# A Multi-Bit Fully Homomorphic Encryption with Shorter Public Key from LWE

**Zhigang Chen[1], Xinxia Song[2]**

1. College of Computer and Information, Zhejiang Wanli University, Zhejiang NingBo 315100, China.

zhig.chen@foxmail.com

2. College of Junior，Zhejiang Wanli University, Zhejiang Ningbo 315100, China

## ABSTRACT

The efficiency of fully homomorphic encryption is a big question at present. To improve efficiency of fully homomorphic encryption, we use the technique of packed ciphertext to construct a multi-bit fully homomorphic encryption based on Learning with Errors problem. Our scheme has a short public key. Since our fully homomorphic encryption scheme builds on the basic encryption scheme that choose Learning with Errors samples from Gaussian distribution and add Gaussian error to it, which result in that the number of Learning with Errors samples decrease from $2n\log q$ to $n+1$. We prove that our fully homomorphic encryption scheme is feasible and its security relies on the hardness of Learning with Errors problem. In addition we adapt the optimization for the process of key switching from GHS13 and formal this new process of key switching for multi-bit fully homomorphic encryption. At last, we analyze the concert parameters and compare these parameters between our scheme and GHS13 scheme. The data show that our scheme has public key smaller by a factor of about $\log q$ than it in GHS13 scheme.

## Keywords

Fully Homomorphic Encryption; Public Key Encryption; Multi-Bit Plaintext; Concert Security Parameters.

## 1. INTRODUCTION

Fully homomorphic encryption (FHE) supports arbitrarily computation on encrypted data without using secret key. FHE has a number of potential applications such as private cloud computing. The first FHE scheme was proposed by Gentry in 2009 [1]. Then numerous schemes based on different hardness assumptions have been proposed [1, 2, 3, 4, 5, 6, 7] and some techniques have been developed to improve efficiency [8, 9,10,11].

FHE is still quite expensive following its invention, which hinder application of FHE in practical. Specially, the ciphertext contain noise due to security consideration so that each homomorphic operation will increase the noise in ciphertext. In particularly, homomorphic multiplication increases the noise significantly. When the noise exceeds the bound of correct decryption, homomorphic operation cannot be performed.

To perform more homomorphic operations, we must set large parameters so that the ciphertext has enough space to accommodate noise, which lead to large ciphertext size. To improve efficiency of FHE, there is a technique named packed ciphertext proposed in [12], which can pack some plaintext values into one ciphertext. Performing once homomorphic operation for a packed ciphertext is equivalent to performing the same operation for these plaintext values simultaneously. The technique of packed ciphertext is originally based on polynomial Chinese reminder theorem (CRT) [12], which can be applied in the FHE based on ring Learning with Errors so as to achieve a nearly optimal homomorphic evaluation in [8]. In addition, Brakerski et al. describe how to apply the technique of packed ciphertext in FHE based on Learning with Errors (LWE) [9], and we refer their scheme as GHS13. However, GHS13 scheme is only a symmetric FHE and they don't describe how to achieve FHE in detail.

The goal of this paper is to construct a multi-bit FHE with short public key using packed ciphertext. Note that our FHE scheme is not the asymmetric version of GHS13, since both build on the different basic encryption schemes that results in different size of parameters in both FHE schemes. In GHS13 scheme, Brakerski et al. use Regev-type cryptosystem to construct FHE. In this paper our scheme build on the Linder and Peikert's encryption scheme (LP10) proposed in [13], which is different with GHS13. In our basic encryption scheme, we choose LWE samples from Gaussian distribution and add Gaussian error to it, which result in that the number of LWE samples decrease from $2n\log q$ to $n+1$. The smaller public key comes from the different style of the basic encryption scheme.

Furthermore, it is well known that key switching is a critical technique to achieve LWE-based FHE. However, using key switching to construct FHE is expensive. To improve the efficiency of key switching, we optimize the process of key switching as in [9], and we formal this new process of key switching in term of multi-bit FHE. For example, a key switching matrix for a multi-bit FHE is $(n+t)^2\lceil \log q \rceil \times (n+t)$matrix in the traditional process of key switching, where $t$ is the length of message. In our scheme, a key switching matrix is only $(n+t)^2 \times (n+t)$ matrix. Since key switching needs to be performed after each homomorphic multiplication, thus this optimization for key switching is important to improve efficiency of FHE.

For application of FHE, it is also very important to analyze how to estimate parameters of a FHE scheme to ensure correctness and security against lattice attacks. Given a security level required by a real-world application, we analyze the concert parameters for fully homomorphic encryption based on Learning with Error problem. We obtain concert parameters of our scheme and GHS13 scheme by this method. The data shows our scheme has a better public key size than the asymmetric version of GHS13 scheme.

This paper is organized as follows. Section 2 introduces the LWE assumption and defines homomorphic encryption and its related terms. Section 3 describes the basic encryption scheme. Section 4 defines homomorphic addition and homomorphic multiplication. The new key switching process is introduced in this section. Section 5 describes a FHE scheme. Section 6 analyzes the noise growth in homomorphic addition and homomorphic multiplication, which show it is possible to achieve a leveled FHE scheme. Section 7 gives the parameters property and concert parameters.

## 2. PRELIMINARIES

### 2.1 Basic Notation

We use $\lfloor x \rceil$ to indicate rounding $x$ to the nearest integer, and $\lfloor x \rfloor$, $\lceil x \rceil$ (for $x \geq 0$) to indicate rounding down or up. When $q$ is not a power of two, we will use $\lceil \log q \rceil$ to denote $1+\lfloor \log q \rfloor$. For an integer $q$, we define the set $\mathbb{Z}_q = (-q/2, q/2] \cap \mathbb{Z}$. For any $x \in \mathbb{Z}$, let $y=[x]_q$ denote the unique value $y \in (-q/2, q/2]$. $x \leftarrow \mathcal{D}$ means that $x$ is a sample from a distribution $\mathcal{D}$. We define $B$-bounded distributions as ones whose magnitudes never exceed $B$.

### 2.2 Learning with Errors

The LWE problem was introduced by Regev in [14] as a generalization of the well-known “learning parity with noise” problem, to larger moduli. This problem was later generalized as the ring LWE problem by Lyubaskevsky, Peikert and Regev in [15].

The LWE problem is parameterized by a dimension $n \geqslant 1$ and integer modulus $q \geqslant 2$, as well as a probability distribution $\chi$ over $\mathbb{Z}$ or $\mathbb{Z}_q$. For a vector $\boldsymbol{s} \in \mathbb{Z}_q^n$, the LWE distribution $\mathcal{A}_{\boldsymbol{s},\chi}$ is obtained by choosing a vector $\boldsymbol{a}$ from $\mathbb{Z}_q^n$ uniformly at random and a noise term $e \leftarrow \chi$, and outputting $(\boldsymbol{a}, b = <\boldsymbol{a}, \boldsymbol{s}> + e \bmod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The search-LWE problem is, given an arbitrary number of independent samples $(\boldsymbol{a}_i, b_i) \leftarrow \mathcal{A}_{\boldsymbol{s},\chi}$, to find $\boldsymbol{s}$. We are primarily interested in the decision-LWE (DLWE) problem for cryptographic applications. The decision-LWE problem is to distinguish with some non-negligible advantage between the two cases. One case is any desired number of independent samples $(\boldsymbol{a}_i, b_i) \leftarrow \mathcal{A}_{\boldsymbol{s},\chi}$. Another case is the same number of independent samples drawn from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

There are two kinds of reductions such as quantum reduction [14] and classical reduction [16, 17] from worst-case lattice problems to the LWE problem. In addition, if the vector $\boldsymbol{s}$ is sampled from the distribution $\chi$, then the LWE problem is still hard.

For a lattice $\Lambda$ and a positive real $r > 0$, we denote $D_{\Lambda,r}$ as the discrete Gaussian distribution over $\Lambda$ and parameter $r$, which is the probability distribution that assigns mass proportional to $\exp(-\pi\|\boldsymbol{x}\|^2/s^2)$ to each point $\boldsymbol{x} \in \Lambda$. For $\Lambda = \mathbb{Z}^n$, the discrete Gaussian $D_{\mathbb{Z}^n,r}$ is simply the product distribution of $n$ independent copies of $D_{\mathbb{Z},r}$.

### 2.3 Leveled Homomorphic Encryption

A homomorphic encryption scheme HE=(Keygen, Enc, Dec, Eval) includes a quadruple of PPT algorithms. For the definition of full homomorphic encryption in detail, readers refer to these papers [1, 5].

There are two types of fully homomorphic encryption schemes. One is the leveled fully homomorphic encryption scheme, in which the parameters of a scheme depend on the multiplication depth that the scheme can evaluate. In this case, any circuit with a polynomial depth can be evaluated. The other one is pure fully homomorphic encryption schemes, which can be built by using bootstrapping method from a leveled fully homomorphic encryption scheme with the assumption of circular security. A pure fully homomorphic encryption scheme can evaluate the circuit whose depth is not limited. The following definitions are taken from [5].

**Definition 1** ($L$-homomorphism). A scheme HE is $L$-homomorphic, for $L$=$L(\lambda)$, if for any depth $L$ arithmetic circuit $f$ (over GF(2)) and any set of inputs $m_1,\ldots,m_l$, it holds that

$\Pr[\text{HE.Dec}_{sk}(\text{HE.Eval}_{evk}(f,c_1,\ldots,c_l))\neq f(m_1,\ldots,m_l)] = \text{negl}(\lambda)$ ,

where $(pk, evk, sk)\leftarrow\text{HE.Keygen}(1^{\lambda})$ and $c_i\leftarrow$ $\text{HE.Enc}_{pk}(m_i)$.

**Definition 2** (compactness, full homomorphism and leveled full homomorphism). A homomorphic scheme is compact if its decryption circuit is independent of the evaluated function. A compact scheme is fully homomorphic if it is $L$-homomorphism for any polynomial $L$. The scheme is leveled fully homomorphic scheme if it takes $1^L$ as additional input in key generation.

## 3. THE BASIC ENCRYPTION SCHEME

At present all of FHE schemes are built on some basic encryption scheme. Our FHE scheme is built on the cryptosystem proposed by Lindner and Peikert [13]. Below we describe this cryptosystem and then analyze encryption noise and decryption noise of this cryptosystem, which is important to construct FHE scheme later. An integer modulus $q \geq 2$, integer dimension $n_1$, $n_2$ and a Gaussian distribution $D_{\mathbb{Z},r}$ denoted as $\chi$ , which relate to the underlying LWE problem. In order for the smallest public keys, a uniformly random public matrix $\mathbf{A}\in\mathbb{Z}_q^{n_1\times n_2}$ can be generated by a trusted source, and is used by all parties in the system. If the trusted source is not got in the system, **A** may be generated in the step of key generation and as part of public key.

**SecretKeygen**($1^{n_2}$ ): Choose a matrix $\mathbf{S}\leftarrow$ $\chi^{t\times n_2}$ . Output $sk$ = $\mathbf{S}'\leftarrow(\mathbf{I}\mid\mathbf{-S})$, where **I** is the $t\times t$ identity matrix. Thus the secret key $sk$ is a $t\times(t+n_2)$ matrix in which each row can be viewed as a secret key that can recover one bit of multi-bit message.

**PublicKeygen**(**A,** $sk$): Choose $\mathbf{E}\leftarrow$ $\chi^{n_1\times t}$ , and let $\mathbf{B}=\mathbf{A}\mathbf{S}^{\mathrm{T}}+\mathbf{E}\in\mathbb{Z}_q^{n_1\times t}$ . Set the public key $pk$ = **B**.

**Enc**(**A,** $pk$, $\boldsymbol{m}$)**:** To encrypt a multi-bit message $\boldsymbol{m}\in\mathbb{Z}_2^t$ , sample $\boldsymbol{e}_1\leftarrow$ $\chi^{n_1}$ , $\boldsymbol{e}_2\leftarrow$ $\chi^{t}$, and $\boldsymbol{e}_3\leftarrow$ $\chi^{n_2}$ , and output $\boldsymbol{c}\leftarrow(\left\lfloor\frac{q}{2}\right\rfloor\cdot\boldsymbol{m}+$ $\mathbf{B}^{\mathrm{t}}\cdot\boldsymbol{e}_1+\boldsymbol{e}_2,\mathbf{A}^{\mathrm{t}}\cdot\boldsymbol{e}_1+\boldsymbol{e}_3)\ \in\mathbb{Z}_q^{n_2+t}$ .

**Dec**($sk$, $\boldsymbol{c}$)**:** Compute $\boldsymbol{v}\longleftarrow\mathbf{S}'\boldsymbol{c}$ mod $q$ and output $\boldsymbol{m}\longleftarrow\lfloor\frac{2}{q}\cdot\boldsymbol{v}\rceil\bmod 2$ .

For security purpose the noise is added at encryption and correct decryption depend on the noise magnitude. Next we analyze the noise magnitude at encryption and decryption.

**Lemma 3.1** (encryption noise). Let $q$, $n_2$, **A**, $|\chi| \le B$ be parameters in above encryption scheme. The secret key **S′** and public key **B** are generated from **SecretKeygen**($1^n$) and **PublicKeygen(A, S′)**. Set $\boldsymbol{c} \leftarrow$ **Enc(A, B,** $\boldsymbol{m}$**)**. Then for some $\boldsymbol{e}$ with $\|\boldsymbol{e}\|_\infty \leq E < (n_1+n_2)B^2 + B$, it holds that

$$\mathbf{S}'\boldsymbol{c} = \left\lfloor \frac{q}{2} \right\rfloor \cdot \boldsymbol{m} + \boldsymbol{e} \ (mod\, q) .$$

**Proof**. By definition

$$\mathbf{S}'\boldsymbol{c} = \left\lfloor \frac{q}{2} \right\rfloor \cdot \boldsymbol{m} + \mathbf{B}^{\mathrm{t}} \cdot \boldsymbol{e}_1 + \boldsymbol{e}_2 - \mathbf{S}\mathbf{A}^{\mathrm{t}} \cdot \boldsymbol{e}_1 - \mathbf{S}\boldsymbol{e}_3 \ (mod\, q)$$

$$= \left\lfloor \frac{q}{2} \right\rfloor \cdot \boldsymbol{m} + (\mathbf{B}^{\mathrm{t}} - \mathbf{S}\mathbf{A}^{\mathrm{t}}) \cdot \boldsymbol{e}_1 - \mathbf{S}\boldsymbol{e}_3 + \boldsymbol{e}_2 \ (mod\, q)$$

$$= \left\lfloor \frac{q}{2} \right\rfloor \cdot \boldsymbol{m} + \mathbf{E}^{\mathrm{T}} \cdot \boldsymbol{e}_1 - \mathbf{S}\boldsymbol{e}_3 + \boldsymbol{e}_2 \ (mod\, q) .$$

Since $|\chi| \le B$, we have $\left\| \mathbf{E}^{\mathrm{T}}\boldsymbol{e}_1 - \mathbf{S}\boldsymbol{e}_3 + \boldsymbol{e}_2 \right\|_\infty \leq (n_1+n_2)B^2 + B$ and the lemma follows.

We refer to $\boldsymbol{e}$ as the noise in ciphertext. The above Lemma give the bound of noise magnitude in "fresh ciphertext" that is the result of encryption and not the result of homomorphic operations on encrypted data.

**Lemma 3.2** (decryption noise). Choose a matrix $\mathbf{S} \leftarrow \chi^{t \times n_2}$ . Let $\boldsymbol{c} \in \mathbb{Z}_q^{n_2+t}$ be a vector such that

$$\mathbf{S}'\boldsymbol{c} = \left\lfloor \frac{q}{2} \right\rfloor \cdot \boldsymbol{m} + \boldsymbol{e} \ (\mathrm{mod}\, q) ,$$

where $\boldsymbol{m} \in \mathbb{Z}_2^t$ and $\mathbf{S}' \leftarrow (\mathbf{I} \mid -\mathbf{S})$. If $\|\boldsymbol{e}\|_\infty < \left\lfloor \frac{q}{4} \right\rfloor$, then we have $\boldsymbol{m} \leftarrow$ **E.Dec(S′,** $\boldsymbol{c}$**)**.

The decryption is as same as Regev's encryption scheme in [14]. We omit the proof of above Lemma. In order to recover message, $|\boldsymbol{e} / \left\lfloor \frac{q}{2} \right\rfloor|$ should be less than 1/2. Thus the condition for correct decryption is $|\boldsymbol{e}| < \left\lfloor \frac{q}{2} \right\rfloor /2$. Since $\left\lfloor \frac{q}{4} \right\rfloor \le \left\lfloor \frac{q}{2} \right\rfloor /2$, we can also take the bound of noise magnitude as $\left\lfloor \frac{q}{4} \right\rfloor$.

## 4. HOMOMORPHIC OPERATION

Suppose $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ encrypt $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ under the secret key **S′** respectively; that is, $\mathbf{S}'\boldsymbol{c}_i = \left\lfloor \frac{q}{2} \right\rfloor \cdot \boldsymbol{m}_i + \boldsymbol{e}_1 (\mathrm{mod}\, q)$ with small $\boldsymbol{e}_i$ for $i$={1,2}.

If the ciphertext $\boldsymbol{c}$ resulted from addition or multiplication of two ciphertext $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ can hold $\mathbf{S}'\boldsymbol{c} = \left\lfloor \frac{q}{2} \right\rfloor \cdot (\boldsymbol{m}_1 + \boldsymbol{m}_2) + \boldsymbol{e} \ (\mathrm{mod}\, q)$ or

$\mathbf{S}'\boldsymbol{c} = \left\lfloor \frac{q}{2} \right\rfloor \cdot (\boldsymbol{m}_1 \odot \boldsymbol{m}_2\ ) + \boldsymbol{e}$（mod $q$） for small $\boldsymbol{e}$, where $\boldsymbol{m}_1 \odot \boldsymbol{m}_2$ means the bitwise product, we say that additive or multiplicative homomorphism could be achieved.

The above basic encryption scheme has additive homomorphic property itself. To obtain multiplicative homomorphic property, we define the ciphertext for multiplication as $\lfloor \frac{2}{q} \cdot (\boldsymbol{c}_1 \otimes \boldsymbol{c}_2) \rceil$ like definition in paper [5]. However, the secret key is the matrix and is not the vector in the above basic encryption scheme, what is the form of the secret key corresponding to the multiplication of two ciphertexts? In fact, each row in the secret key matrix can be used to recover a bit of message. If the length of message is $t$, the secret key matrix is viewed as $t$ row vectors. We refer to $\boldsymbol{s}_i$ as the $i$-th row in the secret key matrix $\mathbf{S}'$. According to the above explain, decrypting the ciphertext $\lfloor \frac{2}{q} \cdot (\boldsymbol{c}_1 \otimes \boldsymbol{c}_2) \rceil$ by the tensor vector $\boldsymbol{s}_i \otimes \boldsymbol{s}_i$ will result in a product of the $i$-th bit of two messages with respect to two ciphertexts $\boldsymbol{c}_1$, $\boldsymbol{c}_2$. We store the tensor vector $\boldsymbol{s}_i \otimes \boldsymbol{s}_i$ as the rows of the matrix ST that is the secret key matrix relative to ciphertext $\lfloor \frac{2}{q} \cdot (\boldsymbol{c}_1 \otimes \boldsymbol{c}_2) \rceil$.

Thus the secret key matrix ST is a $t \times (t+n_2)^2$ matrix. We next analyze the condition of correct decryption for homomorphic operation.

## 4.1 Homomorphic Addition

By definition

$$\mathbf{S}'(\boldsymbol{c}_1+\boldsymbol{c}_2) = \mathbf{S}'\boldsymbol{c}_1 + \mathbf{S}'\boldsymbol{c}_2 = \left\lfloor \frac{q}{2} \right\rfloor \bullet (\boldsymbol{m}_1 + \boldsymbol{m}_2) + \boldsymbol{e}_1 + \boldsymbol{e}_2 \pmod q.$$

The noise increase a little in homomorphic addition. If the noise magnitude is small, namely, $\|\boldsymbol{e_1} + \boldsymbol{e_2}\|_\infty < \left\lfloor \frac{q}{4} \right\rfloor$, the ciphertext $\boldsymbol{c}_1+\boldsymbol{c}_2$ can be decrypted correctly. It means the sum of ciphertexts encrypts the sum of messages.

## 4.2 Homomorphic Multiplication

Let an error $\boldsymbol{r} = \lfloor \frac{2}{q} \cdot (\boldsymbol{c}_1 \otimes \boldsymbol{c}_2) \rceil - \frac{2}{q} \cdot (\boldsymbol{c}_1 \otimes \boldsymbol{c}_2)$. Recall that the secret key is the matrix ST relative to the ciphertext vector $\lfloor \frac{2}{q} \cdot (\boldsymbol{c}_1 \otimes \boldsymbol{c}_2) \rceil$. By definition, we have

$$\mathrm{ST} \cdot \lfloor \frac{2}{q} \cdot (\boldsymbol{c}_1 \otimes \boldsymbol{c}_2) \rceil = \mathrm{ST} \cdot \frac{2}{q} \cdot (\boldsymbol{c}_1 \otimes \boldsymbol{c}_2) + \mathrm{ST} \cdot \boldsymbol{r} \pmod q$$

$$= \left\lfloor \frac{q}{2} \right\rfloor \cdot (\boldsymbol{m}_1 \odot \boldsymbol{m}_2) + \boldsymbol{e}_1^{mult} + \mathrm{ST} \cdot \boldsymbol{r} \pmod q \qquad (4.1)$$

$$= \left\lfloor \frac{q}{2} \right\rfloor \cdot (\boldsymbol{m}_1 \odot \boldsymbol{m}_2) + \boldsymbol{e}_1^{mult} + \boldsymbol{e}_2^{mult} \pmod q$$

where $\boldsymbol{e}_1^{mult}$ is the noise in the ciphertext $\frac{2}{q} \cdot (\boldsymbol{c}_1 \otimes \boldsymbol{c}_2)$ and $\boldsymbol{e}_2^{mult} = \mathrm{ST} \cdot \boldsymbol{r}$.

If $\left\| \boldsymbol{e}_1^{mult} + \boldsymbol{e}_2^{mult} \right\|_\infty < \left\lfloor \frac{q}{4} \right\rfloor$, the tensored ciphertext for multiplication $\lfloor \frac{2}{q} \cdot (\boldsymbol{c}_1 \otimes \boldsymbol{c}_2) \rceil$ can be decrypted correctly under the secret key ST.

## 4.3 Key Switching

Even though the tensored ciphertext for multiplication enable to achieve the property of homomorphic multiplication, it leads to the expansion of dimension of ciphertext and secret key. Thus key switching technique was introduced in [3, 4], which can convert one ciphertext of high dimension under the secret key of high dimension into another ciphertext of normal dimension under the secret key of normal dimension. However the key switching described in [3, 4] is not efficient. Since the secret key need to be represented as binary bit in order to reduce the noise in the process of key switching, which result in expansion of the dimension of ciphertext and secret key. Here we apply the technique proposed by Gentry et al. in [18] to improve efficient of key switching and formal this new key switching for multi-bit FHE.

In addition, if it only put key switching matrixes corresponding to the rows in the secret key matrix ST together to form a new key switching matrix, the result of key switching will be the collection of ciphertexts of normal dimension. To get only a single ciphertext resulted from key switching, we apply the method of multi-bit encryption in key switching as same as in [9] to yield key switching matrix that lets us convert the single ciphertext of high dimension into a single ciphertext of normal dimension. The process of key switching is described as below.

**SwitchKeyGen**($\mathbf{S}_1 \leftarrow \chi^{t\times n_s}$, $\mathbf{S}_2 \leftarrow \chi^{t\times n_t}$)：The parameters is described below, which allow to switch ciphertext under the secret key $\mathbf{S}_1$ into the ciphertext under the secret key [$\mathbf{I}|\mathbf{S}_2$], where $\mathbf{I}$ is the identity matrix and [$\mathbf{I}|\mathbf{S}_2$] means the horizontal concatenation of the matrix $\mathbf{I}$ and $\mathbf{S}_2$. Let $l = \lceil \log q \rceil$, and let $\chi$ be an error distribution for which the decision-LWE problem with modulus $P = 2^l q$ is hard.

Choose a uniform matrix $\mathbf{A} \in \mathbb{Z}_P^{n_t \times n_s}$. Sample $\mathbf{E} \leftarrow \chi^{t\times n_s}$.

Set $\mathbf{B} \leftarrow \mathbf{S}_2\mathbf{A} + \mathbf{E} + \mathbf{2}^l\mathbf{S}_1 \in \mathbb{Z}_P^{t\times n_s}$. Output $\mathbf{W} = \left[\frac{\mathbf{B}}{\mathbf{A}}\right] \cdot \mathbf{2}^{-l} \in \mathbb{Q}^{(t+n_t)\times n_s}$, where $\left[\frac{\mathbf{B}}{\mathbf{A}}\right]$ means the vertical concatenation of the matrix A and B.

**SwitchKey** ($\mathbf{W} \in \mathbb{Q}^{(t+n_t)\times n_s}$, $\boldsymbol{c}_1 \in \mathbb{Z}_q^{n_s}$)：Output $\boldsymbol{c}_2 \leftarrow \lceil \mathbf{W}\boldsymbol{c}_1 \rfloor \bmod q \in \mathbb{Z}_q^{t+n_t}$.

We call $\mathbf{W}$ as key switching matrix. The process of key switching is essentially the product of an $(t+n_t)$×$n_s$ key switching matrix and an $n_s$-dimensional ciphertext vector. Next, we describe the correctness of key switching, namely the decryption of the resulting ciphertext after key switching can preserve correctness.

**Lemma 4.1** Let $\mathbf{S}_1$, $\mathbf{S}_2$, $q$, $\mathbf{A}$, $\mathbf{W}$ be parameters as described in **SwitchKeyGen**. Let $\boldsymbol{c}_1 \in \mathbb{Z}^{n_s}$ and $\boldsymbol{c}_2 \leftarrow$ **SwitchKey(W,** $\boldsymbol{c}_1$**)** . Then, $[\mathbf{I}|\mathbf{S}_2] \cdot \boldsymbol{c}_2 = \boldsymbol{e}_t + \mathbf{S}_1\boldsymbol{c}_1 \pmod{q}$, where $\boldsymbol{e}_t = \mathbf{2}^{-l} \cdot \mathbf{E}\boldsymbol{c}_1 + [\mathbf{I}|\mathbf{S}_2]\boldsymbol{e}_w$ is the noise in the ciphertext $\boldsymbol{c}_2$.

**Proof**. Let $\boldsymbol{e}_w = \lceil \mathbf{W}\boldsymbol{c}_1 \rfloor - \mathbf{W}\boldsymbol{c}_1$. By definition

$$[\mathbf{I}|\mathbf{S}_2] \cdot \boldsymbol{c}_2 = [\mathbf{I}|\mathbf{S}_2] \cdot \lceil \mathbf{W}\boldsymbol{c}_1 \rfloor \pmod{q}$$

$$= [\mathbf{I}|\mathbf{S}_2] \cdot \mathbf{W}\boldsymbol{c}_1 + [\mathbf{I}|\mathbf{S}_2]\boldsymbol{e}_w \pmod{q}$$

$$= [\mathbf{I}|\mathbf{S}_2] \left[\frac{\mathbf{B}}{\mathbf{A}}\right] \cdot \mathbf{2}^{-l} \cdot \boldsymbol{c}_1 + [\mathbf{I}|\mathbf{S}_2]\boldsymbol{e}_w \pmod{q}$$

$$= \mathbf{2}^{-l} \cdot \mathbf{E}\boldsymbol{c}_1 + [\mathbf{I}|\mathbf{S}_2]\boldsymbol{e}_w + \mathbf{S}_1\boldsymbol{c}_1 \pmod{q}$$

$$= \boldsymbol{e}_t + \mathbf{S}_1\boldsymbol{c}_1 \pmod{q}.$$

Note that since $\mathbf{E}$, $\mathbf{2}^{-l}\boldsymbol{c}_1$ and $[\mathbf{I}|\mathbf{S}_2]\boldsymbol{e}_w$ is small, $\boldsymbol{e}_t$ is also small. The above Lemma tell us that the noise magnitude in the resulting ciphertext $\boldsymbol{c}_2$ increase a little, but the resulting ciphertext still can be decrypted correctly as long as the noise in the source ciphertext is small. Next we consider the security for the key switching.

**Lemma 4.2** Let $\mathbf{S}_1 \leftarrow \chi^{t \times n_s}$ **,** $\mathbf{S}_2 \leftarrow$ **SecretKeygen(** $1^{n_t}$ **)** and $\mathbf{W} \leftarrow$ **SwitchKeyGen(S**$_1$**, S**$_2$**)**. Then **W** is computationally indistinguishable from uniform over $\mathbb{Q}^{(t+n_t) \times n_s}$ assuming decision-LWE problem is hardness.

**Proof**. We have $\mathbf{W} = \left[\frac{\mathbf{B}}{\mathbf{A}}\right] \cdot \mathbf{2}^{-l} \in \mathbb{Q}^{(t+n_t) \times n_s}$ from above key switching, where **A** is a uniform matrix and $\mathbf{B} \leftarrow \mathbf{S}_2\mathbf{A} + \mathbf{E} + \mathbf{2}^l\mathbf{S}_1$. Because **B** is a matrix whose entries are the ciphertext of Regev's scheme, **B** is computationally indistinguishable from uniform over $\mathbb{Z}_P^{t \times n_s}$ **.** Therefore **W** is computationally indistinguishable from uniform over $\mathbb{Q}^{(t+n_t) \times n_s}$ .

## 5. A HOMOMORPHIC ENCRYPTION SCHEME

A leveled homomorphic encryption scheme we describe as below. For a leveled homomorphic encryption scheme, the circuit depth $L$ is first be given before homomorphic evaluation. Each level in circuit has a different secret key. Homomorphic operations are just to be performed from level $L$ to 1. The first level is level $L$, and the last level is level 0. The level 0 is only used to switch key. After each homomorphic operation, we need to transform the result to enter into the next level of circuit. Before each homomorphic operation, it requires that the two ciphertext have the same secret key (namely, the same level). Otherwise, we need transform the higher level ciphertext into lower level. The function of **FHE.RefreshNextLevel** is to do it. We note the key switching is just used for tensored ciphertext. Thus the ciphertext of normal dimension need to tensor with a trivial ciphertext (1,0,…,0) before using key switching.

**FHE.Setup(** $\lambda$ , $L$ **)**: Input the security parameter $\lambda$ and the circuit level $L$, output the noise distribution $\chi$ with $|\chi| < B$ , and the dimension $n_1$, $n_2$. Let $l = \lceil \log q \rceil$, and the noise distribution $\chi$ ensure that the decision-LWE problem with modulus $P=2^l q$ is hard. If there is a trusted source in the system, all parties in the system would the trusted source to generate a uniformly random public matrix $\mathbf{A} \in \mathbb{Z}_q^{n_1 \times n_2}$ . If not, **A** may be generated in the step of key generation and as part of public key.

**FHE.KeyGen**($n_1$, $n_2$, $L$ )：For $i$ =$L$ down to 0, do the following:：

(1) Run $\mathbf{S}'_i$←SecretKeygen( $1^{n_2}$ )   where $\mathbf{S}'_i$ =[**I**|$\mathbf{S}_i$]. Let $sk$={ $\mathbf{S}'_i$ }.

(2) When $i$ =$L$ do this step. Run $\mathbf{B}_L$←PublicKeygen(**A,** $\mathbf{S}'_L$). Let $pk_1$={ $\mathbf{B}_L$ }.

(3) Let $\boldsymbol{s}_j$ be the $j$-th row of the secret key matrix $\mathbf{S}'_i$. Let $\mathbf{ST}_i$ be the matrix that store the tensor vector $\boldsymbol{s}_j \otimes \boldsymbol{s}_j$ as its rows.（Omit this step when $i$=0.）

(4) Run $\mathbf{W}_{i \to i-1}$ ← SwitchKeyGen($\mathbf{S}'_i$, $\mathbf{S}_{i\text{-}1}$).（Omit this step when $i$=0.）Let $pk_2$={ $\mathbf{W}_{i \to i-1}$ } .

Then output $sk$={ $\mathbf{S}'_i$ } and $pk$={$pk_1$， $pk_2$} for $i$∈{0,…$L$}.

**FHE.Enc**($pk_1$, $\boldsymbol{m}$)：Take a message $\boldsymbol{m} \in \mathbb{Z}_2^t$ . Run **Enc**($pk_1$, $\boldsymbol{m}$).

**FHE.Dec**($sk$, $\boldsymbol{c}_i$)：Assume that $\boldsymbol{c}_i$ is a ciphertext under the secret key $\mathbf{S}'_i$. Run **Dec**($sk$, $\boldsymbol{c}_i$).

**FHE.Add**($pk_2$, $\boldsymbol{c}_1$, $\boldsymbol{c}_2$)：Do the following steps.

(1) If ciphertexts $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ has the same secret key $\mathbf{S}'_i$, first compute $\boldsymbol{c}_3$←$\boldsymbol{c}_1$+$\boldsymbol{c}_2$. In order to provide an output that corresponds to the next level key $\mathbf{S}'_{i\text{-}1}$ and rather than $\mathbf{S}'_i$, we call FHE.RefreshNextLevel to do it. Output $\boldsymbol{c}_{\text{add}}$ ← FHE.RefreshNextLevel($i$, $\boldsymbol{c}_3$, $\mathbf{W}_{i \to i-1}$ ) $\in \mathbb{Z}_q^{n_2+t}$ .

(2) If ciphertexts $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ has the different secret key, we choose the ciphertext with higher level and input into FHE.RefreshNextLevel such that the two ciphertexts have the same secret key. We can repeat to call FHE.RefreshNextLevel until the output from FHE.RefreshNextLevel has the same secret key with another ciphertext of lower level. Then go to step (1).

**FHE.Mult**($pk_2$, $\boldsymbol{c}_1$, $\boldsymbol{c}_2$)：Do the following steps.

(1) If ciphertexts $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ has the same secret key $\mathbf{S}'_i$ , first compute $\boldsymbol{c}_3$← $\lfloor \frac{2}{q} \cdot (\boldsymbol{c}_1 \otimes \boldsymbol{c}_2) \rceil$ under the secret key $\mathbf{ST}_i$. Then output $\boldsymbol{c}_{\text{mult}}$← SwitchKey ( $\mathbf{W}_{i \to i-1}$ , $\boldsymbol{c}_3$).

(2) If ciphertexts $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ has the different secret key, what we do as same as the step (2) in FHE.Add($pk_2$, $\boldsymbol{c}_1$, $\boldsymbol{c}_2$).

**FHE.RefreshNextLevel**($i$, $\boldsymbol{c}$, $\mathbf{W}_{i \to i-1}$ ): First compute $\boldsymbol{c}'$ = $\boldsymbol{c} \otimes$ (1,0,…,0), then output SwitchKey( $\mathbf{W}_{i \to i-1}$ ,$\boldsymbol{c}'$ ).

The below lemma 5.1 illustrate the security of the above FHE scheme.

**Lemma 5.1** (security). Let $n_1$, $n_2$, $q$, $\chi$ be some parameters such that decision-LWE problem is hardness. Let $L$ be polynomial depth. Then for any message $\boldsymbol{m} \in \mathbb{Z}_2^t$, if ($pk_1$, $pk_2$, $sk$)←FHE.KeyGen($n_1$, $n_2$, $L$), $\boldsymbol{c}$←FHE.Enc($pk_1$, $\boldsymbol{m}$), it holds that the joint distribution ($pk_1$, $pk_2$, $\boldsymbol{c}$) is computationally indistinguishable from uniform.

**Proof**. Since $pk_2$={ $\mathbf{W}_{L\to L-1}, \mathbf{W}_{L-1\to L-2}, \cdots, \mathbf{W}_{1\to 0}$ } and $pk_1$={ $\mathbf{B}_L$ }, we consider the distribution ($\mathbf{B}_L$, $\mathbf{W}_{L\to L-1}, \mathbf{W}_{L-1\to L-2}, \cdots, \mathbf{W}_{1\to 0}$ , $\boldsymbol{c}$) and apply a hybrid argument as in paper [3]. First, $\mathbf{W}_{1\to 0}$ is indistinguishable from uniform according to the Lemma 4.2. Then all $\mathbf{W}_{i\to i-1}$ can be replaced with uniform in ascending order according to the same argument. At last, the remainder are ($\mathbf{B}_L$, $\boldsymbol{c}$). Since ($\mathbf{B}_L$, $\boldsymbol{c}$) are a public key and ciphertext of the basic encryption described in section 3, ($\mathbf{B}_L$, $\boldsymbol{c}$) are indistinguishable from uniform. Therefore we have that the joint distribution ($pk_1$, $pk_2$, $\boldsymbol{c}$) is computationally indistinguishable from uniform.

## 6. NOISE ANALYSIS

Homomorphic addition and multiplication increase the noise in ciphertexts. In particularly, homomorphic multiplication increases the noise significantly. The analysis for homomorphic addition is simple. That is only the sum of the noise in two ciphertexts. We next analyze the noise growth in homomorphic multiplication.

Suppose ciphertext $\boldsymbol{c}_i$ under the secret key $\mathbf{S}'_L$ is a fresh ciphertext for i∈{1,2}, namely, $\boldsymbol{c}_i$←**FHE.Enc**($pk_1$, $\boldsymbol{m}_i$). By lemma 3.1, we have $\mathbf{S}'_L\ \boldsymbol{c}_i = \left\lfloor \frac{q}{2} \right\rfloor \cdot \boldsymbol{m} + \boldsymbol{e} \pmod q$, where $\|\boldsymbol{e}\|_\infty \le E<(n_1+n_2)B^2+B$. Let $\boldsymbol{c}_{\text{mult}}$ be the output of **FHE.Mult**($pk_2$, $\boldsymbol{c}_1$, $\boldsymbol{c}_2$) under the secret key $\mathbf{S}'_{L-1}$. According to the result in section 4.2 and Lemma 4.1, we have

$$\mathbf{S}'_{L-1} \cdot \boldsymbol{c}_{\text{mult}} = \mathbf{S}'_L \cdot \boldsymbol{c}_3 + \boldsymbol{e}_t \pmod q$$

$$= \left\lfloor \frac{q}{2} \right\rfloor \cdot (\boldsymbol{m}_1 \odot \boldsymbol{m}_2) + \boldsymbol{e}_1^{mult} + \boldsymbol{e}_2^{mult} + \boldsymbol{e}_t \pmod q.$$

According to the analysis in [20,21,22,23], we get $\|\boldsymbol{e}_1^{mult}\|_\infty < 5(n_2+t)BE$, $\|\boldsymbol{e}_2^{mult}\|_\infty < (1/2)(n_2+t)^2B^2$ and $\|\boldsymbol{e}_t\|_\infty < (n_2+t)^2B + (1/2)n_2B$. Putting these together, we get the bound of noise magnitude after once homomorphic multiplication between two fresh ciphertexts such as

$$\|\boldsymbol{e}_1^{mult} + \boldsymbol{e}_2^{mult} + \boldsymbol{e}_t\|_\infty < 5(n_2+t)BE + (1/2)(n_2+t)^2B^2 + (n_2+t)^2B + (1/2)n_2B < 5(n_2+t)BE + 2(n_2+t)^2B^2.$$

After we evaluate a circuit of depth $L$, the upper bound on the noise magnitude in resulting ciphertext is $t_1^L \cdot E + L \cdot t_1^{L-1} \cdot t_2$, where $t_1=5(n_2+t)B$, $t_2=2(n_2+t)^2B^2$. As long as the parameters of this scheme satisfy

$$t_1^L \cdot E + L \cdot t_1^{L-1} \cdot t_2 < \left\lfloor \frac{q}{4} \right\rfloor, \qquad (6.1)$$

we can evaluate homomorphic operation in circuit of depth $L$. For appropriate parameters, we obtain a leveled fully homomorphic encryption scheme.

# 7. PARAMETERS SETTING

In this section, we estimate the concert parameters for our scheme. These parameters include circuit depth $L$, dimension $n$, modulus $q$ and Gaussian parameter $r$. By these parameters, we can obtain concert public key size, secret key size and ciphertext size. Since GHS13 scheme is also a multi-bit FHE scheme and similar with our scheme, we compare these parameters between our scheme and GHS13 scheme.

## 7.1 Parameters Property

Some properties of our scheme and GHS13 scheme are listed in Table 1. All sizes are in bits. The number of LWE sample is $N$=2$n$log$q$ in GHS13 scheme and is $n_1$ in our scheme. We assume the circuit depth is $L$. Thus there is $L$+1 private keys and $L$+1 key switching matrixes. Note that key switching matrixes is viewed as a kind of public key, namely evaluation keys, for evaluation on ciphertext. If one assume circular security, the number of evaluation keys is one rather than $L$+1. But we here do not assume circular security.

We set parameter as $n_1= n_2 = n$ and $t$=$n$ in our scheme so that the two LWE hardness assumptions is equivalent. It is obvious that our public key size is better than it in GHS13 scheme. Specially, our public key size improves a factor log$q$.

## 7.2 Concert parameters

It is a general method to use distinguishing attack to estimate concert parameters of cryptosystem based on LWE. The distinguishing attack means that the adversary distinguishes an LWE instance from uniformly random with some noticeable advantage. The essential of distinguishing attack is to find a short nonzero integral vector in $\Lambda^{\perp}(\boldsymbol{A})$ . According to the result in [19], if one wants to find a short vector of length $\beta$ using state of the art lattice reduction algorithms, the required root-Hermite factor is $\delta = 2^{(\log^2 \beta)/(4n\log q)}$ . The time (in seconds) that it takes to compute a reduced basis with root-Hermite factor $\delta$ for a random LWE instance was estimated in [13] to be at least log(time) ≥1.8/log( $\delta$ ) − 110. Thus a lower-bound on the dimension $n$ required to get any given security level was derived in [18] as

$$n \geq \log(q/r)(\lambda+110)/7.2. \qquad (7.1)$$

Given security level, modulus $q$ and Gaussian parameter $r$, we obtain the minimal values of dimension $n$ to ensure the corresponding security level from inequation (7.1). Some values are presented in Table 2 for $\lambda$=80 and $r$=8 .

**Table 1. Some properties of our scheme and Bra12 scheme**

| | Message<br>Bit | Public Key<br>B | Full Public key<br>B & A | Secret keys | Evaluation keys | Ciphertext |
|---|---|---|---|---|---|---|
| Our scheme | $t$ | $n_1t\log q$ | $n_1(n_2+t)\log q$ | $t(L+1)(n_2+t)$ | $(L+1)(n_2+t)^3\log q$ | $(n_2+t)\log q$ |
| BGH13 | $t$ | $2nt\log^2 q$ | $2n(n+\mathrm{t})\log^2 q$ | $t(L+1)(n+t)\log q$ | $(L+1)(n+t)^3\log^2 q$ | $(n+t)\log q$ |

**Table 2. Minimal values of dimension *n* to ensure $\lambda$ =80 with *r*=8**

| log *q* | 8 | 13 | 22 | 42 | 81 |
|---|---|---|---|---|---|
| *n* | 132 | 264 | 501 | 1029 | 2058 |

**Table 3. The sizes of parameters in our scheme and GHS13 scheme.**

**(a) Our scheme**

| *L* | *n* | log*q* | Public Key | Full Public Key | Evaluation Keys | Secret keys | Ciphertext |
|---|---|---|---|---|---|---|---|
| 0 | 554 | 24 | 900 | 1799 | 3988710 | 1799 | 3.25 |
| 1 | 1082 | 44 | 6287 | 12575 | 108842272 | 25150 | 11.6 |
| 5 | 3351 | 130 | 21240 | 42479 | 834013416 | 127438 | 26 |
| 10 | 6333 | 243 | 1189819 | 2379639 | 663125976563 | 26176025 | 376 |

**(b) GHS13 scheme**

| *L* | *n* | log*q* | Public Key | Full Public Key | Evaluation Keys | Secret key | Ciphertext |
|---|---|---|---|---|---|---|---|
| 0 | 528 | 23 | 35975 | 71950 | 75946719 | 1564 | 3 |
| 1 | 1188 | 48 | 793212 | 1586425 | 7535522461 | 33051 | 14 |
| 5 | 3800 | 147 | 76180166 | 152360332 | 6947631140625 | 3109395 | 136 |
| 10 | 11004 | 420 | 5214983658 | 10429967316 | 3672739157425943 | 198666044 | 1128 |

For a leveled FHE, the circuit depth *L* has to be specified before performing homomorphic operations. In order to evaluate homomorphic operations in circuit of depth *L*, we need to take appropriate modulus *q* according to inequation (6.1) so that noise growth cannot exceed the bound of correct decryption. For GHS13 scheme, even though their scheme is symmetric encryption, it is easy to translate their scheme to asymmetric encryption. In the asymmetric version of GHS13, the modulus *q* needs to satisfy $t_3^L \cdot E' + L \cdot t_3^{L-1} \cdot t_4 < \left\lfloor \frac{q}{4} \right\rfloor$ where $t_3$=4($n$+$t$)log$q$, $t_4$=2($n$+$t$)$^2B$log$^3q$ and the noise of fresh ciphertext $E'$= 2$nB$log$q$.

In Table 3, when the security level is 80 bit, we provide some values for modulus *q* and dimension *n* under the different circuit depth *L*=0, 1, 5, 10. Note that the size of public key, secret key and ciphertext is kilobyte. The data in Table 3 shows that the concert sizes of all parameters in our scheme are smaller than in GHS13 scheme.

## 8. CONCLUSION

The goal of this paper is to construct a multi-bit FHE scheme with short public key from Learning with Errors. The short public key comes from the different style of the basic encryption scheme. We analyze the correctness and give the proof of the security of our scheme. In addition, we optimize the process of key switching and formal this new process of key switching in term of multi-bit FHE. At last, we estimate the concert parameters for our scheme. We compare these parameters between our scheme and BHS13 scheme. Our scheme have public key smaller by a factor of about log*q* than in GHS13 scheme.

## 9. REFERENCES

[1] Craig Gentry. Fully Homomorphic Encryption Using Ideal Lattices [M]. *Proceedings of the 41st annual ACM symposium on Theory of computing*. Bethesda, MD, USA; ACM. 2009: 169-178.

[2] Marten van Dijk, Craig Gentry, Shai Halevi, Vinod Vaikuntanathan. Fully Homomorphic Encryption over the Integers [M]//GILBERT H. *Advances in Cryptology – Eurocrypt 2010*. Springer Berlin / Heidelberg. 2010: 24-43.

[3] Z. Brakerski, V. Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE [M]//OSTROVSKY R. *IEEE 52nd Annual Symposium on Foundations of Computer Science*. Los Alamitos; IEEE Computer Society. 2011: 97-106.

[4] Zvika Brakerski, Craig Gentry, Vinod Vaikuntanathan. (Leveled) Fully Homomorphic Encryption without Bootstrapping [M]. *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. Cambridge, Massachusetts; ACM. 2012: 309-325.

[5] Zvika Brakerski. Fully Homomorphic Encryption without Modulus Switching from Classical Gapsvp [M]//SAFAVI-NAINI R, CANETTI R. *Advances in Cryptology – Crypto 2012*. Springer Berlin Heidelberg. 2012: 868-886.

[6] Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan. On-the-Fly Multiparty Computation on the Cloud Via Multikey Fully Homomorphic Encryption [M]. *Proceedings of the 44th symposium on Theory of Computing*. New York, New York, USA; ACM. 2012: 1219-1234.

[7] Craig Gentry, Amit Sahai, Brent Waters. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based [M]//CANETTI R, GARAY J. *Advances in Cryptology – Crypto 2013*. Springer Berlin Heidelberg. 2013: 75-92.

[8] Craig Gentry, Shai Halevi, Nigel Smart. Fully Homomorphic Encryption with Polylog Overhead [M]//POINTCHEVAL D, JOHANSSON T. *Advances in Cryptology– Eurocrypt 2012*. Springer Berlin / Heidelberg. 2012: 465-482.

[9] Zvika Brakerski, Craig Gentry, Shai Halevi. Packed Ciphertexts in Lwe-Based Homomorphic Encryption [M]//KUROSAWA K, HANAOKA G. *Public-Key Cryptography – PKC 2013*. Springer Berlin Heidelberg. 2013: 1-13.

[10] Jacob Alperin-Sheriff, Chris Peikert. Faster Bootstrapping with Polynomial Error [M]//GARAY J, GENNARO R. *Advances in Cryptology – CRYPTO 2014*. Springer Berlin Heidelberg. 2014: 297-314.

[11] Ryo Hiromasa, Masayuki Abe, Tatsuaki Okamoto. Packing Messages and Optimizing Bootstrapping in GSW-FHE [M]//KATZ J. *Public-Key Cryptography -- PKC 2015*. Springer Berlin Heidelberg. 2015: 699-715.

[12] N. P. Smart, F. Vercauteren. Fully homomorphic SIMD operations [J]. *Designs, Codes and Cryptography*, 2012, 1-25.

[13] Richard Lindner, Chris Peikert. Better Key Sizes (and Attacks) for Lwe-Based Encryption [M]//KIAYIAS A. *Topics in Cryptology – CT-RSA 2011*. Springer Berlin Heidelberg. 2011: 319-339.

[14] Oded Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography [M]. *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*. Baltimore, MD, USA; ACM. 2005: 84-93.

[15] Vadim Lyubashevsky, Chris Peikert, Oded Regev. On Ideal Lattices and Learning with Errors over Rings [M]//GILBERT H. *Advances in Cryptology – Eurocrypt 2010*. Springer Berlin Heidelberg. 2010: 1-23.

[16] Chris Peikert. Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem: Extended Abstract [M]. *Proceedings of the 41st annual ACM symposium on Theory of computing*. Bethesda, MD, USA; ACM. 2009: 333-342.

[17] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, Damien Stehl. Classical hardness of learning with errors [M]. *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*. Palo Alto, California, USA; ACM. 2013: 575-584.

[18] Craig Gentry, Shai Halevi, NigelP Smart. Homomorphic Evaluation of the AES Circuit [M]//SAFAVI-NAINI R, CANETTI R. *Advances in Cryptology – CRYPTO 2012*. Springer Berlin Heidelberg. 2012: 850-867.

[19] Daniele Micciancio, Oded Regev. Lattice-Based Cryptography [M]//BERNSTEIN D, BUCHMANN J, DAHMEN E. *Post-Quantum Cryptography*. Springer Berlin Heidelberg. 2009: 147-191.

[20] Zhigang Chen, Jian Wang, ZengNian Zhang, Xinxia Song. A Fully Homomorphic Encryption Scheme with Better Key Size [J]. *China Communications*, 2014, 11(9): 82-92.

[21] Zhigang Chen, Xinxia Song, Yanhong Zhang. A fully homomorphic encryption scheme based on binary-LWE and analysis of security parameters [J]. *Journal of Sichuan University (Engineering Science Edition)*, 2015, (2): 75-81.

[22] Zhigang Chen, Jian Wang, Liqun Chen, Xinxia Song. Review of how to construct a fully homomorphic encryption scheme [J]. *International Journal of Security and its Applications*, 2014, 8(2): 221-230.

[23] Zhigang Chen, Jian Wang, Liqun Chen, Xinxia Song. A Regev-Type Fully Homomorphic Encryption Scheme Using Modulus Switching [J]. *Scientific World Journal*, 2014.