High-precision Leveled Homomorphic Encryption with Batching

Long Nie[1], ShaoWen Yao[1] and Jing Liu[1*]

[1*]National Pilot School of software, Yunnan University, Kunming, 650000, China.

*Corresponding author(s). E-mail(s): liujing@ynu.edu.com;
Contributing authors: 1796859140@qq.com; yaosw@ynu.edu.com;

Abstract

In most homomorphic encryption schemes based on the RLWE, the native plaintexts are represented as polynomials in a ring $\boldsymbol{Z_t[x]/x^N+1}$ where $\boldsymbol{t}$ is a plaintext modulus and $\boldsymbol{x^N+1}$ is a cyclotomic polynomial with degree power of two. An encoding scheme should be used to transform some natural data types(such as integers and rational numbers) into polynomials in the ring. After a homomorphic computation on the polynomial is finished, the decoding procedure is invoked to obtain the result. However, conditions for decoding correctly are strict in a way. For example, the overflows of computation modulo both the plaintext modulus $\boldsymbol{t}$ and the cyclotomic polynomial $\boldsymbol{x^N+1}$ will result in a unexpected result for decoding. The reason is that decoding the part which is discarded by modular reduction is not 0.

We combine number theory transformation with Hensel Codes to construct a scheme. Intuitively, decoding the discarded part will yield 0 so the limitations are overcome naturally in our scheme. On the other hand, rational numbers can be handled with high precision in parallel.

Keywords: Homomorphic encryption, hensel codes, batching , number theoretic transforms

1 Introduction

1.1 Background

Fully homomorphic encryption (FHE) is a cryptographic scheme that allows us to evaluate an arbitrary boolean or arithmetic circuit on data in encrypted state directly without decryption. This notation, introduced by Rivest et al.[38], is first implemented by Gentry[33] with ideal lattices. All of schemes before Gentry either support homomorphic operation(homomorphic addition or multiplication)with a single type[5, 29, 36] or have some fatal drawbacks[7, 32](i.e. ciphertext size blows up exponentially with the depth of circuits).
A new construction is proposed by Brakerski et al.[13] with the assumption of the Learning With Error(LWE)[37]. A lot of efforts have been made to improve its efficiency and make it simple [10, 11, 31, 34]. The security of constructions mentioned above is based on either the LWE or the Ring Learning With Error(RLWE)[35]. All of schemes above for fully homomorphic encryption add noise into a ciphertext for security. The noise increases after homomorphic operations and destroys the plaintext once it reachs a certain threshold value which is related to parameters used in the scheme. A bootstrapping can be employed to refresh ciphertexts by calculating the decryption circuit homomorphically and reduce the noise to a small value subject to the depth of decryption circuit. There are a lot of works to improve the efficiency of the bootstrapping [6, 20, 23, 28]. However, in some circumstances where the depth of circuits is predetermined, the costly bootstrapping procedure can be avoided by using a so-called leveled homomorphic encryption scheme(LHE). To prevent plaintexts from being destroyed by the noise the LHE increases the threshold value by simply setting corresponding parameters large enough.
Practically, a computation over bitwise encryptions is not efficient and we are inclined to construct a scheme which manipulates integers directly. Other types of data such as real numbers, complex numbers, rational numbers can be handled by encoding them as integers. The efficiency of homomorphic encryption schemes can be improved significantly by a judicious choice of plaintext space and encoding techniques. There are a number of works which focus on how to encode different types of data efficiently[3, 8, 9, 15–17, 22, 24–27]. One useful technique adopted by previous works is to 'spread out' the numerical input data as evenly as possible over the whole plaintext space, allowing for a smaller value of plaintext modulus. The other main approach used for a short amortized time employs the single instruction multiple data(SIMD)[39] which is compatible with some encoding techniques.

1.2 Encodinng for integers and real numbers

In most schemes based on the RLWE assumption, the plaintext elements are represented as polynomials in a ring $R_t = Z_t[X]/\Phi_m(x)$, where $\Phi_m(x)$ denotes the m-th cyclotomic polynomial. Integers and real numbers should be transformed into polynomials in the ring before encryption. As an example, let $z, B \in Z$, encode z as $\sum_{i=0}^{n-1} a_i X^i$ such that $z = \sum_{i=0}^{n-1} a_i B^i$ where n is the degree of $\Phi_m(x)$ and $a_i \in [-(B-1), B-1]$. The above approach is called as

non-balanced base-B encoding. The other way is simply to encode an integer as a constant polynomial, which is referred to as scalar encoding. Dowlin et al.[27] present two efficient methods to encode fixed-point numbers. In the first a fixed point number is encoded via multiplying by a factor to obtain a scaled integer (which then is encoded as a polynomial), whilst in the second they utilize a fractional representation (which is similar to the non-balanced base-B encoding and allows the exponent to be negative). Costache et al.[25] show that the two representations are in fact isomorphic when the same power of 2 cyclotomic ring is used. A lot of works develop the fractional representation[8, 15]. Another useful way to encode rational numbers is Hensel codes which is used for encoding in some homomorphic encryption schemes[17, 26].

Which way for encoding depends on problems at hand. The scalar encoding is inefficient in its use of available space in the plaintext polynomial(only the constant coefficient is used). The non-balanced base-B encoding and some variants[21, 25](most of them focus on how to choose B, the range of coefficients, the space between entries) make full use of the space in the plaintext polynomial. However, there are many severe limitations. When one of the coefficients of the plaintext polynomial exceeds the plaintext modulus t or the degree of the plaintext polynomial exceeds the degree of $\Phi_m(x)$, a unexpected result will occur and we say the computation overflows modulo t and modulo $\Phi_m(x)$ respectively. As an example, let $n = 4, t = 4, B = 2$ where n is the degree of $\Phi_m(x)$ and $\Phi_m(x) = x^4 + 1$. For a given $z = 9$, we have $z = B^3 + 1$ and therefore encode z as $x^3 + 1$. Decoding is finished by simply replacing x in the plaintext polynomial with B. Add $3x + 3$ to z and get $x^3 + 3x \ mod \ 4$. Decoding $x^3 + 3x$ will yield the number 14 but not 18. Similarly, multiply z by $x + 1$ and get $x^3 + x$. Decoding it leads the number 10 but not 27 although the maximum value we can obtain after decoding is $(t-1)\sum_{i=0}^{3} 2^i = 45$. The reason why a unexpected result occurs is that the modular reduction after the addition or multiplication discards a part whose decoding is not 0. The previous works using similar encodings(include the fractional representations for real numbers) suffer from the limitations. The scaling approach is adopted by Cheon et al.[22] proposing a scheme to handle real numbers with batching. Therefore a rescaling operation should be performed to keep the factor of the result consistent after we do multiplication. For security the ciphertext modulus should be divided by the factor(since the ciphertext must look random in the ciphertext space)and the multiplication can't be performed once the ciphertext modulus reaches some small value. The plaintext modulus which is removed in[16, 22] seems to be necessary for Hensel Codes since a bound should be assigned before we use it.

1.3 Single instruction multiple data

The Chinese Remainder Theorem(CRT)[15, 27, 39] and Fast Fourier Transform(FFT)[16, 22] are two important ways to implement SIMD for homomorphic encryption. The former decomposes the cyclotomic polynomial in the

field Z_t where t is the plaintext modulus by choosing the cyclotomic polynomial and the plaintext modulus carefully, and builds an isomorphism between $R_t = Z_t[X]/\Phi_m(x)$ and $\prod Z_t[x]/Q_i(x)$ such that $\Phi_m(x) = \prod Q_i(x)\ mod\ t$. The latter takes a message vector as input, and then performs the inverse of FFT on it and output the result as a plaintext polynomial. SIMD is used for a short amortized time in general.

1.4 Our contribution

In this paper, we construct a simple leveled homomorphic encryption scheme which supports SIMD and overcomes the limitations(i.e. the overflows of computation modulo both the plaintext modulus and the cyclotomic polynomial will result in a unexpected result). More precisely, we use the number theory transformation(NTT) to implement SIMD so modular reduction can be performed on plaintext polynomials naturally. Then we can use the properties of Hensel Codes(which implies we can process rational numbers with high precision)to overcome these limitations. The merit of our scheme can be summarized as follows:

1. Our scheme supports SIMD by employing NTT and makes full use of plaintext space
2. Our scheme handles rational numbers by Hensel Codes with high precision
3. We show that decoding $(m_1 + m_2)$ and $(m_1 + m_2)\ mod\ (\Phi_m(x), t)$ will yield an equivalent result. The similar result also holds for multiplication
4. The rescaling after multiplication can be avoided

Intuitively, the cyclotomic polynomial $x^n + 1$ where n is a power of two should be decomposed in forms of $\prod(x - g^i)$, so the NTT can be applied. We can achieve the requirement by choosing corresponding parameters elaborately. We show that decoding the discarded part will yield 0 so the limitations are overcome naturally. Hensel Codes is used to handle rational numbers, so the rescaling is avoided and high precision is guaranteed.

1.5 Related work

The scaling approach to encode fix-point numbers is first used to construct homomorphic encryption in [3]. As mentioned above, the rescaling operation should be performed to keep the factor consistent after multiplication. In their work a complex extraction used to extract bits is employed to finish the rescaling. Instead Cheon et al.[22] remove the plaintext modulus to prevent MSBs from being destroyed and use simple division for the rescaling. The non-balanced base-B encoding and some variants[8, 15, 21] suffer from similar limitations discussed in section 1.2. The condition for decoding correctly in [15] is relaxed to some extent(the bounding-box of result is covered by the plaintext space). A rational number is encoded into a continued fraction(which can be represented as integers)in [24]. However, this encoding technique requires

performing very complex arithmetic operations, such as division and modular reduction.

A variant of the FV scheme[31] is proposed by Chen et al[17]. The plaintext space in their construction is isomorphic to $Z/(b^n + 1)Z$. A factor $x - B$ is used for decryption. Intuitively, decrypting the part which is discarded by the reduction modulo $x^n + 1$ will yield 0 so the overflow of computation modulo the cyclotomic polynomial need not be considered. On the other hand, the decryption in the work doesn't involve reduction modulo a plaintext modulus. In fact, our scheme which overcomes the limitations by different approach looks like a version with full SIMD for a short amortized time and the plaintext space in our scheme is isomorphic to Z_t^n(Note that a weak SIMD can also be implemented by CRT in the work). A new HE scheme with Hensel Codes is proposed in [26]. However, the security is not based on the RLWE and the scheme is substantially different from ours.

The construction proposed by Cheon et al.[22] supports SIMD implemented by the FFT different from the CRT adopted in previous works[27, 39]. The overflow modulo the cyclotomic polynomial doesn't influence the decoding because of the deployment of the FFT. The plaintext modulus is removed in the work to prevent the MSBs of the result from being destroyed so the limitation about the overflow modulo the plaintext modulus is overcome(i.e. the plaintext space is R but not R_t). The rescaling need be performed after multiplication since real numbers are handled by scaling. Chen et al.[16] employ a new plaintext space and build a ring homomorphism between it and the plaintext space used in[22]. Therefore they construct a HE scheme supporting SIMD by combining the variant of FV by Bootland et al.[9] with the batching in[22]. However, the plaintext modulus removed in[16, 22] is necessary to employ Hensel Codes which is used in [17] to handle rational numbers for high precision. The batching technique can't be applied for the scheme in[17] because of the modification to the plaintext space. There are many drawbacks for the FFT. For example, a loss of precision is inevitable and modular reduction could not be performed naturally.

1.6 Organization

The paper is organized as follows. In Sec.2, We first introduce how to select corresponding parameters for the NTT, and then review the FV scheme, the batching and hensel codes. In Sec.3, we construct our scheme for rational numbers with high precision in parallel and analyze the correctness and security. Sec.4 presents some techniques for optimizations.

2 Preliminaries

All logarithms are base 2 unless otherwise indicated. We denote vectors in bold, e.g.$\boldsymbol{a}$, and every vector in this paper is a column vector. For simplicity, we make no distinction between a polynomial $c(x)$ and a vector $\boldsymbol{c}$ by the coefficients embedding and use them alternately according to the context. For

a vector $\boldsymbol{a}$ with dimension m and a vector $\boldsymbol{b}$ with dimension n, $(\boldsymbol{a};\ \boldsymbol{b})$ denotes the vector with dimension $m+n$ obtained by concatenating vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ in a vertical direction. We denote by $a \mid b$ that a divides b. For a real number r, $\lfloor r \rceil$ denotes the nearest integer to r and $\lfloor r \rfloor$ denotes the largest integer less than r, rounding upwards in case of a tie. Multiplication of vectors in component-wise way is denoted by $\bigotimes$. For integers modulo $q \in Z_{>0}$, we always use representatives in the symmetric interval $(-q/2, q/2]$. $[\cdot]_q$ and $mod\ q$ denote reduction modulo q. We denote by $\bar{\xi}$ the conjugation of ξ. Operations defined in scalars can be extended to vectors by component-wise way. We use $x \leftarrow D$ to denote sampling x according to a distribution D. $x \leftarrow U(D)$ denotes sampling from the uniform distribution over D when D is a finite set. We let λ denote the security parameter throughout the paper: all known valid attacks against the cryptographic scheme under scope should take 2^λ bit operations.

2.1 Notations

An algebraic number $\xi \in C$ is any root of a polynomial $f(x) \in Q[x]$. The minimal polynomial of ξ is the unique monic irreducible $f(x) \in Q[x]$ with minimal degree having ξ as a root. An algebraic integer is an algebraic number whose minimal polynomial $f(x)$ is in $Z[x]$. The quotient ring $R = Z[x]/f(x)$ where $f(x)$ is a monic irreducible polynomial can be obtained by adjoining an algebraic integer ξ(i.e. $Z[\xi] \cong Z[x]/f(x)$). The residue ring modulo an integer q is denoted by $R_q = R/qR$. An element a in R_q can be represented as $a(\xi) = \sum_{i=0}^{N-1} a_i \xi^i$ whose corresponding vector is denoted by $\boldsymbol{a} = (a_0, a_1, \ldots, a_{N-1})$ where $a_i \in (-q/2, q/2]$ and N is the degree of $f(x)$. The infinity norm $\|a(\xi)\|$ is defined as $max(|a_i|)$ and the expansion factor σ_R is defined as $max(\|ab\|)/(\|a\| \cdot \|b\|)$. In our case, we use a cyclotomic polynomial with degree N power of 2 to generate the ring and set the expansion factor N simply. We denote by χ a discrete Gaussian distribution having standard deviation σ. A distribution over the integers is called B-bounded if it is only supported on $[-B, B]$(with overwhelming probability). The Gaussian distribution with deviation σ is B-bounded and we set $B = 8\sigma$ simply.
The semantic security of encryption schemes presented in this paper is based on the RLWE problem introduced in [35].

Definition 1 (The decision RLWE problem) Let $f(x)$ be a cyclotomic polynomial with degree power of 2. Let $s \in R_q$ where $R = Z[x]/f(x)$ be a random element, $a, a', b' \leftarrow U(R_q)$,$e \leftarrow \chi$ where χ is a Gaussian distribution with some deviation σ. The RLWE problem is to distinguish between $(a, b = a \cdot s + e)$ and (a', b').

RLWE assumption requires that there is no such probabilistic polynomial adversary can solve the problem with non-negligible probability. Let $f(x) = x^N + 1$ where $N = 2^k$ and t be a prime. We decompose $f(x)$ in group Z_t^* in forms of $\prod(x - g^i)$ for NTT.

Lemma 1 Let $2N \mid (t-1)$. There exists an element $g \in Z_t^*$ such that $f(x) = \prod_{i=0}^{N-1}(x - g^{2i+1}) \ mod\ t$.

Proof Let $h(x) = x^{2N} - 1$ and $g \in Z_t^*$ be an element with order $2N$(i.e. $g^{2N} = 1 \ mod\ t$). Note that such g must exist since $2N \mid (t-1)$. We have

$$\begin{aligned} h(x) &= \prod_{i=0}^{2N-1}(x - g^i) \ mod\ t \\ &= (x^N + 1)(x^N - 1) \\ &= f(x) \prod_{i=0}^{N-1}(x - g^{2i}) \end{aligned}$$

It is obvious that the set $\{1, g, \ldots, g^{2N-1}\}$ includes all roots of $h(x)$ in the field Z_t^* so the first equality holds naturally. The third equality holds since g^2 is an element in Z_t^* with order N. We deduce

$$f(x) = \prod_{i=0}^{N-1}(x - g^{2i+1}) \ mod\ t$$

□

As an example, let $N = 4$ and $t = 17$. We have $x^4 + 1 = \prod_{i=0}^{3}(x - 2^{2i+1}) \ mod\ 17$ such that $2^8 = 1 \ mod\ 17$.

2.2 FV scheme

Here we recall the FV scheme[31]. The plaintext space in the FV scheme is R_t where t is referred to as the plaintext modulus and $R_t = Z_t[x]/f(x)$. Cyclotomic polynomials with degree power of 2 are used to construct the ring in general for security and efficiency. In practice error distributions of small width are employed to produce noise for convenience. When using error distributions with small width and considering other rings besides the 2-power cyclotomic rings, there are better known attacks on the RLWE problem[14, 18, 19, 30]. The ciphertext space is $R_q \times R_q$ and $q \gg t$ so there is enough space for the noise to grow.
For simplicity, here we handle integers by the scalar encoding. Let $m \in Z_t$ be a random element. At first, we transform it into a plaintext polynomial $m(x)$ in R_t with the scalar encoding and obtain the corresponding coefficients representation $(m, 0 \ldots, 0)$. Then we invoke the encryption algorithm to get the ciphertext. The following set of algorithms describes the leveled FV scheme.

- FV.SecretKeyGen(1^λ): Sample $s \in R$ with coefficients uniform in $\{-1, 0, 1\}$. Output $sk = s$

- FV.PublicKeyGen(s): Let $s = sk$. Sample $a \leftarrow U(R_q)$, and $e \leftarrow \chi$. Output $([-(a \cdot s + e)]_q, a) \in R_q \times R_q$

- FV.EvaluateKeyGen(sk, T): For $i = 0, 1, \ldots, l = \lfloor log_T q \rfloor$, sample $a_i \leftarrow R_q, e_i \leftarrow \chi$ and return $rlk = [([-(a_i \cdot s + e_i) + T^i \cdot s^2]_q, a_i) : i = 1, \ldots, l]$

- FV.Encrypt(pk, m): To encrypt message $m \in R_t$, let $p_0 = pk[0], p_1 = pk[1]$, $\Delta = \lfloor q/t \rfloor$, sample $u \leftarrow R_2, e_1, e_2 \leftarrow \chi$ and return $ct = ([p_0 \cdot u + e_1 + \Delta \cdot m]_q, [p_1 \cdot u + e_1]_q)$

- FV.Add(ct_1, ct_2): Return $([ct_0[0] + ct_1[0]]_q, [ct_0[1] + ct_1[1]]_q)$

- FV.Mul(ct_1, ct_2, rlk): Compute

$$c_0 = [\lfloor \frac{t(ct_1[0] \cdot ct_2[0])}{q} \rceil]_q$$
$$c_1 = [\lfloor \frac{t(ct_1[0] \cdot ct_2[1] + ct_1[1] \cdot ct_2[0])}{q} \rceil]_q$$
$$c_2 = [\lfloor \frac{t(ct_1[1] \cdot ct_2[1])}{q} \rceil]_q$$

  Write c_2 in base T,i.e. $c_2 = \sum_{i=0}^{l} c_2^{(i)} T^i$ with $c_2 \in R_t$ and set

$$c_0' = [c_0 + \sum_{i=0}^{l} rlk[i][0] \cdot c_2^{(i)}]_q \; and \; c_1' = [c_1 + \sum_{i=0}^{l} rlk[i][1]]_q$$

  return (c_0', c_1')

- FV.Decrypt(sk, ct): Let $s = sk, c_0 = ct[0], c_1 = ct[1]$. Output

$$[\lfloor \frac{t[c_0 + c_1 \cdot s]_q}{q} \rceil]_q \in R_t$$

We refer $(ct[0] + ct[1] \cdot s - \Delta \cdot m)$ as the noise in the ciphertext ct. The condition for correct decryption is that the size of noise in a ciphertext is less than $\Delta/2$ and thus the noise can be removed after rounding. In fact, not only the size of noise but also the encoding scheme can lead a unexpected result as mentioned before. The security of the scheme depends on the hardness of the decision RLWE problem. The following lemma is obtained from standard noise growth argument for the FV[31].

Lemma 2 Let ct_i for $i = 1, 2$ be two ciphertexts, with $[ct_i[s]]_q = [(ct_i[0] + ct_i[1] \cdot s)]_q = \Delta \cdot m + v_i$, and $\|v_i\| < E < \Delta/2$. Set $ct_{add} = FV.Add(ct_1, ct_2)$ and $ct_{mul} = FV.Mul(ct_1, ct_2, rlk)$ then

$$[ct_{add}(s)]_q = \Delta \cdot [m_1 + m_2]_t + v_{add}$$

$$[ct_{mul}(s)]_q = \Delta \cdot [m_1 \cdot m_2]_t + v_{mul}$$
$$\text{with } \|v_{add}\| < 2 \cdot E + t \text{ and } \|v_{mul}\| < E \cdot t \cdot \delta_R(\delta_R + 1.25) + (l+1)B \cdot T \cdot \delta_R/2$$

Assuming that $\|\chi\| < B$, the FV can correctly evaluate circuits of multiplicative depth L with

$$4\delta_R^L(\delta_R + 1.25)^{L+1}t^{L+1} < \lfloor q/B \rfloor$$

2.3 batching in HEAAN

Here, we describe the batching technique employed in the work by Cheon et al.[22] referred to as HEAAN in a simpler way. Instead of encoding one message in a single plaintext polynomial(by the scalar encoding, or other ways), the batching technique allows us to encrypt multiple messages in a plaintext polynomial. Write Z_m^* for the multiplicative group of units in Z_m. The m-th cyclotomic polynomial $\Phi_m(x)$ is defined as $\prod_{k \in Z_m^*}(x - \xi_m^k)$ where $\xi_m = \exp^{2\pi i/m}$. Recall that we have $\Phi_m(x) = x^{m/2} + 1 = \prod_{k=0}^{m/2-1}(x - \xi_m^{2k+1})$ for a power of two integer m. Let $\boldsymbol{z}$ be a vector of complex numbers with dimension $N/2$. We show how HEAAN encodes $\boldsymbol{z}$ as a plaintext polynomial in $R = Z[x]/(X^N + 1)$(Note that the plaintext modulus is removed to prevent the MSBs of result from being destroyed). Intuitively, at most $N/2$ messages can be packed in a plaintext polynomial with degree N since the values of the polynomial at some root ξ_{2N}^{2k+1} and its conjugation $\xi_{2N}^{2N-2k-1}$ are conjugate(Recall that the values of a plaintext polynomial at all roots of $X^N + 1$ are just the messages). The inverse of FFT(IFFT) can be applied to calculate the corresponding coefficients vector $\boldsymbol{c}$ with degree N such that

$$\begin{pmatrix} 1 & \xi_{2N} & \cdots & \xi_{2N}^{N-1} \\ 1 & \xi_{2N}^3 & \cdots & \xi_{2N}^{3(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \xi_{2N}^{2N-1} & \cdots & \xi_{2N}^{(2N-1)(N-1)} \end{pmatrix} \times \boldsymbol{c} = (\boldsymbol{z};\ \bar{\boldsymbol{z}})$$

- Encode(N,$\boldsymbol{z}$): Let $\boldsymbol{Z} = (\boldsymbol{z};\ \bar{\boldsymbol{z}})$ be the vector with dimension N. Let the vector $\boldsymbol{Z'} = (0, \boldsymbol{Z}[0], 0, \boldsymbol{Z}[1], \ldots, \boldsymbol{Z}[N-1])$ with dimension $2N$. Invoke $IFFT_{2N}(\boldsymbol{Z'})$ and get $r(x) - x^N r(x)$. Return the coefficients of $2r(x)$ as a vector $\boldsymbol{c}$ with dimension N.

- Decode(N,$\boldsymbol{c}$): Let $\boldsymbol{C} = (\boldsymbol{c};\ \boldsymbol{0})$ be the vector with dimension $2N$. Invoke $FFT_{2N}(\boldsymbol{C})$ and output a vector $\boldsymbol{Z'}$ with dimension $2N$. Return the vector $[\boldsymbol{Z'}[1], \boldsymbol{Z'}[3], \ldots, \boldsymbol{Z'}[N-1]]$ with dimension $N/2$.

As an example, let $\Phi_8(x) = x^4 + 1$. For a given $\boldsymbol{z} = (3 + 4i, 2 - i)$, let $\boldsymbol{Z'} = [0, 3 + 4i, 0, 2 - i, 0, 2 + i, 0, 3 - 4i]$, invoke $IFFT_8(\boldsymbol{Z'})$ and obtain

$$1.25 - 0.3536x - 1.25x^2 - 0.707x^3 - 1.25x^4 + 0.3536x^5 + 1.25x^6 + 0.707x^7$$

We get $2r(x) = 2.5 - 0.707x - 2.5x^2 - 1.414x^3$ and return $\boldsymbol{c} = (2.5, -0.707, -2.5, -1.414)$.

Lemma 3 Let $\boldsymbol{z}$ be a vector of complex numbers with dimension $N/2$, $\boldsymbol{c} = Encode(N, \boldsymbol{z})$ and $c(x)$ be the corresponding polynomial of $\boldsymbol{c}$(by the coefficients embedding). We have Decode(N,$\boldsymbol{c}$) = $\boldsymbol{z}$ and $c(\xi_{2N}^{2k+1}) = 2r(\xi_{2N}^{2k+1}) = \boldsymbol{z}[k]$ where $k = 0, 1, \ldots, N/2 - 1$.

Proof Recall that we have $m(\xi_{2N}^{j}) = \boldsymbol{Z'}[j]$ $(j = 0, 1, \ldots, 2N - 1)$ where $m(x) = IFFT_{2N}(\boldsymbol{Z'})$ (According to our agreement, we make no distinction between a polynomial and its coefficients vector i.e. $\boldsymbol{c}(x) = c(x)$). It's easy to see that roots of $m(x)$ in the field $\mathbb{C}$ consist of $\{1, \xi_{2N}^{2}, \xi_{2N}^{4}, \ldots, \xi_{2N}^{2N-2}\}$ since $\boldsymbol{Z'}[j] = 0$ for $2 \mid j$. Thus we write $m(x) = (1 - x^N)r(x) = c(x)$. We have $r(\xi_{2N}^{j}) - \xi_{2N}^{j \cdot N} r(\xi_{2N}^{j}) = \boldsymbol{Z'}[j]$. It is obvious $\xi_{2N}^{j \cdot N} = -1$ for an odd number j. We make a conclusion that

$$c(\xi_{2N}^{2k+1}) = 2r(\xi_{2N}^{2k+1}) = \boldsymbol{z}[k] \quad k = 0, 1, \ldots, N/2 - 1$$

We have $\boldsymbol{C}(\xi_{2N}^{j}) = \boldsymbol{c}(\xi_{2N}^{j})$ and $\boldsymbol{c}(\xi_{2N}^{2k+1}) = \boldsymbol{z}[k]$. It's easy to verify Decode(N,$\boldsymbol{c}$) = $\boldsymbol{z}$. □

To finish the batching, the vector $\boldsymbol{c}$ with dimension N should be mapped as a polynomial in R. This can be done by rounding coefficients to the nearest integers. However, this rounding introduces an error that might damage significant bits of input values. To eliminate this error, an input vector is scaled up by some value Δ.

2.4 Hensel Codes

Hensel Codes is used to construct a leveled fully homomorphic encryption with the property of error-free computation(or high precision)[17, 26]. The main idea is to build an isomorphism between a fraction set F_M and Z_p.

$$F_M = \{ \frac{x}{y} | \ |x| \leq M, |y| \leq M \ \}$$

We define a map

$$\Psi_p : \ F_M \to Z_p \quad \frac{x}{y} \to h = x \cdot y^{-1} \ mod \ p$$

where $M = \lfloor \sqrt{(p-1)/2} \rfloor$, and p is a prime. We write Ψ_p as Ψ for simplicity. The inverse of the map is implemented by modified extended Euclidean algorithm. At first, we review how extended Euclidean algorithm(EEA) runs. EEA takes as input two integers x_0 and x_1 and evaluates the greatest common divisor, y and z such that $y \cdot x_0 + z \cdot x_1 = gcd(x_0, x_1)$. The computation generates

the tuples$(x_2, \ldots x_n), (y_2, \ldots y_n), (z_2, \ldots z_n)$ and $q_i = \lfloor x_{i-1}/x_i \rfloor$ such that:

$$\begin{aligned} x_{i+1} &= x_{i-1} - q_i x_i \\ y_{i+1} &= y_{i-1} - q_i y_i \ \ with\ y_0 = 0, y_1 = 1 \\ z_{i+1} &= z_{i-1} - q_i z_i \ \ with\ z_0 = 1, z_1 = 0 \end{aligned}$$

Moreover, for each $i \leq n$, we have $y_i x_1 + z_i x_0 = x_i$. The computation stops with $x_n = 0$ and then x_{n-1} is equal to $gcd(x_0, x_1)$.

Definition 2 (Modified Extended Euclidean Algorithm) Let p be an odd prime, $h \in Z$, and $M = \lfloor \sqrt{(p-1)/2} \rfloor$. Run EEA with $x_0 = p$ and $x_1 = h$(if $h > p$ we simply swap them). Once $|x_i| \leq M$, output $(x, y) = ((-1)^{i+1} x_i, (-1)^{i+1} y_i)$. We write MEEA$(p, h) = (x, y)$.

Now we define the inverse of Ψ

$$\Psi^{-1} : Z_p \to F_M \ \ h \to \frac{x}{y}$$

such that

$$(x, y) = MEEA(p, h)$$

Given $x/y \in F_M$ and an integer k, we have $\Psi^{-1}(\Psi(x/y) + k \cdot p) = x/y$ because MEEA$(p, k \cdot p) = 0$, and $\Psi(\Psi^{-1}(h)) = h$ if $h \in Z_p$[26].

Lemma 4 Let p be an odd prime, $M = \lfloor \sqrt{(p-1)/2} \rfloor$. The following hold:

1. for x_1/y_1 and $x_2/y_2 \in F_M$ such that $x_1/y_1 \neq x_2/y_2$, we have $x_1 y_1^{-1} \neq x_2 y_2^{-1}\ mod\ p$
2. for a given $h \in Z_p$, there exists $x/y \in F_M$ such that $xy^{-1}\ mod\ p = h$
3. Ψ can be seen as an isomorphism between F_M and Z_p when evaluation in F_M is closed

Proof 1. From lemma 1(ii) in [26].
2. It's easy to verify that MEEA(p, h) will stop and return $(x = (-1)^{i+1} x_i, y = (-1)^{i+1} y_i)$ since $\gcd(p, h) = 1 < M$. Moreover, because $y_i h + z_i p = x_i$, then we have $h = y_i^{-1} x_i\ mod\ p = xy^{-1}\ mod\ p$ with $|x_i| < M$.
3. From proposition 3 in [26], we have that $\Psi(x_1/y_1 + x_2/y_2) = \Psi(x_1/y_1) + \Psi(x_2/y_2), \Psi(x_1/y_1 \cdot x_2/y_2) = \Psi(x_1/y_1) \cdot \Psi(x_2/y_2)$ if $x_1/y_1 + x_2/y_2$ and $x_1/y_1 \cdot x_2/y_2$ belong to F_M,. We complete the proof of (3) by combining with (1)(2).

□

3 Leveled homomorphic encryption scheme

3.1 Encoding by NTT

In this section, we show how to encode an integral vector as a plaintext polynomial in R_t. For $2N \mid (t-1)$ we have $x^N + 1 = \prod_{i=0}^{N-1}(x - g^{2i+1}) \ mod\ t$ such that $g^{2N} = 1 \ mod\ t$. Intuitively, for a given integral vector $\boldsymbol{z}$ we can use a similar method in Sec 2.3 to obtain the corresponding coefficients vector $\boldsymbol{c}$ with dimension N such that

$$(\boldsymbol{U} = \begin{pmatrix} 1 & g & \cdots & g^{N-1} \\ 1 & g^3 & \cdots & g^{3(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & g^{2N-1} & \cdots & g^{(2N-1)(N-1)} \end{pmatrix}) \times \boldsymbol{c} = \boldsymbol{z}$$

- EncodeINTT$(N, \boldsymbol{z})$: Let $\boldsymbol{Z} = [0, \boldsymbol{z}[0], 0, \ldots, \boldsymbol{z}[N-1]$. Invoke $INTT_{2N}(\boldsymbol{Z})$ and obtain $r(x) - x^N r(x)$. Return the coefficients $\boldsymbol{c}$ of $2r(x)$.

- DecodeNTT$(N, \boldsymbol{c})$: Let $\boldsymbol{C} = [\boldsymbol{c}, \boldsymbol{0}]$ with dimension $2N$. Invoke $NTT_{2N}(\boldsymbol{C})$ and obtain $\boldsymbol{Z'}$. Return $[\boldsymbol{Z'}[1], \boldsymbol{Z'}[3], \ldots, \boldsymbol{Z'}[2N-1]]$.

Similarly, we can show that $2r(g^{2k+1}) = \boldsymbol{z}[k]$ for $k = 0, 1, \ldots, N-1$. The main observation is that $\{1, g^2, \ldots, g^{2N-2}\}$ consists of all roots of $1 - x^N$ in the field Z_t^*. We have $g^{(2k+1)N} = -1 \ mod\ t$. The correctness of decoding is natural with the relation of NTT and INTT. Different from in FFT and its inverse, computation in NTT and INTT is without loss of precision.

3.2 A concrete scheme

We construct a leveled homomorphic encryption scheme based on the FV scheme with batching. Rational numbers can be handled with high precision. The overflows of computation modulo both plaintext modulus and cyclotomic polynomial will not influence the correctness of decoding. For a given vector of rational numbers, we first encode it as an integral vector with Hensel Codes in component-wise way. Then we employ INTT for batching and get a plaintext polynomial. Finally, the plaintext polynomial is encrypted with the FV scheme.

- SetUp(1^λ): Given the security parameter λ. Choose an integer N(N is a power of two), an integer q, an odd prime t such that $2N|(t-1)$ and $t|q$, set $\Delta = q/t$, $M = \lfloor\sqrt{(t-1)/2}\rfloor$. Set the distributions χ_{key},χ_{err} on $R = Z[x]/f(x)$ where $f(x) = x^N + 1$ for secrets and error, respectively. Choose an integer T.

- KeyGen(1^λ): $sk =$ FV.SecretKeyGen(1^λ), $pk =$ FV.PublicKeyGen(sk), $rlk =$ FV.EvaluateKeyGen(sk, T).

- Ecd$(\boldsymbol{z})$: Given a vector of rational numbers $\boldsymbol{z} \in F_M^N$ with dimension N, compute the integral vector $\boldsymbol{z'} = \Psi_t(\boldsymbol{z})$. Return the plaintext polynomial

$\boldsymbol{c} = EncodeINTT(N, \boldsymbol{z}')$.

- Enc$(pk, \boldsymbol{c})$: $ct =$ FV.Encrypt$(pk, \boldsymbol{c})$.
- Add(ct_0, ct_1): $ct =$ FV.Add(ct_0, ct_0).
- Mul(ct_0, ct_1, rlk): $ct =$ FV.Mul(ct_0, ct_1, rlk).
- Dec(sk, ct): $\boldsymbol{c} =$ FV.Decrypt(sk, ct).
- Dcd$(\boldsymbol{c})$: $\boldsymbol{z}' =$ DecodeNTT$(N, \boldsymbol{c})$. Return $\Psi_t^{-1}(\boldsymbol{z}')$.

It is easy to see the noise growth can also be described by lemma 2. SIMD is implemented by NTT and its inverse and rational numbers are handled by Hensel Codes with high precision. The limitations for decoding correctly can be overcome by combining NTT with Hensel Codes. At first, no rescaling is needed after homomorphic multiplication. Secondly, the overflows of computation modulo both the plaintext modulus and the cyclotomic polynomial have no effect on the correctness of decoding since decoding the discarded part will yield 0. In fact, the condition for decoding correctly is that the result is in F_M^N, which can be met by choosing parameters properly. Note that the condition implies that it's not necessary to pay attention to plaintext polynomials but to consider the range of result directly.

3.3 Correctness and security analysis

Theorem 1 (Correctness) Let sk, pk, rlk be the keys output by KeyGen(1^λ), $\boldsymbol{z}_i$ be a vector where $\boldsymbol{z}_i \in F_M^N (i = 1, 2)$, ct_i be the ciphertext such that $ct_i =$ Enc$(pk$,Ecd$(\boldsymbol{z}_i))$. The HE scheme is correct if the following hold:

1. Dcd(Dec$(sk, ct_i)) = \boldsymbol{z}_i$ for $i = 1, 2$
2. Dcd(Dec$(sk, ct_1 + ct_2)) = \boldsymbol{z}_1 + \boldsymbol{z}_2$ if $\boldsymbol{z}_1 + \boldsymbol{z}_2 \in F_M^N$
3. Dcd(Dec$(sk$,Mul$(ct_1, ct_2, rlk))) = \boldsymbol{z}_1 \bigotimes \boldsymbol{z}_2$ if $\boldsymbol{z}_1 \bigotimes \boldsymbol{z}_2 \in F_M^N$

Proof 1. We have Dec$(sk, ct_i) = c_i(x) = \mathbf{U}^{-1} \cdot \Psi_t(\boldsymbol{z}_i)$ since $e_i < B < \Delta/2$. We can deduce that Dcd$(c_i) = \Psi_t^{-1} \cdot \mathbf{U}(\mathbf{U}^{-1} \cdot \Psi_t(\boldsymbol{z}_i)) = \boldsymbol{z}_i$.

2. Because the encryption scheme is based on the FV scheme, we claim that Dec$(sk, (ct_1 + ct_2)) = (c_1 + c_2)$ *mod* t and Dec$(sk,$ Mul$(ct_1, ct_2)) = (c_1 \cdot c_2)$ *mod* $f(x)$ *mod* t. We complete the proof by showing Dcd$(\boldsymbol{c}_1 + \boldsymbol{c}_2 + \boldsymbol{k} \cdot t)$ $= \boldsymbol{z}_1 + \boldsymbol{z}_2$ and Dcd$((c_1 \cdot c_2 + c(x) \cdot f(x))$ *mod* $t) = \boldsymbol{z}_1 \bigotimes \boldsymbol{z}_2$ respectively where the degree of the polynomial $c_1(x)c_2(x) + c(x) \cdot f(x)$ is less than $f(x)$ and $\boldsymbol{k}$ is an integral vector. We have

$$\begin{aligned} Dcd(Dec(sk, (ct_1 + ct_2)) &= Dcd(\boldsymbol{c}_1 + \boldsymbol{c}_2 + \boldsymbol{k} \cdot t) \\ &= \Psi_t^{-1} \cdot \mathbf{U}(\mathbf{U}^{-1} \cdot \Psi_t(\boldsymbol{z}_1) + \mathbf{U}^{-1} \cdot \Psi_t(\boldsymbol{z}_2) + \boldsymbol{k} \cdot t) \end{aligned}$$

$$
\begin{aligned}
&= \Psi_t^{-1}(\mathbf{U} \cdot \mathbf{U}^{-1}(\Psi_t(\boldsymbol{z}_1) + \Psi_t(\boldsymbol{z}_2)) \\
&= \Psi_t^{-1}(\Psi_t(\boldsymbol{z}_1 + \boldsymbol{z}_2)) \\
&= \boldsymbol{z}_1 + \boldsymbol{z}_2
\end{aligned}
$$

where in the third equality we use the property of Hensel Codes that $\Psi_t^{-1}(\boldsymbol{b} \cdot t + \Psi_t(\boldsymbol{d})) = \boldsymbol{d}$ if $\boldsymbol{d}$ is in F_M^N and $\boldsymbol{b}$ is an integral vector, and the last equality holds since $\boldsymbol{z}_1 + \boldsymbol{z}_2$ is in F_M^N.

3. Let $\boldsymbol{coef}$ be the coefficients vector of the polynomial $c_1(x)c_2(x)+c(x)\cdot f(x)$. We have

$$
\mathbf{U} \cdot \boldsymbol{coef} = \begin{pmatrix} 1 & g & \cdots & g^{N-1} \\ 1 & g^3 & \cdots & g^{3(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & g^{2N-1} & \cdots & g^{(2N-1)(N-1)} \end{pmatrix} \cdot \boldsymbol{coef}
$$

We interpret the vector $\mathbf{U}\cdot\boldsymbol{coef}$ as the values vector of the polynomial $(c_1c_2+c\cdot f)(x)$ at $\{g, g^3, \ldots, g^{2N-1}\}$. We have

$$
(\mathbf{U} \cdot \boldsymbol{coef})[i] = (c_1c_2 + c \cdot f)(g^{2i+1})
$$

On the other hand, $\{g, g^3, \ldots, g^{2N-1}\}$ are all roots of $f(x)$ in Z_t^*. We make a conclusion that

$$
(\mathbf{U} \cdot \boldsymbol{coef})[i] = (c_1c_2)(g^{2i+1}) + k_i \cdot t \cdot c(g^{2i+1})
$$

and

$$
(\mathbf{U} \cdot \boldsymbol{coef})[i] \; mod \; t = (c_1c_2)(g^{2i+1})
$$

The following holds:

$$
Dcd((c_1(x)c(x)_2 + c(x) \cdot f(x)) \; mod \; t) = \Psi_t^{-1} \cdot \mathbf{U}(\boldsymbol{coef} \; mod \; t)
$$

We claim that

$$
\Psi_t^{-1}(\mathbf{U} \cdot \boldsymbol{coef} \; mod \; t)[i] = \boldsymbol{z}_1[i] \cdot \boldsymbol{z}_2[i]
$$

since

$$
\begin{aligned}
\Psi_t^{-1}(\mathbf{U} \cdot \boldsymbol{coef} \; mod \; t)[i] &= \Psi_t^{-1}((\mathbf{U} \cdot \boldsymbol{coef})[i] \; mod \; t) \\
&= \Psi_t^{-1}((c_1c_2)(g^{2i+1})) \\
&= \Psi_t^{-1}(c_1(g^{2i+1}) \cdot c_2(g^{2i+1})) \\
&= \Psi_t^{-1}(\Psi_t(\boldsymbol{z}_1[i]) \cdot \Psi_t(\boldsymbol{z}_2[i])) \\
&= \boldsymbol{z}_1[i] \cdot \boldsymbol{z}_2[i]
\end{aligned}
$$

where the last equality holds since $\boldsymbol{z}_1[i]\cdot\boldsymbol{z}_2[i]$ is in F_M^N. We deduce $Dcd((c_1(x)c_2(x)+c(x)\cdot f(x)) \; mod \; t) = \boldsymbol{z}_1 \bigotimes \boldsymbol{z}_2$. □

Our construction is based on the FV homomorphic encryption scheme whose security is based on the hardness of the RLWE. By the RLWE assumption, the distribution $(b = a \cdot s + e, a)$ is computational indistinguishable from the uniform distribution $U(R_q \times R_q)$. More attacks apply when the secret key is sampled from R_2[2]. There are theoretical results showing that certain small secret RLWE variants are as hard as those with $sk \leftarrow \chi_{err}$, if the dimension N is increased sufficiently[12].

4 Discussion

4.1 Choice of parameters

In this section we discuss how to choose parameters and guarantee a given level of security, and allow a depth L circuit to be evaluated. The discrete gaussian distribution with small width(the deviation $\sigma = 3.2$) is employed to sample the error in general. For a given security level, the homomorphic encryption standardization[1] gives pairs of (N, q) which achieve the security level. The choice of the plaintext modulus depends on the depth L of circuits and the precision needed. On the one hand, we should ensure that the noise doesn't exceed $\Delta/2$ for correct decryption. On the other hand, the result of computation should be in F_M^N for decoding correctly. Only the level of multiplication need be considered in general since the noise growth caused by multiplication is much quicker than addition. However, the level of addition should also be taken into account when Hensel Codes is employed for encoding since the growth of numerator caused by addition may be quicker than multiplication. To alleviate this problem, we choose a chain of denominators such as $\{2, 2^2, \ldots, 2^\varepsilon\}$. Given a vector of rational numbers $\boldsymbol{z}$, before applying Ψ_t we transform each entry of the vector into the rational number $a_i/2^{k_i}$ with the smallest k_i such that $\mid a_i/2^{k_i} - \boldsymbol{z}[i] \mid < \epsilon$ for some small value ϵ. In fact, it's easy to show that $\mid a_i/2^\varepsilon - \boldsymbol{z}[i] \mid < 2^{-\varepsilon}$ for $a_i = \lfloor 2^\varepsilon \cdot \boldsymbol{z}[i] \rceil$. Intuitively, the plaintext modulus should be large enough to ensure the result of computation in F_M^N with high precision. At the same time, the plaintext modulus should be small enough to guarantee the noise is less than $\Delta/2$. A tradeoff should be made to determine the plaintext modulus. In table 1, we present the parameter setting for homomorphic evaluation of power functions with different degrees. The inputs for computation are sampled from the fraction set F_V uniformly(i.e. the numerator and denominator of $\boldsymbol{z}[i] \in [-V, V]$). Noted that in some cases the choice of t presented in the table is not optimal. Some other functions such as exponential functions and sine functions can be evaluated by the Taylor expansion.

The homomorphic evaluation of the circuit x^4 with depth $L = 2$ can be computed simultaneously over 8192 slots. An element with order $2N$ can be obtained easily for NTT according the generator of the group Z_t presented in the table. We show that the parameters are chosen correctly for decryption and decoding. At first, the choice of (N, q) with 128-bit security level follows the homomorphic encryption standardization. Secondly, we have $4\delta_R^L(\delta_R + 1.25)^{L+1}t^{L+1} < \lfloor q/B \rfloor$ with $t = 15 \cdot 2^{44} + 1$. Finally, it's easy to verify $V^4 < \sqrt{(t-1)/2}$ with $V = 58$ so the result of computation is in F_M^N. We make a conclusion that the decryption and decoding can be performed correctly. The others can be analysed in the same way.

Table 1: Choice of parameters for evaluation of typical functions

Fun	L	N	t	generator	q	V
$2x$	0	1024	$3 \cdot 2^{12} + 1$	11	$t \cdot 2^{17}$	39
x^2	1	4096	$3 \cdot 2^{30} + 1$	5	$t \cdot 2^{78}$	200
x^4	2	8192	$15 \cdot 2^{44} + 1$	7	$t \cdot 2^{171}$	58
x^8	3	16384	$27 \cdot 2^{56} + 1$	5	$t \cdot 2^{380}$	13

4.2 Techniques for optimizations

It's obvious that our scheme can be implemented with a short amortized time[22]. Moreover, the NTT and its inverse are employed to implement the decoding and encoding and the algorithm MEEA runs in time $\Theta(\log t)$. We make a conclusion that our scheme is efficient. Here we introduce some techniques for further optimizations.

To handle large numbers, the CRT can be applied with a small plaintext modulus[27]. The main idea is to encrypt one message into multiple ciphertexts, which is not inconsistent with SIMD. On the contrary, the scheme can be modified with flexibility. We encode a message vector multiple times with several co-prime plaintext moduli $t_0, t_1, \ldots, t_k$. Then decoding can be done using the CRT(i.e. $Z_t^N \cong \prod Z_{t_k}^N$ for $t = \prod t_i$). Relatively minor modifications to our scheme are required for the technique.

On the other hand, the CRT representation(a.k.a. Residue Number Systems, or RNS) can also be used for efficiency. The ciphertext modulus can be chosen as the product of some small moduli fitting with practical hardware requirements (machine word, etc.). The need of multi-precision arithmetic can be avoided in almost the whole scheme. The technique used by Bajard et al.[4] for a full variant of FV can be applied in our scheme easily.

5 Conclusion

In this paper, we construct a leveled homomorphic encryption scheme based on the FV scheme with batching by NTT. The deployment of NTT allows us to handle rational numbers by Hensel Codes with high precision in parallel. The limitations about a plaintext polynomial for decoding correctly in some previous works are overcome naturally by combining NTT with Hensel Codes. A unexpected result will never occur in our scheme if parameters are chosen correctly, which is just the main idea of the leveled homomorphic encryption. Some techniques for optimizations on the FV can be applied in our scheme easily.

Declarations

Partial financial support was received from the fundamental research plan of "Release Management Service" in Yunnan Province: Research on Multi-source Data Platform and Situation Awareness Application for Cross-border

Cyberspace Security(No.202001BB050076).
The authors have no competing interests to declare that are relevant to the content of this article. All data generated or analysed during this study are included in this manuscript.

References

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

[2] Martin R Albrecht. On dual lattice attacks against small-secret lwe and parameter choices in helib and seal. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 103–129. Springer, 2017.

[3] Seiko Arita and Shota Nakasato. Fully homomorphic encryption for point numbers. In *International Conference on Information Security and Cryptology*, pages 253–270. Springer, 2016.

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

[5] Josh Daniel Cohen Benaloh. *Verifiable secret-ballot elections*. PhD thesis, Yale University, 1987.

[6] Fabrice Benhamouda, Tancrède Lepoint, Claire Mathieu, and Hang Zhou. Optimization of bootstrapping in circuits. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2423–2433. SIAM, 2017.

[7] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In *Theory of cryptography conference*, pages 325–341. Springer, 2005.

[8] Charlotte Bonte, Carl Bootland, Joppe W Bos, Wouter Castryck, Ilia Iliashenko, and Frederik Vercauteren. Faster homomorphic function evaluation using non-integral base encoding. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 579–600. Springer, 2017.

[9] Carl Bootland, Wouter Castryck, Ilia Iliashenko, and Frederik Vercauteren. Efficiently processing complex-valued data in homomorphic encryption. *Journal of Mathematical Cryptology*, 14(1):55–65, 2020.

[10] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *Annual Cryptology Conference*, pages 868–886. Springer, 2012.

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

[12] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 575–584, 2013.

[13] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on computing*, 43(2):831–871, 2014.

[14] Wouter Castryck, Ilia Iliashenko, and Frederik Vercauteren. Provably weak instances of ring-lwe revisited. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 147–167. Springer, 2016.

[15] Wouter Castryck, Ilia Iliashenko, and Frederik Vercauteren. Homomorphic sim ^{2} d operations: Single instruction much more data. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 338–359. Springer, 2018.

[16] Hao Chen, Ilia Iliashenko, and Kim Laine. When heaan meets fv: a new somewhat homomorphic encryption with reduced memory overhead. In *IMA International Conference on Cryptography and Coding*, pages 265–285. Springer, 2021.

[17] Hao Chen, Kim Laine, Rachel Player, and Yuhou Xia. High-precision arithmetic in homomorphic encryption. In *Cryptographers' Track at the RSA Conference*, pages 116–136. Springer, 2018.

[18] Hao Chen, Kristin Lauter, and Katherine E Stange. Security considerations for galois non-dual rlwe families. In *International Conference on Selected Areas in Cryptography*, pages 443–462. Springer, 2016.

[19] Hao Chen, Kristin Lauter, and Katherine E Stange. Attacks on the search rlwe problem with small errors. *SIAM Journal on Applied Algebra and Geometry*, 1(1):665–682, 2017.

[20] Jung Hee Cheon, Kyoohyung Han, and Duhyeong Kim. Faster bootstrapping of fhe over the integers. In *International Conference on Information Security and Cryptology*, pages 242–259. Springer, 2019.

[21] Jung Hee Cheon, Jinhyuck Jeong, Joohee Lee, and Keewoo Lee. Privacy-preserving computations of predictive medical models with minimax approximation and non-adjacent form. In *International Conference on Financial Cryptography and Data Security*, pages 53–74. Springer, 2017.

[22] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 409–437. Springer, 2017.

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

[24] HeeWon Chung and Myungsun Kim. Encoding rational numbers for fhe-based applications. *Cryptology ePrint Archive*, 2016.

[25] Anamaria Costache, Nigel P Smart, Srinivas Vivek, and Adrian Waller. Fixed-point arithmetic in she schemes. In *International Conference on Selected Areas in Cryptography*, pages 401–422. Springer, 2016.

[26] David W. H. A. da Silva, Luke Harmon, Gaetan Delavignette, and Carlos Araujo. Leveled fully homomorphic encryption schemes with hensel codes. Cryptology ePrint Archive, Report 2021/1281, 2021. https://ia.cr/2021/1281.

[27] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Manual for using homomorphic encryption for bioinformatics. *Proceedings of the IEEE*, 105(3):552–567, 2017.

[28] Léo Ducas and Daniele Micciancio. Fhew: bootstrapping homomorphic encryption in less than a second. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 617–640. Springer, 2015.

[29] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[30] Yara Elias, Kristin E Lauter, Ekin Ozman, and Katherine E Stange. Provably weak instances of ring-lwe. In *Annual Cryptology Conference*, pages 63–92. Springer, 2015.

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

[32] Michael Fellows and Neal Koblitz. Combinatorial cryptosystems galore! *Contemporary Mathematics*, 168:51–51, 1994.

[33] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

[34] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Annual Cryptology Conference*, pages 75–92. Springer, 2013.

[35] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 1–23. Springer, 2010.

[36] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer, 1999.

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

[38] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[39] Nigel P Smart and Frederik Vercauteren. Fully homomorphic simd operations. *Designs, codes and cryptography*, 71(1):57–81, 2014.