(\zeta_k)$ is the slot value. Since the TRLWE part does not change, the native plaintext does not change either and the bound 2^{-L} is preserved. □

4 The identities below are approximate since the sum of two normal distributions $X = N(0, \sigma_X)$ and $Y = N(0, \sigma_Y)$ is not necessarily a normal distribution (unless the variables are independent which needs not be the case as they are distributions over ciphertexts), but, for any practical purposes, is close enough to a normal distribution with zero mean and standard deviation $\sigma = \sqrt{\sigma_X^2 + \sigma_Y^2}$. In addition, the Gaussian distributions that appear in practice are not even continuous Gaussian distributions and this makes the estimate even more complex.

Multiplication by a constant $\texttt{HeaanMultCst}(a \in \mathbb{Z} \quad \text{s.t.}|a| \le 2^\rho, (\mathbf{c}, \tau, L) \to (a \cdot \mathbf{c} \mod \mathbb{Z}, \tau + \rho, L - \rho)$

Proof. We can check that multiplication with $a \le 2^\rho$ transforms the slot value into $z' = 2^{\tau+L} a\mu(\zeta_k)$. □

Constant slot-wise multiplication $\texttt{HeaanSlotMult}((u_1, \ldots, u_{N/2}), (\mathbf{c}, \tau, L))$

Let $u_1, \ldots, u_{N/2}$ be N fixed-point complex slots of the same order of magnitude (e.g. $u_k = (x_k + iy_k).2^{-\rho}$ where x_k, y_k are integers in $[-2^\rho, 2^\rho]$.

Interpolate (or find by least mean square) an integer polynomial $d(x)$ with coefficients in $[-2^\rho, 2^\rho]$ and t an integer exponent such that the slots of $d(X)2^t$ are all good approximations of $z_1, \ldots, z_{N/2}$, up to precision $2^{t-\rho}$. Namely,

$$|d(\zeta_k)2^t - u_k| \le 2^{t-\rho} \text{ for all } k \in [1, N/2]. \tag{8}$$

Then all we need is to multiply the input ciphertext by $d(x)$ and shift the result by τ bits. The level decreases by ρ bits, where 2^ρ is the norm of d.

$$d(X) \in \mathcal{R}_\mathbb{Z}(\|d\|_\infty \le 2^\rho), (\mathbf{c}, \tau, L) : \begin{cases} d(X) \cdot \mathbf{c} \mod \mathbb{Z}, \\ \tau' = \tau + \rho + t, \\ L' = L - \rho. \end{cases}$$

Proof. It follows from (8) that $z'_k = 2^{\tau'+L'}\mu(\zeta_k)d(\zeta_k) = 2^{\tau+L}\mu(\zeta_k)d(\zeta_k)2^t = z_k \cdot (u_k + \varepsilon_k)$ where $|z_k\varepsilon_k| \le 2^{\tau'-\rho}$ and that the native plaintext norm verifies $||\mu'(X)||_\infty \le 2^{-L+\rho} = 2^{-L'}$. □

Slot-wise precomputed secret multiplication :

$\texttt{HeaanPrivSlotMult}(\mathsf{TRGSW}(D), (\mathbf{c}, \tau, L))$

In the previous multiplication, $d(x)$ can be provided encrypted as a TRGSW ciphertext of D.

General multiplication $\texttt{HeaanMult}((\mathbf{c}_1, \tau_1, L'), (\mathbf{c}_2, \tau_2, L'))$ use the Algorithm 3 below, proved in Proposition 3.

Algorithm 3 HEAAN homomorphic product on $\mathbb{T}_\mathcal{R}$

Input: Two HEAAN ciphertexts $(a_1, b_1, \tau_1, L_1), (a_2, b_2, \tau_2, L_2) \in \mathbb{T}^2 \times \mathbb{Z} \times \mathbb{N}$ whose slots are $(z_1^{(1)}, \ldots, z_{N/2}^{(1)})$ and $(z_1^{(2)}, \ldots, z_{N/2}^{(2)})$ under the same key s and precision $\rho > \log_2 N$.

Output: a HEAAN ciphertext (a, b, τ, L) whose slots are $z_j = z_j^{(1)} z_j^{(2)}$ for $j \in [1, N/2]$ with the same key s

1: Set $\tau = \tau_1 + \tau_2$ (slot exponent)
2: Set $L' = \min(L_1, L_2)$ and use $\texttt{HeaanRS}_{L_i \to L'}$ to decrease both ciphertexts to level L'
3: Let $q = 2^{L'+\rho}$, $\alpha = q^{-1}$, and $L = L' - \rho$
4: Round (a_i, b_i) to the nearest multiple of $\alpha = q^{-1}$.
5: Let $(a, b) = (a_1, b_1) \boxtimes_{q,\alpha} (a_2, b_2)$ (with $\boxtimes_{q,\alpha}$ the internal homomorphic product defined in the Definition 3)
6: **return** (a, b, τ, L)

Proposition 3 (HEAAN product) *Let* $(a_1, b_1, \tau_1, L_1), (a_2, b_2, \tau_2, L_2) \in \mathbb{T}^2_\mathcal{R} \times \mathbb{Z} \times \mathbb{N}$ *whose slots are* $(z_1^{(1)}, \ldots, z_{N/2}^{(1)})$ *and* $(z_1^{(2)}, \ldots, z_{N/2}^{(2)})$ *under the same key* s. *Suppose that the precision* ρ *is larger than* $\log_2 N$. *Algorithm 3 computes a* HEAAN *ciphertext* (a, b, τ, L) *whose slots are* $z_j = z_j^{(1)} z_j^{(2)}$ *for* $j \in [1, N/2]$ *with the same key* s *such that* $Var(Err((a, b)))$ *remains implicitly* $4^{-L-\rho}$.

Proof. (sketch) Since Algorithm 3 rescales both ciphertexts to the same level, we can assume that both inputs have the same level L'. Compared to the proof of Lemma 1, defining the same auxiliary quantities C_0, C_1, C_2, we have

$$\begin{aligned} \varphi_s(a, b) &= \mu_1 \boxtimes_q \mu_2 + e_2\mu_1 q + e_1\mu_2 q + e_1 e_2 q + e_2 I_1 q + e_1 I_2 q + \mathsf{Err}(RK \boxdot_\alpha (C_2, 0)) \mod 1 \\ &= \mu_1 \boxtimes_q \mu_2 + e_2\mu_1 q + e_1\mu_2 q + e_1 e_2 q + \mathsf{Err}(RK \boxdot_\alpha (C_2, 0)) \mod 1 \end{aligned}$$

Here, the terms $e_1 I_2 q$ and $e_2 I_1 q$ disappear because e_i are exact multiples of $\frac{1}{q}$ after the rounding. The expectation of the phase is still $\mu_1 \boxtimes_q \mu_2$, so the output slots contain $z_k = q\mu_1(\zeta_k)\mu_2(\zeta_k)2^{\tau+L} = z_{1,k}z_{2,k}$ since $L = 2L' - \log_2(q)$. The native plaintext $\mu_1 \boxtimes_q \mu_2$ itself is bounded by $q2^{-2L'} = 2^{L'+\rho-2L'} = 2^{-L}$. The phase variances of $e_2\mu_1 q$, $e_1\mu_2 q$ are bounded by $\left(q2^{-L'}2^{-L'-\rho}\right)^2 = 4^{-L-\rho}$, $e_1 e_2 q$ by $4^{-L-2\rho}$, and $\mathsf{Var}(\mathsf{Err}(RK\boxdot_\alpha(C_2,0)) \le \left(2\ell N + \frac{N^2+N}{4}\right)\alpha^2 \le N^2\alpha^2 \le 4^{\log_2(N)-L'-\rho} < 4^{-L'} \le 4^{-L-\rho}$ because $\rho > \log_2(N)$. Overall, the output noise standard deviation is $2^{-L-\rho}$, which corresponds to ρ bits of fixed-point precision. □ □

4.1 Scheme switching between TFHE and HEAAN

4.1.1 TFHE → HEAAN

Let $S = (S_1, \ldots, S_n) \in \{0,1\}^n$ be a TLWE key and let N be the power of 2 in the HEAAN cryptosystem. Let $(a_1, b_1), \ldots, (a_{N/2}, b_{N/2}) \in \mathbb{T}^n \times \mathbb{T}$ be TLWE ciphertexts encrypted under the secret key S. For every $1 \le k \le N/2$, let $v_k = \varphi_S(a_k, b_k)$ be their phases. Suppose that N is the modulus used in HEAAN and $K \in \mathcal{B}$ is a HEAAN secret key. Here, we describe an algorithm (functional switching from TFHE to HEAAN) that outputs a HEAAN ciphertext (a, b, τ, L) which, when decrypted with the key K, yields a vector of size $N/2$ whose components are the complex values $z_k = \exp(2\pi i v_k)$ for $1 \le k \le N/2$. This allows us to evaluate trigonometric polynomials and have various practical applications such as evaluating continuous functions via rapid convergence of Fourier series. We describe this algorithm as a variant of the bootstrapping for HEAAN [15].

Letting $\mathrm{BK}_i = \mathsf{TRGSW}_K(S_i)$ be the components of the bootstrapping key BK, this algorithm is a combination of a homomorphic evaluation of the mod $\mathbb{Z}$ operation (the bootstrapping proposed by HEAAN in [15]) and a multiplication with the DFT matrix in order to switch between coefficients and slots.

Once we have the complex exponential, we can represent any other piecewise continuous function f from $\mathbb{T}$ to $\mathbb{C}$ and obtain $f(v_1), \ldots, f(v_{N/2})$ in the slots, by just evaluating the first terms of the Fourier series of f.

Proposition 4 (Functional switching TFHE to HEAAN) *In the setting described above where the noise standard deviation is α and the precision parameter is $\rho \in \mathbb{N}$, Algorithm 4 computes a* HEAAN *ciphertext (a, b, τ, L) decrypting to $(z_1, \ldots, z_{N/2})$ under K and having precision ρ, $p = \sqrt{\rho} + \log_2(2\pi n/\sqrt{\rho})$ and level ciphertext exponent*

$$L = -\log_2 \alpha - (p + \log_2 \rho)\rho - \frac{1}{2}\left(\log_2\left(-2nN\log_2\alpha + n\frac{1+N}{4}\right)\right).$$

Proof. (sketch) We approximate $\exp(2\pi i v_k)$ using the idea of [15] by first taking a small real representative of the input ciphertexts and divide them by 2^p for a suitable p (that depends on the target precision). This way, the (real) phase of the rescaled ciphertext $v_k/2^p$ is guaranteed to be bounded by $n/2^p$. We first compute a good approximation, up to an error $\le 2^{-\rho}$, for $\exp(2\pi i v_k/2^p)$ using the first $\sqrt{\rho}$ terms of the Taylor expansion of the exponential function. For instance, the Taylor–Lagrange inequality gives $\left|\exp(ix) - \left(\sum_{k=0}^{\sqrt{\rho}-1}\frac{(ix)^k}{k!}\right)\right| \le \frac{|x|^{\sqrt{\rho}}}{\sqrt{\rho}!}$, so for $x \le n/2^p$, it suffices to choose $p = \sqrt{\rho} + \log_2(2\pi n/\sqrt{\rho})$ to get the required approximation within $2^{-\rho}$. Then, we square and multiply (we square and square in this case) the result to raise $\exp(2\pi i v_k/2^p)$ to the power 2^p to obtain the desired plaintext $\exp(2\pi i v_k)$. From Theorem 1 on the external product for TFHE, the noise of the ciphertext $\mathbf{c}$ of line 4 is

$$\mathsf{Var}(\mathsf{Err}(\mathbf{c})) \le (-2n\log_2\alpha N + n\frac{1+N}{4})\alpha^2 \le 4^{-L-(p+\log_2(\rho))\rho}.$$

We then evaluate the Taylor expansion of the complex exponential (up to degree $d = \sqrt{\rho}$) via the algorithm of Paterson–Stockmayer [27]: this requires a depth of $2\log_2 d = \log_2\rho$, it uses $3\sqrt{d}$ non-constant (`HeaanMult`) as well as d constant multiplications (`HeaanSlotMult`). After this step, the level decreases by ρ

Algorithm 4 Switching TFHE to HEAAN

Input: $N/2$ TLWE ciphertexts (a_k, b_k), $1 \le k \le N/2$ whose phases are $v_k \in \mathbb{T}$ under the same key $S \in \{0,1\}^n$ and $BK_i = \mathsf{TRGSW}_K(S_i)$.

Output: A HEAAN ciphertext $(a, b) \in \mathbb{T}^2_{\mathcal{R}}$ at level L that decrypts (under K) to the slots $(z_1, \ldots, z_{N/2})$ where $z_k = e^{2\pi i v_k}$ and $v_k = \varphi_S(a_k, b_k)$.

1: Let A be the $N/2 \times (n+1)$ real matrix where $A_{i,j}$ is the representative of the j-th coefficient of $a_i \in [-1/2, 1/2)$ and the last column $A_{j,n+1}$ contains the representative of $b_j \in [-1/2, 1/2)$.

2: Let $p = \sqrt{\rho} + \log_2(2\pi n/\sqrt{\rho})$

3: Compute $P_j \leftarrow \frac{1}{2^p N}\mathrm{Re}(\mathsf{IDFT} * A_j)$ for $1 \le j \le n+1$. Here, P_j is the polynomial whose slots are $\frac{1}{2^p}A_j$.

4: $c \leftarrow (0, 2^{-(L+(p+1)\rho)}P_{n+1}) - \sum_{j=1}^{n} BK_j \boxdot_\alpha (0, 2^{-(L+(p+1)\rho)}P_j)$.

5: Let $C = (c, \tau = 0, L + (p+1)\rho)$

6: Evaluate homomorphically $E = \sum_{k=0}^{\sqrt{\rho}-1} \frac{i^k}{k!} C^k$ using Paterson–Stockmayer algorithm [27] (in depth $\log_2 \rho$), `HeaanMult` for non-constant multiplications and `HeaanSlotMult` for constant multiplications. Here, E has parameters $\tau = 0$ and level $L + p\rho$.

7: **for** $j = 1$ to p **do**

8: $\quad E \leftarrow \mathtt{HeaanMult}(\mathtt{E}, \mathtt{E})$ (the new E has parameters $\tau = 0$ and level $L + (p-j)\rho$)

9: **end for**

10: **return** E at level L

times the multiplicative depth, so the level of E is $\le L + p\rho$. Finally, we square the ciphertext p times to obtain the desired result.

□ □

4.1.2 HEAAN → TFHE.

Conversely, to switch from HEAAN to TFHE ciphertexts, we use the observation that a HEAAN ciphertext of level 1 is automatically a TFHE ciphertext. Starting from a ciphertext HEAAN of level L, we use the level decrease function $\mathsf{HeaanRS}_{L\to 1}$ followed by a slot extraction to obtain a TFHE ciphertext of μ. Here, by slot extraction, we mean the slots to coefficients procedure described in [15] to extract HEAAN slots into coefficients of TRLWE (i.e. applying the IDFT complex transformation homomorphically).

4.2 Non-linear functions in HEAAN

In HEAAN, non-linear functions can be evaluated via approximations by either complex-valued polynomials (via traditional products) or trigonometric polynomials (Fourier approach within the bootstrapping).

As explained in [4], Fourier series of smooth and regular functions converge rapidly: for instance, the Fourier series of a C^∞-function converges super-algebraically and if one smooths any periodic function by convolution with a small Gaussian, its Fourier series converges exponentially fast. However, the convergence is slower if the function has discontinuities (pointwise convergence in $\Omega(1/k)$), or discontinuities in its derivative (uniform convergence in $\Omega(1/k^2)$) where k is the number of harmonics used in the series.

Compared to classical approximations of functions by polynomials in [10, 24] (i.e. Taylor series or Weierstrass approximation theorem), Fourier series have three main advantages: they do not diverge to ∞ outside of the interval (better numerical stability), the Fourier coefficients are small (square integrable), and the series converge uniformly to the function on any interval that does not contain any discontinuity in the derivative.

With this bootstrapping trick, HEAAN can at the same time evaluate a non-linear function and bootstrap its output to a level L even higher than its input. Taking this fact into account, instead of writing $\mathtt{ReLU}(x) =$

$\max(0, x)$ as $\frac{1}{2}(|x| + x)$ like in TFHE, where the term $+x/2$ is not bootstrapped, it is actually better to extend the graph of ReLU from a half period $(-1/4, 1/4)$ directly to a 1-periodic continuous function and to decompose the whole graph as a Fourier series. In the latter case, the output level L can be freely set to an arbitrary large value. Figure 6 shows a degree-7 approximation of the odd-even periodic extension of the graph of $\mathtt{ReLU}(x)$.

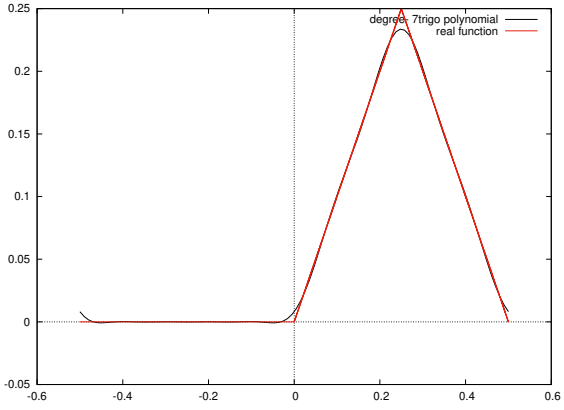

Figure 6: ReLU for HEAAN

5 Implementation of some major components of Chimera

Some of the algorithms presented in this paper have been implemented and tested in the context of an IDash 2018 [1] submission described in [9]. This submission was selected in October 2018 to be among the finalists of the competition.

Since the challenge was mostly about real-valued algorithms, we implemented the TFHE/HEAAN bridges, and we left the implementation of the TFHE-B/FV bridges as a future work. In the implementation, torus arithmetic is represented either by double-precision floats (for $\alpha \geq 2^{-52}$), or by a fixed-size array of 64-bit gmp limbs, and polynomial multiplications are implemented via the complex FFT. Since the total precision required was always lower than 128 bits for external products and 192 bits for internal products (so at most two or three gmp limbs), there was no advantage in switching to an alternative RNS/NTT representation. All bit-decompositions for the TRGSW external products (also used in the internal product) have been carried-on in base 2^{16} and 2^{32} rather than in base 2. The implemented algorithms are:

- **Evaluation of a sigmoid** on a TLWE ciphertext via gate bootstrapping followed by a functional key-switch to HEAAN (noise reduction from $(\alpha = 2^{-7}, N = 650)$ to $(\alpha = 2^{-80}, N = 4096)$). The time for the evaluation of the bootstrapped sigmoid is: 32s for the bootstrapping to TLWE and 7s for the public key-switch TLWE to TRLWE.
- **External product** to multiply a HEAAN ciphertext by a fresh TRGSW ciphertext, using both the slot packed ciphertexts ($N/2$ slots) and coefficient-packed ciphertexts (N slots). The time is 80ms with parameters $\alpha = 2^{-80}$, $N = 4096$.
- **TRLWE internal product** for HEAAN slotwise multiplication, using the tagging system proposed here, and relinearization via the external product formula. The time is 160ms for parameters $\alpha = 2^{-80}$, $N = 4096$.
- **Evaluation of the non-linear function** $\log|x|$ on 1024 slots in parallel during the bootstrapping for HEAAN (Algorithm 4), versus the traditional bootstrapping followed by a Taylor approximation.

The implementation of the algorithms described in this section is now public and available at: `https://github.com/DPPH/chimera-iDash2018`. Other directions that we leave as a future work is the elaboration of a bridge towards the `BGV` scheme as well as the introduction of RNS in this framework. The presented framework has already been applied to some concrete use-cases in the domain of machine-learning [5, 9] but we are wishing to provide more applications in this or other directions. The remaining bridges will be implemented when specific use cases are identified for them.

Conclusion

Some recent works introduce alternatives to the methods presented in this article. In [17], the authors propose a new method for the evaluation of the sign function by staying in the HEAAN encoding by multiple compositions of polynomials. This extends to the evaluation of the RELU function and can be very efficient in amortized running time. This SIMD sign evaluation, can be viewed as a bootstrapping for BFV via HEAAN, which is not covered in this work.

Furthermore, in [11], an alternative method for switching between RLWE ciphertexts is presented. This is an orthogonal direction to ours and uses composition with Galois polynomials. In this work, the authors obtain speed-up by one order of magnitude in particular on the LWE-RLWE conversion. As a counterpart the noise overhead is no longer additive since the phase is multiplied by a power of 2. This method can speed-up the Chimera framework by replacing some of the underlying KS methods.

References

[1] Track 2: Secure parallel genome wide association studies using homomorphic encryption. www.humangenomeprivacy.org/2018/competition-tasks.html, 2018.

[2] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. Lauter, S. Lokam, D. Micciancio, D. Moody, T. Morrison, A. Sahai, and V. Vaikuntanathan. Homomorphic encryption security standard. Technical report, HomomorphicEncryption.org, Toronto, Canada, November 2018.

[3] M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. *J. Mathematical Cryptology*, 9(3):169–203, 2015.

[4] C. Boura, I. Chillotti, N. Gama, D. Jetchev, S. Peceny, and A. Petric. High-precision privacy-preserving real-valued function evaluation. *IACR Cryptology ePrint Archive*, 2017:1234, 2017.

[5] C. Boura, N. Gama, M. Georgieva, and D. Jetchev. Simulating homomorphic evaluation of deep learning predictions. In *CSCML 2019*, volume 11527 of *LNCS*, pages 212–230. Springer, 2019.

[6] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *CRYPTO 2012*, volume 7417 of *LNCS*, pages 868–886. Springer, 2012.

[7] Z. Brakerski and R. Perlman. Lattice-based fully dynamic multi-key FHE with short ciphertexts. In *CRYPTO 2016, Part I*, LNCS, pages 190–213. Springer, 2016.

[8] M. Brenner, W. Dai, S. Halevi, K. Han, A. Jalali, M. Kim, K. Laine, A. Malozemoff, P. Paillier, Y. Polyakov, K. Rohloff, E. Savaş, and B. Sunar. A standard api for rlwe-based homomorphic encryption. Technical report, HomomorphicEncryption.org, Redmond WA, July 2017.

[9] S. Carpov, N. Gama, M. Georgieva, and J. R. Troncoso-Pastoriza. Privacy-preserving semi-parallel logistic regression training with fully homomorphic encryption. Cryptology ePrint Archive, Report 2019/101, 2019. https://eprint.iacr.org/2019/101.

[10] H. Chabanne, A. de Wargny, J. Milgram, C. Morel, and E. Prouff. Privacy-preserving classification on deep neural network. Cryptology ePrint Archive, Report 2017/035, 2017. https://eprint.iacr.org/2017/035.

[11] H. Chen, W. Dai, M. Kim, and Y. Song. Efficient homomorphic conversion between (ring) LWE ciphertexts. *IACR Cryptology ePrint Archive*, 2020:15, 2020.

[12] H. Chen and K. Han. Homomorphic lower digits removal and improved FHE bootstrapping. In *EUROCRYPT 2018, Proceedings, Part I*, volume 10820 of *LNCS*, pages 315–337. Springer, 2018.

[13] H. Chen, K. Laine, and R. Player. Simple encrypted arithmetic library - SEAL v2.1. In *Financial Cryptography and Data Security - FC 2017*, pages 3–18, 2017.

[14] H. Chen, K. Laine, R. Player, and Y. Xia. High-precision arithmetic in homomorphic encryption. In *CT-RSA 2018*, LNCS, pages 116–136. Springer, 2018.

[15] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song. Bootstrapping for approximate homomorphic encryption. In *EUROCRYPT 2018, Proceedings, Part I*, volume 10820 of *LNCS*, pages 360–384. Springer, 2018.

[16] J. H. Cheon, A. Kim, M. Kim, and Y. S. Song. Homomorphic encryption for arithmetic of approximate numbers. In *ASIACRYPT 2017, Proceedings, Part I*, volume 10624 of *LNCS*, pages 409–437. Springer, 2017.

[17] J. H. Cheon, D. Kim, and D. Kim. Efficient homomorphic comparison methods with optimal complexity. *IACR Cryptology ePrint Archive*, 2019:1234, 2019.

[18] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. TFHE: Fast Fully Homomorphic Encryption over the Torus. *IACR Cryptology ePrint Archive*, 2018:421.

[19] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *ASIACRYPT 2016, Proceedings, Part I*, volume 10031 of *LNCS*, pages 3–33. Springer, 2016.

[20] L. Ducas and D. Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. In *EUROCRYPT 2015, Proceedings, Part I*, volume 9056 of *LNCS*, pages 617–640. Springer, 2015.

[21] J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.

[22] M. Geihs and D. Cabarcas. Efficient integer encoding for homomorphic encryption via ring isomorphisms. In *LATINCRYPT 2014*, LNCS, pages 48–63. Springer, 2015.

[23] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC 2009*, pages 169–178, 2009.

[24] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. E. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 201–210.

[25] J. Hoffstein and J. Silverman. Optimizations for NTRU, January 2000.

[26] V. Lyubashevsky, C. Peikert, and O. Regev. *EUROCRYPT*, chapter On Ideal Lattices and Learning with Errors over Rings, pages 1–23. Springer, 2010.

[27] M. Paterson and L. J. Stockmeyer. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J. Comput.*, 2(1):60–66, 1973.