Multi-bit homomorphic encryption based on learning with errors over rings

Zhang Wei, Liu Shuguang, Yang Xiaoyuan
Key Lab of Computer Network and Information Security under CAPF, Shaanxi 710086, China
zhaangweei@yeah.net

Abstract: Basing on Learning with errors over rings (RLWE) assumption, we provide a new multi-bit somewhat homomorphic encryption scheme. We introduce canonical embedding to transform a ring element into a vector, such that polynomial multiplication can be performed in $\widetilde{O}(n\log n)$ scalar operations, and ciphertext size is reduced at the same time. The CPA security of this scheme can be reduced into RLWE assumption.

Key words: homomorphic encryption; RLWE assumption; canonical embedding

1. Introduction

The idea of homomorphic encryption can be traced back to 1978 by Ridest et al.[1], it means that an entity can carry out computations on encrypted data without decryption. This trait of encryption scheme sounds appealing in network services. People need to conduct various kinds of operations, such as search, sum up or computing the average value, on data stored in a remote server. Suppose every operation requires a series of works including downloading the ciphertext, decryption, compute the target value, encryption and upload the new ciphertext, the communication and computation cost may grow very large and become unbearable. Homomorphic encryption permits operations on ciphertext directly, and thus reduce communication and computation cost. The most prominent application of homomorphic encryption is the outsourcing of data and computation on clouds. Besides these, there are some other interesting applications including private information retrieval (PIR), electronic voting, database encryption delegate computation and secure multiparty computation.

If an encryption scheme can compute any function of the ciphertexts, then it is called a *Fully Homomorphic Encryption* (FHE) scheme. Otherwise, if it can only evaluate a limited set of circuits about ciphertexts, then it is called a *Somewhat Homomorphic Encryption* (SHE) scheme. The existence and construction of FHE schemes remains an open problem in cryptography. The substantial progress was achieved by Gentry in STOC'2009[2]. Basing on hard problems on ideal lattices, Gentry proposed the first FHE scheme. Following this work, there have appeared some improvements with higher efficiency and better performance. In 2010, Smart and Vercauteren improved Gentry's scheme by shorten the key size and ciphertext size[3]. In Asiacrypt'2010, Stehle and Steinfeld also proposed an improved scheme of Gentry's scheme[4], which introduced decryption errors to reduce computation cost.

In the past years, people also proposed new schemes, some of which are quite potential in improving the performance and thus suitable for applications. In Eurocrypt 2010, Dijk, Gentry and Halevi et al. [5] promoted another construction of fully homomorphic encryption scheme, called DGHV. Using a Somewhat homomorphic encryption scheme on integers other than on ideal lattice, DGHV scheme was more succinct than Gentry's scheme. In 2011, Brakerski et al. [6] proposed two schemes that are based on Learning with Errors problem over Rings (RLWE). They also presented new techniques called re-linearization and dimension-modulus reduction to control noise and the length of encrypted data.

In the year 2011, Bogdanov et.al.[7] proposed an algorithm based on codes, this work has both a clear concept and a concise technique to bootstrapping without squashing the decryption algorithm. More recent studies by Rothblum[8] and Goldwasser[9] deals with changing a private key homomorphic encryption scheme into a public key one.

However, all of the algorithms are far away from practical uses. There are still a lot of works to do in this area. In order to promote performance and make homomorphic encryption practical, some researchers try a different way from the existing works. Other than designing a perfect scheme that is ideal in theory but not practical, they prefer to construct encryption schemes that can only evaluate low-degree polynomials about the ciphertexts, namely, Somewhat homomorphic encryption schemes. The predominance of SHE lies in a higher performance and efficiency. However, most of the existing schemes, including Gentry's SHE scheme, are designed on message space {0,1}, this means that the scheme can only encrypt one bit in each encryption. In 2011, Gentry et al. [10] presented a practical SHE scheme based on the GPV scheme which was constructed by Gentry, Peikert and Vaikuntanathan in 2008[11]. The new scheme was based on RLWE assumption, it allows any times of addition and one multiplication, and can also operate in a larger message space. But it has a deficiency that can only allow one multiplication.

Although any feasible computation can be expressed as a Boolean circuit in theory, we still need to design homomorphic encryption schemes for larger message spaces for practical uses, and at the same time, allowing more multiplication so as to evaluate a larger set of circuits. Following the idea of SHE and aiming at encryption on a larger message space, we present a multi-bit SHE scheme that is basing on RLWE (Learning With Error over Rings) problem. Compared to existing schemes, our scheme has the following advantages:

(1) Most of the existing works can only encrypt one bit in an encryption operation, while our scheme can encrypt multi-bits, this is suitable for a large message space.

(2) The scheme is constructed basing on RLWE assumption, RLWE assumption is tighter than standard LWE assumption, so schemes basing on RLWE has a higher performance with same security requirements. Among the existing schems, BV11 was constructed on RLWE, but it can only encrypt one bit. Moreover, we use canonical embedding to reduce key size and computation cost, thus can achieve a more time efficient scheme and also small key length.

(3) The scheme allow any times of addition and more than one times of multiplication.

2. Preliminaries

2.1 Homomorphic encryption schemes

Definition 1 A Homomorphic Encryption scheme (HE) can be described as a 4-tuple of algorithms HE=(KeyGen, Enc, Dec, Eval). The algorithms are probabilistic polynomial time and satisfy the following properties:

KeyGen(1^λ)：input a security parameter λ, output (*pk*, *sk*, *evk*), where *pk* and *sk* are the public encryption key and private decryption key, and *evk* is the public homomorphic evaluation key.

Enc(*pk*, *m*): input the encryption key *pk* and a message *m*, the encryption algorithm outputs a ciphertext *c*, denoted as *c*=Enc(*pk*, *m*).

Dec(*sk*, *c*): input a ciphertext *c* and decryption key *sk*, output a plaintext *m*.

Eval(*evk*, *f*, c_1, c_2…, c_l): input the homomorphic evaluation key *hk*, a function *f* and *l* ciphertexts c_1, c_2…, c_l, output a ciphertext c_f, satisfing

$$c_f = \mathrm{Enc}(pk, f(\mathrm{Dec}(sk, c_1), \mathrm{Dec}(sk, c_2), \ldots\ldots, \mathrm{Dec}(sk, c_l)))$$

The above definition is a generic description of homomorphic encryption schemes, and the material of function *f* is omitted. Usually *f* can be expressed as a Boolean circuit on field GF(2^n), and only contains ADD and OR operations. This means that *f* is made up of addition and multiplication of ciphertexts.

We say an encryption scheme is strongly homomorphic if a homomorphicly evaluated ciphertext *c** is indistinguishable with a normal ciphertext *c* which is output by the Enc algorithm. We say an encryption scheme is weakly homomorphic if the length of *c** only depends on the depth of the circuit to be evaluated.

2.2 LWE and RLWE assumptions

As a widely used tool for constructing cryptographic schemes on lattices, the *Learning With Errors* (LWE) problem has gained a universal notice since it is being introduced by Regev in 2005[12]. LWE assumption is defined as the following:

Definition 2 (Decisional LWE assumption) Let *n* be security parameter, *q*=poly(*n*) is a prime, and $s \in Z_q^n$ is a secret vector. Then any linear combination of elements of *s* are computational indistinguishable with a uniformly random element in Z_q, namely

$$\left\{a_i, b_i = \langle a_i, s\rangle + e_i\right\}_{i=1}^{poly(n)} \overset{C}{\approx} \left\{a_i, u_i\right\}_{i=1}^{poly(n)}$$

where $a_i \in Z_q^n$, e_i are sampled from some error distribution, $\sqrt{n} \le e_i << q$. A typical error distribution is the discrete Gauss distribution on Z_q with expectation being 0 and standard deviation being $\alpha q > 2\sqrt{n}$.

The search version of LWE is to find *s* from several given pairs (a_i, b_i).

In 2005, Regev has proved that LWE problem is at least as hard as the shortest vector problem in any lattice. Since then, LWE assumption has been used to construct public key encryption schemes, identity based encryption schemes, oblivious transfer protocol, key dependent message security encryption schemes and homomorphic encryption schemes[13].

In Eurocrypt 2010, Lyubashevsky, Peikert and Regev[14] discussed the efficiency of LWE assumptions. For a standard LWE assumption, getting one pseudorandom scalar $b_i \in Z_q$ requires an *n*-dim inner production computation. They propose a more compact version of LWE called RLWE assumption, that is, *LWE assumptions on a given ring*, where conducting an *n*-dim inner production can get another *n*-dim vector. Thus make an efficiency improvement by *n* times.

Definition 3 (*RLWE* assumption) Let *f*(*x*) be an *n*-degree polynomial with integer coefficients, *q* is a prime, and define a ring R_q as $R_q = Z_q[x]/\langle f(x)\rangle$. Let χ be an error distribution on R_q, $s \overset{\$}{\leftarrow} R_q$, $a_i \overset{\$}{\leftarrow} R_q$, *k*=poly(*n*). For any given *k* pairs $(a_i, b_i = a_i s + e_i)_{i=1}^k$, where e_i abide distribution χ, then b_i is computational indistinguishable with a random uniformly chosen element from R_q.

Lyubashevsky, Peikert and Regev[14] have proven that, the *Shortest Independent Vector Problem* (SIVP) or *Shortest Vector Problem*(SVP) in the worst case on ideal lattices can be reduced into RLWE. Their main result is described in the following lemma 1.

Lemma 1 Let K be the *m*th cyclotomic number field having dimension $n=\varphi(\mathrm{m})$ and $R = \mathrm{O}_K$ be its ring of integers. Let $\alpha = \alpha(n) > 0$, and let $q = q(n) \geq 2$, q=1 mod m be a poly(n)-bounded prime such that $\alpha q \geq \omega\left(\sqrt{\log n}\right)$. Then there is a polynomial-time quantum reduction from $\tilde{\mathrm{O}}\left(\sqrt{n}/\alpha\right)$-approximate SIVP (or SVP) to $\mathrm{R-DLWE}_{q,\gamma_\alpha}$. Alternatively, for any $l \geq 1$, we can replace the target problem by the problem of solving $\mathrm{R-DLWE}_{q,D_\xi}$ given only *l* samples, where $\xi = \alpha \cdot (nl/\log(nl))^{1/4}$ is the standard deviation of Gauss distribution D_ξ.

From lemma 1 we can immediately get to a conclusion: with error distribution be D_ξ and $\xi = \alpha \cdot (nl/\log(nl))^{1/4}$, given *l* samples, the RLWE problem is at least as hard as SIVP problem in a lattice.

In the above conclusion, *f*(*x*) is the *m*th cyclotomic polynomial $\Phi_m(x)$ with *m*=2*n*. While if let *f*(*x*)=x^n+1 then we can make the a slower norm increase when conducting multiplication of ring elements. On account of a more clear description, we only use RLWE assumption on a special polynomial $R = Z_q[x]/\langle x^n + 1 \rangle$ where *n* is a power of 2 and *q*=1 mod 2*n*.

2.3 Canonical embedding in polynomial rings

Let $n=2^k$, *q*=1 mod 2*n* is a prime, there are two ways to map a polynomial in R_q into a Ring vector: coefficient embedding and canonical embedding.

Coefficient embedding is a "naive" method, let $a(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$, then coefficient embedding can be simply defined as:

$$a(x) \mapsto (a_0, \cdots, a_{n-1}) \in Z_q^n$$

This is a mapping from ring element to element in an ideal lattice. The demerit of coefficient embedding is quite clear: add operation can be conducted coefficient-wise, while multiplication is miscellaneous.

Canonical embedding was first proposed by Minkowski[14]. Let $\omega = \exp(\pi i/n)$, then we can define canonical map as

$$a(x) \mapsto \left(a(\omega^1), a(\omega^3), \cdots, a(\omega^{2n-1})\right) \in C^n$$

Here C is the complex number field. When a polynomial is mapped into a vector in C^n, both add and multiplication can be conducted coordinate-wisely, thus make computation more convenient. Especially when q is a prime and q=1 mod 2n, ω^{2i-1}, i=1,…,n-1 are just the n roots of x^n+1 in Z_q, so a polynomial $a(x) \in Z_q[x]/\langle x^n + 1\rangle$ can be mapped into an elements in Z_q^n or a n-dim vector on Z_q.

Example1: Let n=4, q=17, x^4+1 has 4 roots in Z_{17}: $\omega, \omega^3, \omega^5, \omega^7$. In fact, we can let ω=2, then the 4 roots are: 2, 8, 15 and 9.

Let $a(x)=x^3+3x^2+1$, $b(x)=2x^3+6x+9$ be polynomials chosen from $R = Z_{17}[x]/\langle x^4 + 1\rangle$, through canonical embedding, they can be mapped into two factors:

$$\sigma(a(x)) = (a(\omega), a(\omega^3), a(\omega^5), a(\omega^7)) = (4,8,5,4)$$

$$\sigma(b(x)) = (b(\omega), b(\omega^3), b(\omega^5), b(\omega^7)) = (3,10,15,8)$$

Next, we compute add and multiply in R and Z_q^n respectively.

Computations in R are the generic polynomial operation, that is

$$a(x)+ b(x)=3x^3+3x^2+6x+10$$

$$a(x)* b(x)=2x^6+6x^5+6x^4+12x^3+10x^2+6x+9=12x^3+8x^2+3$$

Computing in Z_q^n are coordinate-wise add and multiply, that is

$$\sigma(a(x)) + \sigma(b(x)) = (7,1,3,12)$$

$$\sigma(a(x)) \cdot \sigma(b(x)) = (12,12,7,15)$$

It can be easily validated that σ is a homomorphic mapping from R to Z_q^n, namely satisfying:

$$\sigma(a(x) + b(x)) = \sigma(a(x)) + \sigma(b(x))$$

And

$$\sigma(a(x) * b(x)) = \sigma(a(x)) \cdot \sigma(b(x))$$

For a given $\sigma(a(x)) = (a(\omega^1), a(\omega^3), \cdots, a(\omega^{2n-1})) \in Z_q^n$, we can get to its preimage $a(x)$ through solving a linear equation set of n variables.

3 Multi-bit homomorphic encryption schemes based on the original Regev scheme

3.1 The basic scheme

The first single-bit public key encryption scheme basing on LWE assumption was proposed by Regev in 2005[12], and basing on this scheme, people have promoted some other constructions and applications. The multi-bit version of Regev's scheme is denoted as the following scheme1.

Scheme 1 (Regev's original LWE based multi-bit encryption scheme)

Parameters: Suppose n is an integer and q a prime, satisfying $q \in (n^2, 2n^2) \geq 2$, $k = (1+\varepsilon)(1+n)\log q$, here ε>0 is a constant, the error distribution is discrete Gauss distribution, noted by $\chi = \overline{\Psi}_{\alpha q}$, and $\alpha = O(1/(\sqrt{n}\log n))$. Define a set $D_r = \left(Z \bigcap \left\{ -\left\lfloor \frac{r}{2} \right\rfloor, \cdots, \left\lceil \frac{r}{2} \right\rceil \right\} \right)$, $r \geq 1$. Let the plaintext length be *l* bits.

- Private key: $S \overset{\$}{\leftarrow} Z_q^{n \times l}$, the private key S is a $n \times l$ matrix on Z_q, |*S*|=*nl*log*q*.

- Public key: $A \overset{\$}{\leftarrow} Z_q^{n \times k}$, $P = A^t S + E \in Z_q^{k \times l}$, where *E* is error matrix, and $E \leftarrow \chi_\alpha^{k \times l}$, in practical use, to reduce key length, we can make all users share the same *A*, and transform *B* into its Hermit standard form, thus the length of public key is $(k-n) l \log_2 q$ bits.

- Encryption: A message $m \in Z_2^l$ is encrypted into a pair (c_0, c_1), where

$$c_0 = Aa \in Z_q^n,\ c_1 = P^t a + \frac{q-1}{2} m$$

and $a \overset{\$}{\leftarrow} D_r^k$. The ciphertext has a length of $(n+l)\log_2 q$ bits.

- Decryption: $c_1 - S^t c_0 \approx \frac{q-1}{2} m$.

The above scheme can be implemented in a polynomial ring, which is depicted in[15] by Rückert and Schneider as the following scheme 2.

Scheme 2 (Polynomial ring implementation of scheme 1)

Parameters: let q be a prime, $q \equiv 1 \bmod 2n$, $R = Z_q[x] / \langle x^n + 1 \rangle$, χ is discrete Gauss distribution. A sample that abides to χ is noted by $e(x) \in R$ with *r*≥1. Define a set D_r as

$$D_r = \left(Z \bigcap \left\{ -\left\lfloor \frac{r}{2} \right\rfloor, \cdots, \left\lceil \frac{r}{2} \right\rceil \right\} \right) / \langle x^n + 1 \rangle$$

For a natural number *k*, define two operations on R^k:

(1) Multiplication of two polynomial vectors: For any $\hat{x}, \hat{y} \in R^k$, $\otimes : R^k \times R^k \to R$,

$$\hat{x} \otimes \hat{y} = \sum_{i=1}^{k} x_i y_i$$

(2) Multiplication of one polynomial vector and one polynomial: for any $\hat{x} \in R^k$, $y \in R$,

$$\hat{x} y = (x_1 y, \cdots, x_k y) \in R^k$$

- Private key: Randomly choose $s \overset{\$}{\leftarrow} R$, the length of s is $n\log_2 q$ bits

- Public key: Randomly choose a k-dim vector $\hat{a} \overset{\$}{\leftarrow} R^k$, choose error vector $\hat{e} \leftarrow \chi_{R,\alpha}^k$, $\chi_{R,\alpha}^k$ is a k-dim discrete Gauss distribution with its value come from R^k and with 0 as the expectation and $\alpha \le 1/t\left(\sqrt{nk}\left\lceil \frac{r}{2} \right\rceil + 1\right)$ as the standard deviation. Computing a vector $\hat{b} = \hat{a}s + \hat{e} \in R^k$, and the public key is $\left(\hat{a}, \hat{b}\right)$. To decrease key length, we can also let all of the users share the same $\hat{a}$, and the length of public key is $kn\log_2 q$ bits.

- Plaintext: $m \in D_1 = Z_2[x]/\langle x^n + 1 \rangle$, the length of plaintext is n bits.

- Encryption: Randomly choose $\hat{r} \overset{\$}{\leftarrow} D_r^k$, compute a pair (c_0, c_1) as the ciphertext, here $c_0 = \hat{a} \otimes \hat{r} \in R$ and $c_1 = \hat{b} \otimes \hat{r} + m\frac{q-1}{2} \in R$, the length of (c_0, c_1) is $2n\log_2 q$ bits.

- Decryption: Compute $c_1 - c_0 s = m\frac{q-1}{2} + \hat{e} \otimes \hat{r} \approx m\frac{q-1}{2}$.

The correctness of scheme2 is shown in [15], the authors also have pointed out that when $\alpha \le 1/30\sqrt{nk}\left\lceil \frac{r}{2} \right\rceil$, the scheme can decrypt correctly. We give a brief discussion in the following.

To decrypt correctly, it is required that $\|\hat{e} \otimes \hat{r}\|_\infty \le \frac{q}{4}$, here $\hat{e}$ is the initial error, abides to $\chi_{R,\alpha}^k$, $\hat{r} \overset{\$}{\leftarrow} D_r^k$ and obviously $|r_i| \le \frac{r}{2}$. On account of Chebyshev's law, for n independent samples that abiding the same Gauss distribution $X_i \leftarrow N(\mu, \sigma^2)$, $1 \le i \le n$, the summation $\sum_{i=1}^{n} X_i \leftarrow N(n\mu, n\sigma^2)$. So every coefficient of $\hat{e} \otimes \hat{r}$ abides the same Gauss distribution with 0 as the expectation and $\sqrt{\sum_{i=1}^{k}\left(\frac{r\sqrt{n}\alpha}{2}\right)^2} = \frac{\sqrt{nk}r\alpha}{2} \le \sqrt{nk}\left\lceil \frac{r}{2} \right\rceil \alpha \le \frac{1}{t}$ as the standard deviation. Utilizing the truncated inequality of Gauss distribution, the probability that all of the coefficients of $\hat{e} \otimes \hat{r}$ is greater than q /4 is $\frac{4}{t}\sqrt{\frac{2}{\pi}}e^{-\frac{t^2}{32}}$. When $t \ge 30$, this value can be neglected. So $\|\hat{e} \otimes \hat{r}\|_\infty \le \frac{q}{4}$ will sure to happen, and thus scheme 2 can decrypt correctly.

3.2 A somewhat homomorphic encryption scheme basing on scheme 2

We propose a somewhat homomorphic encryption scheme that is constructed basing on scheme 2.

3.2.1 Add operation

Given two ciphertext pairs $C=(c_0,c_1)$ and $C'=\left(c_0^{'},c_1^{'}\right)$, the add operation is simple and quite directly.

$$\begin{aligned}C_{add}(C,C') &= C+C' \\ &= \left(c_0+c_0^{'},c_1+c_1^{'}\right) \\ &= \left(\hat{a}\otimes\hat{r}_1+\hat{a}\otimes\hat{r}_2,\hat{b}\otimes\hat{r}_1+m_1\frac{q-1}{2}+\hat{b}\otimes\hat{r}_2+m_2\frac{q-1}{2}\right) \\ &= \left(\hat{a}\otimes(\hat{r}_1+\hat{r}_2),\hat{b}\otimes(\hat{r}_1+\hat{r}_2)+(m_1+m_2)\frac{q-1}{2}\right) \\ &= (c_{add,0},c_{add,1})\end{aligned}$$

The decryption process includes computing $c_{add,1}-c_{add,0}s\approx(m_1+m_2)\frac{q-1}{2}$, and get the plaintext of the sum of two plaintexts. This adding operation does not increase the length of ciphertext and the amount of coordinates. Here the error may have a slight increase after adding, but it has little impact on decryption.

3.2.2 Multiply operation

Firstly, we multiply two initial ciphertexts.

Taking into account the decryption process:

$$c_1-c_0s=m\frac{q-1}{2}+\hat{e}\otimes\hat{r}$$

$$c_1^{'}-c_0^{'}s=m'\frac{q-1}{2}+\hat{e}\otimes\hat{r}'$$

From the above formula, we can get

$$\begin{aligned}\left(m\frac{q-1}{2}+\hat{e}\otimes\hat{r}\right)\cdot\left(m'\frac{q-1}{2}+\hat{e}\otimes\hat{r}'\right) &= (c_1-c_0s)\cdot\left(c_1^{'}-c_0^{'}s\right) \\ &= c_1c_1^{'}-\left(c_1c_0^{'}+c_0c_1^{'}\right)s+c_0c_0^{'}s^2\end{aligned} \quad (3\text{-}1)$$

Let $C_{mult}(C,C')=(c_{mult,0},c_{mult,1},c_{mult,2})$, where

$$c_{mult,0}=c_1c_1^{'}$$

$$c_{mult,1}=-c_1c_0^{'}-c_1^{'}c_0$$

$$c_{mult,2}=c_0c_0^{'}$$

So after one multiplication, the ciphertext is changed into a three-tuple $C_{mult}=(c_{mult,0},c_{mult,1},c_{mult,2})$. To decrypt this new ciphertext, we need to compute:

$$
\begin{aligned}
& c_{mult,0} + c_{mult,1}s + c_{mult,2}s^2 \\
& = \left(m\frac{q-1}{2} + \hat{e} \otimes \hat{r} \right) \cdot \left(m'\frac{q-1}{2} + \hat{e} \otimes \hat{r}' \right) \\
& = mm'\frac{(q-1)^2}{4} + m\frac{q-1}{2}\hat{e} \otimes \hat{r}' + m'\frac{q-1}{2}\hat{e} \otimes \hat{r} + (\hat{e} \otimes \hat{r}) \cdot (\hat{e} \otimes \hat{r}') \\
& = mm'\frac{(q-1)^2}{4} + \Delta \\
& = M
\end{aligned}
$$

Next we discuss the value of *m* and *m'*.

(1) When *m*=0 and *m'*=0, $M = (\hat{e} \otimes \hat{r}) \cdot (\hat{e} \otimes \hat{r}')$ is the product of two polynomials in R, and each coefficient of *M* is less than $\frac{q}{4} \cdot \frac{q}{4} = \frac{q^2}{16}$;

(2) When *m*=1and *m'*=0 (or *m*=0, *m'*=1), $M = \frac{q-1}{2}(\hat{e} \otimes \hat{r}') + (\hat{e} \otimes \hat{r}) \cdot (\hat{e} \otimes \hat{r}')$, each coefficient of *M* is less than $\frac{q-1}{2} \cdot \frac{q}{4} + \frac{q^2}{16} = \frac{3q^2 - q}{16}$;

(3) When *m*=1and *m'*=1, each coefficient of *M* is greater than $\frac{(q-1)^2}{4}$, but is less than

$$\frac{(q-1)^2}{4} + \left(\frac{q}{4}\right)^2 + \frac{q(q-1)}{4} = \frac{9q^2 - 12q + 4}{16}.$$

According to the above discussion, in order to decrypt correctly, we should take the coefficients of *M* into consideration, when the value of a certain coefficient is within $\left[\frac{(q-1)^2}{4}, \frac{9q^2 - 12q + 4}{16}\right]$, then the decryption result will be 1, else it will be 0. It should be noted that the last operation of decryption is done in $R' = Z_{q^2} / \langle x^n + 1 \rangle$, but not R, thus doubles the length of ciphertext, that is $4n \log_2 q$.

Through the above method, after one multiplication, the amount of ciphertext elements will increase by 1. So given two generic ciphertexts: $C = (c_0, c_1, \cdots, c_{l-1})$ and $C' = \left(c_0', c_1', \cdots, c_{t-1}'\right)$, without generality, we can let $l \geq t$. When doing addition, can pad C' with *l* – *t* zeros, namely, let $C' = \left(c_0', c_1', \cdots, c_{t-1}', c_t', \cdots, c_{l-1}'\right)$. When doing multiplication, use the method similar to formula（3-1）, we could get to a polynomial about s with a degree of *l* + *t*-2. In decryption, it needs to compute this polynomial, and then compare each coefficient. While the discussion domain is changed into $Z_{q^{l+t-2}}$, and the length of ciphertext is now $2n(l+t-2)\log_2 q$.

This scheme has two shortcomings in efficiency:

(1) The length of ciphertext is doubled after multiplication, and will be $4n\log_2 q$.

(2) The amount of ciphertext elements is increased in multiplication.

In brief, multiplication will cause a great decrease in efficiency, so multiplication can only be conducted by a limited times in this scheme.

4. Homomorphic encryption scheme basing on an improved scheme.

In this section, we make some important mortification to scheme2, and put forward a new homomorphic encryption scheme noted as scheme3 .

4.1 Mortification to scheme 2

We use canonical mapping to construct a new scheme basing on scheme2, in this new shceme, the operation is quite time-efficient and the amount of ciphertext elements will not increase after homomorphic evaluations.

4.1.1 An elaborated description of the new scheme

Scheme 3

- Parameters: q is a prime and $q \equiv 1 \bmod 2n$, let ω be a root of x^n+1 in Z_q, and ω is not a divisor of $\frac{q-1}{2}$, error distribution $\chi_{R,\alpha}^k$ is still a discrete Gauss distribution on R^k, with expectation 0 and standard deviation $\alpha \le 1/t\left(\sqrt{nk}\left\lceil \frac{r}{2} \right\rceil + 1\right)$. Definition of D_r and polynomial vector operations are the same as in scheme2.

- Private key: $s \overset{\$}{\leftarrow} R$, satisfying that $s(0)$ is not a divisor of $\frac{q-1}{2}$. The length of private key is $n\log_2 q$ bits.

- Public key: Randomly choose a k-dim polynomial vector $\hat{a} \overset{\$}{\leftarrow} R^k$. Choose error vector $\hat{e} \leftarrow \chi_{R,\alpha}^k$ and compute $\hat{b} = \hat{a}s + \hat{e} \in R^k$. To shorten the key length, we can also make all of the users share a same $\hat{a}$, and the public key is $(\hat{a}, \hat{b})$ with length of $kn\log_2 q$ bits.

- Encryption: Encryption includes three steps.

(1) For any given n-bits plaintext $m \in D_1$, let $m=(m_0, m_1, \ldots, m_{n-1})$ and randomly choose $\hat{r} \overset{\$}{\leftarrow} D_r^k$;

(2) Compute $c_0 = \hat{b} \otimes \hat{r}$, $c_1 = \hat{a} \otimes \hat{r}$. Noticing that c_0, c_1 are two polynomials in R, we

can use canonical mapping to change them into vectors in Z_q^n, namely

$$c_0 \mapsto \left(c_0(\omega), c_0\left(\omega^3\right), \cdots, c_0\left(\omega^{2n-1}\right)\right) = C_0$$

$$c_1 \mapsto \left(c_1(\omega), c_1\left(\omega^3\right), \cdots, c_1\left(\omega^{2n-1}\right)\right) = C_1$$

(3) Compute $C_2 = C_0 + \frac{q-1}{2}\left(m_0, \cdots m_{n-1}\right)$, and output the ciphertext $\left(C_1, C_2\right)$.

- Decryption: Also includes three steps.

(1) Use the inverse mapping of canonical mapping to change C_1 into a polynomial $c_1(x) = \hat{a} \otimes \hat{r}$;

(2) Compute $c_1(x) \cdot s = \hat{a} \otimes \hat{r} \cdot s = \hat{b} \otimes \hat{r} - \hat{e} \otimes \hat{r} \approx c_0(x)$, and change $c_1(x) \cdot s$ into a vector S;

(3) Compute $\left(C_2 - S\right) \mod \omega \approx \frac{q-1}{2} m$

4.1.2 Correctness

Theorem 1 When the parameters are chosen properly, Scheme 3 can decrypt correct.

Proof:

Consider the decryption process,

$$\begin{aligned} C_2 - S &= C_0 + \frac{q-1}{2} m - \sigma\left(c_1(x) \cdot s\right) \\ &= \sigma\left(c_0(x)\right) + \frac{q-1}{2} m - \sigma\left(c_1(x) \cdot s\right) \\ &= \left(c_0(\omega), \cdots, c_0\left(\omega^{2n-1}\right)\right) + \frac{q-1}{2} m - \left(c_1 s(\omega), \cdots, c_1 s\left(\omega^{2n-1}\right)\right) \\ &= \left(\left(\hat{b} \otimes \hat{r}\right)(\omega), \cdots, \left(\hat{b} \otimes \hat{r}\right)\left(\omega^{2n-1}\right)\right) - \left(\left(\hat{a} \otimes \hat{r} s\right)(\omega), \cdots, \left(\hat{a} \otimes \hat{r} s\right)\left(\omega^{2n-1}\right)\right) + \frac{q-1}{2} m \end{aligned}$$

We only discuss the first item, and conclusions about other items are the same. The first item of the above formula is

$$\left(\hat{a} \otimes \hat{r} s\right)(\omega) + \left(\hat{e} \otimes \hat{r}\right)(\omega) - \left(\hat{a} \otimes \hat{r} s\right)(\omega) + \frac{q-1}{2} m_0 = \left(\hat{e} \otimes \hat{r}\right)(\omega) + \frac{q-1}{2} m_0$$

Here $\left(\hat{e} \otimes \hat{r}\right)(\omega)$ is a polynomial about ω in R, and after a mod operation, it only remain the constant term. Let $\hat{e} = \left(e_1, \cdots, e_k\right)$, $\hat{r} = \left(r_1, \cdots, r_k\right)$, then $\hat{e} \otimes \hat{r} = \sum_{i=1}^{k} e_i r_i$. And because $\hat{e} \leftarrow \chi_{R,\alpha}^k$, following the discussion of section 4.1, $\sum_{i=1}^{k} e_i r_i$ abides a Normal distribution with expectation 0 and standard deviation $\sqrt{\sum_{i=1}^{k} \left(\frac{r\sqrt{n}\alpha}{2}\right)^2} = \frac{\sqrt{nk} r \alpha}{2} \leq \sqrt{nk} \left\lceil \frac{r}{2} \right\rceil \alpha \leq \frac{1}{t}$. According to the truncated inequality of Normal distribution， $\Pr\left[\sum_{i=1}^{k} e_i(0)\right] > \frac{q}{4} = \frac{4}{t}\sqrt{\frac{2}{\pi}}e^{-\frac{t^2}{32}}$. When t≥30, this value can be ignored, so $\Pr\left[\sum_{i=1}^{k} e_i(0)\right] \le \frac{q}{4} \approx 1$.

According to the parameter requirement of scheme3, ω is not a divisor of $\frac{q-1}{2}$, so the first item of $(C_2 - S) \bmod \omega$ is not greater than $\frac{q}{4} + \frac{q-1}{2}m_0$. This completes the correctness proof of scheme3.

4.1.3 Security

Here we give a reduction about the CPA security of scheme3 into the difficulty of decisional RLWE assumption.

Theorem 2 For any $\varepsilon > 0$ and $m \ge (1+\varepsilon)(1+n)\log q$, if there exists a PPT algorithm that can attack the CPA security of scheme 3 with advantage ϵ, then there exist a poly-time distinguisher V that for any possible private key *s*, can distinguish distribution $\left\{(\hat{a}, \hat{a}s + \hat{e}) \mid \hat{a} \overset{\$}{\leftarrow} R^k, \hat{e} \leftarrow D_{R,\xi}, s \overset{\$}{\leftarrow} R\right\}$ and the uniform distribution U on $R^k \times R^k$, here $\xi = \alpha \cdot (nk/\log(nk))^{1/4}$.

Proof:

We only discuss the first bit *m*0 of a plaintext. Suppose there exists a CPA attacker A that can distinguish the ciphertext of *m*0=0 and *m*0=1 with advantage ϵ. We construct a distinguisher V which can distinguish these two distributions with advantage at least ϵ/2:

$$\left\{(\hat{a}, \hat{a}s + \hat{e}) \mid \hat{a} \overset{\$}{\leftarrow} R^k, a_i(0) = 1, i = 1, \cdots, k, \hat{e} \leftarrow D_{R,\xi}, s \overset{\$}{\leftarrow} R, s(0) = 1\right\}$$

and Uniform distribution U on $R^k \times R^k$.

The distinguisher V is constructed as following:

Input of V are two polynomial vectors $(\hat{a}, \hat{b})$ in $R^k \times R^k$, and satisfying that each constant term of $\hat{a}$ is 1. Now V will call for A to judge that whether $(\hat{a}, \hat{b})$ is abide to uniform distribution or is a RLER vector.

Using $(\hat{a}, \hat{b})$ as private key, V invokes A, the latter generate two message bits *m*0, *m*1and send to V. V randomly choose $i \in \{0,1\}$, encrypt *m*i and send the ciphertext back to A. If A can return the correct I, then V will output 1, else output 0.

Let the challenging ciphertext be (C_1, C_2), if σ is canonical mapping, then the first bit of C1 and C₂ are $(\hat{a}\otimes\hat{r})(\omega)$ and $(\hat{b}\otimes\hat{r})(\omega)+\frac{q-1}{2}m_0$ respectively. If $\hat{b}$ is chosen randomly and uniformly in R^k and is independent with $\hat{a}$, then the first bit of the challenging ciphertext is also random and uniform. In this situation, the probability of V output 1 is at most 1/2. On the other hand, if $\hat{b}=\hat{a}s+\hat{e}$ and the parameters are chosen according to the requirement, then by assumption, probability of A correctly guess *i* is (1+ϵ)/2, so V can output 1 with the same probability. Thus completes the proof, namely, V can distinguish two distributions with advantage ϵ/2. ■

4.2 Homomorphic evaluations

Given two ciphertexts (C_1,C_2) and $\left(C_1^{'},C_2^{'}\right)$, here

$$C_1=\left(c_1(\omega),c_1(\omega^3),\cdots,c_1(\omega^{2n-1})\right),$$

$$C_2=C_0+\frac{q-1}{2}(m_0,\cdots m_{n-1})$$

$$=\left(c_0(\omega)+\frac{q-1}{2}m_0,c_0(\omega^3)+\frac{q-1}{2}m_1,\cdots,c_0(\omega^{2n-1})+\frac{q-1}{2}m_{n-1}\right),$$

$$C_1^{'}=\left(c_1^{'}(\omega),c_1^{'}(\omega^3),\cdots,c_1^{'}(\omega^{2n-1})\right),$$

$$C_2^{'}=\left(c_0^{'}(\omega)+\frac{q-1}{2}m_0^{'},c_0^{'}(\omega^3)+\frac{q-1}{2}m_1^{'},\cdots,c_0^{'}(\omega^{2n-1})+\frac{q-1}{2}m_{n-1}^{'}\right)$$

4.2.1 Addition

When computing the sum of two ciphertexts, we could simply add them coordinate-wise, and get

$$(C_{add1},C_{add2})=\left(C_1+C_1^{'},C_2+C_2^{'}\right)$$

Where

$$C_1+C_1^{'}=\left(c_1(\omega)+c_1^{'}(\omega),c_1(\omega^3)+c_1^{'}(\omega),\cdots,c_1(\omega^{2n-1})+c_1^{'}(\omega^{2n-1})\right)$$

$$C_2+C_2^{'}=\left(c_0(\omega)+c_0^{'}(\omega)+\frac{q-1}{2}\left(m_0+m_0^{'}\right),\cdots\cdots,c_0(\omega^{2n-1})+c_0^{'}(\omega^{2n-1})+\frac{q-1}{2}\left(m_{n-1}+m_{n-1}^{'}\right)\right)$$

are exactly the encryption of the sum of two plaintexts.

4.2.2 Multiplication

According to the features of canonical mapping, multiplication of two vectors could also done coordinate-wisely. Let "*" denote the multiplication of vectors coordinate-wise, then

$$(C_{mult1},C_{mult2})=\left(C_1*C_1^{'},C_2*C_2^{'}\right)$$

and

$$C_1 * C_1' = \left(c_1(\omega)c_1'(\omega), c_1(\omega^3)c_1'(\omega), \cdots, c_1(\omega^{2n-1})c_1'(\omega^{2n-1})\right)$$

$$C_2 * C_2'$$

$$= \left(\left(c_0(\omega) + \frac{q-1}{2}m_0\right)\left(c_0'(\omega) + \frac{q-1}{2}m_0'\right), \cdots\cdots, \left(c_0(\omega^{2n-1}) + \frac{q-1}{2}m_{n-1}\right)\left(c_0'(\omega^{2n-1}) + \frac{q-1}{2}m_{n-1}'\right)\right)$$

We discuss the decryption of the first item.

The first item of $C_2 * C_2'$ is

$$c_0(\omega)c_0'(\omega) + \frac{q-1}{2}m_0 c_0'(\omega) + \frac{q-1}{2}m_0' c_0(\omega) + \frac{(q-1)^2}{4}m_0 m_0'$$

In the decryption process, we need to change $C_1 * C_1'$ into a polynomial, multiply it with s^2 and then transform into a vector S_{mult}, the first item of which is

$$s^2(\omega)c_1(\omega)c_1'(\omega) = \hat{a} \otimes \hat{r}(\omega)s(\omega) \cdot \hat{a} \otimes \hat{r}'(\omega)s(\omega) \qquad （4\text{-}1）$$

Noticing that

$$c_0(\omega)c_0'(\omega) = [(\hat{a} \otimes \hat{r})(\omega)s(\omega) + (\hat{e} \otimes \hat{r})(\omega)][(\hat{a} \otimes \hat{r}')(\omega)s(\omega) + (\hat{e} \otimes \hat{r}')(\omega)] \qquad （4\text{-}2）$$

Subtract （4-2） by （4-1）， we can get

$$(\hat{e} \otimes \hat{r})(\omega)(\hat{e} \otimes \hat{r}')(\omega) + (\hat{a} \otimes \hat{r})(\omega)s(\omega)(\hat{e} \otimes \hat{r}')(\omega) + (\hat{a} \otimes \hat{r}')(\omega)s(\omega)(\hat{e} \otimes \hat{r})(\omega) = D$$

The last decryption step in scheme 3 is compute *C*₂-*S*, and after a homomorphic multiplication, it needs to compute $C_2 * C_2' - S_{mult}$. Then the first item is

$$D + \frac{q-1}{2}m_0 c_0'(\omega) + \frac{q-1}{2}m_0' c_0(\omega) + \frac{(q-1)^2}{4}m_0 m_0'$$

Where

$$c_0(\omega) = (\hat{a} \otimes \hat{r})(\omega)s(\omega) + (\hat{e} \otimes \hat{r})(\omega)$$

$$c_0'(\omega) = (\hat{a} \otimes \hat{r}')(\omega)s(\omega) + (\hat{e} \otimes \hat{r}')(\omega)$$

Noticing that in the first item, besides the first item, all of the other items are multiples of ω, at the same time, ω is not a divisor of $\frac{(q-1)^2}{4}$, so we can divide the first item by ω, and get the residue:

$$(\hat{e} \otimes \hat{r})(0)(\hat{e} \otimes \hat{r}')(0) + (\hat{a} \otimes \hat{r})(0)s(0)(\hat{e} \otimes \hat{r}')(0) + (\hat{a} \otimes \hat{r}')(0)s(0)(\hat{e} \otimes \hat{r})(0)$$

$$+ \frac{q-1}{2}m_0[(\hat{a} \otimes \hat{r})(0)s(0) + (\hat{e} \otimes \hat{r})(0)] + \frac{q-1}{2}m_0'[(\hat{a} \otimes \hat{r}')(0)s(0) + (\hat{e} \otimes \hat{r}')(0)] + \frac{(q-1)^2}{4}m_0 m_0'$$

Also noticing that *s*(0) is not a divisor of $\frac{q-1}{2}$, dividing the above formula by s(0) and get the residue, the first item is turned into

$$(\hat{e}\otimes\hat{r})(0)(\hat{e}\otimes\hat{r}')(0)+\frac{q-1}{2}m_0(\hat{e}\otimes\hat{r})(0)+\frac{q-1}{2}m_0{}'(\hat{e}\otimes\hat{r}')(0)+\frac{(q-1)^2}{4}m_0m_0{}'=\Delta$$

Here $(\hat{e}\otimes\hat{r})(0)$ and $(\hat{e}\otimes\hat{r}')(0)$ are the constant items of $\hat{e}\otimes\hat{r}$ and $\hat{e}\otimes\hat{r}'$ respectively. According to the discussion in section 3.1, the probability of each coefficient in $\hat{e}\otimes\hat{r}$ greater than q /4 is $\frac{4}{t}\sqrt{\frac{2}{\pi}}e^{-\frac{t^2}{32}}$, when t≥30, this value can be ignored. So $\|\hat{e}\otimes\hat{r}\|_\infty\le\frac{q}{4}$ holds with a probability close to 1. Consulting also the discussion in 4.2.2, when Δ>$\frac{(q-1)^2}{4}$ the decryption result is 1 in Z_{q^2}, else, is 0. Thus obtain the multiplication of two bits. This argument can be extended to other elements of the ciphertext, and we could soon get to the result that in scheme 3, $C_2 * C_2{}'$ can be correctly decrypted and get to the multiplication of two plaintexts.

4.3 Efficiency of scheme 3

The advantage of scheme3 lies in a shorter key length and small computation cost, we give a detailed analysis below.

Length of public key: the public key is a pair of polynomial vectors $(\hat{a},\hat{b})$ in R^k, if $\hat{a}$ is shared by all users, then it only need to take into consideration of $\hat{b}$, and the length of public key is $kn\log_2 q$ bits.

Length of private key: the private key is a polynomial in *R* with constant item 1, and the length of private key is $n\log_2 q$ bits.

Length of ciphertext: In scheme3, an *n* bits plaintext is encrypted into a ciphertext of $2n\log_2 q$ bits.

Computation cost of encryption: It needs to compute a polynomial convolution, then two canonical mapping and finally a vector addition on Z_q^n. Here the computing cost of polynomial convolution can be reduced through a fast Fourier transformation. And the total computation cost of encryption is $\tilde{O}(n\log n)$.

Computation cost of decryption: It needs to compute the inverse of canonical mapping, namely to solve a linear equation set on Z_q, then compute a polynomial multiplication and one canonical mapping, finally a vector subtraction. And the total computation cost of encryption is $\tilde{O}(n\log n)$.

Homomorphic addition: The addition of two ciphertexts is simply vector addition by coordinate wise, the computation cost is $\widetilde{\mathrm{O}}(n)$. After an addition, the length of ciphertext is not increased, and accordingly the computation cost of decryption remains the same.

Homomorphic multiplication: When multiplying two ciphertexts, it needs to directly compute vector multiplication on Z_q^n coordinate-wise, the computing cost is $\widetilde{\mathrm{O}}(n\log n)$. After one multiplication, the length of ciphertext is increased to $4n\log_2 q$ bits, namely doubled. In decryption phase, for each ciphertext element, it needs to solve a linear equation set, then compute one polynomial multiplication and one subtraction, the total computation cost of decryption is $\widetilde{\mathrm{O}}(n^2)$.

To sum up, we confirm that comparing with scheme 2, scheme 3 has an obvious advantage in efficiency. The key length and computation cost is controlled in a rational bound, and, after one multiplication, the ciphertext vector still remains two elements, though there is an increase in length. We believe that scheme 3 is a practical homomorphic encryption scheme.

5 Conclusion

This paper provides a somewhat homomorphic multi-bit encryption scheme that is basing on RLWE assumption. We use canonical mapping to change polynomial computation into vector operation that is coordinate-wise. Due to this technique, the computation cost of our scheme is very limited, and the key length is short. We give a security proof to show that the scheme is CPA secure on condition that RLWE assumption holds.

Homomorphic encryption scheme is a new hot point in cryptography. There has been abundant works in recent years focusing on scheme construction and application, and new methods and new ideas have appeared continuously. However there still leaves a lot of problems to solve in this area, both in theoretical and practical.

Aiming on performance improvement, we use a new technique to construct scheme, and our scheme is practical due to its computation cost and key length, while because homomorphic multiplication can cause an increase in ciphertext length, the scheme is somewhat but not fully homomorphic. However, scheme 3 in this paper can be a potential somewhat homomorphic encryption scheme. Further studies on controlling ciphertext length and ultimately constructing fully homomorphic encryption schemes will be our target in the future.

Acknowledgements

This work was financially supported by the National Natural Science Foundation of China (Grant No. 61272492, 61103230) and Natural Science Basic Research Foundation of Engineering University of Chinese People’s Armed Police Force(Grant No. WJY201121)

References

[1] R.L.Rivest, L. Adleman, M.L.Dertouzos. On Data Banks and Privacy Homomorphisms.

Foundations of Secure Computation, pp.169-177, 1978.

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

[3] N.P.Smart, F. Vercauteren. Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. PKC 2010, LNCS 6056, pp.420-443, 2010

[4] D. Stehle, R.Steinfeld. Faster Fully Homomorphic Encryption. ASIACRYPT 2010, LNCS 6477, pp.377-394, 2010.

[5] M. Dijk, C.Gentry, S.Halevi, V.Vaikuntanathan, Fully Homomorphic Encryption over the Integers. Advancesin Cryptology-EUROCRYPT 2010, pp.24-43, 2010. 1, 7, 8

[6] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. Advances in Cryptology-CRYPTO2011, pp.505-524, 2011. 1, 9, 13

[7] A.Bogdanov and C.Lee, Homomorphic encryption from codes, Arxiv preprint arXiv:1111.4301, 2011. 1, 9, 14

[8] R.Rothblum. Homomorphic encryption: from private-key to public-key. Theory of cryptography, pp,219-234, 2011.

[9] S. Goldwasser, B. Barak, L.Reyzin, Y.Kalai, and S. Vadhan. New developments in cryptography, Lectures of MIT's 6.889 and BU's CAS CS 937, Spring 2011. http://www.cs.bu.edu/~reyzin/teaching/s11cs937/, 2011. 2,3,4,5,6

[10] C.Gentry, S. Halevi, V. Vaikuntanathan. A simple BGN-type cryptosystem from LEW. In Eurocrypt2010, LNCS vol 6110. pp. 506-522. Springer, Heidelberg 2010.

[11] C.Gentry, C.Peikert and V.Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In STOC, pages197-206, 2008.

[12] O.Regev. On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM, 56(6):34,2009. Preliminary version in STOC'05.

[13] C.Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In STOC, pages 333–342. 2009.

[14] V. Lyubashevsky, C. Peikert, O. Regev. On Ideal Lattices and Learning With Errors Over Rings, Eurocryt2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp.1-23.

[15] Markus Rückert, Michael Schneider. Estimating the Security of Lattice Based cryptosystems. http://eprint.icur.org/2010/137.pdf