Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 983862, 12 pages
http://dx.doi.org/10.1155/2014/983862

Research Article

A Regev-Type Fully Homomorphic Encryption Scheme Using Modulus Switching

Zhigang Chen,[1,2,3] Jian Wang,[1] Liqun Chen,[4] and Xinxia Song[5]

[1] *College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[2] *College of Computer and Information, Zhejiang Wanli University, Ningbo, Zhejiang 315100, China*
[3] *Information Security Group, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK*
[4] *HP Labs, Bristol BS34 8QZ, UK*
[5] *College of Junior, Zhejiang Wanli University, Ningbo, Zhejiang 315101, China*

Correspondence should be addressed to Zhigang Chen; chzg99@gmail.com

Received 9 March 2014; Accepted 14 May 2014; Published 25 June 2014

Academic Editor: Tianjie Cao

A critical challenge in a fully homomorphic encryption (FHE) scheme is to manage noise. Modulus switching technique is currently the most efficient noise management technique. When using the modulus switching technique to design and implement a FHE scheme, how to choose concrete parameters is an important step, but to our best knowledge, this step has drawn very little attention to the existing FHE researches in the literature. The contributions of this paper are twofold. On one hand, we propose a function of the lower bound of dimension value in the switching techniques depending on the LWE specific security levels. On the other hand, as a case study, we modify the Brakerski FHE scheme (in Crypto 2012) by using the modulus switching technique. We recommend concrete parameter values of our proposed scheme and provide security analysis. Our result shows that the modified FHE scheme is more efficient than the original Brakerski scheme in the same security level.

1. Introduction

A fully homomorphic encryption (FHE) scheme allows arbitrary functions on certain data (referred to as plaintexts) to be performed via their ciphertexts (the encrypted version of the plaintexts) without decrypting the ciphertexts first; therefore, performing these functions does not require one to hold the secret decryption key corresponding to the encryption algorithm. This cryptographic primitive has shown a variety of attractive applications both in theory and in practice. A typical application example is to outsource a computational job to a mistrusted remote server without compromising data privacy.

Since Gentry constructed the first FHE scheme in 2009 [1], a number of FHE schemes including various optimizations of the Gentry original scheme have been proposed. Gentry and colleagues developed several FHE schemes with different improvement, for example, [2–6]; one of them is how to bootstrap "packed" ciphertexts [6]. Smart and Vercauteren modified the Gentry scheme with the purpose of reducing the key and ciphertext sizes [7]. Stehlé and Steinfeld provides two improvements, respectively, on more aggressive analysis and probabilistic decryption algorithm in order to make the Gentry type of FHE schemes faster [8]. Brakerski et al. made a number of important contributions to this research field, such as [9–13], the details of which will be discussed more in the late part of this paper. Furthermore, van Dijk et al. proposed a new FHE construction over the integers [14], and Coron et al. further suggested on how to optimize this idea with shorter keys [15, 16]. López-Alt et al. constructed a multikey FHE scheme, which allows multiple ciphertexts under different keys to be decrypted jointly [17]. Alperin-Sheriff and Peikert introduced a method to achieve practical bootstrapping in Quasilinear time [18].

One critical challenge when constructing a FHE scheme is managing the noise growth in the process of homomorphic additions and multiplications. To our best knowledge, so far,

there exist three techniques to manage the noise growth as follows.

The first technique is bootstrapping that was used in the first FHE scheme introduced by Gentry. Bootstrapping means to evaluate its own decryption circuit homomorphically. One can use a bootstrapping process to get a new ciphertext after each homomorphic addition or homomorphic multiplication. The noise level in the new ciphertext is maintained in a fixed level. As long as this noise level permits, one can handle the next homomorphic addition or multiplication. By recursing this process a leveled FHE scheme can be developed, and the number of levels (although say the depth of the levels) for a computational circuit could be arbitrary with an assumption of circular security. A FHE scheme with the property of having an arbitrary depth of leveled circuits is referred to as a "pure" FHE scheme.

The second technique is modulus switching. This technique was developed by Brakerski and Vaikuntanathan in [10] and improved in [11]. The main idea of modulus switching is to scale down the ciphertext vector $\mathbf{c}$ over $\mathbb{Z}_q$ or a factor B after each multiplication, which results in a new ciphertext vector $\mathbf{c}/B$ over $\mathbb{Z}_{q/B}$. A scaling process switches the first modulus q to the second modulus q/B and also reduces the noise E in the ciphertext vector $\mathbf{c}$ to the new noise E/B in the new ciphertext vector $\mathbf{c}/B$. By following this process, the absolute magnitude of the new noise in the new ciphertext actually decreases. Modulus switching therefore can be used to manage noise at the cost of sacrificing the size of modulus. A leveled FHE scheme without bootstrapping can be achieved by modulus switching. In this technique, the depth of leveled computational circuits is prearranged before the computation starts. The depth is presented as a polynomial. For any prearranged polynomial denoted by L, one can evaluate circuits of depth L by carefully choosing the ladder of decreasing modulus.

The third technique is called Flatten, developed by Gentry et al. in [19]. It is designed for the case that an encryption key is presented as a vector and a ciphertext is presented as a matrix. It makes the coefficients of a vector or matrix small by using a flattening technique.

Among the three techniques for noise management, bootstrapping is a general technique that can be used to manage noise in any FHE scheme, but it is very costly! The technique of Flatten is only used in the case where ciphertexts are matrices and the secret keys are vectors. Modulus switching is a lightweight and very powerful way to manage noise and one can efficiently evaluate an arithmetic circuit with an arbitrary polynomial size without resorting to bootstrapping. In this paper, we will focus on modulus switching for noise management and consider the case that ciphertexts and the secret keys are both vectors.

In terms of noise growth, the noise grows from E to E^2 with every multiplication in most of the existing FHE schemes, where E denotes the noise magnitude in ciphertext. However, in the FHE scheme [12] by Brakerski in 2012 (we call it Bra12 for short), each homomorphic multiplication does not square the noise, and instead of the noise grows from E to $poly(n) \cdot E$ after each homomorphic multiplication. From this point of view, it looks like that the Bra12 scheme is more efficient, but in fact it is not true. Since the Bra12 scheme makes use of bootstrapping to manage noise, it requires modulus q must be big in order to achieve the result that the scheme has a circuit with enough depth to evaluate its own decryption circuit, for example, $q = \widetilde{O}(2^{n/2})$. The security of the scheme depends on the ratio q/B, where B is an initial magnitude of noise; therefore, B cannot be small. These reasons result in the secret key $\mathbf{s}$ sampled uniformly from $\mathbb{Z}_q^n$ rather than from the error distribution χ in the Bra12 scheme. In addition, the noise mainly depends on one norm of $\mathbf{s}$ writing as $\|\mathbf{s}\|_1$ in homomorphic multiplication in the Bra12 scheme. In order to reduce the noise, the scheme uses binary decomposition of the secret key $\mathbf{s}$ to reduce the norm $\|\mathbf{s}\|_1$. This means that a ciphertext $\mathbf{c}$ under the key $\mathbf{s}$ is converted into a new form of ciphertext, denoted by Powerof2($\mathbf{c}$) under key BitDecomp($\mathbf{s}$). Although the new form of the ciphertext and the secret key can effectively reduce the noise, it increases the dimension of ciphertext and the secret key. In particular, the dimension of the ciphertext and secret key can further blow up in homomorphic multiplication and key switching, which lead to a fatal result when evaluating deep circuits, since it may need too much memory to compute. This feature considerably affects efficiency in the Bra12 scheme.

In this paper, we use modulus switching and an additional technique to improve the efficiency of the Bra12 scheme. Our scheme has the following properties:

(1) There is lower dimension of the ciphertext and the secret key in homomorphic multiplication and key switching than in the Bra12 scheme. The ciphertext for homomorphic multiplication is defined as $\lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \rceil$ that corresponds to the secret key $\mathbf{s} \otimes \mathbf{s}$ in our scheme, while the ciphertext for homomorphic multiplication is defined as $\lfloor 2/q \cdot (\text{Poweof2}(\mathbf{c}_1) \otimes \text{Powerof2}(\mathbf{c}_2)) \rceil$ that corresponds to the secret key BitDecomp($\mathbf{s}$) $\otimes$ BitDecomp($\mathbf{s}$) in the Bra12 scheme.

(2) The secret key $\mathbf{s}$ is sampled from a Gaussian distribution χ in our scheme, which can enable us to get small coefficients of $\mathbf{s}$. In the Bra12 scheme, the secret key $\mathbf{s}$ is sampled uniformly from $\mathbb{Z}_q^n$.

(3) Our scheme uses modulus switching to manage noise, while the Bra12 scheme uses bootstrapping to manage noise.

(4) In our scheme the initial modulus is that $q \approx 2^{n^\varepsilon}$ for every $\varepsilon < 1$, while in the Bra12 scheme the modulus is that $q \approx \widetilde{O}(2^{n/2})$. The small modulus q makes our scheme considerably efficient.

For a FHE scheme using modulus switching, it is very important to choose a ladder of gradually decreasing moduli $\{q_i\}$. However, so far there has not been a concrete method to tell how to choose these parameters in terms of a certain security level, even in the BGV scheme [11] that just provided a general method to choose moduli $\{q_i\}$. In this paper, we provide a solution to this problem. We first derive a function between the lower bound on the dimension of the LWE problem and the security level. Then we can choose every

concrete modulus q_i and other parameters for a certain security level (e.g., the security level is 80 bit) according to this function.

The rest of this paper is organized as follows. Section 2 defines notational conventions, introduces the LWE assumption, and defines homomorphic encryption and its related terms. Section 3 introduces the Regev encryption scheme that our scheme is based on and defines invariant structure. There is a minor change in the Regev encryption scheme that we describe here. We sample the secret key from a Gauss distribution rather than sample uniformly from $\mathbb{Z}_q^n$ in the Regev encryption scheme. Section 4 analyzes the homomorphic properties by the opinion of invariant structure and the noise growth in homomorphic addition and multiplication. Section 5 introduces key switching and modulus switching. Our FHE scheme based on the modified Regev encryption scheme is presented in Section 6. We analyze how to enable the correctness of our scheme in Section 7. The security and the parameters of our scheme are presented in Section 8. We conclude the paper with a performance comparison between our scheme and the Bar12 scheme in Section 9.

2. Preliminaries

2.1. Basic Notation. 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 $y \in (-q/2, q/2]$. We use $\lfloor x \rceil$ to indicate rounding x to the nearest integer, and $\lfloor x \rfloor$, $\lceil x \rceil$ (for $x \geqslant 0$) to indicate rounding down or up. When q is not a power of two, we will use $\lceil \log q \rceil$ to denote $1 + \lfloor \log q \rfloor$.

We use $x \leftarrow \mathcal{D}$ to denote that x is a sample from a distribution $\mathcal{D}$. We define B-bounded distributions as ones whose magnitudes never exceed B.

The inner product of two vectors $\mathbf{v}$, $\mathbf{u}$ of dimension n is denoted by $\langle \mathbf{v}, \mathbf{u} \rangle$, recalling that $\langle \mathbf{v}, \mathbf{u} \rangle = \mathbf{v}^T \cdot \mathbf{u}$. The tensor product of two vectors $\mathbf{v}, \mathbf{u}$ of dimension n, denoted by $\mathbf{v} \otimes \mathbf{u}$, is the n^2 dimensional vector containing all elements of the form $\mathbf{v}[i]\mathbf{u}[j]$. Note that $\langle \mathbf{v} \otimes \mathbf{u}, \mathbf{x} \otimes \mathbf{y} \rangle = \langle \mathbf{v}, \mathbf{x} \rangle \cdot \langle \mathbf{u}, \mathbf{y} \rangle$.

A lattice is defined as the set of all integer combinations $\Lambda = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^n x_i \mathbf{b}_i : x_i \in \mathbb{Z} \text{ for } 1 \leq i \leq n\}$ of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ in $\mathbb{R}^n$. The set of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is called a basis for the lattice. A basis can be represented by the matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{n \times n}$. The determinant of a lattice is the absolute value of the determinant of the basis matrix $\det(\Lambda) = |\det(\mathbf{B})|$.

q-ary lattices are most important in lattice-based cryptography. Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for integers q, m, n, there are two kinds of m-dimensional q-ary lattices

$$\Lambda_q(\mathbf{A}) = \left\{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} = \mathbf{A}^T \mathbf{s} \mod q \text{ for some } \mathbf{s} \in \mathbb{Z}^n\right\},$$

$$\Lambda_q^{\perp}(\mathbf{A}) = \left\{\mathbf{y} \in \mathbb{Z}^m : \mathbf{A}^T \mathbf{s} = 0 \mod q\right\}. \tag{1}$$

The two kinds of q-ary lattices are dual to each other, namely, $\Lambda_q(\mathbf{A}) = q \cdot \Lambda_q^{\perp}(\mathbf{A})^*$ and $\Lambda_q^{\perp}(\mathbf{A}) = q \cdot \Lambda_q(\mathbf{A})^*$.

2.2. Learning with Errors (LWE). The learning with errors (LWE) problem was introduced by Regev [20]. This problem was later generalized as the ring learning with errors (RLWE) problem by Lyubashevsky et al. [21]. For security parameter λ, let $n = n(\lambda)$ be an integer dimension, let $q = q(\lambda) \geqslant 2$ be an integer, a vector $\mathbf{s} \in \mathbb{Z}_{\mathbf{q}}^{\mathbf{n}}$, and let $\chi = \chi(\lambda)$ be a distribution over $\mathbb{Z}$. Let $\mathcal{A}_{\mathbf{s},\chi}$ be the distribution obtained by choosing a vector $\mathbf{a}$ from $\mathbb{Z}_q^n$ uniformly at random and a noise term $e \leftarrow \chi$, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The LWE problem includes the search-LWE problem and the decision-LWE problem. The search-LWE problem is giving an arbitrary number of independent samples from $\mathcal{A}_{\mathbf{s},\chi}$, output $\mathbf{s}$ with a high probability. We are primarily interested in the decision-LWE (DLWE) problem for cryptographic applications. The DLWE problem is defined as follows.

Definition 1 (DLWE). For an integer $q = q(\lambda)$ and an error distribution $\chi = \chi(\lambda)$ over $\mathbb{Z}$, the decision-LWE problem, denoted by $\mathrm{DLWE}_{n,q,\chi}$, is to distinguish the following two distributions: in the first distribution, one sample from $\mathcal{A}_{\mathbf{s},\chi}$; in the second distribution, one sample uniformly from $\mathbb{Z}_q^{n+1}$. The $\mathrm{DLWE}_{n,q,\chi}$ assumption is that solving $\mathrm{DLWE}_{n,q,\chi}$ is computationally infeasible.

Two kinds of reductions are known, namely, the quantum reduction [20] and classical [22, 23] reduction, between $\mathrm{DLWE}_{n,q,\chi}$ and approximating short vector problems in lattices. Particularly, a probability distribution χ is taken to be the Gaussian distribution, which is statistically indistinguishable from the B-bound distribution for an appropriate value B.

Note that the DLWE problem can be seen as a bound distance decoding problem in q-ary lattices. The second component of LWE instance can be seen as a perturbed lattice point in $\Lambda_q(\mathbf{A}^t)$, to be decoded.

We now state the quantum reduction from worst-case lattice problems to the LWE problem introduced in [20].

Theorem 2. *For any integer dimension n, prime integer $q = q(\lambda)$, and $B = B(\lambda) \geqslant 2n$, there is an efficiently samplable B-bound distribution such that if there exists an efficient (possibly quantum) algorithm that solves $DLWE_{n,q,\chi}$, then there is an efficient quantum algorithm for solving $\widetilde{O}(q \cdot n^{1.5}/B)$-approximate worst-case SIVP and gapSVP.*

There are other forms of q (see [24, 25]). In addition, if the vector $\mathbf{s}$ *is sampled from the distribution χ, then the LWE problem is still hard. We sample* $\mathbf{s}$ *from the Gaussian distribution χ in our scheme.*

2.3. Leveled Fully Homomorphic Encryption. A homomorphic encryption scheme HE = (Keygen, Enc, Dec, Eval) includes a quadruple of PPT algorithms. For the definition of full homomorphic encryption, readers can refer to these papers [1, 12].

At present, there are two types of fully homomorphic encryption schemes. One is leveled fully homomorphic encryption schemes, in which the parameters of a scheme depend on the depth of the circuits that the scheme can evaluate. In that case any circuit with a polynomial depth can be evaluated. The other is pure fully homomorphic

encryption schemes, which can be built 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 [12].

Definition 3 (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

$$\Pr\left[\mathrm{HE.Dec}_{sk}\left(\mathrm{HE.Eval}_{evk}\left(f, c_1, \ldots, c_l\right)\right) \neq f\left(m_1, \ldots, m_l\right)\right] = \mathrm{negl}\left(\lambda\right), \tag{2}$$

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

Definition 4 (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-homomorphic for any polynomial L. The scheme is *leveled fully homomorphic* if it takes 1^L as additional input in key generation.

3. The Basic Encryption Scheme

As same as the Bra12 scheme, our scheme is based on Regev's encryption scheme [20]. We now describe the Regev encryption scheme, but we sample the secret key $\mathbf{s}$ from a Gauss distribution while it was sampled uniformly from $\mathbb{Z}_q^n$ in the Regev encryption scheme. This modification allows us to achieve our goal that the error distribution χ can be set to be as small as possible in our scheme. We call this modified Regev encryption scheme the basic encryption scheme.

Let $n = n(\lambda)$ be the dimension of lattice, an odd modulus $q = q(\lambda)$, and an error distribution $\chi = \chi(\lambda)$. The basic encryption scheme is described as follows.

E.SecretKeygen(1^λ): sample $\mathbf{s}' \leftarrow \chi^n$. Output $\mathbf{sk} = \mathbf{s} \leftarrow (1, \mathbf{s}')$.

E.PublicKeygen$(\mathbf{s})$: let $N \geqslant 2\,(n \log q)$. Sample $\mathbf{A}' \leftarrow \mathbb{Z}_q^{N \times n}$ and $\mathbf{e} \leftarrow \chi^N$. Compute $\mathbf{b} \leftarrow \mathbf{A}'\mathbf{s}' + \mathbf{e}$. Set $\mathbf{A}$ to be the $(n+1)$-column matrix consisting of $\mathbf{b}$ followed by the n columns of $-\mathbf{A}'$, namely $\mathbf{A} = [\mathbf{b} \mid -\mathbf{A}'] \in \mathbb{Z}_q^{N \times (n+1)}$. Note that $\mathbf{A} \cdot \mathbf{s} = \mathbf{e}$. Set the public key $\mathbf{pk} = \mathbf{A}$.

E.Enc$(\mathbf{pk}, m)$: to encrypt a message $m \in \{0,1\}$, set $\mathbf{m} \leftarrow (m, 0, \ldots, 0) \in \{0,1\}^{n+1}$, sample $\mathbf{r} \in \{0,1\}^N$, and output $\mathbf{c} \leftarrow \lfloor q/2 \rfloor \cdot \mathbf{m} + \mathbf{A}^T \cdot \mathbf{r} \in \mathbb{Z}_q^{n+1}$.

E.Dec$(\mathbf{sk}, \mathbf{c})$: output $m \leftarrow \lfloor (2/q)[\langle \mathbf{c}, \mathbf{s} \rangle]_q \rceil \bmod 2$.

The basic encryption scheme above is semantic security based on the hardness of the LWE problem. The proof of this statement follows the proof of security of the original Regev encryption scheme given in [20].

A FHE scheme needs to maintain an invariant structure in decryption that is composed of plaintext and noise. The scheme must keep the invariant structure in the process of homomorphic addition and homomorphic multiplication in order to achieve homomorphism. Next, we define the invariant structure in the above basic encryption scheme and explain the relationship between the correctness of decryption and the noise magnitude in ciphertext.

Lemma 5. *Let* $\mathbf{c} \in \mathbb{Z}_q^{n+1}$ *and* $\mathbf{s} \in \mathbb{Z}_q^{n+1}$ *be two vectors such that*

$$\langle \mathbf{c}, \mathbf{s} \rangle = \left\lfloor \frac{q}{2} \right\rfloor \cdot m + e \pmod q, \tag{3}$$

where $m \in \{0,1\}$. *If* $|e| < \lfloor q/2 \rfloor/2$, *then we have* $m \leftarrow \mathbf{E.Dec}(\mathbf{s}, \mathbf{c})$.

Proof. By definition

$$\begin{aligned} \langle \mathbf{c}, \mathbf{s} \rangle &= \left\langle \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{m} + \mathbf{A}^T \cdot \mathbf{r}, \mathbf{s} \right\rangle \pmod q \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \mathbf{r}^T \cdot \mathbf{A} \cdot \mathbf{s} \pmod q \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \langle \mathbf{r}, \mathbf{e} \rangle \pmod q \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + e \pmod q. \end{aligned} \tag{4}$$

Since the coefficients of $\mathbf{e}$ are taken from a Gaussian distribution χ, $e = \langle \mathbf{r}, \mathbf{e} \rangle$ is also subject to a Gaussian distribution according to the standard fact from the Gaussian distribution. The Claim 5.2 in [20] showed that $|e| < \lfloor q/2 \rfloor/2$ with high probability. Consider an encryption of 0 now; it is closer to 0 than to $\lfloor q/2 \rfloor$ in this case and therefore the decryption is correct. The proof for an encryption of 1 is similar.

The term e is called the noise. $\lfloor q/2 \rfloor \cdot m + e \pmod q$ is called the invariant structure. The above Lemma 5 shows that the invariant structure will be hold as long as $|e| < \lfloor q/2 \rfloor/2$, which can ensure the correctness of decryption. Note that it is very important to keep the invariant structure in ciphertexts generated in homomorphic evaluation. □

4. Homomorphic Properties and Noise Analysis

We take the definition of homomorphic addition and homomorphic multiplication from the Bra12 scheme, but here we analyze the homomorphic properties of the above scheme by the approach of the invariant structure. Now we analyze the noise growth in the homomorphic addition and multiplication.

Let $\mathbf{c}_1$ and $\mathbf{c}_2$ be two ciphertexts under the same secret key $\mathbf{s}$ for modulus q such that

$$\begin{aligned} \langle \mathbf{c}_1, \mathbf{s} \rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 + e_1 \pmod q = \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 + e_1 + k_1 q, \\ \langle \mathbf{c}_2, \mathbf{s} \rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_2 + e_2 \pmod q = \left\lfloor \frac{q}{2} \right\rfloor \cdot m_2 + e_2 + k_2 q, \end{aligned} \tag{5}$$

for some e_1 and e_2.

4.1. Homomorphic Addition. Let $\mathbf{c}^{\text{add}} = \mathbf{c}_1 + \mathbf{c}_2$. If the invariant structure $\langle \mathbf{c}_1 + \mathbf{c}_2, \mathbf{s}\rangle = \lfloor q/2\rfloor \cdot (m_1 + m_2) + e \pmod q$ can be held during the decryption of $\mathbf{c}^{\text{add}}$ for some e, the decryption would be correct such that homomorphic addition is obtained.

By definition

$$\begin{aligned}\langle \mathbf{c}_1 + \mathbf{c}_2, \mathbf{s}\rangle &= \langle \mathbf{c}_1, \mathbf{s}\rangle + \langle \mathbf{c}_2, \mathbf{s}\rangle \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot (m_1 + m_2) \\ &\quad + e_1 + e_2 + k_1 q + k_2 q.\end{aligned} \tag{6}$$

Let $e^{\text{add}} = e_1 + e_2$. According to the Lemma 5, if $|e^{\text{add}}| < \lfloor q/2\rfloor/2$, then $m_1 + m_2 \leftarrow \mathbf{E.Dec}(\mathbf{sk}, \mathbf{c}_1 + \mathbf{c}_2)$. It also means that the invariant structure $\lfloor q/2\rfloor \cdot (m_1 + m_2) + e_1 + e_2 \pmod q$ can be kept in the decryption of $\mathbf{c}^{\text{add}}$. We note that the noise term of output is the sum of input noises.

4.2. Homomorphic Multiplication. Multiplicative homomorphism cannot be straightforwardly achieved. We need to construct a form of the two input ciphertexts to represent the homomorphic multiplication such that we can get the product of the two plaintexts with respect to the input ciphertexts after decrypting the homomorphic multiplication. For this purpose, we now focus on the invariant structure in the process of decryption. If the invariant structure $\lfloor q/2\rfloor \cdot (m_1 \cdot m_2) + e$ for some e is kept in the decryption of the homomorphic multiplication, we could achieve multiplicative homomorphism. Next, we describe how to achieve multiplicative homomorphism by the approach of the invariant structure.

Consider the multiplication of $\langle \mathbf{c}_1, \mathbf{s}\rangle$ and $\langle \mathbf{c}_2, \mathbf{s}\rangle$ now, we have:

$$\begin{aligned}\langle \mathbf{c}_1, \mathbf{s}\rangle \cdot \langle \mathbf{c}_2, \mathbf{s}\rangle &= \left(\left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 + e_1 + k_1 q\right) \\ &\quad \times \left(\left\lfloor \frac{q}{2} \right\rfloor \cdot m_2 + e_2 + k_2 q\right) \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot \left\lfloor \frac{q}{2} \right\rfloor \cdot (m_1 m_2) + \cdots.\end{aligned} \tag{7}$$

In order to keep the invariant structure $\lfloor q/2\rfloor \cdot (m_1 \cdot m_2) + e$, we multiple the above equation by $2/q$:

$$\begin{aligned}&\frac{2}{q} \cdot \langle \mathbf{c}_1, \mathbf{s}\rangle \cdot \langle \mathbf{c}_2, \mathbf{s}\rangle \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 m_2 + m_1 e_2 + m_2 e_1 + 2(e_1 k_2 + k_1 e_2) \\ &\quad + q \cdot (m_1 k_2 + k_1 m_2 + 2 k_1 k_2) - [q]_2 \cdot (m_1 k_2 + k_1 m_2) \\ &\quad + \frac{[q]_2}{q} \cdot \left(m_1 e_2 - m_2 e_1 - \left\lfloor \frac{q}{2} \right\rfloor \cdot (m_1 m_2)\right) + \frac{2}{q} \cdot e_1 e_2 \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 m_2 + e,\end{aligned} \tag{8}$$

where $e = m_1 e_2 + m_2 e_1 + 2(e_1 k_2 + k_1 e_2) + q \cdot (m_1 k_2 + k_1 m_2 + 2 k_1 k_2) - [q]_2 \cdot (m_1 k_2 + k_1 m_2) + [q]_2/q \cdot (m_1 e_2 - m_2 e_1 - \lfloor q/2\rfloor \cdot (m_1 m_2)) + 2/q \cdot e_1 e_2$.

The invariant structure appears in (8). Since $2/q \cdot \langle \mathbf{c}_1, \mathbf{s}\rangle \cdot \langle \mathbf{c}_2, \mathbf{s}\rangle = \langle 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2), \mathbf{s} \otimes \mathbf{s}\rangle$, multiplicative homomorphism is achieved by tensoring the input ciphertext $2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2)$. We note the ciphertext is fraction. For the sake of simplicity, we round the ciphertext for multiplication to the nearest integer ciphertext $\lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2)\rceil$, which will bring out an error $\mathbf{r} = \lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2)\rceil - 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2)$. Thus we get

$$\begin{aligned}&\left\langle \left\lfloor \frac{2}{q} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \right\rceil, \mathbf{s} \otimes \mathbf{s} \right\rangle \\ &= \left\langle \frac{2}{q} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) + \mathbf{r}, \mathbf{s} \otimes \mathbf{s} \right\rangle \\ &= \left\langle \frac{2}{q} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2), \mathbf{s} \otimes \mathbf{s} \right\rangle + \langle \mathbf{r}, \mathbf{s} \otimes \mathbf{s}\rangle.\end{aligned} \tag{9}$$

Plugging (8) into above equation, we have

$$\begin{aligned}&\left\langle \left\lfloor \frac{2}{q} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \right\rceil, \mathbf{s} \otimes \mathbf{s} \right\rangle \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 m_2 + m_1 e_2 + m_2 e_1 + 2(e_1 k_2 + k_1 e_2) \\ &\quad - [q]_2 \cdot (m_1 k_2 + k_1 m_2) + \frac{[q]_2}{q} \\ &\quad \cdot \left(m_1 e_2 - m_2 e_1 - \left\lfloor \frac{q}{2} \right\rfloor \cdot (m_1 m_2)\right) + \frac{2}{q} \cdot e_1 e_2 \\ &\quad + \langle \mathbf{r}, \mathbf{s} \otimes \mathbf{s}\rangle \pmod q \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 m_2 + e_1^{\text{mult}} + e_2^{\text{mult}} \pmod q,\end{aligned} \tag{10}$$

where $e_1^{\text{mult}} = m_1 e_2 + m_2 e_1 + 2(e_1 k_2 + k_1 e_2) - [q]_2 \cdot (m_1 k_2 + k_1 m_2) + [q]_2/q \cdot (m_1 e_2 - m_2 e_1 - \lfloor q/2\rfloor \cdot (m_1 m_2)) + 2/q \cdot e_1 e_2$ and $e_2^{\text{mult}} = |\langle \mathbf{r}, \mathbf{s} \otimes \mathbf{s}\rangle|$. The noise is $e_1^{\text{mult}} + e_2^{\text{mult}}$ in the ciphertext for multiplication. Particularly, the significant noise term of e_1^{mult} is $2(e_1 k_2 + k_1 e_2)$, which is not like the many previous FHE schemes whose homomorphic multiplication operation squares the noise.

The ciphertext for multiplication can thus be defined as $\lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2)\rceil$ that can be decrypted using a tensored secret key $\mathbf{s} \otimes \mathbf{s}$. The invariant structure in the decryption of the homomorphic multiplication is $\lfloor q/2\rfloor \cdot m_1 m_2 + e_1^{\text{mult}} + e_2^{\text{mult}}$. If $|e_1^{\text{mult}} + e_2^{\text{mult}}| < \lfloor q/2\rfloor/2$, according to Lemma 5, the invariant structure can be kept such that the correctness of decryption can hold. So we have $m_1 m_2 \leftarrow \mathbf{E.Dec}(\mathbf{sk}, \lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2)\rceil)$, where $\mathbf{sk}$ is $\mathbf{s} \otimes \mathbf{s}$. So far, we have finished the construction for the ciphertext for multiplication. We have achieved homomorphic addition and homomorphic multiplication. However, the noise growth is caused in the homomorphic addition and homomorphic multiplication.

The problem of noise growth in the homomorphic evaluation affects directly the homomorphic ability of the above

basic encryption scheme, so it is critical to manage noise growth for constructing the FHE scheme. Before we solve the problem of noise growth, we in the next subsection analyze the noise growth in a homomorphic addition and homomorphic multiplication. Note that our analysis method for the noise growth is different from the one used in the Bra12 scheme, as the secret key $\mathbf{s}$ is sampled from a Gaussian distribution which results in the secret key $\mathbf{s}$ is B-bounded. In addition, we give a tighter noise analysis than it in [12].

4.3. Noise Analysis

Lemma 6. *Let $q, n, |\chi| \leqslant B, N$ be parameters as described in the basic encryption scheme. Let $\mathbf{c}_1$, $\mathbf{c}_2$ be the ciphertexts under the secret key $\mathbf{s}$ such that*

$$\begin{aligned} \langle \mathbf{c}_1, \mathbf{s} \rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 + e_1 \pmod q, \\ \langle \mathbf{c}_2, \mathbf{s} \rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_2 + e_2 \pmod q, \end{aligned} \tag{11}$$

with $|e_1|, |e_2| \leqslant E < \lfloor q/2 \rfloor / 2$. Then

$$\begin{aligned} \langle \mathbf{c}_1 + \mathbf{c}_2, \mathbf{s} \rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot [m_1 + m_2]_2 + e^{add} \pmod q, \\ \left\langle \left\lfloor \frac{2}{q} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \right\rceil, \mathbf{s} \otimes \mathbf{s} \right\rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 m_2 + e^{mult} \pmod q, \end{aligned} \tag{12}$$

where $|e^{add}| \leqslant 1 + 2E$, $|e^{mult}| \leqslant 12nBE$.

Proof

Analysis for Addition. By definition

$$\begin{aligned} \langle \mathbf{c}_1 + \mathbf{c}_2, \mathbf{s} \rangle &= \langle \mathbf{c}_1, \mathbf{s} \rangle + \langle \mathbf{c}_2, \mathbf{s} \rangle \pmod q \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 + e_1 + \left\lfloor \frac{q}{2} \right\rfloor \cdot m_2 + e_2 \pmod q \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot (m_1 + m_2) + e_1 + e_2 \pmod q \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot [m_1 + m_2]_2 + 2 \cdot \left\lfloor \frac{q}{2} \right\rfloor \cdot \left\lfloor \frac{(m_1 + m_2)}{2} \right\rfloor \\ &\quad + e_1 + e_2 \pmod q. \end{aligned} \tag{13}$$

Then we get $|e^{\text{add}}| = |2 \cdot \lfloor q/2 \rfloor \cdot \lfloor (m_1 + m_2)/2 \rfloor + e_1 + e_2| \leqslant 1 + 2 \cdot E$.

Analysis for Multiplication. By (10)

$$\begin{aligned} \left\langle \left\lfloor \frac{2}{q} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \right\rceil, \mathbf{s} \otimes \mathbf{s} \right\rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 m_2 \\ &\quad + e_1^{\text{mult}} + e_2^{\text{mult}} \pmod q. \end{aligned} \tag{14}$$

We first analyze the bound of e_1^{mult}. The magnitude of e_1^{mult} mainly depends on the term $2(e_1 k_2 + k_1 e_2)$, so we check the bound of the absolute value of k_1 (the same bound also holds for k_2):

$$\begin{aligned} |k_1| &= \frac{|\langle \mathbf{c}_1, \mathbf{s} \rangle - \lfloor q/2 \rfloor \cdot m_1 - e_1|}{q} \\ &\leqslant \frac{|\langle \mathbf{c}_1, \mathbf{s} \rangle|}{q} + 1 \\ &\leqslant \left(\frac{\|\mathbf{c}_1\|_\infty}{q} \right) \cdot \|\mathbf{s}\|_1 + 1 \\ &\leqslant \left(\frac{1}{2} \right) \cdot \|\mathbf{s}\|_1 + 1 \\ &\leqslant nB. \end{aligned} \tag{15}$$

The absolute value of k_1 depends on $\|\mathbf{s}\|_1$ from above inequality, then the bound of k_1 is $O(E \cdot \|\mathbf{s}\|_1)$. The tighter bound is described as follows:

$$\left| e_1^{\text{mult}} \right| \leqslant 2E + 4nBE + 2nB + 3 \leqslant 8nBE. \tag{16}$$

Next, we analyze the bound of e_2^{mult}. According to the definition of an error $\mathbf{r} = \lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \rceil - 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2)$ and the secret key sampled from a B-bounded Gaussian distribution, we get $\|\mathbf{r}\|_\infty \leqslant 1/2$ and $|\mathbf{s} \otimes \mathbf{s}| \leqslant (n+1)^2 B^2$. Then

$$\begin{aligned} \left| e_2^{\text{mult}} \right| &= |\langle \mathbf{r}, \mathbf{s} \otimes \mathbf{s} \rangle| \leqslant \|\mathbf{r}\|_\infty \cdot |\mathbf{s} \otimes \mathbf{s}| \\ &\leqslant (n+1)^2 B^2. \end{aligned} \tag{17}$$

Putting these together, we get

$$\begin{aligned} \left| e_1^{\text{mult}} + e_2^{\text{mult}} \right| &\leqslant 8nBE + (n+1)^2 B^2 \\ &\leqslant 8nBE + 4nBE \\ &\leqslant 12nBE. \end{aligned} \tag{18}$$

We see that the significant noise term in the homomorphic multiplication depends on $\|\mathbf{s}\|_1$ from Lemma 6, which also happens in the Bra12 scheme. In order to reduce the norm, the secret key $\mathbf{s}$ is expressed in the form of binary, namely, BitDecomp($\mathbf{s}$), then the ciphertext corresponding to the $\mathbf{s}$ is expressed in Powerof2($\mathbf{c}$). The side effect is to produce the ciphertext vector and the secret key vector of a high dimension. In particularly, the ciphertext is the form of $\lfloor 2/q \cdot (Powerof2(\mathbf{c}_1) \otimes Powerof2(\mathbf{c}_2)) \rceil$ under the key $\text{BitDecomp}(\mathbf{s}_1) \otimes \text{BitDecomp}(\mathbf{s}_2)$ after homomorphic multiplication, which results in a large amount of computation that requires a large memory. The process cannot be practical. However, our scheme does not have this result. Since we sample the secret key from a Gaussian distribution that enables the coefficients of the secret key to be as small as possible, the secret key $\mathbf{s}$ needs not to be expressed in the form of binary, so the ciphertext. That is the reason why it can improve performance.

Under the above definition of homomorphic addition and homomorphic multiplication, we can perform only

a bounded number of homomorphic operations (namely, a somewhat homomorphic encryption scheme), because the noise and the dimension grow as a result of performing homomorphic operations. Therefore, there are two problems that should be solved in order to achieve a FHE scheme based on the somewhat homomorphic encryption scheme.

First, we need to control the dimension of the ciphertext that increases from $n+1$ to $(n+1)^2$ after a homomorphic multiplication. We use the key switching technique to solve this problem.

Second, we need to manage the noise growth in homomorphic operations. We use modulus switching to solve this problem. □

5. Key Switching and Modulus Switching

We describe the two techniques: key switching and modulus switching. Our notation is adopted from [11].

5.1. Key Switching. Key switching can transform a ciphertext $\mathbf{c}_1$ under a secret key $\mathbf{s}_1$ to a new ciphertext $\mathbf{c}_2$ under a secret key $\mathbf{s}_2$, in which $\mathbf{c}_1$ and $\mathbf{c}_2$ encrypt the same message. If the dimension of $\mathbf{c}_2$ and $\mathbf{s}_2$ is lower than the dimension of $\mathbf{c}_1$ and $\mathbf{s}_1$, the dimension of the key and ciphertect vectors is reduced by key switching.

Key switching consists of two procedures. The first procedure is denoted by $\mathbf{SwitchKeyGen}(\mathbf{s}_1, \mathbf{s}_2, n_1, n_2, q)$, which takes as input the two secret key vectors, the respective dimension of these vectors, the corresponding modulus q, and outputs some auxiliary information τ that is a matrix. The second procedure is denoted by $\mathbf{SwitchKey}(\tau, \mathbf{c}_1, n_1, n_2, q)$, which takes as input the auxiliary information τ, a ciphertext $\mathbf{c}_1$, and its dimension n_1, the dimension of the output ciphertext n_2, and the modulus q, and outputs a new ciphertext $\mathbf{c}_2$ whose dimension is n_2.

$\mathbf{SwitchKeyGen}(\mathbf{s}_1 \in \mathbb{Z}_q^{n_1}, \mathbf{s}_2 \in \mathbb{Z}_q^{n_2})$:

(1) Run $\mathbf{A} \leftarrow \mathbf{E.PublicKeygen}(\mathbf{s}_2)$ for $N = n_1 \cdot \lceil \log q \rceil$, namely, $\mathbf{A} = [\mathbf{b} \mid -\mathbf{A}']$.

(2) Set $\mathbf{B} \leftarrow [(\text{Powerof2}(\mathbf{s}_1) + \mathbf{b}) - \mathbf{A}']$, which means to add the $\text{Powerof2}(\mathbf{s}_1) \in \mathbb{Z}_q^N$ to $-\mathbf{A}'$'s first column and add $\mathbf{b}$ to $-\mathbf{A}'$'s second column. Output $\tau_{\mathbf{s}_1 \to \mathbf{s}_2} = \mathbf{B}$.

$\mathbf{SwitchKey}(\tau_{\mathbf{s}_1 \to \mathbf{s}_2}, \mathbf{c}_1, q)$: output $\mathbf{c}_2 = \text{BitDecomp}(\mathbf{c}_1)^T \cdot \mathbf{B} \in \mathbb{Z}_q^{n_2}$.

Key switching is essentially the product of a high dimension vector and a high dimension matrix. Next, we describe the correctness of key switching; namely, the decryption of the new ciphertext can preserve correctness. The proof is based on the definition (see [11]).

Lemma 7. *Let* $\mathbf{s}_1, \mathbf{s}_2, q, \mathbf{A}, \mathbf{B} = \tau_{\mathbf{s}_1 \to \mathbf{s}_2}$ *be parameters as described in* **SwitchKeyGen** *and have* $\mathbf{A} \cdot \mathbf{s}_2 = \mathbf{e}_2 \in \mathbb{Z}_q^N$. *Let* $\mathbf{c}_1 \in \mathbb{Z}_q^N$ *and* $\mathbf{c}_2 \leftarrow \mathbf{SwitchKey}(\tau_{\mathbf{s}_1 \to \mathbf{s}_2}, \mathbf{c}_1)$. *Then,*

$$\langle \mathbf{c}_2, \mathbf{s}_2 \rangle = \langle \mathit{BitDecomp}(\mathbf{c}_1), \mathbf{e}_2 \rangle + \langle \mathbf{c}_1, \mathbf{s}_1 \rangle \pmod q. \tag{19}$$

5.2. Modulus Switching

Definition 8 (Scale). For integer vector $\mathbf{x}$ and integers $q > p > m$, we define $\mathbf{x}' \leftarrow \mathbf{Scale}(\mathbf{x}, q, p, 2)$ to be the vector closest to $(q/p) \cdot \mathbf{x}$ that satisfies $\mathbf{x} = \mathbf{x}' \pmod 2$.

The next lemma shows that it is possible to transform a ciphertext $\mathbf{c}$ that encrypts m under key $\mathbf{s}$ for modulus q into a ciphertext $\mathbf{c}'$ that encrypts m under the same key $\mathbf{s}$ for modulus p. Since our basic encryption scheme is different from the basic scheme in the BGV scheme [11], the proof of Lemma 9 is slightly different from the proof in [11].

Lemma 9. *Let* q, p *be odd and* $q > p > 2$. *Let* $\mathbf{c} \in \mathbb{Z}_q^{n+1}$ *and* $\mathbf{c}' \leftarrow \mathbf{Scale}(\mathbf{x}, q, p, 2)$. *Then, for any* $\mathbf{s} \in \mathbb{Z}_q^{n+1}$, *if* $\langle \mathbf{c}, \mathbf{s} \rangle = \lfloor q/2 \rfloor \cdot m + E \pmod q$ *and* $\langle \mathbf{c}', \mathbf{s} \rangle = \lfloor p/2 \rfloor \cdot m + E' \pmod p$, *with* $|E| < q/4 - (q/p) \cdot \|\mathbf{s}\|_1 - q/(2p)$, *we have*

$$|E'| < \left(\frac{p}{q}\right) \cdot |E| + \|\mathbf{s}\|_1 + \frac{1}{2},$$
$$\left\lfloor \frac{2}{p} \left(\left\langle \mathbf{c}', \mathbf{s} \right\rangle \bmod p \right) \right\rceil = \left\lfloor \frac{2}{q} \left\langle \mathbf{c}', \mathbf{s} \right\rangle \bmod q \right\rceil \bmod 2. \tag{20}$$

Proof. By $\langle \mathbf{c}, \mathbf{s} \rangle = \lfloor q/2 \rfloor \cdot m + E \pmod q$, we have $\langle \mathbf{c}, \mathbf{s} \rangle - \lfloor q/2 \rfloor \cdot m - kq = E$ for some $k \in \mathbb{Z}$. For the same k, let $\langle \mathbf{c}', \mathbf{s} \rangle - \lfloor p/2 \rfloor \cdot m - kp$. Next we just prove $|\langle \mathbf{c}', \mathbf{s} \rangle - \lfloor p/2 \rfloor \cdot m - kp| < p/2$ in order to prove $\langle \mathbf{c}', \mathbf{s} \rangle - \lfloor p/2 \rfloor \cdot m - kp = E'$.

Since $\langle \mathbf{c}', \mathbf{s} \rangle = \langle (p/q) \cdot \mathbf{c}, \mathbf{s} \rangle + \langle \varepsilon, \mathbf{s} \rangle$, where $\|\varepsilon\|_\infty < 1$, we have

$$\begin{aligned}
&\left| \left\langle \mathbf{c}', \mathbf{s} \right\rangle - \left\lfloor \frac{q}{2} \right\rfloor \cdot m - kp \right| \\
&= \left| \left\langle \left(\frac{p}{q}\right) \cdot \mathbf{c}, \mathbf{s} \right\rangle - \left\lfloor \frac{q}{2} \right\rfloor \cdot m - kp + \langle \varepsilon, \mathbf{s} \rangle \right| \\
&= \left| \left(\frac{p}{q}\right) \cdot \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \left(\frac{p}{q}\right) \cdot E \right. \\
&\quad \left. + kp - \left\lfloor \frac{p}{2} \right\rfloor \cdot m - kp + \langle \varepsilon, \mathbf{s} \rangle \right| \\
&= \left| \left\lfloor \frac{p}{2} \right\rfloor \cdot m + \left(\frac{1}{2} - \frac{p}{2q}\right) \cdot m + \left(\frac{p}{q}\right) \cdot E \right. \\
&\quad \left. - \left\lfloor \frac{p}{2} \right\rfloor \cdot m + \langle \varepsilon, \mathbf{s} \rangle \right| \\
&= \left| \left(\frac{1}{2} - \frac{p}{2q}\right) \cdot m + \left(\frac{p}{q}\right) \cdot E + \langle \varepsilon, \mathbf{s} \rangle \right| \\
&< \left(\frac{p}{q}\right) \cdot |E| + \|\mathbf{s}\|_1 + \frac{1}{2} \\
&< \frac{p}{4}.
\end{aligned} \tag{21}$$

We thus have $\langle \mathbf{c}', \mathbf{s} \rangle - \lfloor p/2 \rfloor \cdot m - kp = E'$ and $|E'| < (p/q)|E| + \|\mathbf{s}\|_1 + 1/2$.

Since $\langle \mathbf{c}', \mathbf{s} \rangle \bmod p = \langle \mathbf{c}', \mathbf{s} \rangle - kp$ and $\mathbf{c} = \mathbf{c}' \pmod 2$, we have $\langle \mathbf{c}', \mathbf{s} \rangle \bmod p = \langle \mathbf{c}', \mathbf{s} \rangle - kp = \langle \mathbf{c}, \mathbf{s} \rangle - kq \pmod 2 = \langle \mathbf{c}, \mathbf{s} \rangle \bmod q \pmod 2$. By definition, $2p = 2q \pmod 2$. Since pq and 2 are coprime, it follows that $2/q = 2/p \pmod 2$. Modulo 2,

we have $(2/p)\cdot(\langle \mathbf{c}', \mathbf{s}\rangle \bmod p) = (2/q)\cdot(\langle \mathbf{c}, \mathbf{s}\rangle \bmod q) \pmod 2$. We thus get $\lfloor (2/p)(\langle \mathbf{c}', \mathbf{s}\rangle \bmod p)\rceil = \lfloor (2/q)(\langle \mathbf{c}, \mathbf{s}\rangle \bmod q)\rceil \bmod 2$. □

The following corollary follows immediately from Lemma 9.

Corollary 10. *Let q and p be two odd moduli. Let $\mathbf{c}$ be a ciphertext under the key $\mathbf{s}$ for the modulus q, where $m \leftarrow \lfloor (2/q)(\langle \mathbf{c}, \mathbf{s}\rangle \bmod q)\rceil \bmod 2$. Suppose that $\mathbf{s}$ is a completely short key, and assume that $|E| < q/4 - (q/p)\cdot\|\mathbf{s}\|_1 - q/(2p)$. Then we have $\mathbf{c}' \leftarrow \mathbf{Scale}(\mathbf{x}, q, p, 2)$, where $\mathbf{c}'$ is a ciphertext that encrypts the same message m under the key $\mathbf{s}$ for the modulus p, namely, $m \leftarrow \lfloor (2/p)(\langle \mathbf{c}', \mathbf{s}\rangle \bmod p)\rceil \bmod 2$. The noise of the new ciphertext $\mathbf{c}'$ has magnitude at most $(p/q)\cdot|E| + \|\mathbf{s}\|_1 + 1/2$.*

Since the noise magnitude in the ciphertext $\mathbf{c}$ depends on the length of the key vector $\mathbf{s}$, we must make the length of the key vector $\mathbf{s}$ short in order to use modulus switching to reduce the magnitude of the noise. For this purpose, we sample the key $\mathbf{s}$ from Gaussian distribution that is set to be as small as possible.

6. A Regev-Type FHE Scheme Using Modulus Switching

Next, we use modulus switching to construct a Regev-type FHE. This scheme is a leveled FHE scheme, in which the ith level needs a modulus q_i. The parameters in our scheme includes a ladder of decreasing modulus $\{q_i\}$ $(i = L-1, \ldots, 0)$, where a parameter L indicates the depth that a circuit can be evaluated. It is very important to choose reasonable modulus from q_L to q_0, and we will focus on the details on how to choose reasonable modulus in Section 8. Since the magnitude of q_L is related to the security parameter λ and different circuit depths result in different magnitude values of q_L, the performance of our scheme depends on the security parameter λ and the circuit depth L.

FHE.Setup(λ, L): input the security parameter λ and the circuit level L, output a ladder of decreasing modulus $\{q_i\}$ $(i = L-1, \ldots, 0)$, the noise distribution χ, and the dimension n. Note that χ and n are the same as in the previous basic encryption scheme.

FHE.KeyGen(n, i): For $i = L-1$ down to 0, do the following.

(1) Run $\mathbf{s}_i \leftarrow \mathbf{E.SecretKeygen}(1^n)$. Let $\mathbf{sk} = \{\mathbf{s}_i\}$.
(2) Run $\mathbf{A}_i \leftarrow \mathbf{E.PublicKeygen}(\mathbf{s}_i)$. Let $\mathbf{pk}_1 = \{\mathbf{A}_i\}$ $(i = L-1, \ldots, 0)$.
(3) Set $\mathbf{s}_i' \leftarrow \mathbf{s}_i \otimes \mathbf{s}_i \in \mathbb{Z}_q^{(n+1)^2}$.
(4) Run $\tau_{\mathbf{s}_{i+1}' \to \mathbf{s}_i} \leftarrow \mathbf{SwitchKeyGen}(\mathbf{s}_{i+1}', \mathbf{s}_i)$. (Omit this step when $i = L-1$.) Let $\mathbf{pk}_2 = \{\tau_{\mathbf{s}_{i+1}' \to \mathbf{s}_i}\}$ $(i = L-1, \ldots, 0)$.

Then output $\mathbf{sk} = \{\mathbf{s}_i\}$ and $\mathbf{pk} = (\mathbf{pk}_1, \mathbf{pk}_2)$.

FHE.Enc$(\mathbf{pk}, m)$: take a message $m \in \{0, 1\}$. Run $\mathbf{E.Enc}(\mathbf{A}_{L-1}, m)$.

FHE.Dec$(\mathbf{sk}, \mathbf{c}_i)$: assume that $\mathbf{c}_i$ is a ciphertext under the secret key $\mathbf{s}_i$. Run $\mathbf{E.Dec}(\mathbf{sk}, \mathbf{c}_i)$.

FHE.Add$(\mathbf{pk}, \mathbf{c}_1, \mathbf{c}_2)$: input two ciphertexts $\mathbf{c}_1$, $\mathbf{c}_2$ under the same secret key $\mathbf{s}_i$. If the two secret keys are different, we can use **FHE.Refresh** to refresh the two ciphertexts to the new two ciphertexts under the same secret key. Then, output $\mathbf{c}_3 \leftarrow \mathbf{c}_1 + \mathbf{c}_2$.

FHE.Mult$(\mathbf{pk}, \mathbf{c}_1, \mathbf{c}_2)$: input two ciphertexts $\mathbf{c}_1$, $\mathbf{c}_2$ under the same secret key $\mathbf{s}_i$. If the two secret keys are different, we can use **FHE.Refresh** to make it so. Compute $\mathbf{c}_3 \leftarrow \lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2)\rceil$, and the relative secret key is $\mathbf{s}_i' = \mathbf{s}_i \otimes \mathbf{s}_i$. Then, output $\mathbf{c}_4 \leftarrow \mathbf{FHE.Refresh}(\mathbf{c}_3, \tau_{\mathbf{s}_{i+1}' \to \mathbf{s}_i}, q_i, q_{i-1})$.

FHE.Refresh$(\mathbf{c}, \tau_{\mathbf{s}_i' \to \mathbf{s}_{i-1}}, q_i, q_{i-1})$: input ciphertext $\mathbf{c}$ under the secret key $\mathbf{s}_i'$ for modulus q_i. $\tau_{\mathbf{s}_i' \to \mathbf{s}_{i-1}}$ is the auxiliary information for key switching. The current and next modulus are q_i and q_{i-1}. Do the following.

(1) Key switching: comput $\mathbf{c}_1 \leftarrow \mathbf{SwitchKey}(\tau_{\mathbf{s}_i' \to \mathbf{s}_{i-1}}, \mathbf{c}, q_i)$, a ciphertext under the key $\mathbf{s}_{i-1}$ for q_i.
(2) Modulus switching: compute $\mathbf{c}_2 \leftarrow \mathbf{Scale}(\mathbf{c}_1, q_i, q_{i-1}, 2)$, a ciphertext under the key $\mathbf{s}_{i-1}$ for q_{i-1}.

In order to enable the correctness of the above leveled FHE scheme, we must choose the correct parameters. Next, we describe how to enable the correctness of this scheme.

7. Correctness

The correctness of the above leveled FHE scheme comes from the correctness of each step in homomorphic operations, that is, each step in **FHE.Add** and **FHE.Mult**. If the noise magnitude in ciphertext is below $q_{i-1}/4$ or $q_i/4$ after each step in homomorphic operations, correct decryption is guaranteed.

7.1. The Initial Noise. The initial ciphertext is output by **FHE.Enc** that just invokes **E.Enc**.

Lemma 11. *Let q_{L-1}, n, N be the parameters associated with* **FHE.Enc**. *χ is a B-bounded Gaussian distribution. The length of the noise in ciphertexts output by* **FHE.Enc** *is at most NB. If $NB < \lfloor q_{L-1}/2\rfloor/2$, correct decryption is guaranteed.*

Proof. $\mathbf{E.Enc}(\mathbf{A}_{L-1}, m)$ shows that $\lfloor q/2\rfloor\cdot\mathbf{m} + \mathbf{A}_{L-1}^T\cdot\mathbf{r} \in \mathbb{Z}_{q_{L-1}}^{(n+1)}$, where $\mathbf{r} \in \{0, 1\}^N$. We have $\mathbf{A}_{L-1}\cdot\mathbf{s}_{L-1} = \mathbf{e}$, where $\mathbf{e} \leftarrow \chi^N$ and $\mathbf{s}_{L-1} \leftarrow \mathbf{FHE.KeyGen}$. Then we get

$$\begin{aligned}
&\langle \mathbf{c}, \mathbf{s}_{L-1}\rangle \pmod{q_{L-1}} \\
&\quad = \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \mathbf{r}^T\mathbf{A}_{L-1}\mathbf{s}_{L-1} \pmod{q_{L-1}} \\
&\quad = \left\lfloor \frac{q}{2} \right\rfloor \cdot m + NB \pmod{q_{L-1}}.
\end{aligned} \tag{22}$$

According to Lemma 5, if $NB < \lfloor q_{L-1}/2\rfloor/2$, then correct decryption is guaranteed. □

7.2. The Correctness of Homomorphic Operations

Lemma 12. *Let* $\mathbf{c}_1$ *and* $\mathbf{c}_2$ *be two ciphertexts under* $\mathbf{s}_i$ *for* q_i*, where* $\langle \mathbf{c}_1, \mathbf{s}_i \rangle = \lfloor q_i/2 \rfloor \cdot m_1 + e_1 (\mathrm{mod}\ q_i)$ *and* $\langle \mathbf{c}_2, \mathbf{s}_i \rangle = \lfloor q_i/2 \rfloor \cdot m_2 + e_2 (\mathrm{mod}\ q_i)$ *with* $|e_1|, |e_2| \leqslant E < \lfloor q_i/2 \rfloor/2$*. Let* $\mathbf{c}_3 = \mathbf{c}_1 + \mathbf{c}_2$*. The noise magnitude of* $\mathbf{c}_3$ *is at most* $2E$*. If* $2E < \lfloor q_i/2 \rfloor/2$*, we have* $m_1 + m_2 \leftarrow \mathbf{FHE.Dec}(\mathbf{s}_i, \mathbf{c}_3)$*; namely,* $\mathbf{c}_3$ *can be correctly decrypted.*

Proof. The proof can be obtained easily from Lemma 6. □

The procedure of **FHE.Mult** consists of three steps, namely, the multiplication, and then the key switching and modulus switching. Next, we analyze the correctness of each step.

Lemma 13. *Let* $\mathbf{c}_1$ *and* $\mathbf{c}_2$ *be two ciphertexts under* $\mathbf{s}_i$ *for* q_i*, where* $\langle \mathbf{c}_1, \mathbf{s}_i \rangle = \lfloor q_i/2 \rfloor \cdot m_1 + e_1 (\mathrm{mod}\ q_i)$ *and* $\langle \mathbf{c}_2, \mathbf{s}_i \rangle = \lfloor q_i/2 \rfloor \cdot m_2 + e_2 (\mathrm{mod}\ q_i)$ *with* $|e_1|, |e_2| \leqslant E < \lfloor q_i/2 \rfloor/2$*. Let* $\mathbf{c}_3 = \lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \rceil$*, and let* $\mathbf{s}_i' = \mathbf{s}_i \otimes \mathbf{s}_i$*. The noise magnitude of* $\mathbf{c}_3$ *is at most* $12nBE$*. If* $12nBE < \lfloor q_i/2 \rfloor/2$*, we have* $m_1 m_2 \leftarrow \mathbf{FHE.Dec}(\mathbf{s}_i', \mathbf{c}_3)$*; that is,* $\mathbf{c}_3$ *can be correctly decrypted.*

Proof. The proof can be obtained easily from Lemma 6. □

We note that the noise after multiplication is $12nBE$ rather than E^2 like in many of the previous FHE schemes.

Lemma 14. *Let* $\mathbf{c}$ *be a ciphertext under* $\mathbf{s}_i' = \mathbf{s}_i \otimes \mathbf{s}_i$ *for* q_i*, where* $\langle \mathbf{c}, \mathbf{s}_i' \rangle = \lfloor q_i/2 \rfloor \cdot m + e (\mathrm{mod}\ q_i)$ *with* $|e| \leqslant E < \lfloor q_i/2 \rfloor/2$*. Let* $\mathbf{c}_1 \leftarrow \mathbf{SwitchKey}(\tau_{\mathbf{s}_i' \to \mathbf{s}_{i-1}}, \mathbf{c}, q_i)$*. The noise magnitude of* c_1 *is at most* $E + B(n+1)^2 \log q_i$*. If* $E + B(n+1)^2 \log q_i < \lfloor q_i/2 \rfloor/2$*, we have* $m \leftarrow \mathbf{FHE.Dec}(\mathbf{s}_{i-1}, \mathbf{c}_1)$*; namely,* $\mathbf{c}_1$ *can be correctly decrypted.*

Proof. By Lemma 7

$$\begin{aligned} \langle \mathbf{c}_1, \mathbf{s}_{i-1} \rangle &= \langle BitDecomp(\mathbf{c}), \mathbf{e}_1 \rangle + \langle \mathbf{c}, \mathbf{s}_i' \rangle (\mathrm{mod}\ q_i) \\ &= B(n+1)^2 \cdot \lceil \log q_i \rceil + \langle \mathbf{c}, \mathbf{s}_i' \rangle (\mathrm{mod}\ q_i) \\ &= B(n+1)^2 \cdot \lceil \log q_i \rceil + E + \left\lfloor \frac{q_i}{2} \right\rfloor \cdot m\ (\mathrm{mod}\ q_i), \end{aligned} \tag{23}$$

where $\mathbf{e}_1 \leftarrow \chi^N$ and $N = (n+1)^2 \cdot \lceil \log q_i \rceil$. If $E + B(n+1)^2 \log q_i < \lfloor q_i/2 \rfloor/2$, we have $m \leftarrow \mathbf{FHE.Dec}(\mathbf{s}_{i-1}, \mathbf{c}_1)$; namely, $\mathbf{c}_1$ can be correctly decrypted. □

Lemma 15. *Let* $\mathbf{c}_1$ *be a ciphertext of dimension* $n+1$ *under* $\mathbf{s}_{i-1}$ *for* q_i*, where* $\langle \mathbf{c}_1, \mathbf{s}_i \rangle = \lfloor q_i/2 \rfloor \cdot m_1 + e_1 (\mathrm{mod}\ q_i)$ *with* $|e_1| \leqslant E < \lfloor q_i/2 \rfloor/2$*. Let* $\mathbf{c}_2 \leftarrow \mathbf{Scale}(\mathbf{c}_1, q_i, q_{i-1}, 2)$*, The noise magnitude of* $\mathbf{c}_2$ *is at most* $(q_{i-1}/q_i) \cdot E + (n+1) \cdot B + 1/2$*. If* $(q_{i-1}/q_i) \cdot E + (n+1) \cdot B + 1/2 < \lfloor q_{i-1}/2 \rfloor/2$*, we have* $m \leftarrow \mathbf{FHE.Dec}(\mathbf{s}_{i-1}, \mathbf{c}_2)$*; that is,* $\mathbf{c}_2$ *can be correctly decrypted.*

Proof. By Corollary 10

$$\begin{aligned} \left| E' \right| &< \left(\frac{q_{i-1}}{q_i} \right) \cdot |E| + \left\| \mathbf{s}_{i-1} \right\|_1 + \frac{1}{2} \\ &< \left(\frac{q_{i-1}}{q_i} \right) \cdot E + (n+1) \cdot B + \frac{1}{2}. \end{aligned} \tag{24}$$

If $(q_{i-1}/q_i) \cdot E + (n+1) \cdot B + 1/2 < \lfloor q_{i-1}/2 \rfloor/2$, we have $m \leftarrow \mathbf{FHE.Dec}(\mathbf{s}_{i-1}, \mathbf{c}_2)$. □

8. Security and Parameters Settings

For a FHE scheme using modulus switching, it is most important to set up a reasonable ladder of decreasing modulus. The size of modulus is related to the dimension n of the LWE problem and the circuit depth L. Furthermore, the underlying security parameter λ is related to the dimension n of the LWE problem. However, it does not provide the concrete connection between the underlying security parameter and the dimension of the LWE problem in Regev's paper, nor the concrete parameters setting on its encryption scheme. It also does not provide the concrete method to set a concrete ladder of decreasing modulus based on a concrete security level and other parameters in the BGV scheme, even though BGV scheme is the first FHE scheme using modulus switching.

In this section, we will analyze the function between the lower bound in the dimension n of the LWE problem and the security level. Then we will give the method how to set the concrete ladder of decreasing modulus based on a certain security level and other parameters in our scheme.

8.1. The Dimension of the LWE Problem and the Security Level. In order to estimate the hardness of LWE for a concert set of parameters, we first consider the distinguishing attack LWE; namely, the adversary distinguishes (with some noticeable advantage) an LWE instance from uniformly random, which can result in that the semantic security of an LWE-based cryptosystem is to be broken with the same advantage. Given a point $\mathbf{b}$ that is either LWE instance or uniformly random. In order to do this attack, the adversary needs to find a short nonzero integral vector $\mathbf{v}$ such that $\mathbf{A}\mathbf{v} = 0 \bmod q$; namely, $\mathbf{v}$ is a short vector in $\Lambda_q^{\perp}(\mathbf{A}^t)$. Since $\Lambda_q^{\perp}(\mathbf{A}^t) = q \cdot \Lambda_q(\mathbf{A}^t)^*$, we have $\mathbf{v} = q \cdot \mathbf{y}$, where $\mathbf{y}$ is a short vector in the dual of the lattice $\Lambda_q^{\perp}(\mathbf{A}^t)$. Then the adversary tries to test whether the inner product $\langle \mathbf{v}, \mathbf{b} \rangle$ is close to zero modulo q. When $\mathbf{b}$ is a uniformly random instance, the test accepts with the probability exactly $1/2$. When $\mathbf{b} = \mathbf{A}^t\mathbf{s} + \mathbf{e}$, where $\mathbf{e}$ is sampled from a Gaussian distribution with standard deviation σ, we have $\langle \mathbf{v}, \mathbf{b} \rangle = \langle \mathbf{v}, \mathbf{A}^t\mathbf{s} \rangle + \langle \mathbf{v}, \mathbf{e} \rangle = \langle q \cdot \mathbf{y}, \mathbf{A}^t\mathbf{s} \rangle + \langle \mathbf{v}, \mathbf{e} \rangle = \langle \mathbf{v}, \mathbf{e} \rangle \bmod q$, which is essentially Gaussian with standard deviation $\|\mathbf{v}\| \cdot \sigma$. When $\|\mathbf{v}\| \cdot \sigma$ is not much larger than q, the adversary can distinguish the Gaussian from the uniform with advantage of being very close to $\exp(-\pi \cdot (\|\mathbf{v}\| \cdot \sigma/q)^2)$. In general, in order to do the distinguishing attack with high confidence, one needs $\|\mathbf{v}\| \leqslant q/(2\sigma)$, which need to reduce the basis well enough such that the shortest vector is of size roughly q/σ. We assume that the security depends on the ratio

q/σ. Furthermore, we assume that the adversary will spend all the attack running time doing lattice reduction according to the paper [26].

The key point is to compute inner product $\langle \mathbf{v}, \mathbf{e} \rangle$ modulo q for a enough short vector in the distinguishing attack described above, which do not use the secret $\mathbf{s}$ of LWE sample. It means that the distinguishing attack still work whether the secret $\mathbf{s}$ is sampled from a Gaussian distribution or uniform. Next, we analyze the relation between the dimension of LWE and the security level.

A short vector used in the distinguishing attack can be got from lattice reduction algorithm. From the analysis of lattice reduction algorithms by Gama and Nguyen [27], the Hermite factor is regarded as the dominant parameter in the runtime of the reduction and the quality of the reduced basis. A reduced basis $\mathbf{B}(\|\mathbf{b}_1\| \leqslant \|\mathbf{b}_2\|, \ldots, \leqslant \|\mathbf{b}_m\|)$ of an m-dimensional lattice Λ has the Hermite factor δ^m for $\delta \geq 1$ if $\|\mathbf{b}_1\| = \delta^m \cdot \det(\Lambda)^{1/m}$. The term δ is called a quality parameter. In addition, Lindner and Peikert perform the experiments in the paper [26], which predict the runtime required to achieve a given root-Hermite factor δ in random q-ary lattices arising from LWE. The result of their experiments show that the logarithm of the runtime should grow roughly linearly in $1/\log(\delta)$. In particular, for a random q-ary lattices arising from LWE, the time (in seconds) that is spent to compute a reduced basis of quality δ is conservatively estimated at least as follows:

$$\log(\text{time}) \geqslant \frac{1.8}{\log(\delta)} - 110. \tag{25}$$

We note that the runtime estimated in (25) can be also applied in here to analyze our scheme. First, the random q-ary lattices for experiments in the paper [26] include the random q-ary lattices arising from LWE where the secret was sampled from a Gaussian distribution. Second, the encryption scheme described in the paper [26] is also based on the same LWE problem like our scheme; namely, the secret is choose from a Gaussian distribution.

Recall that the basis is required to be reduced well enough such that the shortest vector is of size roughly q/σ in the distinguishing attack. Thus the adversary needs to reduce the basis enough so that $\|\mathbf{b}_1\| = q/\sigma$. Moreover, for a random q-ary lattice of rank n, the determinant is q^n with high probability. By the definition of quality parameter δ, a basis $\mathbf{B}$ that has quality parameter δ has $\|\mathbf{b}_1\| = \delta^m \cdot \det(\Lambda)^{1/m} = \delta^m \cdot q^{n/m}$. From the result in paper [28], when lattice reduction algorithms is applied to $\Lambda^{\perp}(\mathbf{A}^t)$, the shortest vectors are produced when $m = (n \cdot \log q / \log \delta)^{1/2}$. For simplicity, we take $\sigma = 1$ such that $\|\mathbf{b}_1\| = q/\sigma = q$, then we have

$$\begin{aligned} \log q &= \log\left(\delta^m \cdot q^{n/m}\right) \\ &= m \cdot \log \delta + \left(\frac{n}{m}\right) \log q \\ &= 2\left(\frac{n \cdot \log q}{\log \delta}\right)^{1/2}. \end{aligned} \tag{26}$$

We can solve for n and plug Equation (25) into it, then get $n = \log q \cdot (\log(\text{time}) + 110)/7.2$ which is a function between n and q/σ (recall $q = q/\sigma$). In order to ensure the time that is spent to reduce the basis at least 10^k, we need to set n to be at least

$$n \geqslant \frac{(\log(q/\sigma) \cdot (k + 110))}{7.2}. \tag{27}$$

We thus obtain the relation between the dimension of LWE and the security level. If we want to get 80 bit security level we need to set $n \geqslant \log(q/\sigma) \cdot 26.4$, for 128 bit security level we need to set $n \geqslant \log(q/\sigma) \cdot 33.1$.

8.2. Setting Concrete Parameters. Based on our scheme, we first set a concrete ladder of decreasing modulus. For a certain security level, we recommend specific dimension and modulus values for a specific circuit level L.

8.2.1. The Upper Bound of Noise. In order to obtain a suitable modulus, we need to find a common upper bound of noise for each circuit level.

Assume that we have a common upper bound E on noise magnitude, which means that the noise magnitude is at most E for all ciphertexts in all levels. Let $\mathbf{c}_1$ and $\mathbf{c}_2$ be two ciphertexts at level i. The noise magnitude is at most $12nB\,E$ after multiplication by following Lemma 11. Then, we apply the key switching, and the noise magnitude is at most $12nB\,E + B(n+1)^2 \log q_i$ by following Lemma 12. Finally, we apply modulus switching, and the noise magnitude in this stage is at most

$$\left(\frac{q_{i-1}}{q_i}\right) \cdot \left(12nB\,E + B(n+1)^2 \log q_i\right) + (n+1) \cdot B + \frac{1}{2}. \tag{28}$$

According to our assumption, the above equation is less than E. The dominant term is $12nB\,E$; thus, we have

$$\left(\frac{q_{i-1}}{q_i}\right) \cdot 12nB\,E < \frac{E}{2}, \tag{29}$$

$$\left(\frac{q_{i-1}}{q_i}\right) \cdot \left(B(n+1)^2 \log q_i\right) + (n+1) \cdot B + \frac{1}{2} < \frac{E}{2}. \tag{30}$$

We get $q_{i-1}/q_i < 1/(24nB)$ from Inequality (29), and we plug it into Inequality (30); then we have

$$\begin{aligned} &\left(\frac{1}{24nB}\right) \cdot \left(B(n+1)^2 \log q_i\right) + (n+1) \cdot B + \frac{1}{2} \\ &\quad \approx \left(\frac{1}{24}\right) \cdot (n+1) \cdot \log q_i + (n+1) \cdot B \\ &\quad \approx 2\,(n+1) \cdot B \\ &\quad < \frac{E}{2}. \end{aligned} \tag{31}$$

We thus set $E \approx 8(n+1) \cdot B$, which is the approximate common upper bound. We also get the ratio of q_{i-1} and q_i that is approximately $1/(24nB)$. Next we can set a concrete ladder of decreasing modulus.

Table 1: The parameters of our scheme.

L	n	$\log q_0$	$\log q_{L-1}$
10	9400	63	149
20	19100	67	714
30	29200	70	1092
40	39500	72	1446

8.2.2. A Concrete Ladder of Decreasing Modulus. We first consider the smallest modulus. At the level 0, the noise magnitude is at most $12nBE$ after multiplication. In order for the correction of decryption to occur, we need to ensure $12nBE < q_0/4$. We can take $q_0 \approx 48nBE \approx 384n^2B^2$, which is approximately the smallest modulus.

Since $q_{i-1}/q_i \approx 1/(24nB)$, we can derive $q_1, \ldots, q_{L-1}$; for example, $q_1 = 24nBq_0$, $q_2 = (24nB)^2 \cdot q_0, \ldots, q_{L-1} = (24nB)^{L-1} \cdot q_0$. We thus obtain a concrete ladder of decreasing modulus.

8.2.3. The Concrete Parameters of Our Scheme. According to (27) and the largest modulus q_{L-1}, we have $n \geqslant (\log(q_{L-1}/\sigma) \cdot (k + 110))/7.2$, which is the lower bound of the dimension of LWE. We use the Gaussian parameter $\sigma = 7$ from the experiment in [28]. Since $\log(q_{L-1}/\sigma) = \log((384 \cdot 24^{L-1} \cdot (nB)^{L+1})/7) \approx 8 + 5(L-1) + (L+1)(\log n + \log B)$, we have

$$n \geqslant \frac{(3 + 5L + (L+1)(\log n + \log B)) \cdot (k + 110)}{7.2}. \quad (32)$$

B is the bound of Gaussian, and we use $B = 2n$ from the statement in [20]. We then can obtain the lower bound of the dimension n from the circuit depth L as well as the security level.

For an 80-bit security level ($k = 80$) and different circuit depth L, we derive the parameters of our scheme, as shown in the Table 1.

8.3. Performance. The computational complexity of our scheme comes from homomorphic multiplication which includes three steps. The computational cost that computes the tensored ciphertext is $\widetilde{O}(n^2\log^2 q_j)$. The computational cost in the step of key switching is $\widetilde{O}(n^3\log^2 q_j)$. The computational cost in the step of modulus switching is $\widetilde{O}(n\log q_j)$. As a result, the per-gate computation in our scheme is $\widetilde{O}(n^3\log^2 q_j) = \widetilde{O}(n^3L^2)$. As a comparison, in the Bra12 scheme the per-gate computation is $\widetilde{O}(n^3\log^4 q_j)$. This shows that our scheme is more efficient than the Bra12 scheme.

8.4. Bootstrapping. We also can use bootstrapping to achieve a leveled FHE scheme. Furthermore, by using bootstrapping, we can obtain a pure FHE scheme with an assumption of circular security. There is a detailed explanation about bootstrapping in paper [29].

In our scheme the depth of a decryption circuit is $O(\log n + \log\log q)$. We can regard the above leveled FHE scheme as a somewhat homomorphic encryption scheme. As long as we set the depth of circuit $L > O(\log n + \log\log q)$, our scheme is bootstrappable.

9. Conclusions

We have constructed a leveled FHE scheme using modulus switching based on the Bra12 scheme, and our scheme improves the efficiency of the Bra12 scheme. The per-gate computation in our scheme is $\widetilde{O}(n^3\log^2 q_j) = \widetilde{O}(n^3L^2)$, while it is $\widetilde{O}(n^3\log^4 q_j)$ in the Bra12 scheme. Furthermore, we have derived a function of the lower bound in the dimension n of the LWE problem and the security parameter. For an 80-bit security level and several different depth parameters, we have shown the concrete values of the dimension n of the LWE problem and the modulus q in each level. These concrete values for different parameters are very important in the fully homomorphic scheme that leverages modulus switching technique for noise management, which cannot be solved before.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The first author would like to thank the Fund of Jiangsu Innovation Program for Graduate Education (no. CXLX12_0162), the Fundamental Research Funds for the Central Universities, Ningbo Natural Science Foundation (no. 2012A610067), and the Chinese National Scholarship Fund and also appreciate the benefit to this work from Projects in science and technique of Ningbo municipal (no. 2012B82003). The forth author would like to thank Ningbo Natural Science Foundation (no. 2013A610071).

References

[1] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09)*, pp. 169–178, ACM, Bethesda, Md, USA, June 2009.

[2] C. Gentry and S. Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits," in *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS '11)*, pp. 107–109, IEEE Computer Society, October 2011.

[3] C. Gentry and S. Halevi, "Implementing Gentry's fully-homomorphic encryption scheme," in *Advances in Cryptology—Eurocrypt 2011*, K. Paterson, Ed., pp. 129–148, Springer, Berlin, Germany, 2011.

[4] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the AES circuit," in *Advances in Cryptology—Crypto 2012*, R. Safavi-Naini and R. Canetti, Eds., pp. 850–867, Springer, Berlin, Germany, 2012.

[5] C. Gentry, S. Halevi, C. Peikert, and N. P. Smart, "Ring switching in BGV-style homomorphic encryption," in *Security*

and Cryptography for Networks, I. Visconti and R. Prisco, Eds., pp. 19–37, Springer, Berlin, Germany, 2012.

[6] C. Gentry, S. Halevi, and N. P. Smart, "Better bootstrapping in fully homomorphic encryption," in *Public Key Cryptography—Pkc 2012*, M. Fischlin, J. Buchmann, and M. Manulis, Eds., pp. 1–16, Springer, Berlin, Germany, 2012.

[7] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *Public Key Cryptography—Pkc 2010*, P. Nguyen and D. Pointcheval, Eds., pp. 420–443, Springer, Berlin, Germany, 2010.

[8] D. Stehlé and R. Steinfeld, "Faster fully homomorphic encryption," in *Advances in Cryptology—Asiacrypt 2010*, M. Abe, Ed., pp. 377–394, Springer, Berlin, Germany, 2010.

[9] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Advances in Cryptology—Crypto 2011*, P. Rogaway, Ed., pp. 505–524, Springer, Berlin, Germany, 2011.

[10] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS '11)*, pp. 97–106, IEEE Computer Society, October 2011.

[11] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proceedings of the 3rd Conference on Innovations in Theoretical Computer Science*, pp. 309–325, ACM, Cambridge, Mass, USA, January 2012.

[12] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *Advances in Cryptology—Crypto 2012*, R. Safavi-Naini and R. Canetti, Eds., pp. 868–886, Springer, Berlin, Germany, 2012.

[13] Z. Brakerski, C. Gentry, and S. Halevi, "Packed ciphertexts in LWE-based homomorphic encryption," in *Public-Key Cryptography—Pkc 2013*, K. Kurosawa and G. Hanaoka, Eds., pp. 1–13, Springer, Berlin, Germany, 2013.

[14] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," in *Advances in Cryptology—Eurocrypt 2010*, H. Gilbert, Ed., pp. 24–43, Springer, Berlin, Germany, 2010.

[15] J. Coron, A. Mandal, D. Naccache, and M. Tibouchi, "Fully homomorphic encryption over the integers with shorter public keys," in *Advances in Cryptology—Crypto 2011*, P. Rogaway, Ed., pp. 487–504, Springer, Berlin, Germany, 2011.

[16] J. Coron, D. Naccache, and M. Tibouchi, "Public key compression and modulus switching for fully homomorphic encryption over the integers," in *Advances in Cryptology—Eurocrypt 2012*, D. Pointvheval and T. Johansson, Eds., pp. 446–464, Springer, Berlin, Germany, 2012.

[17] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pp. 1219–1234, ACM, New York, NY, USA, May 2012.

[18] J. Alperin-Sheriff and C. Peikert, "Practical bootstrapping in quasilinear time," in *Advances in Cryptology—Crypto 2013*, R. Canetti and J. Garay, Eds., pp. 1–20, Springer, Berlin, Germany, 2013.

[19] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology—Crypto 2013*, R. Canetti and J. Garay, Eds., pp. 75–92, Springer, Berlin, Germany, 2013.

[20] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pp. 84–93, ACM, Baltimore, Md, USA, November 2005.

[21] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Advances in Cryptology—Eurocrypt 2010*, H. Gilbert, Ed., pp. 1–23, Springer, Berlin, Germany, 2010.

[22] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09)*, pp. 333–342, ACM, Bethesda, Md, USA, June 2009.

[23] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, "Classical hardness of learning with errors," in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pp. 575–584, ACM, June 2013.

[24] D. Micciancio and P. Mol, "Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions," in *Advances in Cryptology—Crypto 2011*, P. Rogaway, Ed., pp. 465–484, Springer, Berlin, Germany, 2011.

[25] D. Micciancio and C. Peikert, "Trapdoors for lattices: simpler, tighter, faster, smaller," in *Advances in Cryptology—Eurocrypt 2012*, D. Pointcheval and T. Johansson, Eds., pp. 700–718, Springer, Berlin, Germany, 2012.

[26] R. Lindner and C. Peikert, "Better key sizes (and Attacks) for LWE-based encryption," in *Topics in Cryptology—Ct-Rsa 2011*, A. Kiayias, Ed., pp. 319–339, Springer, Berlin, Germany, 2011.

[27] N. Gama and P. Q. Nguyen, "Predicting lattice reduction," in *Advances in Cryptology—Eurocrypt 2008*, N. Smart, Ed., pp. 31–51, Springer, Berlin, Germany, 2008.

[28] D. Micciancio and O. Regev, "Lattice-based cryptography," in *Post-Quantum Cryptography*, D. Bernstein, J. Buchmann, and E. Dahmen, Eds., pp. 147–191, Springer, Berlin, Germany, 2009.

[29] C. Zhigang, W. Jian, C. Liqun, and S. Xinxia, "Review of how to construct a fully homomorphic encryption scheme," *International Journal of Security and Its Applications*, vol. 8, no. 2, pp. 221–230, 2014.