New Fully Homomorphic Encryption over the Integers

Gu Chunsheng
School of Computer Engineering
Jiangsu Teachers University of Technology
Changzhou, China, 213001
guchunsheng@gmail.com

Abstract: We first present a fully homomorphic encryption scheme over the integers, which modifies the fully homomorphic encryption scheme in [vDGHV10]. The security of our scheme is merely based on the hardness of finding an approximate-GCD problem over the integers, which is given a list of integers perturbed by the small error noises, removing the assumption of the sparse subset sum problem in the origin scheme [vDGHV10].

Then, we construct a new fully homomorphic encryption scheme, which extends the above scheme from approximate GCD over the ring of integers to approximate principal ideal lattice over the polynomial integer ring. The security of our scheme depends on the hardness of the decisional approximate principle ideal lattice polynomial (APIP), given a list of approximate multiples of a principal ideal lattice. At the same time, we also provide APIP-based fully homomorphic encryption by introducing the sparse subset sum problem.

Finally, we design a new fully homomorphic encryption scheme, whose security is based on the hardness assumption of approximate lattice problem and the decisional SSSP.

Keywords: Fully Homomorphic Encryption, Approximate Lattice Problem, Approximate Principal Ideal Lattice, Approximate GCD, BDDP, SSSP

1. Introduction

We construct a new fully homomorphic encryption schemes based on approximate lattice problem over the integers. Our scheme directly works on the integers without modulus. Our first scheme is to modify their scheme [vDGHV10] to a FHE without the sparse subset sum problem. Then we construct a new APIP-based FHE over the integers. Finally, we design a FHE based on approximate general lattice problem. Now, we describe the second scheme. Assume n the parameter of security, $R = Z[x]/ < x^n + 1 >$ is a ring over the integers. The public key is a list of approximate multiples $\left\{b_i = (a_i f + 2e_i) \bmod (x^n + 1)\right\}_{i=0}^{\tau}, \tau = O(n)$ for a polynomial $f \in R$, where a_i, e_i is the uniformly random elements over R such that $\|a_i\|_\infty \le n^{O(1)}$ and $\|e_i\|_\infty \le n/2$. The secret key is a polynomial $s \in Z[x]$ such that

$(f \times s / p) \bmod (x^n + 1) = 1$, where p is the determinant of the circulant matrix of s, namely $p = \det(Rot(s))$. To encrypt a message bit m, the ciphertext is computed as $c = \sum_{i \in T, T \subseteq \{0,\dots,\tau\}} b_i + 2e + m$, where $\|e\|_\infty \le n/2$. To obtain addition or multiplication of the messages in the ciphertexts, we simply add/multiply the ciphertexts as the addition/multiplication over R. To decrypt a ciphertext c, we compute the message bit $m = (c - f \times \lfloor c \times s / p + 0.5h \rfloor) \bmod x \bmod 2$, where $h = \sum_{i=0}^{n-1} x^i$.

It is easy to see that if we set $n = 1$, $s = 1$, $f = p$, then our scheme in this paper becomes that in [vDGHV10]. So, our scheme adapts their scheme from one dimension to multiple dimensions.

In the above scheme, we use the matries $A \in \mathbb{Z}^{n \times m}$, $T \in \mathbb{Z}^{m \times m}$ to substitute f, s, then we can obtain a FHE based on approximate lattice problem over the integers.

1.1 Our Contribution

Our schemes are different from the previous both underlying the hardness assumption and implementing the method of FHE. Our first scheme removes the hardness assumption of the SSSP in [vDGHV10] to implement FHE. Our second scheme constructs a new FHE based on approximate principal ideal lattice problem over the integers, which extends the scheme of [vDGHV10] from one dimension to multiple dimensions. Our third scheme design a new FHE based on approximate general lattice problem. As far as we know, this approximate lattice problem does not consider in the previous work. The size of the public key in our scheme is $O(n^3 \log n)$ bits, and the expansion factor of ciphertext is $O(n \log n)$. The security of our first scheme relies on the hardness assumption of finding an approximate principle ideal lattice problem (APIP), given a list of approximate multiples of a polynomial f, and solving the sparse subset sum problem. To remove the hardness assumption of SSSP, we design a new fully homomorphic encryption merely based on decisional approximate principle ideal lattice problem. In fact, the objective we hide modulus p is to prevent adversary factoring $x^n + 1 \bmod p$, since $x^n + 1$ and s have a common factor.

In this paper, we design fully homomorphic encryption scheme based on approximate lattice problem over the integers by using self-loop method or circle encrypted secret key. So, we assume our schemes are KDM-secure.

1.2 Related work

Rivest, Adleman, and Dertouzos [RAD78] first investigated a privacy homomorphism, which now is called the fully homomorphic encryption (FHE). Many researchers [BGN05, ACG08, SYY99, Yao82] have worked at this open problem. Until 2009, Gentry [Gen09] constructed the first fully homomorphic encryption using ideal lattice. In Gentry's scheme, the public key is approximately n^7 bits, the computation per gate costs $O(n^6)$ operations. Smart and Vercauteren [SV10] presented a fully homomorphic encryption scheme with both relatively small key $O(n^3)$ bits , ciphertext size $O(n^{1.5})$ bits and computation per gate at least $O(n^3)$ operations, which is in some sense a specialization and optimization of Gentry's scheme. Dijk, Gentry, Halevi, and Vaikuntanathan [vDGHV10] proposed a simple fully homomorphic encryption scheme over the integers, whose security depends on the hardness of finding an approximate integer gcd. Stehle and Steinfeld [SS10] improved Gentry's fully homomorphic scheme and obtained to a faster fully homomorphic scheme, with $O(n^{3.5})$ bits complexity per elementary binary addition/multiplication gate, but the hardness assumption of the security of the scheme in [SS10] is stronger than that in [Gen09].

1.3 Outline

We recalls some notations and definitions in Section 2, and then this paper is organized in two parts. In Part I, we construct a fully homomorphic encryption based on hidden odd integers. We first describe a somewhat homomorphic encryption scheme in Section 3, then transform it into a FHE in Section 4, and finally give its security in Section 5. In Part II, we adapt the above FHE from one dimension to multiple dimensions. First, we construct a new somewhat homomorphic encryption in Section 6, then transform it into a fully homomorphic encryption by introducing the hardness of SSSP in Section 7. What is more, we describe a new FHE by using method of re-randomizing the secret key 1/*p*, and give the hardness assumption of the security of scheme. In part III, we construct a new FHE based on approximate lattice problem over the integers. In Section 13, we give further direction.

2. Preliminaries

2.1 Notations

Let n with the power of 2 be a security parameter. $[n]=\{0,1,...,n\}$. Let $R=Z[x]/<x^n+1>$. For $f\in R$, we denote by $\|f\|_\infty$ the infinity norm of its coefficient vector, $[f]_2$ the polynomial of its coefficient modulo 2. For R, its expansion factor γ_{mul} is n, that is, $\|u \times v\|_\infty \leq n \cdot \|u\|_\infty \cdot \|v\|_\infty$, where $\times$ is multiplication in R.

Let $w \leftarrow_\psi S$ denote to choose an element w in S according to the distribution ψ. For the distributions A, B, $A \equiv_c B$ is computationally indistinguishing by arbitrary probabilistic polynomial time algorithm.

2.2 Lattice

Given n linearly independent vectors $b_1, b_2, ..., b_m \in \mathbb{R}^n$, the lattice is equal to the set $L(b_1, b_2, ..., b_m) = \{\sum_{i=1}^m x_i b_i, x_i \in \mathbb{Z}\}$ of all integer linear combinations of the b_i's. We also denote by matrix B the b_i's. In this paper, we only consider the lattice over the integers, i.e., $b_i \in \mathbb{Z}^n$.

An ideal $I \subseteq R$ is a principal if it only has a single generator. For the coefficient vector $\vec{u} = (u_0, u_1, ..., u_{n-1})^T$ of $u \in R$, we define the cyclic rotation $rot(\vec{u}) = (-u_{n-1}, u_0, ..., u_{n-2})^T$, and its corresponding circulant matrix $Rot(u) = (\vec{u}, rot(\vec{u}), ..., rot^{n-1}(\vec{u}))^T$. $Rot(u)$ is called the rotation basis of the ideal lattice (u). The detail may be found in the [Mic07]. For $f, u \in R$, $[f]_u$ is the coefficient vector of f modulo the rotation basis of u, namely, $\vec{f} \bmod Rot(u)$.

2.3 Approximate Lattice Problem

In the following, we define the approximate-GCD from [vDGHV10], and extend it to the approximate principal ideal lattice problem in this paper.

Definition 2.1. (Approximate-GCD over the Integers (AGCD)). Given a list of approximate multiples of p: $\{b_i = a_i p + e_i : a_i \in Z_+, e_i \in Z, |e_i| < 2^{n-1}\}_{i=0}^{\tau}$, find p.

Definition 2.2. (Approximate Principal Ideal Lattice Problem (APIP)). For a polynomial $f \in R$ and a distribution φ over R subject to $e \leftarrow_\varphi R$ and $||e||_\infty \leq n/2$, the distribution $H_{f,\varphi}$ over R is generated by choosing uniformly random element $a \leftarrow_U R$

and $e \leftarrow_{\varphi} R$, and outputting $b=(a \times f+2e)\bmod(x^n+1)$. The APIP problem, denoted $APIP_{f,\varphi}$, is defined as follows: Given access to arbitrary many independent samples from $H_{f,\varphi}$, find f. The decision version of APIP, denoted $dAPIP_{f,\varphi}$, is to distinguish $H_{f,\varphi}$ from $g \leftarrow_U R$.

Definition 2.3. (Approximate Lattice Problem (ALP)). Let n,m be integers related to security parameter λ, and χ a distribution over $\mathbb{Z}^m$. Given a list samples b_i of the distribution $D_{n,m,\chi}$ over $\mathbb{Z}^m$ such that $A \leftarrow \mathbb{Z}^{n\times m}$, $s_i \leftarrow \mathbb{Z}^n$, $e_i \leftarrow \chi$ and $b_i = s_i A + 2e_i$, the ALP $ALP_{n,m,\chi}$ is to distinguish the distribution $D_{n,m,\chi}$ from the uniform distribution over $\mathbb{Z}^m$.

Definition 2.4. (Decision Bounded Distance Decoding Problem). For R, the challenger sets $\alpha \leftarrow_R \{0,1\}$ and $b_0=(a_0 \times f)\bmod(x^n+1)$. If $\alpha=0$, it samples $r_1 \leftarrow_R H_{f,\varphi}$ and sets $r = r_1 \bmod Rot(b_0)$. If $\alpha=1$, it samples r uniformly from $R \bmod Rot(b_0)$. The problem is to guess α given (r,b_0).

Part I FHE-1 Based on Approximate GCD Problem

3. Somewhat Homomorphic Encryption (SHE-1)

In this section, we present a somewhat homomorphic encryption, which is similar to that in [vDGHV10] and simply analyze its performace in this section.

3.1 Construction

Key Generating Algorithm (SHE-1.KeyGen).

(1) Select an odd integer $p > 2^{n^2+3}$ such that $s \approx 1/p$, $sp = 1+O(2^{-n^2-3})$, and $h(s)=\omega(\log n)$, where $h(s)$ is the number of 1 in the binary representation of s.

(2) Pick random integers $a_i \in (2^{O(n)}, 2^{n^2})$ subject to the largest a_0 is an odd integer, $e_i \in Z, i \in [\tau]$ with $|e_i| < 2^{n-1}$. Then compute $b_0 = a_0 p + 2e_0$, and $[b_i = a_i p + 2e_i]_{b_0}$.

(3) Choose $t = O(n)$ approximate integers $\{d_i = a_i \times p + 2e_i\}_{i=0}^{t}$ with $|e_i| < 2^{n-1}$ such that $d_{i+1} / d_i < 2^n$, $b_0^2 / d_t < 2^n$, and $d_0 = b_0$.

(4) Output the public key $pk = (n, \{b_i\}_{i=0}^{\tau}, \{d_i\}_{i=0}^{t})$, and the secret key $sk = (p)$.

Encryption Algorithm (SHE-1.Enc). Given the public key pk and an message bit $m \in \{0,1\}$, choose a random subset $T \subseteq [\tau]$ and an independent perturbed error polynomial e with $|e| < 2^{n-1}$. Compute the ciphertext $c = \left[\sum_{i \in T} b_i + 2e + m\right]_{b_0}$.

Add Operation (SHE-1.Add). Given the public key pk, and the ciphertexts c_1, c_2, evaluate the ciphertext $c = \left[c_1 + c_2\right]_{b_0}$.

Multiplication Operation (SHE-1.Mul). Given the public key pk, and the ciphertexts c_1, c_2, evaluate the ciphertext $c = \left[Opt(c_1 \times c_2)\right]_{b_0}$, where Opt is same as the optimizations of Section 3.3 in [vDGHV10].

Decryption Algorithm (SHE-1.Dec). Given the secret key sk, and a ciphertext c, decipher $m = \left[c\right]_p \bmod 2$.

Remark 3.1: To quickly generate p, we may select $s = \sum_{j=n^2+3}^{2n^2+6} s_j 2^{-j}$ with $h(s) = \omega(\log n)$ and $len(s) = 2n^2 + 6$, such that its inverse p is an odd integer and $sp = 1 + O(2^{-n^2-3})$, where *len* is the length of *s* in binary representation.

Example 3.1. Let $n = 4$. We select at random $s = \sum_{j=19}^{38} s_j 2^{-j} = 2^{-22} + 2^{-28} + 2^{-29}$, where $h(s) = \omega(\log n) = 3$, $len(s) = 2n^2 + 6 = 38$, and compute $p = \lfloor 1/s \rfloor = 4098251$. It is easy to verify that $s \bullet p = 0.9999999423 = 1 + O(2^{-19})$. Now, we can use $p = 4098251$ as the secret key in the above SHE.

3.2 Performance of SHE-1

The size of the public key $pk = (n, \{b_i\}_{i=0}^{\tau}, \{d_i\}_{i=0}^{t})$ is $O(n^3)$ bits, the size of the secret key $sk = (p)$ is $O(n^2)$. The running times of Enc, Dec, Add, Mul are $O(n^3)$, $O(n^2)$,

$O(n^2)$, and $O(n^2 \log n)$, respectively. The expansion factor of ciphertext is $O(n^2)$.

4. Fully Homomorphic Encryption (FHE-1)

We first construct a new fully homomorphic shceme from SHE-1 by applying self-loop Gentry's bootstrappable technique, then discusses how to remove self-loop bootstrappable technique. Since the multiplication operation increase the degree of perturbed error noise, we require to reduce it to obtain fully homomorphic encryption. We refresh a ciphertext c to a new ciphertext c_{new} with the smaller error noise by using Gentry's bootstrappable technique. To implement this function, we encrypt the secret key s generated by KeyGen and add the ciphertexts of s to the public key.

4.1 FHE-1 Scheme

FHE-1.KeyGen Algorithm.

(1) First, generate pk and sk as SHE.

(2) Assume $s = \sum_{j=n^2+3}^{2n^2+6} s_j 2^{-j}$. Choose random integers $a_j \in (2^{O(n)}, 2^{n^2})$, $e_j \in Z$ with $|e_j| < 2^{n-1}$, $j \in [n^2+3]$, and compute $\left[\vec{s}_j = a_j p + 2e_j + s_{j+n^2+3}\right]_{b_0}$.

(3) Output the public key $pk^* = (n, w, \{b_i\}_{i=0}^{\tau}, \{d_i\}_{i=0}^{t}, \vec{s} = \sum_{j=0}^{n^2+3} \vec{s}_j 2^{-(j+n^2+3)})$, and the secret key $sk^* = (p)$, where $w = h(s)$.

The Enc, Dec, Add, Mul algorithms are identical to ones in the above SHE.

Remark 4.1: We may also generate the secret key as follows. Choose an arbitrary odd integer p and a random fraction s_1 with $h(s_1) = \omega(\log n)$, and compute p's inverse $s = s_1 + s_2 \approx 1/p$. The public key includes s_2 and the ciphertexts $\vec{s}_1$ of the bits of s_1. Now, the public key is modified into $pk = (n, w, \{b_i\}_{i=0}^{\tau}, \{d_i\}_{i=0}^{t}, \vec{s}_1 = \sum_{j=0}^{n^2+3} \vec{s}_j 2^{-(j+n^2+3)}, s_2)$. It is not difficult to verify that the above parameters can implement FHE.

Example 4.1. Let $n = 4$. We select at random an odd integer $p = 534019$ and a fraction $s_1 = \sum_{j=19}^{38} s_j 2^{-j} = 2^{-22} + 2^{-25} + 2^{-34}$ with $h(s_1) = 3$, set $s = s_1 + s_2 \approx 1/p$, and compute $s_2 = 2^{-20} + 2^{-21} + 2^{-23} + 2^{-25} + 2^{-26} + 2^{-27} + 2^{-29} + 2^{-31} + 2^{-34} + 2^{-35} + 2^{-36} + 2^{-37}$. It is easy to verify that $s \bullet p = 0.9999999423 = 1 + O(2^{-19})$. Now, we can use $p = 534019$ as the secret key in the above SHE.

Recrypting algorithm (FHE-1.Recrypt). Evaluate a new ciphertext

$$c_n = \lfloor c \times \bar{s} + 0.5 \rfloor \bmod 2 \oplus c \bmod 2 .$$

Theorem 4.1. FHE-1.Recrypt correctly generates a 'fresh' ciphertext c_{new} with the same message of c and the perturbed error noise e subject to $|2e| < (p/8)^{1/2}$.

Proof: We know the general form of ciphertext $c = ap + 2e + m$ subject to $|2e| \leq p/8$. So,

$$\lfloor c \times s + 0.5 \rfloor \bmod 2 = \lfloor (ap + 2e + m) \times s + 0.5 \rfloor \bmod 2 = a \bmod 2 .$$

By using $c_0 = c \bmod 2 = (ap + 2e + m) \bmod 2 = a \bmod 2 + m$, we obtain the message $m = c_0 + a \bmod 2$. Thus, Recrypt only substitutes s with $\bar{s}$, which is the form of the ciphertexts of bits in s. It is not difficult to verify that FHE-1.Recrypt algorithm correctly computes a new ciphertext c_{new} of m in c by using the ciphertext arithmetic circuit and the fact $h(s) = \omega(\log n)$, and c_{new} has the error noise less than $|2e| < (p/8)^{1/2}$, namely, it now can carry out at least one multiplication operation. Notice that FHE-1.Recrypt uses the methods of the hamming weights, the symmetric polynomials and the three-for-two, all of which are explained in [Gen09, vDGHV10]. □

Now we only need to prove our scheme can compute the circuit depth of FHE-1.Recrypt.

Lemma 4.1. The FHE-1.Dec algorithm from the above scheme is correct, if the error noise of ciphertext is less than $p/8$ when decrypted.

Lemma 4.2. The above scheme is correct for arbitrary arithmetic circuit C with addition and multiplication gates, and circuit depth $d = \log_2 n$.

Proof. Assume $c_j = a_j p + 2e_j + m_j$, $j = 1, 2$ are the ciphertexts of arbitrary two bits of s generated by FHE-1.KeyGen in FHE-1. To correctly decrypt, the perturbed error noise of ciphertext output by arithmetic circuit can not be too large. The error noise in addition gate is linearly rising, whereas the error noise in multiplication gate is exponentially increasing. So, the multiplication operation dominates the depth of arithmetic circuit. Now, we estimate the bound of the perturbed error term in the ciphertext generated by one multiplication operation.

$$\begin{aligned} c &= c_1 \times c_2 \\ &= (a_1 p + 2e_1 + m_1) \times (a_2 p + 2e_2 + m_2) . \\ &= (a \times f + 2e + m_1 m_2) \end{aligned}$$

where $a = (a_1 p + 2e_1 + m_1) \times a_2 + 2a_1 e_2 + a_1 m_2$, $e = e_1 \times (2e_2 + m_2) + m_1 e_2$.

So, $|2e| = |2e_1 \times (2e_2 + m_2) + 2m_1 e_2| < 2^{2n}$.

Since the perturbed error noise in the ciphertexts c_1, c_2 are less than 2^n. So, the error term for one multiplication operation is less than $(2^n)^2$. Thus, To correctly decrypt, the depth d of arithmetic circuit must be satisfied inequality $(2^n)^{2^d} \le p/8$, namely,

$d = \log(\log(p/8)/n) = \log_2 n$. □

4.2 Performance of FHE-1

For our FHE, the size of the public key pk^* is $O(n^4)$ the size of the secret key $sk^* = (p)$ is $O(n^2)$. The expansion factor of ciphertext is $O(n^2)$.

5. Security of FHE-1

5.1 Security Reduction

The security of our scheme is based on the hardness of the approximate-GCD over integers, which follows from Theorem 4.2 in [vDGHV10].

Theorem 5.1. Suppose there is an algorithm A which breaks the semantic security of our SHE with advantage ε. Then there is an algorithm D for solving AGCD with advantage at least $\varepsilon/2$. The running time of D is polynomial in the running time of A, and $1/\varepsilon$.

5.2 Known Attack

Since our scheme is similar to the scheme in [vDGHV10], all known attacks for their scheme are also appropriate for our scheme. But for the approximate GCD of many numbers attacked by using the LLL algorithm, we analyze as follows.

To simply describe, we use the same notations as that in [vDGHV10], and only adapt to our corresponding parameters. For the target solution vector

$$\vec{v} = (a_0, a_1, ..., a_t) \times M = (a_0 2^n, b_0 a_0 (b_1/b_0 - a_1/a_0), ..., b_0 a_0 (b_t/b_0 - a_t/a_0)),$$

Where $a_0 2^n \le 2^{O(n^2)}$ and $b_0 a_0 (b_i/b_0 - a_i/a_0) \le 2^{O(n^2)}$. First, $\vec{v}$ maybe is not the shortest nonzero vector in L, because the length of the first row vector in the matrix M is also $\|(2^n, b_1, b_2, ..., b_t)\|_2 = 2^{O(n^2)}$. Second, for large t, there are exponentially many vectors in L of length at most $2^{O(n^2)}$.

Thus, to guarantee the security of our scheme, the parameters in our scheme can resist this attack, and do not need to set n^5 the size of public key in [vDGHV10].

Part II FHE-2 Based on Approximate Ideal Lattice Problem

6. Somewhat Homomorphic Encryption (SHE-2)

In this section, we extends SHE-1 in Section 3 from approximate GCD over the integer ring to approximate principal ideal lattice over the polynomial ring, and construct a new somewhat homomorphic encryption scheme based on approximate principal ideal lattice problem.

6.1 Construction

Key Generating Algorithm (SHE-2.KeyGen).

(1) Select a random polynomial $s=\sum_{i=0}^{n-1} s_i x^i$ such that $p=\det(Rot(s))>2^{n\log n}$ is an odd number and $\|s\|_\infty \le n$.

(2) Evaluate f over R subject to $(s\times f)/p=1\bmod(x^n+1)$.

(3) Compute $b_i=\left[a_i\times f+2e_i\right]_{b_0}$ with $\tau=O(n)$, $\|a_i\|_\infty \le O(2^n)$ and $\|e_i\|_\infty \le n/2$, where $\|b_i\|_\infty \le \|b_0\|_\infty$.

(4) Choose $t=O(n)$ at random $a_i, e_i \in R, i\in[t]$ such that $\|d_i\|_\infty / \|b_0\|_\infty \le n^{i+1}$, and $\|e_i\|_\infty \le n/2$, and then compute $\{d_i=a_i\times f+2e_i\}_{i=1}^t$.

The public key is $pk=(n,\{b_i\}_{i\in[\tau]},\{d_i\}_{i\in[t]})$, the secret key is $sk=(p,s,[f]_2)$.

Encryption Algorithm (SHE-2.Enc). Given the public key pk and an message bit $m\in\{0,1\}$, choose a random subset $T\subseteq[\tau]$ and an independent 'small' error term e with $\|e\|_\infty \le n/2$. Compute the ciphertext $c=\left[\sum_{i\in T} b_i+2e+m\right]_{b_0}$.

Add Operation (SHE-2.Add). Given the public key pk, and the ciphertexts c_1, c_2, evaluate the ciphertext $c = [c_1 + c_2]_{b_0}$.

Multiplication Operation (SHE-2.Mul). Given the public key pk, and the ciphertexts c_1, c_2, evaluate the ciphertext $c = Opt(c_1 \times c_2)$ such that $\|c\|_\infty \le \|b_0\|_\infty$, where Opt is similar as that in [vDGHV10], namely, $c = [[[[[c_1 \times c_2]_{d_t}]_{d_{t-1}}]...]_{d_0}]_{b_0}$.

Decryption Algorithm (SHE-2.Dec). Given the secret key sk, and the ciphertext c, decipher

$m = \left[\lfloor (c \times s / p + 0.5h \rfloor \times [f]_2 \bmod x\right]_2 \oplus [c \bmod x]_2$.

Remark 6.1: It is not difficult to show that the coefficients of quotient all are small for the operation of each modulus d_i according to the size of $\|(d_i)^{-1}\|_\infty$ over the rational number $\mathbb{Q}$.

6.2 Correctness

Lemma 6.l. The SHE-2.Dec is correct, if the infinity norm of the error term in the ciphertext is less than $\lfloor p/(8n) \rfloor$ when decrypted.

Proof. Given the ciphertext c and the secret key sk, it is not difficult to verify that c has the form $c = a \times f + 2e + m$. To decrypt the ciphertext c, we simply compute

$$\begin{aligned}
&\left[\lfloor c \times s / p + 0.5h \rfloor \times [f]_2 \bmod x\right]_2 \oplus [c \bmod x]_2 \\
&= \left[\lfloor (a \times f + 2e + m) \times s / p + 0.5h \rfloor \times [f]_2 \bmod x\right]_2 \oplus [c \bmod x]_2 \\
&= \left[\lfloor a \times (f \times s / p) + (2e + m) \times s / p + 0.5h \rfloor \times [f]_2 \bmod x\right]_2 \oplus [c \bmod x]_2 \\
&= \left[a \times [f]_2 \bmod x\right]_2 \oplus [c \bmod x]_2 \\
&= \left[[a]_2 \times [f]_2 \bmod x\right]_2 \oplus \left[([a]_2 \times [f]_2 + m) \bmod x\right]_2 \\
&= m
\end{aligned}.$$

Since $\|2e\|_\infty < \lfloor p/8n \rfloor$, $\|(2e+m)/p \times s\|_\infty \le 1/(8n) \times \|s\|_1 < 1/8$.

It is easy to verify that all other algorithms are also correct in the above scheme.

6.3 Performance of SHE-2

The size of public key $pk = (n, \{b_i\}_{i \in [\tau]}, \{d_i\}_{i \in [t] \backslash 0})$ is $O(n^3 \log n)$, the size of secret key

$sk = (p, s, [f]_2)$ is $O(n \log n)$. The expansion factor of ciphertext is $O(n^2 \log n)$. The running times of Enc, Dec, Add, Mul algorithm is respectively $O(n^3 \log n)$, $O(n^2 \log n \log\log n)$, $O(n^2 \log n)$, and $O(n^3 \log n \log\log n)$.

7. Fully Homomorphic Encryption (FHE-2)

To construct an FHE from SHE, we need to give a new algorithm Recrypt, which freshens a 'dirty' ciphertext c into a new ciphertext c_{new} with the 'smaller' error term and the same plaintext of c. To do this, we introduce the sparse subset sum problem and add the hint of the secret key to the public key. Now, we modify the SHE as follows:

7.1 Construction

Key Generating Algorithm for FHE-2 (FHE-2.KeyGen).

(1) Generate $pk = (n, \{b_i\}_{i\in[\tau]}, \{d_i\}_{i\in[t]\backslash 0})$ and $sk = (p, s, [f]_2)$ as before.

(2) Choose at random a set S_1 of t_1 polynomials $g_i \in Q[x]$ with $\|g_i\|_\infty < 2$ such that there is a subset S_2 of t_2 polynomials with $\left\| \left[\sum_{i\in S_2} g_i \right]_2 - s/p \right\|_\infty < \frac{1}{p^2}$.

(3) Set $sk_i = 1$ for $i \in S_2$ and $sk_i = 0$ for $i \in S_1 - S_2$.

(4) Encrypt sk_i as $\overrightarrow{sk}_i = a_i \times f + 2e_i + sk_i$ with $\|a_i\|_\infty \le O(2^n)$ and $\|e_i\|_\infty \le n/2$.

(5) Encrypt $[f_j]_2$ as $\vec{f}_j = a_j \times f + 2e_j + [f_j]_2$ with $\|a_j\|_\infty \le O(2^n)$ and $\|e_j\|_\infty \le n/2$.

Let $\overrightarrow{[f]}_2$ denote the ciphertext polynomial of $[f]_2$.

(6) Output the secret key $sk = (p, s, [f]_2)$ and the public key $pk = (n, \{b_i\}_{i=0}^{\tau}, t_1, t_2, \{\overrightarrow{sk}_i, g_i\}_{i\in S_1}, \overrightarrow{[f]}_2)$.

7.2 Recrypt Algorithm

Recrypting Algorithm (FHE-2.Recrypt(*pk*, *c*)).

(1) Compute $r_i = [c \times g_i]_2$, keeping only $\theta = \lceil \log s_2 \rceil + 3$ bits of precision after the

binary point for each coefficient of r_i.

(2) Evaluate $u_i = r_i \times \overline{sk}_i$, $u = \left[\left\lfloor\left[\sum_{i \in S_1} u_i\right]_2 + 0.5h\right\rfloor\right]_2$ by using the symmetric polynomials in [GH10].

(3) Output a new ciphertext $c_{new} = [c \bmod x]_2 \oplus \left[(u \times \overline{[f]_2}) \bmod x\right]_2$.

Theorem 7.1. FHE-2.Recrypt correctly generates a 'fresh' ciphertext c_{new} with the same message of c , and support a product of two recrypting new ciphertexts when $(n^2 t_1 t_2)^{2t_2} < \frac{p}{8n}$.

Proof: It is not difficult to verify that FHE-2.Recrypt correctly generates a new ciphertext of m by using the method of symmetric polynomials over ciphertext operations. So, we only need to analyze the perturbed noise size in FHE-2.Recrypt. To simplify analysis, we set $t_2 = 2^{k_1} - 1$ where k_1 is an integer, and use the similar method of analysis as that in [GH10].

First, we know there are t_2 nonzero ones in t_1 ciphertext numbers. So, by applying the symmetric polynomial technique, we merely need to use the polynomial with total degree-t_2 to compute the sum of t_1 ciphertext numbers. It is easy to verify that the number of degree-t_2 monomials in the polynomial representing ciphertext additions is equal to $\begin{pmatrix} t_2 \\ t_2/2 \end{pmatrix} \times \begin{pmatrix} t_2 \\ t_2/4 \end{pmatrix} \times \ldots \times \begin{pmatrix} t_2 \\ 1 \end{pmatrix}$, which is less than ${t_2}^{(t_2-1)}$. By the induction method, we can show the size of the error term of a degree-t_2 monomial at most n^{2t_2}. Moreover, the number of degree-t_2 monomial in the symmetric polynomial over t_1 variables is at most ${t_1}^{t_2}$. So, the size of the error term of recrypting ciphertext is at most $(n^2 t_1 t_2)^{t_2}$. At the same time, we must support another ciphertext multiplication for freshing ciphertext. Hence, to support fully homomorphic encryption, our scheme must correctly decrypt a ciphertext with error term $(n^2 t_1 t_2)^{2t_2}$. Thus, we have $(n^2 t_1 t_2)^{2t_2} \le \frac{p}{8n}$ by applying Lemma 3.2.■

8. Fully Homomorphic Encryption (FHE-2v)

To implement FHE-2, we introduce the assumption of SSSP. In this section, we first give a new fully homomorphic encryption scheme by using self-loop bootstrappable technique without the assumption SSSP, whose security merely depends on the hardness of solving decisional approximate principal ideal lattice. Then, we discuss how to removing the self-loop of FHE-2.

8.1 Construction

Key Generating Algorithm (FHE-2v.KeyGen).

(1) Select $s=\sum_{i=0}^{n-1}s_i x^i$ with $s_i \in \{0,1,2^1,...,2^{\log n}\}$ and $\|s\|_\infty = n$ such that $p=\det(Rot(s))>2^{n\log n}$ is an odd integer, $k=W(s)=\sum_{i=0}^{n-1}w(s_i)=\omega(\log n)$, where $w(s_i)$ is the hamming weight of s_i and $w(s_i)\le 1$.

(2) Choose a random binary fraction $v_1=\sum_{j=n\log n}^{2n\log n+3}v_{1,j}2^{-j}$ with $w(v_1)=\omega(\log n)$, and compute $v_2=1/p-v_1$ with $2n\lceil \log n\rceil+3$ bits of precision after the binary point.

(3) Compute its inverse f over R subject to $(s\times f)/p=1\bmod(x^n+1)$.

(4) Generate $b_i=\left[a_i\times f+2e_i\right]_{b_0}$ with $\tau=O(n)$, $\|a_i\|_\infty\le O(2^n)$ and $||e_i||_\infty\le n/2$. Assume $\|b_i\|_\infty\le\|b_0\|_\infty$.

(5) Choose $t=O(n)$ at random $a_i,e_i\in R,i\in[t]$ such that $\|d_i\|_\infty/\|b_0\|_\infty\le n^{i+1}$, and $||e_i||_\infty\le n/2$, and then compute $\{d_i=a_i\times f+2e_i\}_{i=1}^{t}$.

(6) Encrypt the *j*-th bit $s_{i,j}$ of s_i as $\vec{s}_{i,j}=a_{i,j}\times f+2e_{i,j}+s_{i,j}$ with $||a_{i,j}||_\infty=O(2^n)$, $||e_{i,j}||_\infty\le n/2$ for $i\in[n]$, $j\in[\log n]$. Let $\vec{s}=\sum_{i=0}^{n-1}\vec{s}_i x^i=\sum_{i=0}^{n-1}(\sum_{j=0}^{\log n}\vec{s}_{i,j}2^j)x^i$.

(7) Encrypt $v_{1,j}$ as $\vec{v}_{1,j}=a_j\times f+2e_j+v_{1,j}$. Let $\vec{v}_1=\sum_{j=n\log n}^{2n\log n+3}\vec{v}_{1,j}2^{-j}$ and $\vec{v}=\vec{v}_1+v_2$.

(8) Encrypt $[f_j]_2$ as $\vec{f}_j=a_j\times f+2e_j+[f_j]_2$ with $||a_j||_\infty\le O(2^n)$ and $||e_j||_\infty\le n/2$, denoted as $\overrightarrow{[f]}_2$.

(9) Output the secret key $sk^*=(p,s,[f]_2)$ and the public key $pk^*=(n,k,\{b_i\}_{i=0}^{\tau},\{d_i\}_{[t]\backslash 0},\vec{s},\vec{v},\overrightarrow{[f]}_2)$.

8.2 Recrypt Algorithm

Recrypting Algorithm (FHE-2v.Recrypt(*pk*, *c*)).

(1) Evaluate $\vec{r}=c*\vec{v}$, $\vec{z}=\left[\left\lfloor\vec{r}\times\vec{s}+0.5h\right\rfloor\right]_2$ and $\vec{u}=\left[(\vec{z}\times\overrightarrow{[f]}_2)\bmod x\right]_2$.

(2) Output a new ciphertext $c_{new} = \vec{u} \oplus [c \bmod x]_2$.

Theorem 8.2. FHE-2v.Recrypt correctly generates a 'fresh' ciphertext c_{new} with the same message of c, and support a product of two recrypting new ciphertexts for the above parameters.

Proof. By Lemma 6.1, we have

$$\begin{aligned} m &= \left[\lfloor c \times s / p + 0.5h \rfloor \times [f]_2 \bmod x\right]_2 \oplus [c \bmod x]_2 \\ &= \left[\lfloor r \times s + 0.5h \rfloor \times [f]_2 \bmod x\right]_2 \oplus [c \bmod x]_2 \\ &= \left[z \times [f]_2 \bmod x\right]_2 \oplus [c \bmod x]_2 \\ &= u \oplus [c \bmod x]_2 \end{aligned}.$$

So, we merely need to prove that FHE-2v.Recrypt can correctly implement the above algorithm when substituting $s, 1/p, [f]_2$ by $\vec{s}, \vec{v}, \overline{[f]}_2$. First, we have

$$\vec{r} = c * \vec{v} = \sum_{i=0}^{n-1} c_i x^i * (\vec{v}_1 + v_2) = \sum_{i=0}^{n-1} c_i \vec{v}_1 x^i + \sum_{i=0}^{n-1} c_i v_2 x^i = \sum_{i=0}^{n-1} \vec{r}_{1,i} x^i + \sum_{i=0}^{n-1} \vec{r}_{2,i} x^i.$$

Second, let $\vec{g} = [\vec{r} \times \vec{s}]_2 = [(\vec{r}_1 + \vec{r}_2) \times \vec{s}]_2 = [\vec{r}_1 \times \vec{s}]_2 + [\vec{r}_2 \times \vec{s}]_2$. Compute $\vec{g}_0 = \vec{g}_{1,0} + \vec{g}_{2,0}$ as follows, all others $\vec{g}_i$ are similar to $\vec{g}_0$. For $\vec{g}_{1,0}$, we have

$$\begin{aligned} \vec{g}_{1,0} &= \left[\vec{r}_{1,0}\vec{s}_0 - \vec{r}_{1,1}\vec{s}_{n-1} - \cdots - \vec{r}_{1,n-1}\vec{s}_1\right]_2 \\ &= \left[\vec{r}_{1,0}\right]_2 \vec{s}_0 + \left[-\vec{r}_{1,1}\right]_2 \vec{s}_{n-1} + \cdots + \left[-\vec{r}_{1,n-1}\right]_2 \vec{s}_1 \\ &= \left[\vec{r}_{1,0}\right]_2 \sum_{j=0}^{\log n} \vec{s}_{0,j} 2^j + \left[-\vec{r}_{1,1}\right]_2 \sum_{j=0}^{\log n} \vec{s}_{n-1,j} 2^j + \cdots + \left[-\vec{r}_{1,n-1}\right]_2 \sum_{j=0}^{\log n} \vec{s}_{1,j} 2^j \\ &= \left[\vec{r}_{1,0}\right]_2 \sum_{j=0}^{\log n} \vec{s}_{0,j} 2^j + \sum_{t=1}^{n-1} \left[-\vec{r}_{1,t}\right]_2 \sum_{j=0}^{\log n} \vec{s}_{n-t,j} 2^j \\ &= [c_0 \vec{v}_1]_2 \sum_{j=0}^{\log n} \vec{s}_{0,j} 2^j + \sum_{t=1}^{n-1} [-c_t \vec{v}_1]_2 \sum_{j=0}^{\log n} \vec{s}_{n-t,j} 2^j \\ &= \left[c_0 \sum_{j=n\log n}^{2n\log n+3} \vec{v}_{1,j} 2^{-j}\right]_2 \sum_{i=0}^{\log n} \vec{s}_{0,i} 2^i + \sum_{t=1}^{n-1} \left[-c_t \sum_{j=n\log n}^{2n\log n+3} \vec{v}_{1,j} 2^{-j}\right]_2 \sum_{i=0}^{\log n} \vec{s}_{n-t,i} 2^i \\ &= \sum_{j=n\log n}^{2n\log n+3} \vec{v}_{1,j} \left[c_0 2^{-j}\right]_2 \sum_{i=0}^{\log n} \vec{s}_{0,i} 2^i + \sum_{t=1}^{n-1} \sum_{j=n\log n}^{2n\log n+3} \vec{v}_{1,j} \left[-c_t 2^{-j}\right]_2 \sum_{i=0}^{\log n} \vec{s}_{n-t,i} 2^i \\ &= \sum_{j=n\log n}^{2n\log n+3} \sum_{i=0}^{\log n} \vec{s}_{0,i} \vec{v}_{1,j} \left[c_0 2^{-j+i}\right]_2 + \sum_{t=1}^{n-1} \sum_{j=n\log n}^{2n\log n+3} \sum_{i=0}^{\log n} \vec{s}_{n-t,i} \vec{v}_{1,j} \left[-c_t 2^{-j+i}\right]_2 \end{aligned}.$$

So, we get $n(n\log n + 4)$ rational numbers denoted by ciphertexts. According to $W(s) = \omega(\log n)$ and $w(s_i) \le 1$, there are only $\omega(\log^2 n)$ non-zero rational numbers among the $n(n\log n + 4)$ rational numbers of ciphertexts.

For $\vec{g}_{2,0}$, we have

$$\begin{aligned}\vec{g}_{2,0} &= \left[\vec{r}_{2,0}\vec{s}_0 - \vec{r}_{2,1}\vec{s}_{n-1} - \cdots - \vec{r}_{2,n-1}\vec{s}_1\right]_2 \\ &= \left[\vec{r}_{2,0}\right]_2 \vec{s}_0 + \left[-\vec{r}_{2,1}\right]_2 \vec{s}_{n-1} + \cdots + \left[-\vec{r}_{2,n-1}\right]_2 \vec{s}_1 \\ &= \left[\vec{r}_{2,0}\right]_2 \sum_{i=0}^{\log n} \vec{s}_{0,i} + \left[-\vec{r}_{2,1}\right]_2 \sum_{i=0}^{\log n} \vec{s}_{n-1,i} + \cdots + \left[-\vec{r}_{2,n-1}\right]_2 \sum_{i=0}^{\log n} \vec{s}_{1,i} \\ &= \sum_{i=0}^{\log n} \vec{s}_{0,i} \left[\vec{r}_{2,0}\right]_2 + \sum_{i=0}^{\log n} \vec{s}_{n-1,i} \left[-\vec{r}_{2,1}\right]_2 + \cdots + \sum_{i=0}^{\log n} \vec{s}_{1,i} \left[-\vec{r}_{2,n-1}\right]_2\end{aligned}.$$

That is, $\vec{g}_{2,0}$ consists of n rational numbers of ciphertexts with $\omega(\log n)$ non-zero numbers. So, we can evaluate $\vec{g}_0$ by using the technique of symmetric polynomial in [GH10, vDGHV10]. Thus, we can compute $\vec{g}$, $\vec{z} = \left[\lfloor \vec{g} + 0.5h \rceil\right]_2$, and $\vec{u} = \left[(\vec{z} \times \overline{[f]_2}) \bmod x\right]_2$, finally output $c_{new} = \vec{u} \oplus [c \bmod x]_2$.■

8.3 Improvement of FHE-2v

For the above FHE-2v, we know there are $\omega(\log^2 n)$ non-zero rational numbers among all ciphertext numbers. Although Recrypt algorithm can evaluate this sum, the degree of decryption algorithm polynomial is too big to make the above scheme be practical. So, to decrease the complexity of decryption algorithm and guarantee the security of our scheme, we induce the dimension of s to a constant k, but increase the size of its coefficients. Concrete Key Generating algorithm consists of as follows.

(1) Select $s = \sum_{i=0}^{k} s_i x^i$ with $\|s\|_\infty = 2^r$ such that $p = \det(Rot(s)) > 2^{kr}$ is an odd integer, where r is a function on n, $r = r(n) \geq n$.

(2) Compute a polynomial f over R subject to $(s \times f)/p = 1 \bmod (x^{k+1} + 1)$.

(3) Choose a random binary fraction $v_{1,i} = \sum_{j=rk}^{2rk+3} v_{1,i,j} 2^{-j}$ with $w(v_{1,i}) = \omega(\log n)$, and set $v_{2,i} = s_i / p - v_{1,i}$ with $2rk+3$ bits of precision after the binary point.

(4) Generate $b_i = \left[a_i \times f + 2e_i\right]_{b_0}$ with $\tau = O(n)$, $\|a_i\|_\infty \leq O(2^n)$ and $\|e_i\|_\infty \leq 2^{r_1}$, where $r_1 = r_1(n) \geq \omega(\log n)$. Assume $\|b_i\|_\infty \leq \|b_0\|_\infty$.

(5) Generate $\{d_i = a_i \times f + 2e_i\}_{i=0}^{t}$ with $t = O(rn)$, such that $\|d_i\|_\infty / \|b_0\|_\infty \leq n^{i+1}$ and $\|e_i\|_\infty \leq 2^{r_1}$.

(6) Encrypt the *j*-th bit of $v_{1,i} = \sum_{j=rk}^{2rk+3} v_{1,i,j} 2^{-j}$ as $\vec{v}_{1,i,j} = a_{i,j} \times f + 2e_{i,j} + v_{1,i,j}$ with $\|a_{i,j}\|_\infty = O(2^n)$, $\|e_{i,j}\|_\infty \le 2^{r_1}$ for $i \in [k]$ and $j \in [rk+3]$. Let $\vec{v}_{1,i} = \sum_{j=rk}^{2rk+3} \vec{v}_{1,i,j} 2^{-j}$, $\overrightarrow{s_i / p} = \vec{v}_{1,i} + v_{2,i}$, and $\overrightarrow{s / p} = \sum_{i=0}^{k} (\overrightarrow{s_i / p}) x^i$.

(7) Encrypt $[f_j]_2$ as $\vec{f}_j = a_j \times f + 2e_j + [f_j]_2$ with $\|a_j\|_\infty \le O(2^n)$ and $\|e_j\|_\infty \le 2^{r_1}$, denoted as $\overrightarrow{[f]}_2$.

(8) Output the secret key $sk^* = (p, s, [f]_2)$ and the public key $pk^* = (n, k, w, \{b_i\}_{i \in [\tau]}, \{d_i\}_{i \in [t]}, \overrightarrow{s / p}, \overrightarrow{[f]}_2)$.

It is easy to verify that there is at most $kw+1$ non-zero rational numbers among all ciphertext numbers. When k is a small constant, the circuit depth of decryption algorithm is dominated by $w(v_{1,i}) = \omega(\log n)$. So, we have obtained some improvement of performance.

Remark 8.1: Indeed, we can use general polynomial $s = \sum_{i=0}^{n-1} s_i x^i$ as secret key, choose at random a polynomial $v_1 = \sum_{i=0}^{n-1} v_{1,i} x^i$ with $h(v_{1,i}) = \omega(\log n)$ for each $v_{1,i}$, and set $v_2 = s / p - \sum_{i=0}^{n-1} v_{2,i} x^i$. However, we now need to take $\|s\|_\infty$ large enough to guarantee computing $\omega(n \log n)$ non-zero rational numbers when performing recrypting algorithm.

8.4 Extension to Large Message Space

In the FHE-2v, we can reduce the expansion factor of ciphertext to $O(n \log n)$ by expanding the plaintext message space. For a message $m \in \{0,1\}^n$, we map it into a polynomial $m(x) = \sum_{i=0}^{n-1} m_i x^i$. Now, the Enc algorithm is $c = \left[\sum_{i \in T} b_i + 2e + m(x)\right]_{b_0}$, the Dec algorithm $m(x) = \left[\left\lfloor (c \times s / p) \bmod (x^n+1) + 0.5h \right\rfloor \times [f]_2 \bmod (x^n+1)\right]_2 \oplus [c]_2$. The Recrypt algorithm is modified into $\vec{M}(x) = [c]_2 \oplus \left[u \times \overrightarrow{[f]}_2\right]_2$. Since $\vec{M}(x)$ consists of n ciphertexts, we require to transform $\vec{M}(x)$ into a new ciphertext $\vec{m}(x)$ as follows: $\vec{m}(x) = \left[\sum_{i=0}^{n-1} (\vec{M}(x))_i \times x^i\right]_{b_0}$.

8.5 Construction of Non-self-loop FHE-2v

According to [Gen09], the above FHE can not prove to be semantically secure by a standard hybrid argument when using self-loop. In fact, the FHE in [Gen09] also reveals the encrypted secret key bits, although it is not direct. Although we do not know any actual attack by using self-loop, we may contain a cycle of encrypted secret key to remove self-loop in our scheme. In this case, one can compute ciphertexts of sk_1 under pk_1, but there is bigger error noise in the ciphertexts of encrypted secret key than that in self-loop scheme. We now modify the self-loop FHE-2v into a non-self-loop FHE-2v as follows.

We generate two keys pk_j, sk_j, $j=1,2$, encrypt the secret key sk_1 under the public key pk_2, the secret key sk_2 under the public key pk_1, and output the public key $pk^* = \{pk_1^*, pk_2^*\}$. Assume we use the public key pk_1 to encrypt message bit. When we refresh a ciphertext by using $pk^* = \{pk_1^*, pk_2^*\}$, we first apply Recrypt of FHE-2 to transform the ciphertext under pk_1 into a ciphertext under pk_2, then again use Recrypt to transform the ciphertext under pk_2 into a new ciphertext under pk_1.

It is easy to see that there is a cycle of encrypted secret key in FHE. We can obtain an encrypted sk_i under pk_i by homomorphic operations. Moreover, an encrypted sk_i under pk_i in the non-self-loop scheme has bigger error noise than that in the self-loop scheme. However, the drawback of our non-self-loop scheme is to require calling Recrypt two times to refresh ciphertext.

9. Security Analysis

In this section, we present the hardness assumption of the security of our scheme, and give possible attack for our scheme.

If we take $n=1$, our scheme is identical to that of [vDGHV10]. On the other hand, our scheme is also an extension for that of [Gen09] by replacing the public basis with an approximate public basis, namely, if we take $b_0 = (a_0 \times f) \bmod (x^n + 1)$ as public key, then our scheme is similar to that of [Gen09], except for the ideal in their scheme may not be a principal ideal.

Theorem 9.1. Suppose there is an algorithm A which breaks the semantic security of our SHE with advantage ε. Then there is a distinguisher D which solves the decisional APIP with advantage at least $\varepsilon / 2$.

Proof. We construct a distinguishing algorithm D with advantage at least $\varepsilon/2$ between two distributions $H_{f,\varphi}$ and $g \leftarrow_U R$. The algorithm D receives as input c and $\alpha \leftarrow_U \{0,1\}$, sends the challenge ciphertext $[2c+\alpha]_{b_0}$ to A, then returns 1 if A guesses right α, and otherwise 0. It is easy to verify that D solves the decisional APIP with advantage at least $\varepsilon/2$.■

From Theorem 9.1, we can directly obtain the following result for the decisional BDDP.

Corollary 9.1. Suppose there is an algorithm A which breaks the semantic security of our SHE with advantage ε. Then there is a distinguisher D which solves DBDDP with advantage at least $\varepsilon/2$.

Since our scheme is to extend that of [vDGHV10] from one dimension to multiple dimensions. So, when n is small, we have the following theorem, whose proof is to adapt from that in [vDGHV10]. In this case, we must enlarge the error terms e to guarantee the security of scheme.

Theorem 9.2. Suppose there is an algorithm A which breaks the semantic security of our SHE with advantage ε. Then there is a distinguisher D which solves the APIP with advantage at least $\varepsilon/2^n$. In particular, then there is an algorithm B which solves AGCD with advantage at least $\varepsilon/2$.

Theorem 9.3. Suppose the decisional APIP is hard, then our SHE-2 is semantic security.

Part III FHE-3 Based on Approximate Lattice Problem

In this part, we will construct a new fully homomorphic encryption scheme based on approximate general lattice problem. We first give a public key encryption scheme based on approximate lattice problem, then provide homomorphic operations over this PKE, and finally present a new FHE-3.

10. Somewhat Homomorphic Encryption (SHE-3)

10.1 Public Key Encryption Scheme (PKE)

To generate our public key encryption scheme, we require this following lemma.

Lemma 10.1. (AP09, Theorem 3.1 and 3.2). There is a probabilistic polynomial-time algorithm that, on input a positive integer *n*, positive integer p, and a poly(*n*)-bounded positive integer $m \geq 8n\log p$, outputs a pair of matries $A \in \mathbb{Z}_p^{n\times m}$, $T \in \mathbb{Z}^{m\times m}$ such that A is statistically close to uniform over $\mathbb{Z}_p^{n\times m}$, $AT = 0 \bmod p$, and $\|T\| = O(n\log p)$.

PKE.KeyGen:

(1) Let n, m, p be integers related to security parameter λ, and p an odd integer. By using Lemma 10.1, one generates a pair of matries $A \in \mathbb{Z}_p^{n \times m}$, $T \in \mathbb{Z}^{m \times m}$ such that A is statistically close to uniform over $\mathbb{Z}_p^{n \times m}$, $AT = 0 \bmod p$, $\det(T)$ is an odd integer, and $\|T\| = O(n \log p)$ (resp. $\|T\| = O(1)$).

(2) Let χ be a distribution over $\mathbb{Z}^m$. Choose a list $\tau = O(\lambda)$ elements $b_i = s_i A + 2e_i$ over $\mathbb{Z}^m$ such that $s_i \leftarrow \mathbb{Z}^n$, $e_i \leftarrow \chi$ with $\|e_i\|_\infty \le \beta / 2$.

(3) Output the public key $pk = (m, b_i, i \in [\tau], \beta)$ and the secret key $sk = (T, p)$.

To reduce the size of the public key, one in general sets $\|s_i\|_\infty \le \lambda^{O(1)}$ in the PKE.KeyGen.

PKE.Enc. Given the public key pk and a message $x \in \mathbb{Z}_2^m$, choose a random subset $S \subseteq [\tau]$ and an independent 'small' error term $e \leftarrow \chi$ with $e \in \mathbb{Z}^m$ and $\|e\|_\infty \le \beta / 2$. Evaluate a ciphertext $c = \sum_{i \in S} b_i + 2e + x$.

PKE.Dec. Given the secret key sk, and the ciphertext c, decipher $x = \left[\left[[c \cdot T]_p\right]_2 ([T]_2)^{-1}\right]_2$.

Correctness: When $p > 2\|(x + \sum_{i \in S} 2e_i) \cdot T\|_\infty$, Dec works correctly because

$$\begin{aligned}
&\left[\left[[c \cdot T]_p\right]_2 ([T]_2)^{-1}\right]_2 \\
&= \left[\left[\left[(x + \textstyle\sum_{i \in S} s_i A + 2e_i) \cdot T\right]_p\right]_2 ([T]_2)^{-1}\right]_2 \\
&= \left[\left[\left[(x + \textstyle\sum_{i \in S} 2e_i) \cdot T\right]_p\right]_2 ([T]_2)^{-1}\right]_2 \\
&= \left[\left[(x + \textstyle\sum_{i \in S} 2e_i) \cdot T\right]_2 ([T]_2)^{-1}\right]_2 \\
&= \left[[x \cdot T]_2 ([T]_2)^{-1}\right]_2 \\
&= \left[[x]_2 \cdot [T]_2 ([T]_2)^{-1}\right]_2 \\
&= x
\end{aligned}.$$

Remark 10.1: We observe that the above PKE itself is very interesting because its public key consists of a list of approximate vectors of the closest vector problem in lattice, but does not provide lattice itself. The expansion rate of ciphertext is $O(\log p)$ in this PKE. It is not difficult to see that the security of PKE is harder than the decisional GapCVP with certain gap parameter.

Remark 10.2: For the above PKE, one can set the public key as $b_i = s_i A + e_i$ over $\mathbb{Z}^m$ such that $s_i \leftarrow \mathbb{Z}^n$, $e_i \leftarrow \chi$ with $\|e_i\|_\infty \le \beta$ and $b_0 = s_0 A + e_0 + \{\lfloor p/2 \rfloor, 0, \ldots, 0\}$. Assume t is first column of T and $t_0 = 1 \bmod 2$. When encrypting, if a message bit is '0', then $S \subseteq [\tau] \backslash 0$, otherwise $S' \subseteq [\tau] \backslash 0, S = S' \bigcup \{0\}$. When deciphering, one decides a message bit is '0' or '1' according to the value of $[<c,t>]_p$ to be the nearest to 0 or $p/2$.

10.2 Homomorphic Operations over Ciphertexts

To discuss simplicity, assume that $t \in \mathbb{Z}^m$ is some column of T such that its first term t_0 is an odd integer. Moreover, we merely use a message bit space $x \in \mathbb{Z}_2$ and set $\overline{x} = \{x, 0, \ldots, 0\}$. When encrypting, one outputs $c = \sum_{i \in S} b_i + 2e + \overline{x}$. When decrypting, one outputs $x = \left[[<c,t>]_p\right]_2$.

It is obvious that the above PKE supports addition operation over the ciphertexts. To perform multiplication operation, Brakerski and Vaikuntanathan [BV11] consider the multiplication operation over ciphertexts as the quadratic equation, that is, given the ciphertexts c_1, c_2 that encrypts x_1, x_2 and the secret key t: $Q_{c_1,c_2}(t) =< c_1, t > \bullet < c_2, t >$. If the noise of c_1, c_2 is small, then we can get $x_1 \bullet x_2$ by computing $\left[\left[Q_{c_1,c_2}(t)\right]_p\right]_2$. The problem is how to perform this function under ciphertexts. In [BV11], they use the tensor product $t \otimes t$ of t to implement dimension reduction (key switching). Here we use another approach. Since $< c_1, t > \bullet < c_2, t > =<< c_1, t > \bullet c_2, t > =< c_2 \sum_{i=0}^{m-1} c_{1,i} t_i, t >$, we only require generate a new ciphertext by evaluating $c_2 \sum_{i=0}^{m-1} c_{1,i} t_i = (\sum_{i=0}^{m-1} c_{2,0} c_{1,i} t_i, \ldots, \sum_{i=1}^{m} c_{2,m-1} c_{1,i} t_i)$. To compute this ciphertext, we adapt the subroutines BitDecomp and Powersof2 introduced by [BV11, Gen11] from $\mathbb{Z}_p$ to $\mathbb{Z}$. Now, we assume $\|c\|_\infty \le q$. In the following we will give an optimization algorithm to reduce the length of ciphertext.

Definition 10.1. (BitDecomp). Let $y \in \mathbb{Z}^m$ and $N = m \cdot \lceil 2 \log q \rceil$. We decompose y into its bit representation $y = \sum_{j \in [\lfloor 2 \log q \rfloor]} 2^j u_j$, where all of the vectors $u_j \in \{0, 1, -1\}^m$. Output

$(u_0, u_1, ..., u_{\lfloor 2\log q \rfloor}) \in \{0,1,-1\}^N$.

Definition 10.2. (Powersof2). Let $y \in \mathbb{Z}^m$ and $N = m \cdot \lceil 2\log q \rceil$. We define Powersof2(*y*) to be the vector $(y, 2 \cdot y, ..., 2^{\lfloor 2\log q \rfloor} \cdot y) \in \mathbb{Z}^N$.

Lemma 10.2. For vectors $c, t \in \mathbb{Z}^m$, we have $< BitDecomp(c), Powersof2(t) >=< c, t >$.

Now, we can evaluate homomorphic multiplication by adding encrypted Powersof2(*t*) to the public key.

11. Fully Homomorphic Encryption (FHE-3)

We can construct a new FHE-3 scheme based on ALP by applying two methods in Part II. To be simple, we merely provide the FHE-3 by using bootstrapping with the sparse subset sum problem. In addition, when generating the public key of FHE-3, we set $q = kp, k = \lambda^{O(1)}$ to control the size of the public key. Our FHE-3 constructs as follows:

FHE-3.KeyGen.

(1) Generate $pk = (m, b_i, i \in [\tau], \beta)$, $sk = (t)$ and A by using PKE.KeyGen in Section 10.1.

(2) Let $N = m \cdot \lceil 2\log q \rceil$. Choose a list elements $b_{i,j} = s_{i,j}A + 2e_{i,j}$ over $\mathbb{Z}_q^m$ such that $s_{i,j} \leftarrow \mathbb{Z}_q^n$, $e_{i,j} \leftarrow \chi$ with $\|e_{i,j}\|_\infty \le \beta / 2$, where $i \in [m-1], j \in [N-1]$.

(3) Let $B_i^{'}$, $i \in [m-1]$ be a matrix with row vectors $b_{i,j}$, $j \in [N-1]$. Evaluate $B_i = B_i^{'} + (Powersof2(t)_i) \bmod p$, where $Powersof2(t)_i$ is added to the *i*-th column of $B_i^{'}$.

(4) Choose $3m$ elements $d_{i,j} = s_{i,j}A + 2e_{i,j}$ over $\mathbb{Z}^m$ for $i \in [2], j \in [m-1]$ with $s_{i,j} \leftarrow \mathbb{Z}^n$, $e_{i,j} \leftarrow \chi$, $\|e_{i,j}\|_\infty \le \beta / 2$ and $\|d_{i,j}\|_\infty / q \approx m^i$. Let D_i, $i \in [2]$ be a matrix with row vectors $d_{i,j}$, $j \in [m-1]$. We require $\|(D_i)^{-1}\|_\infty \approx 1 / \|D_i\|_\infty$.

(5) Choose at random a set S_1 of δ_1 vectors $g_i \in Q^m$ with $\|g_i\|_\infty < 2$ such that there is a subset S_2 of δ_2 vectors with $\left\| \left[\sum_{i \in S_2} g_i \right]_2 - t / p \right\|_\infty < \frac{1}{p^2}$.

(6) Set $sk_i = 1$ for $i \in S_2$ and $sk_i = 0$ for $i \in S_1 - S_2$.

(7) Encrypt sk_i as $\overrightarrow{sk_i} = s_i A + 2e_i + \overline{sk_i}$ with $s_i \leftarrow \mathbb{Z}^n$, $e_i \leftarrow \chi$ and $\|e_i\|_\infty \le \beta / 2$.

(8) Encrypt the *i*-th bit of $[t]_2$ as $(\overrightarrow{[t]_2})_i == s_i A + 2e_i + \overline{([t]_2)_i}$ with $s_i \leftarrow \mathbb{Z}^n$, $e_i \leftarrow \chi$ and $\|e_i\|_\infty \le \beta / 2$, denoted as $\overrightarrow{[t]_2}$.

(9) Output the public key $pk = (m, \{b_i\}_{i=0}^{\tau}, \{B_i\}_{i=0}^{m-1}, \{D_i\}_{i=0}^{2}, \delta_1, \delta_2, \{\overrightarrow{sk_i}, g_i\}_{i \in S_1}, \overrightarrow{[t]_2})$, and the secret key $sk = (t, p)$.

FHE-3.Enc. Given pk and a message bit $x \in \mathbb{Z}_2$, set $\overline{x} = \{x, 0, ..., 0\}$, output $c = (\sum_{i \in S} b_i + 2e + \overline{x}) \bmod D_0$.

FHE-3.Dec. Given sk, and a ciphertext c, output $x = \left[[< c, t >]_p \right]_2$.

FHE-3.Add. Given pk and ciphertexts c_1, c_2, output $c = (c_1 + c_2) \bmod D_0$.

FHE-3.Mul. Given pk and ciphertexts c_1, c_2, output

$$c = (\sum_{i=0}^{m-1} BitDecomp(c_{2,i} c_1) \bullet B_i) \bmod D_2 \bmod D_1 \bmod D_0 .$$

Remark 10.2: To remove D_i in the above algorithms, we may permit to appropriately increase the length of ciphertext. Of course, we must increase the size of Powersof2(y).

FHE-3.Recrypt. Given pk and ciphertext c, compute as follows:

(1) Compute $r_i = < c, g_i >$, keeping only $\theta = \lceil \log \delta_2 \rceil + 3$ bits of precision after the binary point for each coefficient of r_i.

(2) Evaluate $u_i = r_i \times \overrightarrow{sk_i}$, $u = \left[\left\lfloor \sum_{i \in S_1} u_i + 0.5 \right\rfloor \right]_2$ by using the symmetric polynomials in [GH10].

(3) Output a new ciphertext $c_{new} = \left[< [c]_2, \overrightarrow{[t]_2} > \right]_2 \oplus u$.

Correctness: It is easy to verify that the FHE-3.Add and FHE-3.Mul works correctly for appropriate parameters setting.

Now, we estimate the noise bound of the ciphertext after one homomorphic multiplication. Given two ciphertexts c_1, c_2, we have

$$[<c_1,t>\bullet<c_2,t>]_p=\left[<[<c_1,t>]_p\bullet c_2,t>\right]_p=[<<2e_1+\overline{x},t>\bullet c_2,t>]_p.$$

According to FHE-3.Enc, $\|<2e_1+\overline{x},t>\bullet 2e_2\|\le m\beta^2\|t\|$. On the other hand, to compute $<2e_1+\overline{x},t>\bullet c_2$, one requires to sum $2m^2\log q$ ciphertexts, and this increases the noise of ciphertext at most $2m^2\beta\log q$. At the same time, to reduce the size of ciphertext by using modulo D_i each time increases the noise of ciphertext at most $m^3\beta$. So, the noise bound of the ciphertext $c=c_1\times c_2$ is at most $m^2\beta\log p+m\beta^2\|t\|+3m^3\beta\approx O(m^3\beta)$.

Theorem 10.1. When $m^{O(\delta_2)}<p$, the FHE-3.Recrypt correctly generates a ‘fresh’ ciphertext c_{new} with the same message of c and smaller error term, and two homomorphic-decrypted ciphertexts support one multiplication.

Proof: This proof is similar as that of theorem 7.1. □

12. Security Analysis

To give the security of the above scheme, we first define a promise problem and a variant about the closest vector problem in lattice.

Definition 12.1 ($GapCVP_\gamma$)**.** Given $B\in\mathbb{Z}^{n\times m}$, $x\in\mathbb{Z}^m$ and $r\in\mathbb{Q}_+$, the promise problem is to decide the following two cases: In YES inputs, we have $dist(x,L(B))\le r$, whereas in NO inputs, we have $dist(x,L(B))>\gamma\bullet r$.

Definition 12.2 (CVP_r)**.** Given $B\in\mathbb{Z}^{n\times m}$, $x\in\mathbb{Z}^m$ and $r\in\mathbb{Q}_+$, the problem is to decide whether there is a vector $y\in\mathbb{Z}^n$ such that $\|x-yB\|\le r$.

Theorem 12.1. Suppose there is an algorithm A which breaks the semantic security of our PKE with advantage ε. Then there is a decisional algorithm D for $CVP_{p/4mn\log p}$ with running in about the same time A and advantage at least $\varepsilon/2$.

Proof. We construct a decisional algorithm D with advantage at least $\varepsilon/2$ for $CVP_{p/2n\log p}$. The algorithm D receives as input $x\in\mathbb{Z}^m$. D generates the public key as PKE.KeyGen in Section 10.1, then sends the challenge ciphertext $(2x+\alpha)\bmod B$ to A, then returns 1 if A guesses the right α, and otherwise 0. If there is a vector $y\in\mathbb{Z}^n$

such that $\min_{y\in\mathbb{Z}^n}\|x-yB\|_\infty \le p/4n\log p$, then $\|(x-yB)T\| \le (p/4mn\log p)\bullet 2mn\log p$, namely $\|(x-yB)T\| \le p/2$. In this case, A works correctly with advantage ε. Otherwise, A does not have any advantage. ■

Theorem 12.2. Suppose the decisional ALP is hard, then our SHE-2 is semantic security.

13. Further Direction

We have presented a new fully homomorphic encryption scheme based on APIP (resp. ALP), whose security depends upon the hardness assumption of APIP (resp. ALP).

If the decisional APIP is hard, then our scheme is semantic security. In [vDGHV10], they reduce the security of scheme to solving approximate GCD problem. But we do not obtain similar result for our scheme since we can not adapt their reduction proof. An interesting open problem is whether or not there is a reduction from the semantic security of our scheme to solving APIP (resp. ALP)? Our public key has form $sA+2e$, in the following we will establish the relationship between the GapCVP problem and our PKE to support the security of our scheme to the worst-case hardness of some lattice problems.

References

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In Proc. of STOC 1996, pages 99-108, 1996.

[ACG08] C. Aguilar Melchor, G. Castagnos, and G. Gaborit. Lattice-based homomorphic encryption of vector spaces. In IEEE International Symposium on Information Theory, ISIT'2008, pages 1858-1862, 2008.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. Lecture Notes in Computer Science, 2005, Volume 3378, pages 325-341, 2005.

[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In CRYPTO, 2011. To appear.

[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. ePrint Archive: Report 2011/344: http://eprint.iacr.org/2011/344.

[vDGHV10] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In Proc. of Eurocrypt, volume 6110 of LNCS, pages 24-43. Springer, 2010.

[Gen01] C. Gentry. Key Recovery and Message Attacks on NTRU-Composite. Eurocrypt'01, LNCS 2045, pages 182-194.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In Proc. of STOC, pages 169-178, 2009.

[Gen11] C. Gentry. Fully Homomorphic Encryption without Bootstrapping. ePrint Archive: Report 2011/279: http://eprint.iacr.org/2011/277.

[GH10] C. Gentry and S. Halevi. Implementing Gentry's Fully-Homomorphic Encryption

Scheme. Cryptology ePrint Archive: Report 2010/520: http://eprint.iacr.org/2010/520.

[GHV10] C. Gentry and S. Halevi and V. Vaikuntanathan. A Simple BGN-type Cryptosystem from LWE. In Proc. of Eurocrypt, volume 6110, pages 506-522, 2010.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Proc. of STOC, pages 197-206, 2008.

[GS02] C. Gentry, M. Szydlo. Cryptanalysis of the Revised NTRU Signature Scheme. Eurocrypt'02, LNCS 2332, pages 299-320.

[HPS98] J. Hoffstein, J. Pipher, J. H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. LNCS 1423, pages 267-288, 1998.

[LPR10] V. Lyubashevsky and C. Peikert and O. Regev. On Ideal Lattices and Learning with Errors over Rings. In Proc. of Eurocrypt, volume 6110, pages 1–23, 2010.

[Mic07] D. Micciancio Generalized compact knapsaks, cyclic lattices, and efficient one-way functions. Computational Complexity, 16(4):365-411.

[MR07] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussion measures. SIAM Journal Computing, 37(1):267-302, 2007.

[Reg09] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, Journal of the ACM (JACM), v.56 n.6, pages1-40, 2009.

[SS10] D. Stehle and R. Steinfeld. Faster Fully Homomorphic Encryption. Cryptology ePrint Archive: Report 2010/299: http://eprint.iacr.org/2010/299.

[SV10] N. P. Smart and F. Vercauteren Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. Lecture Notes in Computer Science, 2010, Volume 6056/2010, 420-443.

[SYY99] T. Sander, A. Young, and M. Yung. Non-interactive CryptoComputing for NC1. In 40th Annual Symposium on Foundations of Computer Science, pages 554{567. IEEE, 1999.

[RAD78] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In Foundations of Secure Computation, pages 169-180, 1978.

[Yao82] A. C. Yao. Protocols for secure computations (extended abstract). In 23rd Annual Symposium on Foundations of Computer Science (FOCS '82), pages 160-164. IEEE, 1982.