A General Design Method of Constructing Fully Homomorphic Encryption with Ciphertext Matrix

Xinxia Song[1], Zhigang Chen[2,3]

[1] College of Junior, Zhejiang Wanli University
NingBo 315100, Zhejiang - P.R. China
[e-mail: xinxia.song@foxmail.com]

[2]College of Electronic and Computer, Zhejiang Wanli University
NingBo 315100, Zhejiang –P.R. China

[3]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences
Beijing 100093 - P.R. China
[e-mail: zhig.chen@foxmail.com]

*Corresponding author: Zhigang Chen

Received July 12, 2018; revised September 25, 2018; accepted December 5, 2018; published May 31, 2019

Abstract

It is important to construct fully homomorphic encryption with ciphertext matrix that makes fully homomorphic encryption become very nature and simple. We present a general design method of constructing fully homomorphic encryption whose ciphertext is matrix. By using this design method, we can deduce a fully homomorphic encryption scheme step by step based on a basic encryption scheme. The process of deduction is similar to solving equation and the final output result is a fully homomorphic encryption scheme with ciphertext matrix. The idea of constructing ciphertext matrix is ciphertexts stack, which don't simply stack ciphertexts together but is to obtain the desired homomorphic property. We use decryption structure as tool to analyze homomorphic property and noise growth during homomorphic evaluation. By using this design method, we obtain three corresponding fully homomorphic encryption schemes. Our obtained fully homomorphic encryption schemes are more efficient.

This research was supported by the Natural Science Foundation of Zhejiang Province of China [LY17F020002], Public Projects of Zhejiang Province [2017C33079], NingBo Natural Science Foundation[2017A610120] and the State Key Laboratory of Cryptology.

http://doi.org/10.3837/tiis.2019.05.022 **ISSN : 1976-7277**

Finally, we introduce the adversary advantage and improve the previous method of estimating concert parameters of fully homomorphic encryption. We give the concert parameters of these schemes.

Keywords: Fully Homomorphic Encryption, Ciphertext Matrix, Ciphertexts Stack, Decryption Structure, Concert Parameters.

1. Introduction

Fully homomorphic encryption (FHE) can compute arbitrary function on encrypted data without secret key. Such special property enable FHE to be used in a lot of applications such as private cloud computing. FHE was first proposed by Rivest, Adleman and Dertouzos since 1978 [1], and it had been an open hard problem in cryptography community until the first homomorphic encryption scheme was proposed by Gentry in 2009 [2]. Then some FHE schemes were proposed based on different mathematical hard problem. For example, the scheme is based on prime ideal [3], the schemes are based on integer [4-6] and the schemes are based on Learning with errors (LWE) or its ring variant (RLWE) [7-12].

1.1 Related Work

Among these FHE schemes, LWE-based FHE and RLWE-based FHE are very attractive since these schemes are simple and effective, as well as its security can be reduced to the worst case hardness of standard lattice problems that appear to be resistant to attack by both classical and quantum computers. Specially, Gentry, Sahai, and Waters (GSW) used the approximate eigenvector approach to propose a LWE-based FHE scheme in 2013 [10] whose ciphertext is a square matrix, and thus multiplication of ciphertexts is the multiplication of square matrixes that make homomorphic multiplication become very nature and simple.

Before GSW scheme, since the LWE-based cryptosystem [15] itself supports additive homomorphism, the key point of constructing LWE-based FHE is to achieve multiplicative homomorphism. To construct LWE-based FHE, Brakerski and Vaikuntanathan introduced the critical technique of key switching in [7] to reduce the growth of resulting ciphertext size caused by homomorphic multiplication. However, key switching is expensive. On the one hand, after each homomorphic multiplication, we have to do key switching by multiplying the resulting ciphertext by a matrix generated in the process of key switching. On the other hand, the matrix of key switching is as one part of public key. For homomorphic evaluation with circuit depth L, the public key has to include L matrixes of key switching. So key switching needs a lot of space to store matrixes of key switching and greatly affects the computational efficiency. If the ciphertext is matrix, it does not need key switching. Thus it is important to study how to construct the FHE scheme with ciphertext matrix.

Moreover, GSW has an attractive feature observed in [19] that the noise growth is quasi-additive if we multiply GSW ciphertexts in sequence, which can be used to improve the approximation factor [19, 14] and bootstrapping algorithm [13,23,24]. We think this feature is related to the ciphertext structure that is called as decryption structure later.

1.2 Our Contribution

We present a general design method of constructing FHE whose ciphertext is matrix. By using this design method, we can deduce the FHE scheme step by step based on a basic encryption scheme. The process of deduction is similar to solving equation and the final output result is a FHE scheme. As long as the basic encryption scheme satisfies a condition, we can use this design method to construct FHE scheme based on the basic encryption scheme. For example, LWE-based encryption, RLWE-based encryption, NTRU over RLWE [18] and even Integer-based somewhat homomorphic encryption all satisfy conditions and can be as the basic encryption scheme used to construct corresponding FHE schemes. Thus the design method is general. Our design method reveals the essential of constructing FHE with ciphertext matrix.

By using this design method, we obtain three corresponding FHE schemes based on LWE-based encryption scheme [15], RLWE-based encryption scheme [16,25] and NTRU over RLWE, respectively. We also use this method to construct a packing message FHE scheme from LWE. The result is the same as in [14]. It suffices to show that our design method is general.

It is important to work out how to choose parameters of a FHE scheme to ensure correctness and security against lattice attacks. The performance and efficiency of FHE might be reflected by the size of parameters. In order to obtain the concert parameters of FHE, Gentry et.al applied to the LWE-security analysis of Lindner and Peikert [21] to analyze the dimension needed for different security levels [22]. They also analyzed the concert parameters of the BGV scheme [8]. As far as we know, there is not paper to provide the concert parameters of GSW as well as a concert comparison of GSW and other representative FHE scheme such as Bra [9]. We provide the concert parameters in appendix.

Our techniques: Our design method is based on the observation that we can stack some LWE encryptions to form a ciphertext matrix, namely each row in matrix is a piece of ciphertext vector of LWE. We call this idea as ciphertexts stack. If we make the corresponding encryption scheme has homomorphic property, we thus can obtain a FHE scheme with ciphertext matrix.

The idea of ciphertexts stack plays important role in our design method. It is inspired by the paper [17,20] in which the authors improve Regevs's LWE-based cryptosystem that encrypt one bit at a time to a multi-bit version that encrypt a vector at a time. Their method actually can be view as a type of ciphertexts stack. However ciphertexts stack does not mean that it just simply stack some ciphertext, but rather it needs to keep some structure.

Moreover, decryption structure is as a tool used in our design method to analyze homomorphic property and noise growth. We unify three different decryption structures to one, and we call it the *abstract decryption structure*. Then we derive the decryption structure of homomorphic multiplication from the abstract decryption structure, and we call it *expected decryption structure*. It means that if the decryption structure of the multiplication of ciphertext has same structure as expected decryption structure, homomorphic property would be hold.

To construct a FHE with ciphertext matrix, we assume the ciphertext matrix *C* formed by ciphertexts stack. According to the abstract decryption structure, we can derive the decryption structure with respect to *C* that enables to obtain the expected decryption structure. Namely, this decryption structure would result in homomorphic property. However, the noise growth is large in homomorphic multiplication with respect to this decryption structure. We thus need to adjust repeatedly decryption structure till the *final decryption structure* with respect to ciphertext *C** enable us to obtain not only homomorphic property but also small noise growth during homomorphic operations. Moreover, from the view of ciphertexts stack, we can construct the encryption form of *C** in which the part with respect to plaintext is view as an unknown variable *M*. The corresponding decryption structure of *C** is called as the *virtual decryption structure*. Finally, we establish the equation between the *final decryption structure* and the *virtual decryption structure* about unknown variable *M*. We solve for *M* and eventually obtain concert encryption form of *C**. Thus we achieve a FHE with ciphertext matrix.

We assume the ciphertext is a polynomial, e.g., the ciphertext polynomial is taken from the encryption scheme NTRU over RLWE, our design method also can be applied to construct FHE. The resulting ciphertext would not a matrix but a vector in which each element is a polynomial. Homomorphic multiplication is the product of a matrix and a vector where the matrix is that a ciphertext vector is decomposed as binary representation. The appearance, an original ciphertext is transformed from a vector (e.g.,LWE and RLWE encryption) to a matrix or from a polynomial (e.g, NTRU over RLWE) to a vector, sufficiently show that it is the result of ciphertexts stack. The purpose of ciphertexts stack is to control growth in noise and achieve homomorphic property at the same time.

2. The Design Method of Constructing FHE

2.1 Decryption Structure

For LWE-based encryption scheme, its decryption has the form $\lfloor <\boldsymbol{c},\boldsymbol{s}> \bmod q \rceil \bmod 2$ where $\boldsymbol{c}$ encrypt plaintext bit $m \in \{0,1\}$ under the secret key $\boldsymbol{s}$. Specially, there is an important item in the decryption form, namely the inner product <$\boldsymbol{c}$, $\boldsymbol{s}$> = $\lfloor q/2 \rfloor \cdot m + e$ mod q. It connects the ciphertext $\boldsymbol{c}$ with the corresponding plaintext m and the noise e in some sense, which enables us to analyze clearly homomorphic property and noise growth. We call <$\boldsymbol{c}$, $\boldsymbol{s}$> mod q as decryption structure for LWE-based encryption scheme. It is also hold for RLWE-based encryption scheme. Next we introduce three notions about decryption structure.

At present, for LWE-based encryption scheme, there are three types of decryption structure such as $\lfloor q/2 \rfloor \cdot m + e$ mod q [15] , m +2e mod q [7] and $s \cdot m$+e mod q [18]. Both the first two exist in LWE-based encryption scheme and RLWE-based encryption scheme, while only the last one exists in the NTRU over RLWE where s is a secret polynomial. We unify above three types of decryption structure as one, namely $x \cdot m$+e, which is called as *abstract decryption structure*. Here we denote plaintext and noise by m and e respectively, and we view x and e as unknown variables. The abstract decryption structure can be used to analyze what decryption structure with respect to the resulting ciphertext would result in additive and multiplicative homomorphism.

Suppose two ciphertext $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ encrypt m_1, m_2 with abstract decryption structure xm_1+e_1, x·m_2+e_2 respectively. To achieve additive homomorphism, the decryption structure with respect to adding $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ is required to keep the structure as x·(m_1+m_2) + e^+, where e^+ is the noise in the sum and x is an unknown variable. To achieve multiplicative homomorphism, the decryption structure with respect to multiplying $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ is required to keep the structure as x·(m_1m_2) + $e^\times$, where $e^\times$ is the noise in the result of multiplying $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$. We refer to x·(m_1+m_2) + e^+ or x·(m_1·m_2) + $e^\times$ as *expected decryption structure* for the addition and multiplication of ciphertexts. In other words, if the decryption structure of the resulting ciphertext has the same structure as the expected decryption structure during evaluation, homomorphic property would be hold without considering noise.

To design a FHE scheme with ciphertext matrix, we firstly consider what form of decryption structure that the ciphertext matrix has would enable to obtain homomorphic property.

Suppose a ciphertext matrix $\boldsymbol{C}$ encrypt m under the secret key $\boldsymbol{s}$. From above description, the ciphertext $\boldsymbol{C}$ should have the decryption structure of form $\boldsymbol{C}$·$\boldsymbol{s}$ = x·m+e (mod q) where x and e are two unknown variables. Note that additive homomorphism is obtained obviously, we thus only focus on how to obtain multiplicative homomorphism. For two ciphertexts $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ with decryption structure $\boldsymbol{C}_i$·$\boldsymbol{s}$= x·m_i+e_i (mod q) for i=1, 2, their product has the decryption structure of form $\boldsymbol{C}_1$·$\boldsymbol{C}_2$·$\boldsymbol{s}$=$\boldsymbol{C}_1$·(xm_2+e_1)=$\boldsymbol{C}_1$·x·m_2+$\boldsymbol{C}_1$·e_1 (mod q).

In order to achieve multiplicative homomorphism, the decryption structure of the product of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ need to has the same structure as the expected decryption structure x·(m_1·m_2) + $e^\times$. If we set x = $\boldsymbol{s}$, we have

$$\boldsymbol{C}_1\cdot\boldsymbol{C}_2\cdot\boldsymbol{s}=\boldsymbol{C}_1\cdot(\boldsymbol{s}\cdot m_2+e_1)=\boldsymbol{s}\cdot m_1\cdot m_2+m_2\cdot\boldsymbol{e}_1+\boldsymbol{C}_1\cdot\boldsymbol{e}_2=\boldsymbol{s}\cdot m_1\cdot m_2+e^{\times}\ (\text{mod } q) \qquad (1)$$

where $e^{\times}=m_2\cdot\boldsymbol{e}_1+\boldsymbol{C}_1\cdot\boldsymbol{e}_2$. Namely the decryption structure of the product of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ has the same structure as the expected decryption structure $x\cdot(m_1\cdot m_2)+e^{\times}$. Thus when the ciphertext matrix $\boldsymbol{C}$ has the decryption structure of form $\boldsymbol{C}\cdot\boldsymbol{s}=\boldsymbol{s}\cdot m+e$ (mod q), homomorphic property would be hold without considering noise growth. The encryption corresponding to this decryption structure is called as *zero homomorphic encryption* that is similar to the conception of somewhat homomorphic encryption [2] and can be regarded as the extreme case of somewhat homomorphic encryption.

If we consider noise, the growth in noise mainly depends $\left\|\boldsymbol{C}_1\right\|_\infty$ according to the Equation (1). Thus above decryption structure will result in that even one homomorphic multiplication cannot be performed due to large noise growth. Thus we need to take some measure to suppress growth in noise caused by homomorphic multiplication. For example, we represent the ciphertext matrix $\boldsymbol{C}_1$ as binary, namely BitDecomp($\boldsymbol{C}_1$), to reduce the noise magnitude in the product of ciphertext matrixes. Note that BitDecomp($\boldsymbol{C}_1$) is the matrix formed by applying the operation to each row of $\boldsymbol{C}_1$ separately. Thus homomorphic multiplication is defined as BitDecomp($\boldsymbol{C}_1$)$\cdot\boldsymbol{C}_2$. However, the decryption structure corresponding to BitDecomp($\boldsymbol{C}_1$)$\cdot\boldsymbol{C}_2$, namely BitDecomp($\boldsymbol{C}_1$)$\cdot\boldsymbol{C}_2\cdot\boldsymbol{s}$= BitDecomp($\boldsymbol{C}_1$)$\cdot\boldsymbol{s}\cdot m_2$+BitDecomp($\boldsymbol{C}_1$)$\cdot\boldsymbol{e}_2$ (mod q), is not the same structure as the expected decryption structure. It means that multiplicative homomorphism cannot be achieved. The reason is that BitDecomp($\boldsymbol{C}_1$) need the corresponding secret key Powerof2($\boldsymbol{s}$) rather than $\boldsymbol{s}$.

To achieve multiplicative homomorphism, we adjust the decryption structure of ciphertext matrix $\boldsymbol{C}$ as

$$\boldsymbol{C}\cdot\boldsymbol{s} = \text{Powerof2}(\boldsymbol{s})\cdot m+\boldsymbol{e}\ (\text{mod } q). \qquad (2)$$

It is the *final decryption structure* that enables to achieve not only multiplicative homomorphism but also low noise growth during homomorphic operations. The dimension of $\boldsymbol{C}$ can be obtained by the dimension of $\boldsymbol{s}$. This step actually use ciphertexts stack to adjust decryption structure. That is we insert more LWE ciphertexts into original ciphertext matrix.

2.2 Ciphertexts Stack

LWE encryption has the form $\boldsymbol{c}\leftarrow(m, 0,\ldots\ldots,0)+\boldsymbol{A}^{\mathrm{T}}\cdot\boldsymbol{r}=(m, 0,\ldots\ldots,0)+\boldsymbol{c}_0$ (mod q) , where m is a plaintext bit, $\boldsymbol{r}$ is a random vector and $\boldsymbol{A}=[\boldsymbol{b}|\boldsymbol{A}']$ is a LWE matrix ($\boldsymbol{A}$ is also the public key). Here we denote the encryption of 0 by $\boldsymbol{c}_0$. Note that $\boldsymbol{A}\cdot\boldsymbol{s}=\boldsymbol{b}\text{-}\boldsymbol{A}'\cdot\boldsymbol{s}'=2\boldsymbol{e}'$ where $\boldsymbol{e}'$ is an error vector and $\boldsymbol{s}=(1, \boldsymbol{s}')$ is a secret key vector. Note that RLWE encryption also has the similar form.

The idea of ciphertexts stack is inspired by [17,20] in which they proposed a multi-bit version of Regevs's lattice-based cryptosystem, namely $\boldsymbol{c}\leftarrow(m_1,m_2,\ldots m_t\ ,0\ldots0)+\boldsymbol{A}^{\mathrm{T}}\cdot\boldsymbol{r}$ (mod q)= $(m_1,m_2,\ldots m_t\ , 0,\ldots0)+[b_1,b_2,\ldots b_t\ |\boldsymbol{A}']^{\mathrm{T}}\cdot\boldsymbol{r}$ (mod q). Their idea actually can be viewed as a type of ciphertexts stack.

Since we want to design a FHE scheme whose ciphertext is the matrix, the intuition is that the ciphertext matrix could be formed by stacking some LWE ciphertext vectors together (each row of ciphertext matrix is a LWE ciphertext vector). However, ciphertexts stack don't simply stack these ciphertexts together but need to obtain the expected decryption structure, and thus achieve homomorphic property.

Suppose ciphertext matrix $\boldsymbol{C}$ is formed by stacking some LWE ciphertext vectors. According to LWE encryption form， we have $\boldsymbol{C}\leftarrow\boldsymbol{M}+\boldsymbol{C}_0$ (mod q) where $\boldsymbol{M}$ is viewed as unknown variable with respect to plaintext m and each row in the matrix $\boldsymbol{C}_0$ is the encryption of 0. The decryption structure of the ciphertext matrix $\boldsymbol{C}$ has the form $\boldsymbol{C}\cdot\boldsymbol{s}=\boldsymbol{M}\cdot\boldsymbol{s}+\boldsymbol{C}_0\cdot\boldsymbol{s}$ $=\boldsymbol{M}\cdot\boldsymbol{s}+\boldsymbol{e}$ (mod q), where $\boldsymbol{e}$ is an error vector. This decryption structure is called as the *virtual decryption structure*.

Recall that the final decryption structure of the form (2) enables to achieve homomorphic property and low noise growth at the same time. If above virtual decryption structure has the same structure as the final decryption structure, the corresponding encryption scheme would be a FHE scheme with ciphertext matrix. Thus we establish the equation between virtual decryption structure and expected decryption structure, namely $\boldsymbol{M}\cdot\boldsymbol{s}+\boldsymbol{e}=\text{Powerof2}(\boldsymbol{s})^{\mathrm{T}}\cdot m+\boldsymbol{e}$ (mod q) = $\boldsymbol{G}\cdot\boldsymbol{s}\cdot m+\boldsymbol{e}$ (mod q),where $\boldsymbol{G}=\text{Powerof2}(\boldsymbol{I})^{\mathrm{T}}$. We denote identity matrix by $\boldsymbol{I}$.

Then we solve for unknown variable $\boldsymbol{M}$ from above equation and derive $\boldsymbol{M}=\boldsymbol{G}\cdot m$. Thus we obtain the concert encryption form that is $\boldsymbol{C}\leftarrow\boldsymbol{G}\cdot m+\boldsymbol{C}_0$ (mod q).The decryption is same as LWE decryption. We choose the appropriate row from $\boldsymbol{C}$ which corresponds to a LWE ciphertext, and decrypt it. The corresponding encryption scheme is a FHE scheme with ciphertext matrix.

2.3 The Design Method

At present, all of FHE schemes are built on some known encryption scheme that we call it the *basic encryption scheme*. The encryption form in the basic encryption scheme is required to meet a condition when we apply below design method to construct a FHE scheme. The condition is that the encryption has the form $\boldsymbol{c}\leftarrow m+\boldsymbol{c}_0$, where m is the plaintext and $\boldsymbol{c}_0$ is encryption of 0. All of encryption schemes such as LWE encryption [15], ring LWE encryption [16], NTRU encryption over ring LWE [18] as well as DGHV basic encryption scheme [4] meet this condition. It means that the corresponding FHE scheme with ciphertext matrix can be achieved by applying the design method on these basic encryption schemes.

Let ciphertext $\boldsymbol{C}$ be a matrix or polynomial. In the case of ciphertext matrix, we need to construct it from the basic encryption scheme. We assume that ciphertext matrix $\boldsymbol{C}$ is formed by stacking up a certain number of ciphertext vectors that encrypt plaintext m under secret key $\boldsymbol{s}$ using the basic encryption scheme (e.g, LWE-based encryption scheme). That is, each row in ciphertext matrix $\boldsymbol{C}$ is a piece of ciphertext vector. The decryption is same as the basic encryption scheme, i.e., the secret key $\boldsymbol{s}$ of the basic encryption scheme is used to decrypt some row in ciphertext matrix. Thus the ciphertext matrix $\boldsymbol{C}$ is the encryption of plaintext m under the secret key $\boldsymbol{s}$. The corresponding decryption structure is $\boldsymbol{C}\cdot\boldsymbol{s}$.

In the case of ciphertext polynomial, we do not need to construct it as there is the known NTRU encryption scheme based on ring LWE whose ciphertext is a polynomial. The corresponding decryption structure also has the form $\boldsymbol{C}\cdot\boldsymbol{s}$ where $\boldsymbol{C}$ is a ciphertext polynomial and $\boldsymbol{s}$ is a secret key polynomial. Note that, in this case, we can skip step 1 and start directly from step 2 to construct FHE using below design method. The design method is described as follows.

Step 1. Establish decryption structure of the ciphertext $\boldsymbol{C}$ that enable to achieve additive and multiplicative homomorphism without considering noise growth. From the session 2.1, the decryption structure of the ciphertext $\boldsymbol{C}$ should has the form $\boldsymbol{C}\cdot\boldsymbol{s} =\boldsymbol{s}\cdot m+e$ (mod q), where e is a noise variable. The dimension of $\boldsymbol{C}$ can be obtained by the dimension of $\boldsymbol{s}$. This form of decryption structure enables us to achieve potentially homomorphic property without considering noise growth. The encryption scheme with respect to this decryption structure is *zero homomorphic encryption* that is obtained by jumping directly to the step 3 and step 4.

Step 2. Adjust decryption structure, and output the *final decryption structure* that enable to achieve simultaneously homomorphic property and low noise growth during homomorphic evaluation. From the section 2.1, the *final decryption structure* is derived as $\boldsymbol{C}^*\cdot\boldsymbol{s}$ = Powerof2($\boldsymbol{s}$)$^{\mathrm{T}}\cdot m+\boldsymbol{e}$ = $\boldsymbol{G}\cdot\boldsymbol{s}\cdot m+e$(mod q) where e is a noise variable and $\boldsymbol{G}$=Powerof2($\boldsymbol{I}$)$^{\mathrm{T}}$. The dimension of $\boldsymbol{C}$* and $\boldsymbol{G}$ can be obtained by the dimension of $\boldsymbol{s}$. Note that $\boldsymbol{C}^*$is the expansion of $\boldsymbol{C}$ by inserting a certain number of ciphertexts into $\boldsymbol{C}$**.** If $\boldsymbol{C}$ is a matrix, $\boldsymbol{C}^*$is also a matrix. If $\boldsymbol{C}$ is a polynomial, $\boldsymbol{C}^*$ would be a vector. Since a vector can also be seen as a matrix, $\boldsymbol{C}^*$is viewed as a matrix in the later whether it is a matrix or vector.

Step 3. Construct the form of ciphertext matrix $\boldsymbol{C}^*$ by using ciphertexts stack. According to the ciphertext form of the basic encryption scheme, the ciphertext matrix $\boldsymbol{C}$***** can be represented as $\boldsymbol{C}$*****$\leftarrow\boldsymbol{M}+\boldsymbol{C}_0$(mod q) by using ciphertexts satck, where the matrix $\boldsymbol{M}$ is seen as an unknown variable with respect to plaintext m and $\boldsymbol{C}_0$ is a matrix in which each row is an encryption of 0 produced by the basic encryption scheme. The decryption structure of $\boldsymbol{C}$***** has the form $\boldsymbol{C}$*****$\cdot\boldsymbol{s}\leftarrow\boldsymbol{M}\cdot\boldsymbol{s}+e$ (mod q) where e is a noise variable. This decryption structure is also called the *virtual decryption structure*. If we get the concert form of $\boldsymbol{M}$, we can obtain the concert encryption algorithm.

Step 4. Establish an equation about $\boldsymbol{M}$ between the *virtual decryption structure* and the *final decryption structure*, namely $\boldsymbol{M}\cdot\boldsymbol{s}+\boldsymbol{e}$ = $\boldsymbol{G}\cdot\boldsymbol{s}\cdot m+e$ (mod q). We derive $\boldsymbol{M}$=$\boldsymbol{G}\cdot m$ from this equation. Thus the encryption is obtained as $\boldsymbol{C^*}\leftarrow\boldsymbol{G}\cdot m + \mathbf{C}_0 \,(\mathrm{mod}\ q)$.

Step 5. Decryption is the same as Regev's decryption procedure that is applied to one row of $\boldsymbol{C^*}$. Let $\boldsymbol{c}_{l\text{-}1}$ be the l-1th row of $\boldsymbol{C}$, namely the coefficient of the plaintext m is $2^{l\text{-}1}$ where $l=\lceil \log q \rceil -1$. Output $\lfloor \frac{1}{2^{l-1}} < \boldsymbol{c}_{l-1}, \boldsymbol{s} > \bmod q \rceil \bmod 2$.

For the security of the scheme outputted by this design method, on the one hand, $\boldsymbol{G}$ is the "primitive matrix". On the other hand, each row of $\boldsymbol{C^*}$ is a LWE encryption of plaintext bit m, thus the security can be obtained from the security of Regev's encryption scheme. For each scheme obtained by this design method, we give the proof of security in detail in the later.

3. A RLWE-based FHE Scheme with Ciphertext Matrix

The basic encryption scheme that this FHE scheme built on is the RLWE-based encryption scheme [16]. In the ring LWE encryption scheme, the secret key is a 2-dimensional vector $\boldsymbol{s}$=(1,-s') where s' is a polynomial over $\boldsymbol{R}$ and is sampled uniformly from the error distribution. To generate the public key, choose a uniformly random element $a\in R_q$ and a uniformly random small elements $e\in\boldsymbol{R}$ from the error distribution, and output the public key $\boldsymbol{b}$'=(b=$a\cdot s'$+e, a)$\in R_q\times R_q$. To encrypt an n-bit message $\boldsymbol{t}\in\{0,1\}^n$, we use its bits as the 0-1 coefficients of a polynomial $m\in R_2$. The encryption algorithm then chooses three random small elements r, e_1, $e_2\in R$ from the error distribution and outputs $\boldsymbol{c}\leftarrow(\lfloor q/2 \rfloor\cdot m$+$br$+$e_1$, ar+e_2)=($\lfloor q/2 \rfloor\cdot m$, 0)+$\boldsymbol{b'}\cdot r$+$\boldsymbol{e'}$ = $\boldsymbol{m}$+$\boldsymbol{c}_0\in R_q\times R_q$, where $\boldsymbol{e'}$=(e_1,e_2) and $\boldsymbol{m}$=($\lfloor q/2 \rfloor\cdot m$, 0) and $\boldsymbol{c}_0$=$\boldsymbol{b'}\cdot r$+$\boldsymbol{e'}$. Thus the decryption structure is $\boldsymbol{c}\cdot\boldsymbol{s}$=$\boldsymbol{m}\cdot\boldsymbol{s}$ +$\boldsymbol{c}_0\cdot\boldsymbol{s}$=$\lfloor q/2 \rfloor\cdot m$+ $r\cdot e$+e_1-$s'\cdot e_2$. As long as the coefficients of $r\cdot e$+e_1-$s'\cdot e_2$ have magnitudes less than q/4, the message can be recovered by $\lfloor \frac{2}{q}[<\boldsymbol{c},\boldsymbol{s}>]_q \rceil \bmod 2$. The ring LWE encryption obviously satisfies the condition as the basic encryption scheme to construct FHE scheme using the design method. Next we firstly explain how to construct this FHE scheme using the design pattern, and then we give this FHE scheme.

3.1 Using Design Method to Construct A RLWE-Based FHE Scheme with Ciphertext Matrix

Assume that the ciphertext $\boldsymbol{C}$ is a matrix in which each row is an encryption produced by the RLWE-based encryption scheme under the secret key $\boldsymbol{s}$=(1, -s') where $s'\in\boldsymbol{R}_q$.

In step 1, we can obtain the decryption structure of form $\boldsymbol{C}\cdot\boldsymbol{s}=\boldsymbol{s}\cdot m+e=\begin{bmatrix}1\\ s'\end{bmatrix}\cdot m+e \pmod q$, where e is a noise variable. Here we can deduce that e is a 2-dimensional vector and the ciphertext $\boldsymbol{C}$ is a 2×2 matrix.

By step 2, the final decryption structure is outputted and has the form

$$\boldsymbol{C}^{*}\cdot\boldsymbol{s}=\boldsymbol{G}\cdot\boldsymbol{s}\cdot m+e=\begin{bmatrix}1\\ 2\\ \vdots\\ 2^{l}\\ s'\\ 2s'\\ \vdots\\ 2^{l}s'\end{bmatrix}\cdot m+e \pmod q \qquad (3)$$

where e is a noise variable and $l=\lceil \log q\rceil-1$.We can deduce that e is a $2(l+1)$-dimensional vector and $\boldsymbol{C}^{*}$are $2(l+1)\times 2$ matrix.

By step 3 and step 4, we obtain the encryption that has the form of

$$\boldsymbol{C}^{*}\leftarrow\boldsymbol{G}m+\boldsymbol{C}_{0}=\begin{bmatrix}1 & 0\\ 2 & 0\\ \vdots & \vdots\\ 2^{l} & 0\\ 0 & 1\\ 0 & 2\\ \vdots & \vdots\\ 0 & 2^{l}\end{bmatrix}m+\boldsymbol{C}_{0} \pmod q. \qquad (4)$$

This FHE scheme is given as follows in detail.

- **RLFHE.Setup**(λ , L):Input the security parameter λ and the circuit level L. Choose a prime integer modulus $q(\lambda)\geq 2$ and a dimension parameter $n(\lambda)\geq 1$ which is a power of two. Let $\phi(x)=x^n+1$ be the n^{th} cyclotomic polynomial. Let $R=\mathbb{Z}[x]/\phi(x)$ and $R_q=\mathbb{Z}_q[x]/\phi(x)$. Let χ be the *B*-bounded discrete Gaussian distribution over the ring *R*. Let $l=\lceil \log q \rceil -1$. Output *params* = (*n*, *q*, *f*(*x*), χ).

- **RLFHE.SecretKeyGen**(*params*): Sample $s' \leftarrow \chi$. Set sk= $\boldsymbol{s}$ = (1, -*s*')$\in R_q\times R_q$.
- **RLFHE.PublicKeyGen**(*params*,sk): Sample $a\leftarrow R_q$ and $e\leftarrow \chi$. Compute $b=a\cdot s'+e$. Set pk= $\boldsymbol{b'}$=(*b*, *a*)$\in R_q\times R_q$.
- **RLFHE.Enc**(*params*, pk, *m*):To encrypt an *n*-bit message in $\{0,1\}^n$, we use its bits as the 0-1 coefficients of a polynomial $m\in R_2$. Sample $\boldsymbol{R}\leftarrow \chi^{2(l+1)\times 1}$ and $\boldsymbol{E}\leftarrow \chi^{2(l+1)\times 2}$, where $\boldsymbol{R}$ is a 2(*l*+1)×1matrix and $\boldsymbol{E}$ is a 2(*l*+1)×2 matrix in which each entry is sampled from the discrete Gaussian distribution χ. Output the ciphertext：

$$\boldsymbol{C}\leftarrow \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ \vdots & \vdots \\ 2^l & 0 \\ 0 & 1 \\ 0 & 2 \\ \vdots & \vdots \\ 0 & 2^l \end{bmatrix} m+\boldsymbol{Rb'}+\boldsymbol{E}\in R_q^{2(l+1)\times 2}. \qquad (5)$$

- **RLFHE.Dec**(sk,$\boldsymbol{C}$): Let $\boldsymbol{c}_l$ be the *l*-th row of $\boldsymbol{C}$, namely the coefficient of the plaintext*m* is 2^{l-1}. Output $\lfloor \frac{1}{2^{l-1}} < \boldsymbol{c}_l, \boldsymbol{s} > \bmod q \rceil \bmod 2$.
- **RLFHE.Add**($\boldsymbol{C}_1$, $\boldsymbol{C}_2$): Output $\boldsymbol{C}_1+\boldsymbol{C}_2\in R_q^{2(l+1)\times 2}$.
- **RLFHE.Mult**($\boldsymbol{C}_1$, $\boldsymbol{C}_2$): OutputBitDecomp($\boldsymbol{C}_1$)$\cdot\boldsymbol{C}_2\in R_q^{2(l+1)\times 2}$.

Lemma 3.1 (security). Let *params* = (*n*, *q*, *f*(*x*), χ) be such that the ring LWE assumption holds. Then for any $m\in R_2$ ， if $\boldsymbol{s}\leftarrow$**RLFHE.SecretKeyGen**(*params*), $\boldsymbol{b'}\leftarrow$**RLFHE.PublicKeyGen**(*params*, sk)，$\boldsymbol{C}\leftarrow$**RLFHE.Enc**(*params*, pk,*m*), its holds that the joint distribution (***b'***, ***C***) is computationally indistinguishable from uniform over $R_q^2 \times R_q^{2(l+1)\times 2}$.

Proof. The security of above scheme includes two parts. One part is that we need to prove the public key is indistinguishable from uniform over $R_q \times R_q$. Another part is that we need to prove the ciphertext matrix is indistinguishable from uniform over $R_q^{2(l+1)\times 2}$. For the first part, since the public key ***b'***=(b, a) is a ring LWE instance, the public key is indistinguishable from uniform over $R_q \times R_q$ under the ring LWE assumption. For the second part, since each row in ciphertext matrix is a ciphertext produced by the ring LWE encryption scheme, the ciphertext matrix is indistinguishable from uniform over $R_q^{2(l+1)\times 2}$ under the ring LWE assumption. Therefore, the joint distribution (***b'***, ***C***) is computationally indistinguishable from uniform over $R_q^2 \times R_q^{2(l+1)\times 2}$.

3.2 Analysis of Noise

Below we analyze noise growth to show that above scheme is a leveled FHE scheme. We firstly analyze the noise magnitude at encryption and decryption and then analyze noise growth during homomorphic operations.

3.2.1 Encryption Noise and Decryption Noise

Lemma 4.2 (encryption noise). Let *params* = (n, q, $f(x)$, $|\chi| \le B$) be parameters for the above scheme. Sample $s' \leftarrow \chi$. Set ***s***←(1, $-s'$). Let $m \in R_2$ be any polynomial. Set ***b'***←**RLFHE**.**PublicKeyGen**(*params*, sk) and ***C***←**RLFHE**.**Enc**(*params*, pk, m). Then for some $\boldsymbol{e} \in R_q^{2(l+1)}$ with $\|\boldsymbol{e}\|_\infty \le 2nB^2 + B$, it holds that $\boldsymbol{C}\cdot\boldsymbol{s} = m\cdot$ Powerof2($\boldsymbol{s}$)+$\boldsymbol{e}$ (*mod q*).We call ***e*** noise in ciphertext ***C***.

Proof. By definition

$$\boldsymbol{C}\cdot\boldsymbol{s} = m\cdot \text{Powerof2}(\boldsymbol{s}) + \boldsymbol{R}\cdot\boldsymbol{b'}\cdot\boldsymbol{s} + \boldsymbol{E}\cdot\boldsymbol{s}\ (mod\ q) \qquad (6)$$
$$= m\cdot \text{Powerof2}(\boldsymbol{s}) + \boldsymbol{R}\cdot e + \boldsymbol{E}\cdot\boldsymbol{s}\ (mod\ q)$$
$$= m\cdot \text{Powerof2}(\boldsymbol{s}) + \boldsymbol{e}\ (mod\ q).$$

Since $|\chi| \le B$, we have $\|\boldsymbol{e}\|_\infty = \|\boldsymbol{R}\cdot e + \boldsymbol{E}\cdot\boldsymbol{s}\|_\infty \le \|\boldsymbol{R}\cdot e\|_\infty + \|\boldsymbol{E}\cdot\boldsymbol{s}\|_\infty \le nB^2+B+nB^2 = 2nB^2+B$ and the lemma follows.

The above lemma means that the fresh ciphertext, namely the ciphertext is produced by the encryption and not the homomorphic operations, has the noise magnitude at most $2nB^2$

$+B$.

Lemma 4.3 (decryption noise). Let χ be the B-bounded discrete Gaussian distribution over the ring R. Sample $s' \leftarrow \chi$. Set $\boldsymbol{s} \leftarrow (1, -s')$. Let $\boldsymbol{C} \in R_q^{2(l+1)\times 2}$ be such that

$$\boldsymbol{C} \cdot \boldsymbol{s} = m \cdot \text{Powerof2}(\boldsymbol{s}) + \boldsymbol{e}\ (mod\ q),$$

with $m \in R_2$ and $\|\boldsymbol{e}\|_\infty < q/8$. Then

$$m \leftarrow \textbf{RLFHE}.\textbf{Dec}(\boldsymbol{s},\boldsymbol{C}). \qquad (7)$$

Proof. Let $\boldsymbol{c}_{l\text{-}1}$ be the l-1th row of $\boldsymbol{C}$. Then we have $<\boldsymbol{c}_{l\text{-}1}, \boldsymbol{s}>=m \cdot 2^{l\text{-}1}+e(mod\ q)$ where $|e|<q/8$. Since $q/4<2^{l\text{-}1}<q/2$, then $\|e/2^{l-1}\|_\infty <1/2$. Therefore we have $m \leftarrow \lfloor <\boldsymbol{c}_{l-1}, \boldsymbol{s}> \bmod q/2^{l-1} \rceil$, namely $m \leftarrow \textbf{RLFHE}.\textbf{Dec}(\boldsymbol{s},\boldsymbol{C})$.

Above lemma means that the correctness of decryption is guaranteed as long as the noise in ciphertext matrix $\boldsymbol{C}$ has magnitude at most $q/8$.

3.2.2 Analysis of Noise Growth

Homomorphic addition and multiplication increase the noise in ciphertext. Since noises grow slightly with homomorphic additions and substantially with homomorphic multiplications, we just focus on the analysis of noise growth in homomorphic multiplication.

Suppose $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ encrypt m_1 and $m_2 \in R_2$ under the secret key $\boldsymbol{s}$ respectively. It holds that $\boldsymbol{C}_i \cdot \boldsymbol{s} = m_i \cdot \text{Powerof2}(\boldsymbol{s})+\boldsymbol{e}_i$ for $i \in \{1,2\}$ where $\|\boldsymbol{e}_i\|_\infty \le \beta = 2nB^2+B$. Let $\boldsymbol{C}^\times = \text{BitDecomp}(\boldsymbol{C}_1) \cdot \boldsymbol{C}_2$, namely $\boldsymbol{C}^\times$ is the homomorphic multiplication of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. We have

$$\begin{aligned}\boldsymbol{C}^\times \cdot \boldsymbol{s} &= \text{BitDecomp}(\boldsymbol{C}_1) \cdot \boldsymbol{C}_2 \cdot \boldsymbol{s}\\ &= \text{BitDecomp}(\boldsymbol{C}_1) \cdot (m_2 \cdot \text{Powerof2}(\boldsymbol{s}) + \boldsymbol{e}_2)\\ &= m_1 m_2 \cdot \text{Powerof2}(\boldsymbol{s}) + m_2 \cdot \boldsymbol{e}_1 + \text{BitDecomp}(\boldsymbol{C}_1) \cdot \boldsymbol{e}_2\\ &= m_1 m_2 \cdot \text{Powerof2}(\boldsymbol{s}) + \boldsymbol{e}^\times.\end{aligned}$$

Since $\|\boldsymbol{e}_i\|_\infty \le \beta$, we have $\|\boldsymbol{e}^\times\|_\infty \le 2n(l+1)\beta + n\beta$. Set $N=n(l+1)$, then $\|\boldsymbol{e}^\times\|_\infty \le (2N+n)\beta$. It is only the noise caused by one homomorphic multiplication of two fresh ciphertexts. After evaluating depth-L circuit (L levels of multiplication), the noise grows to at most $(2N+n)^L \cdot \beta \sim (2n\log q)^L \cdot \beta$. It means that in order to guarantee correct decryption the final noise magnitude is required below $q/8$.

For security, the best known algorithm for LWE runs in time approximately $2^{n/\log(q/B)}$. This result also holds for ring LWE. Therefore we choose B to be polynomial in n and $q = 2^{n^\varepsilon}$ for every $\varepsilon<1$, we can derive $L \approx \log q \approx n^\varepsilon$ from $(nN+1)^L \cdot \beta < q/8$. It means that we could homomorphically evaluate a circuit of polynomial depth using above scheme from ring LWE. Thus above scheme is a leveled FHE scheme.

4. A NTRU-Type FHE with ciphertext matrix

When we consider the ciphertext is a polynomial, there is a known encryption scheme, a NTRU scheme from ring LWE in [18], whose ciphertext is a polynomial. We take this NTRU scheme as the basic encryption scheme and we construct FHE scheme based on it. In this NTRU basic encryption scheme, the secret key $f=2f'+1$ is invertible in R_q where f' is sampled uniformly from the error distribution, and the public key is $h=2g\cdot f^{-1}\in R_q$ where g is sampled uniformly from the error distribution. To encrypt a message $m\in R_2$, sample s, e from the error distribution and output the ciphertext $c\leftarrow m+h\cdot s+2e\in R_q$. The decryption is $m\leftarrow c\cdot f \bmod q \bmod 2 \in R_2$. Thus the decryption structure has the form $c\cdot f= m\cdot f +2g\cdot s+2e\cdot f\in R_q$ where $g\cdot s+e\cdot f$ is called as the noise in ciphertext. As long as the noise is below $q/4$, correct decryption is guaranteed. It is obvious that the NTRU basic encryption scheme meet the condition of constructing FHE scheme by using design method.

4.1 Using Design Method to Construct A NTRU-Type FHE Scheme

From the decryption structure of the NTRU basic encryption scheme, we know that this scheme is a *zero homomorphic encryption* by nature. Thus we start directly from step 2 to construct FHE by using design method.

In step 2, the final decryption structure is derived as $\boldsymbol{C}^*\cdot f=$ Powerof2$(f)^{\mathrm{T}}\cdot m+e(\bmod q)$ where f is the secret key and e is a noise variable. It is obvious that $\boldsymbol{C}^*$ is a vector of length $l+1=\lceil \log q \rceil$. By step 3 and step 4, we obtain the encryption of the form $\boldsymbol{C}^*\leftarrow \boldsymbol{G}m+\boldsymbol{C}_0=$Powerof2$(1)^{\mathrm{T}}\cdot m+\boldsymbol{C}_0=\begin{bmatrix}1\\2\\\vdots\\2^l\end{bmatrix}\cdot m+\boldsymbol{C}_0 (\bmod q)$. This FHE scheme is omitted in detail.

5. A Packing Messages FHE from LWE

By using the design method, we can construct a FHE scheme that encrypts a plaintext matrix instead of a plaintext bit, namely multiple plaintexts are packed into one ciphertext. This packing message FHE scheme obtained by using design method is the same as in [14]. It suffices to show that this design method is general.

The idea of constructing a packing message FHE scheme by using design method is to stack up a number of encryptions of plaintext vectors instead of a number of encryptions of plaintext bits. We first recall the basic encryption scheme that the packing message FHE scheme is based on. To encrypt a plaintext vector $\boldsymbol{m}$=(m_1,m_2,…m_t) with length t where $m_i \in \{0,1\}$, according to the encryption schemes proposed in [17,20], a matrix $\boldsymbol{A}' \in \mathbb{Z}_q^{m\times n}$ is chosen uniformly , and an error vector $\boldsymbol{e}_i$ is chosen from a Gaussian distribution, and $\boldsymbol{s}_i \in \mathbb{Z}_q^n$ is chosen uniformly at random for $1\leq i\leq t$. Let $\boldsymbol{S}'$=[-$\boldsymbol{s}_1$,-$\boldsymbol{s}_2$,…,-$\boldsymbol{s}_t$] be a matrix whose each column is the vector $\boldsymbol{s}_i$. The secret key is $\boldsymbol{S}=\left[\frac{I}{-S'}\right]$ where I denote an $t\times t$ identity matrix. Set $\boldsymbol{b}_i=\boldsymbol{A}'\boldsymbol{s}_i+\boldsymbol{e}_i \in \mathbb{Z}_q^m$ for $1\leq i\leq t$. The public key is [$\boldsymbol{b}_1$,$\boldsymbol{b}_2$,…$\boldsymbol{b}_t$ |A']. Sample a vector $\boldsymbol{r} \in \{0,1\}^m$, the encryption of a plaintext vector $\boldsymbol{m}$ under the secret key $\boldsymbol{S}$ is $\boldsymbol{c}\leftarrow(\lfloor q/2 \rfloor \cdot \boldsymbol{m}$, 0,…0)+[$\boldsymbol{b}_1$,$\boldsymbol{b}_2$,…$\boldsymbol{b}_t$ |A']$^{\mathrm{T}}\boldsymbol{r}$ (mod q). The corresponding decryption is $\lfloor \frac{2}{q}[\boldsymbol{c}\cdot\boldsymbol{S}]_q \rceil \bmod 2$. If we stack up a number of these ciphertexts produced by the above basic encryption to form a ciphertext matrix, it is possible to construct a packing message FHE scheme with ciphertext matrix. The intuition is that the corresponding plaintext may be a matrix. Below we describe how to use design method to deduce a packing message FHE scheme with ciphertext matrix.

Let $\boldsymbol{C}$ be a $(n+t)\times(n+t)$ square matrix. Suppose each row of ciphertext matrix $\boldsymbol{C}$ is a ciphertext produced by the above basic encryption scheme under the $(n+t)\times t$ secret matrix $\boldsymbol{S}$. From the step 1 described in design method, we can obtain the decryption structure $\boldsymbol{C}\cdot\boldsymbol{S}=\boldsymbol{S}\cdot m+e$ (mod q) where e is a noise variable. We can deduce that m is a $t\times t$ square matrix and e denote a $(n+t)\times t$ matrix. From the step 2, the final decryption structure has the form of $\boldsymbol{C}^{*}\boldsymbol{\cdot}\boldsymbol{S}$=Powerof2($\boldsymbol{S}$) $\cdot m+e=\boldsymbol{G}\cdot\boldsymbol{S}\cdot m+e^*$ (mod q). We can deduce that $\boldsymbol{C}^*$ is a $(n+t)(l+1)\times(n+t)$ matrix and e^* denote a $(n+t)(l+1)\times t$ matrix. According to the step 3, we construct ciphertext matrix $\boldsymbol{C}^*$ by using ciphertexts stack and obtain the corresponding encryption form that is $\boldsymbol{C}^*\leftarrow\boldsymbol{M}+\boldsymbol{C}_0$ (mod q) where the unknown variable $\boldsymbol{M}$ denote a $(n+t)(l+1)\times(n+t)$ matrix with respect to plaintext and $\boldsymbol{C}_0$ is a $(n+t)(l+1)\times(n+t)$ matrix with respect to encryptions of 0. Thus the decryption structure of $\boldsymbol{C}^*$, namely virtual decryption structure, is $\boldsymbol{C}^*\boldsymbol{\cdot}\boldsymbol{S}\leftarrow\boldsymbol{M}\cdot\boldsymbol{S}+\boldsymbol{C}_0\cdot\boldsymbol{S}=\boldsymbol{M}\cdot\boldsymbol{S}+e^*$(mod q) where e^* is a noise variable.

Then we establish an equation between the virtual decryption structure and the final decryption structure, namely $\boldsymbol{M}\cdot\boldsymbol{S}+e^*=\boldsymbol{G}\cdot\boldsymbol{S}\cdot m+e^*$(mod q).We can solve for $\boldsymbol{M}$ from above equation and have $\boldsymbol{M}\cdot\boldsymbol{S}=\boldsymbol{G}\cdot\boldsymbol{S}\cdot m$. Since $\boldsymbol{M}\cdot\boldsymbol{S}=M\cdot\left[\frac{I}{-S'}\right]$, then have $\boldsymbol{M}$=[$\boldsymbol{G}\cdot\boldsymbol{S}\cdot m$|0]. Therefore the concert encryption form is obtained, namely $\boldsymbol{C}^*\leftarrow$[$\boldsymbol{G}\cdot\boldsymbol{S}\cdot m$| 0]+$\boldsymbol{C}_0$(mod q). The decryption is the same as the basic encryption scheme. Note that the result $\boldsymbol{M}$=[$\boldsymbol{G}\cdot\boldsymbol{S}\cdot m|0$] may be just one possible answer for the equation $\boldsymbol{M}\cdot\boldsymbol{S}=\boldsymbol{G}\cdot\boldsymbol{S}\cdot m$. We don't know whether there are other answers.

6. Conclusion

We present a general design method of constructing FHE whose ciphertext is matrix. By using this design method, we can deduce the FHE scheme step by step based on a basic encryption scheme. The process of deduction is similar to solving equation and the final output result is a FHE scheme.

By using this design method, we obtain three corresponding FHE schemes. Our obtained FHE schemes are more efficient than GSW. In addition, we also use this method to construct a packing message FHE scheme from LWE. The result is the same as in [14]. It suffices to show that our design method is general.

Reference

[1] R.L. Rivest, L. Adleman, and M.L. Dertouzos, "On Data Banks and Privacy Homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169-180, 1978.

[2] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," *in Proc. of 41st annual ACM symposium on Theory of computing*, pp. 169-178, 2009. Article (CrossRef Link).

[3] N.P. SmartF. Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes," in *Proc. of International Workshop on Public Key Cryptography*, pp. 420-443, 2010. Article (CrossRef Link).

[4] M. van Dijk, C. Gentry, S. Halevi and et al., "Fully Homomorphic Encryption over the Integers," in *Proc. of Advances in Cryptology – Eurocrypt 2010*, pp. 24-43, 2010. Article (CrossRef Link).

[5] J.-S. Coron, A. Mandal, D. Naccache and et al., "Fully Homomorphic Encryption over the Integers with Shorter Public Keys," in *Proc. of Advances in Cryptology – Crypto 2011*, pp. 487-504, 2011. Article (CrossRef Link).

[6] J.-S. Coron, D. Naccache, and M. Tibouchi, "Public Key Compression and Modulus Switching for Fully Homomorphic Encryption over the Integers," in *Proc. of Advances in Cryptology – Eurocrypt 2012*, pp. 446-464, 2012. Article (CrossRef Link).

[7] Z. BrakerskiV. Vaikuntanathan, "Efficient Fully Homomorphic Encryption from (Standard) Lwe," in *Proc. of 52nd Annual Symposium on Foundations of Computer Science*, pp. 97-106, 2011.

[8] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) Fully Homomorphic Encryption without Bootstrapping," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, pp. 13, 2014.

[9] Z. Brakerski, “Fully Homomorphic Encryption without Modulus Switching from Classical Gapsvp,” *in Proc. of Advances in Cryptology – Crypto 2012*, pp. 868-886, 2012. Article (CrossRef Link).

[10] C. Gentry, A. Sahai, and B. Waters, “Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based,” pp. 75-92, 2013.

[11] A. López-Alt, E. Tromer, and V. Vaikuntanathan, “On-the-Fly Multiparty Computation on the Cloud Via Multikey Fully Homomorphic Encryption,” in *Proc. of 44th symposium on Theory of Computing*, pp. 1219-1234, 2012. Article (CrossRef Link).

[12] Z. Chen, J. Wang, Z. Zhang and et al., “A Fully Homomorphic Encryption Scheme with Better Key Size,” *China Communications*, vol. 11, no. 9, pp. 82-92, 2014. Article (CrossRef Link).

[13] J. Alperin-SheriffC. Peikert, “Faster Bootstrapping with Polynomial Error,” in *Proc. of Advances in Cryptology – Crypto 2014*, pp. 297-314, 2014. Article (CrossRef Link).

[14] R. Hiromasa, M. Abe, and T. Okamoto, “Packing Messages and Optimizing Bootstrapping in Gsw-Fhe,” in *Proc. of Public-Key Cryptography -- Pkc 2015*, pp. 699-715, 2015. Article (CrossRef Link).

[15] O. Regev, “On Lattices, Learning with Errors, Random Linear Codes, and Cryptography,” in *Proc. of the thirty-seventh annual ACM symposium on Theory of computing*, pp. 84-93, 2005. Article (CrossRef Link).

[16] V. Lyubashevsky, C. Peikert, and O. Regev, “On Ideal Lattices and Learning with Errors over Rings,” *in Proc. of Advances in Cryptology – Eurocrypt 2010*, pp. 1-23, 2010. Article (CrossRef Link).

[17] C. Peikert, V. Vaikuntanathan, and B. Waters, “A Framework for Efficient and Composable Oblivious Transfer,” in *Proc. of Advances in Cryptology – Crypto 2008*, pp. 554-571, 2008. Article (CrossRef Link).

[18] D. StehléR. Steinfeld, “Making Ntru as Secure as Worst-Case Problems over Ideal Lattices,” in *Proc. of Advances in Cryptology – Eurocrypt 2011*, pp. 27-47, 2011. Article (CrossRef Link).

[19] Z. BrakerskiV. Vaikuntanathan, “Lattice-Based Fhe as Secure as Pke,” in *Proc. of 5th conference on Innovations in theoretical computer science*, pp. 1-12, 2014. Article (CrossRef Link).

[20] C. Peikert, “A Decade of Lattice Cryptography,” *Found. Trends Theor. Comput. Sci.*, vol. 10, no. 4, pp. 283-424, 2014. Article (CrossRef Link).

[21] M.R. Albrecht, R. Player, and S. Scott, “On the Concrete Hardness of Learning with Errors,” *Journal of Mathematical Cryptology*, vol. 9, no. 3, pp. 169-203, 2015.

[22] C. Gentry, S. Halevi, and N. Smart, “Homomorphic Evaluation of the Aes Circuit,” in *Proc. of Advances in Cryptology – Crypto 2012*, pp. 850-867, 2012. Article (CrossRef Link).

[23] M. PaindavoineB. Vialla, “Minimizing the Number of Bootstrappings in Fully Homomorphic Encryption,” in *Proc. of International Conference on Selected Areas in Cryptography*, pp. 25-43, 2015.

[24] I. Chillotti, N. Gama, M. Georgieva and et al., "Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds," in *Proc. of Advances in Cryptology – Asiacrypt 2016*, pp. 3-33, 2016. Article (CrossRef Link).

[25] J. Bos, K. Lauter, J. Loftus and et al., "Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme," in *Proc. of Cryptography and Coding*, pp. 45-64, 2013. Article (CrossRef Link).

Appendix

A. Analysis of Concert Parameters

In order to be consistent with the practical application, we improve the method of estimating concert parameters for LWE-based FHE scheme in [22], and we introduce the advantage of adversary. Namely, given security level, advantage of adversary and Gaussian parameter, we can derive dimension n, modulus q and the circuit depth L from some formulas and thus get corresponding concert public key size, secret key size and ciphertext size. We apply this method of estimating concert parameters in our three FHE schemes, two GSW13 schemes and two Bra12 schemes. We call our three FHE schemes, namely LWE-based scheme, ring LWE-based scheme and NTRU scheme, obtained by using the design method as matrix-LWE scheme, matrix-RLWE scheme and matrix-NTRU scheme respectively. We call the two GSW13 schemes, namely LWE-based GSW13 scheme and ring LWE-based GSW13 scheme, as GSW13-LWE scheme and GSW13-RLWE scheme respectively. Even though Bra12 scheme is different from our schemes and GSW13 scheme, we still consider it and compare its corresponding parameters size with other schemes in order to study extensively. We call the two Bra12 schemes, namely LWE-based Bra12 scheme and ring LWE-based Bra12 scheme, as Bra12-LWE scheme and Bra12-RLWE scheme respectively.

A.1 Parameters Properties

Here we list the parameters properties of the schemes described above in **Table 1**. Note that only Bra12 scheme has evaluation keys used in key switching process while other schemes don't need to do this operation. Here we don't assume the circular security, thus each multiplicative level need a secret key and evaluation key. We use L to denote the circuit depth.

Table 1. The parameters properties of seven FHE schemes

	Public Key	Secret Key	Ciphertext	Evaluation Keys
matrix-RLWE	$2n\log q$	$(n+1)\log B$	$4n\log^2 q$	
matrix-NTRU	$n\log q$	$n\log(2B)$	$n\log^2 q$	

matrix-LWE	$2n(n+1)\log^2 q$	$(n+1)\log q$	$(n+1)^2\log^2 q$	
GSW13-RLWE	$2n\lceil \log q \rceil$	$(n+1)\lceil \log q \rceil^2$	$4n\lceil \log q \rceil^3$	
GSW13-LWE	$2n(n+1)\log^2 q$	$(n+l)\lceil \log q \rceil^2$	$(n+1)^2\lceil \log q \rceil^3$	
Bra12-RLWE	$2n\lceil \log q \rceil$	$(L+1)(n+l)\log B$	$2n\lceil \log q \rceil$	$6Ln\lceil \log q \rceil^2$
Bra12-LWE	$2n(n+1)\log^2 q$	$(L+1)(n+l)\lceil \log q \rceil$	$(n+1)\lceil \log q \rceil$	$L(n+1)^3\lceil \log q \rceil^4$

Obviously, the parameters sizes of the schemes base on ring LWE are smaller than the schemes based on LWE. Thus we only consider the comparison of the schemes based on the same hard problem, e.g., ring LWE or LWE. The parameters properties listed in table 1 show that the sizes of ciphertexts and secret keys of our schemes are smaller by a factor of about log$q$ than the corresponding GSW schemes. For Bra12 schemes (over ring LWE or LWE), the disadvantage is that the sizes of the public key and secret key are larger than other schemes due to including L evaluation keys and secret keys respectively. However its advantage is that the sizes of ciphertexts are smaller by a factor of about log$q$ and $\log^2 q$ than matrix-RLWE (or matrix-NTRU) and GSW13-RLWE respectively.

A.2 The Relation of Dimension and Modulus

In order to estimate the hardness of LWE for a concert set of parameters, we apply the distinguishing attack against LWE as in [20]. Since it is unknown how to exploit the ring structure of ring LWE to improve lattice reduction, the distinguishing attack can also be applied in ring LWE by embedding ring LWE instance into a LWE lattice.

The distinguishing attack on LWE means that an adversary distinguishes an LWE instance ($\boldsymbol{A}^{\mathrm{T}}$, $\boldsymbol{b}$=$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}$+$\boldsymbol{e}$) from uniform with some noticeable advantage, where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\boldsymbol{s} \in \mathbb{Z}_q^n$ are chosen uniformly and $\boldsymbol{e}$ is sampled from a Gaussian distribution with standard deviation r .

To estimate the concert parameters for a FHE scheme, a natural way is that the security level λ and Gaussian parameter r are firstly be fixed, then the modulus q is derived from the circuit depth required to perform homomorphic evaluation as well as the correctness of decryption, and then the dimension n can be determined by the above parameters such that the scheme has the corresponding security level. Thus the dimension n is the function of the parameters security level λ, Gaussian parameter r and the modulus q. This function was given in the paper [22]. However, this function doesn't reflect the advantage of adversary. One may need to consider the different advantage of adversary in practical application. Therefore, we improve the function by introducing the advantage of adversary.

Suppose an adversary has advantage *adv*. The length of short vector achieved by the adversary is $\beta = (q/r)\sqrt{\ln(1/adv)/\pi}$ from [21]. In addition, root-Hermite factor δ and the length of short vector β has the relation $\delta = 2^{(\log^2 \beta)/(4n\log q)} = 2^{(\log^2((q/r)\sqrt{\ln(1/adv)/\pi}))/(4n\log q)}$ [20]. Then according to the equation log (time)=1.8/log(δ) − 110 described in [21], we have

$$n \geq \log^2((q/r)\cdot\sqrt{\ln(1/adv)/\pi}\,)\cdot(\log(2^{\lambda}\cdot adv)+110)\,/\,(7.2\cdot\log q) \qquad (3)$$

It means that given security level λ, Gaussian parameter r and the advantage of adversary *adv* we can derive the minimal value of the dimension n for a modulus q from equation (3). We provide some values in **Table 2**.

Table 2. The minimal value of the dimension n for different modulus q to ensure 80 bit security with Gaussian parameter r =8 and the adversary advantage adv=2^{-32},2^{-80}

log q	13	22	42	80	158	313
n, adv=2^{-32}	171.64	363.27	798.12	1629.90	3340.41	6741.22
n，adv=2^{-80}	123.34	257.19	560.23	1139.48	2330.43	4698.12

Next we consider how to estimate the value of modulus q and concert parameters.

A.3 Concert Parameters

In the leveled fully homomorphic encryption scheme, the modulus q should ensure that the circuit depth is enough to perform the required homomorphic evaluation as well as the correctness of decryption. In order to ensure the correctness of homomorphic evaluation on circuit depth L, we below give the condition of correct decryption for each scheme that we consider above. The matrix-RLWE scheme need to satisfy condition $(2N+n)^L\cdot\beta_1<q/8$, where $N=n(l+1)$ and $\beta_1=2nB^2+B$. The matrix-NTRU scheme needs to satisfy condition $(N+n)^L\cdot\beta_2<q/8$, where $\beta_2=4nB^2+nB$. The matrix-LWE scheme needs to satisfy condition $(N+n+2)^L\cdot\beta_3<q/8$, where $\beta_3=2nB\log q$. The GSW13-RLWE scheme needs to satisfy condition $(2N+n)^L\cdot\beta_1<q/8$ that is the same as the condition in the matrix-RLWE scheme. The GSW13-LWE scheme needs to satisfy condition $(N+n+2)^L\cdot\beta_3<q/8$, where $\beta_3=2nB\log q$. The Bra12-RLWE

scheme needs to satisfy condition $|t_1^L \cdot \beta_4 + L \cdot t_1^{L-1} \cdot t_2| < \left\lfloor \frac{q}{2} \right\rfloor / 2$, where $t_1=2n^2B+8n$, $t_2=n^2B(4+B)+2nB\log q$ and $\beta_4 = 2n^2B+B$ The Bra12-LWE scheme needs to satisfy condition $|t_1^L \cdot \beta_3 + L \cdot t_1^{L-1} \cdot t_2| < \left\lfloor \frac{q}{2} \right\rfloor / 2$, where $t_1= 4(n+1)\lceil \log q \rceil$, $t_2=2(n+1)^2 \lceil \log q \rceil^3 B$ and $\beta_3=2nB\log q$.

Fix security level and adversarial advantage, we can obtain the concert parameters values for different circuit depth. These concert parameters values are listed in **Table 3** when the security level is chosen as 80 bit and the adversarial advantage is chosen as 2^{-80}. The unit of measurement is Kilobyte.

The data in table 3 show that the size of public key and secret key in our schemes are smaller than other schemes. Specially, the size of public key and secret key in our matrix-NTRU scheme are the smallest among these schemes. The reason is that the public key and secret key are only a polynomial respectively in matrix-NTRU scheme. In terms of the size of ciphertext, Bra scheme has smallest size among these schemes. The reason is that the ciphertext is only a vector while the ciphertext is a matrix in other schemes. The sizes of ciphertexts in our three schemes are smaller than it in GSW schemes. The reason is that the dimension of ciphertext matrix in our schemes is smaller than it in GSW schemes.

Table 3. The concert parameters values of seven FHE schemes

	Circuit Depth	Public Key	Evaluation Keys	Secret Key	Ciphertext
matrix-RLWE	L=0	1.68	0	0.2	80.80
	L=5	51.27	0	1.2	12305.41
	L=10	199.45	0	2.39	93344.52
matrix-NTRU	L=0	1.00	0	0.26	18.09
	L=5	23.67	0	1.36	1893.65
	L=10	92.17	0	2.71	14377.80
matrix-LWE	L=0	9606.79	0	0.77	4821.04
	L=5	9377466	0	23.93	4691509
	L=10	139063692.91	0	92.14	69552578.60
GSW-RLWE	L=0	1.68	0	20.27	1939.30
	L=5	51.27	0	3078.11	1476649.65

	L=10	199.45	0	23342.81	21842616.64
GSW-LWE	*L*=0	9606.79	0	17.64	110883.94
	L=5	9377466.00	0	2776.00	544215043.81
	L=10	139063692.91	0	20732.14	15649330184.86
Bra-RLWE	*L*=0	2.015	0	0.22	2.02
	L=5	115.86	311082.46	10.91	115.86
	L=10	442.01	4601346.26	39.34	442.01
Bra-LWE	*L*=0	6339.80	0	0.62	0.62
	L=5	17437727.71	1574046855853775.20	195.80	32.63
	L=10	238165854.75	302393085215490500.00	1326.41	120.58

Xinxia Song She is an associate professor at Zhejiang Wanli University. Her researches currently focus on cryptography and algebra.

Zhigasng Chen He received Ph.D. in the Nanjing University of Aeronautics and Astronautics. He is a professor at Zhejiang Wanli University now. He was an academic visitor in Information Security Group of Royal Holloway, University of London. Currently his researches focus on fully homomorphic encryption, lattice-based cryptography and blockchain..