IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

Fully homomorphic encryption based on the ring learning with rounding problem

ISSN 1751-8709
Received on 9th August 2018
Revised 22nd May 2019
Accepted on 3rd July 2019
E-First on 12th August 2019
doi: 10.1049/iet-ifs.2018.5427
www.ietdl.org

Fucai Luo[1,2,3] ✉, *Fuqun Wang*[4,5], *Kunpeng Wang*[2,3], *Kefei Chen*[4,5]

[1]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, People's Republic of China*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing, People's Republic of China*
[3]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, People's Republic of China*
[4]*Department of Mathematics, Hangzhou Normal University, Hangzhou, People's Republic of China*
[5]*Westone Cryptologic Research Center, Beijing, People's Republic of China*
✉ *E-mail: luofucai@iie.ac.cn*

Abstract: Almost all existing well-known fully homomorphic encryption (FHE) schemes, which are based on either the learning with errors (LWE) or the ring LWE problem, require expensive Gaussian noise sampling. In this study, the authors propose an FHE scheme based on the ring learning with rounding (RLWR) problem. The learning with rounding (LWR) problem was proposed as a deterministic variant of LWE, while the RLWR is a variant of LWR. Sampling an LWR instance does not require Gaussian noise sampling process, and neither does an RLWR instance. Thus, our FHE scheme can be instantiated without the need for Gaussian noise sampling. To implement homomorphic operations, we devise a specific relinearisation method. Furthermore, we also prove that our RLWR-based FHE scheme is IND-CPA secure under RLWR assumption.

1 Introduction

Fully homomorphic encryption (FHE) is a very attractive cryptographic primitive that allows us to perform arbitrary efficient and compact computations on encrypted data, without decrypting it first. The first candidate FHE construction, which is based on ideal lattices, was proposed by Gentry [1]. In addition, Gentry put forward a remarkable 'bootstrapping' theorem for the first time. The 'bootstrapping' means that the scheme is capable of evaluating its own (augmented) decryption circuit. What is more, if a scheme is bootstrappable and weakly 'circular secure', then it can be converted into a full-fledged one, which enables arbitrary large homomorphic computations on encrypted data. However, his solution is complex and can only homomorphically evaluate 'low degree' polynomials without 'bootstrapping'.

While the scheme proposed in [1] is impractical, a sequence of vastly more efficient schemes has since emerged [2–9]. All of the schemes enjoy higher efficiency and stronger security, as they are based on either the learning with errors (LWE) [10] or the ring learning with errors (RLWE) [11] problem, which has a simple algebraic structure and rigorous security reductions from some apparently intractable worst-case lattice problems [12–14]. Unfortunately, these LWE/RLWE-based FHE schemes suffer complex and time-consuming Gaussian noise sampling (though this is not the main bottleneck). This is mainly because they used LWE/RLWE instances to generate the public and the secret keys, while the LWE/RLWE instances contain a series of noises, which are usually obtained by Gaussian sampling. In particular, some LWE-based PKE schemes (e.g. [6, 15]) need to sample Gaussian noise in the encryption phase, which seriously weakens the schemes' efficiency. Moreover, sampling Gaussian noise in every encryption phase will bring some potential side-channel vulnerabilities [16, 17].

1.1 Motivations

We noted that Micciancio and Walter [18] devised a new discrete Gaussian sampling algorithm over the integers, which is more efficient and more easily implemented in constant time (outperforms the previous algorithms), but their algorithm is very complex and involves a time-memory trade-off. In other words, in order to achieve Gaussian sampling in constant time, their algorithm requires large memory. For the Gaussian noise in RLWE instances that is a polynomial, the Gaussian sampling process involved is more complex. This can be found in [11]. Indeed, even if we use the coefficient-wise Gaussian sampling algorithm proposed in [18] to sample the Gaussian noise and ignore the required memory, the overhead is at least $O(N)$, where N is the degree of the polynomial.

As a matter of course, this raises a question: is it possible to remove the Gaussian noise sampling in building an FHE scheme while preserving (almost) the same security level as those based on the LWE/RLWE problem, and without requiring any other costly conditionsActually, this is valuable theoretically and practically.

Costache and Smart [19] intended to resolve this question in the affirmative by giving the first ring learning with rounding (RLWR)-based FHE scheme. Essentially, their construction is an analogue of Fan–Vercauteren (FV) [7] and Brakerski–Gentry–Vaikuntanathan (BGV) [4], because they chose RLWR instances (we will describe the RLWR problem in Section 2.2) as the public key and the private key therein as the secret key. Then they used the *relinearisation* and *modulus switching* techniques to perform homomorphic operations. More precisely, in their scheme, the public key consists of ℓ pairs of RLWR instances $\{v_i, u_i\}_{i\in[\ell]} \in R_q \times R_p$ (skip to preliminaries for definitions of the ring R and T) satisfying $u_i - (p/q)sv_i = e_i$, where polynomial e_i is distributed uniformly with coefficients in [–1/2, 1/2], and the coefficients of $s \in R$ are chosen from {–1, 0, 1}, with Hamming weight h. Then the secret key is set as $\boldsymbol{s} = (-(p/q)s, 1)$. For any message $\mu \in R_t$, they computed the corresponding ciphertext vector $\boldsymbol{c} = (v, w) \in R_q \times R_p$, using the method similar to Regev's public-key encryption scheme [10]. The correctness of decryption is guaranteed by $\mu = \lceil (1/\Delta_t)\langle \boldsymbol{c}, \boldsymbol{s}\rangle \rfloor = \lceil (1/\Delta_t)(w - (p/q)vs) \rfloor \bmod t$, where $\Delta_t = \lceil (p/t) \rfloor$. For two ciphertext vectors $\boldsymbol{c} = (v, w)$, $\boldsymbol{c}' = (v', w')$ decrypting to messages $\mu, \mu' \in R_t$, where $v, v' \in R_q$ and $w, w' \in R_p$, they performed homomorphic multiplication by three steps: first operated tensor product on ciphertexts, then reduced the result by Δ_t, and relinearised at last.

Table 1 Comparison between our scheme and FV scheme

FHE	sk, pk, $\boldsymbol{c}$	Modulus	Security loss	Gaussian noise
FV scheme	2	$\hat{q} = N(O(N^{3/2}))^{L+O(1)}$	$O(1)$	yes
our scheme	$\ell + 1$	$p = (O(N^{5/2}))^{L+O(1)}$	$O(N^{3c})(c \geq 1)$	no

In the first step, by adding the two middle terms of the tensor product ciphertext $\boldsymbol{c} \otimes \boldsymbol{c}'$ up, they formed the three-element ciphertext

$$\boldsymbol{c}_{\text{mult}} = (vv', wv' + vw', ww') = (a_1, a_2, a_3), \tag{1}$$

which corresponds to the three-element secret key

$$\boldsymbol{s}_{\text{mult}} = \left(\left(\frac{p}{q}\right)^2 s^2, \ -\frac{p}{q}s, 1 \right).$$

Then, the message $\mu\mu'$ can be recovered from the above three-element ciphertext via the following equality:

$$\begin{aligned} \tilde{\mu} &= \lceil \frac{1}{\Delta_t^2}(a_3 - \frac{p}{q}sa_2 + (\frac{p}{q})^2 s^2 a_1) \rfloor \\ &= \lceil \frac{1}{\Delta_t^2}(ww' - \frac{p}{q}s(wv' + vw') + (\frac{p}{q})^2 s^2 vv') \rfloor \\ &= \lceil \frac{1}{\Delta_t^2}(w - \frac{p}{q}vs) \cdot (w' - \frac{p}{q}v's) \rfloor \\ &= \lceil \frac{1}{\Delta_t}(\Delta_t\mu + p\varphi + e) \cdot \frac{1}{\Delta_t}(\Delta_t\mu' + p\varphi' + e') \rfloor \\ &= \mu\mu' + \lceil \hat{e} \rfloor \bmod t . \end{aligned} \tag{2}$$

The correctness of the above decryption requires the last noise to be small, i.e. $\lceil \hat{e} \rfloor = 0$. Here comes the problem. As all polynomial arithmetics are performed over the ring R or T, we only need to focus on the modular operations. We take the term a_1 in (1) into consideration, as it suffices to help us find the flaws of (2).

Since term a_1 should be kept in R_q, it holds that $a_1 = vv' + q\varphi_1$, where $\varphi_1 \in R$ for the components $v, v' \in R_q$. Thus, we rectify (2) by adding the term $q\varphi_1$ into the second equality of (2), then we get

$$\lceil \frac{1}{\Delta_t^2}(ww' - \frac{p}{q}s(wv' + vw') + \left(\frac{p}{q}\right)^2 s^2 vv' + \left(\frac{p}{q}\right)^2 s^2 q\varphi_1) \rfloor .$$

Consequently, the last equality of (2) turns into $\mu\mu' + \lceil \hat{e} \rfloor + \lceil (1/\Delta_t^2)(p/q)^2 s^2 q\varphi_1 \rfloor \bmod t$, of which the component $\lceil (1/\Delta_t^2)(p/q)^2 s^2 q\varphi_1 \rfloor$ is non-zero and cannot be eliminated by taking it modulo t. This leads to the failure of decryption. Essentially, the ciphertext vector $\boldsymbol{c} = (v, w) \in R_q \times R_p$ involves two different moduli, which makes the tensor product intractable. We call this moduli problem.

1.2 Our contributions

In this study, we circumvent the moduli problem existed in [19] by generating the public and the secret keys using the same method as in dual-Regev public-key encryption [20]. To implement homomorphic operations, we devise a specific relinearisation method. We describe in this study, the RLWR-based version, as it is straightforward to construct a LWR-based version of our scheme. Through our delicate analysis for noise growth in homomorphic operations, we have that the noise in our scheme grows linearly with every homomorphic multiplication ($B \to B \cdot \text{poly}(N)$). This growth pattern is the same as in the FV scheme. Specifically, the noise of homomorphic multiplication is of polynomial growth, where the growth factor is $N^{3/2}O(\log p + 4\log N)$. Thus, after L levels of homomorphic multiplications, the magnitude of the noise is at most $(N^{3/2}O(\log p + 4\log N))^{L+1}$.

We remark that our scheme supports Gentry's 'bootstrapping' theorem, and all the 'standard' methodologies used in FV and BGV schemes apply to our scheme. This is due to the fact that the ciphertext structure and the decryption procedure in our scheme are very similar to theirs. Furthermore, we prove that our RLWR-based FHE scheme is IND-CPA secure under the RLWR assumption (its hardness can be reduced to some worst-case ideal lattice problems, relying on the hardness of the RLWE problem [11] and the reduction from RLWE to RLWR [21]).

However, removing the Gaussian noise sampling is not achieved without a penalty: the dimensions of the secret key sk, the public key pk and the ciphertext $\boldsymbol{c}$ in our scheme are all bigger than that in the FV scheme (up to a logarithmic factor), as our key generation method is the same as in dual-Regev pk encryption [20]. This weakness is inherited from the dual-Regev encryption, in which the dimensions of the secret key and ciphertext are larger, by about a logarithmic factor than that in Regev encryption [10]. Moreover, modulus p in our scheme is slightly bigger than the modulus $\hat{q}$ in the FV scheme by a small polynomial factor, and this is also caused by our different key generation method. These can be seen clearly in Table 1. Note that our scheme was compared only with the FV scheme, without comparing with the BGV scheme, because the noise management mode of our scheme is the same as that of the FV scheme, and the FV scheme is superior to the BGV scheme in noise management and security [3, 7].

In [22], we consider the levelled FHE scheme where the depth of circuits is polynomial L. The homomorphic evaluation capability, efficiency and security of the FHE scheme mainly depend on the size of the underlying modulus. Under the same ring of integers $R = \mathbb{Z}[X]/(\Phi_m(X))$ of degree $N = \varphi(m)$, we focus our attention on the dimensions of the secret key sk, the public key pk, and ciphertext $\boldsymbol{c}$. [21] Here, the security loss is caused by the reduction between the security of FHE scheme and the RLWE problem. Though our FHE scheme is based on the RLWR problem, we can get the security loss by the reduction between RLWE and RLWR (see Theorem 1 in Section 2.3).

1.3 Related work

Luo *et al.* [9] also observed the moduli problem existed in [19] and came up with a workable LWR-based FHE scheme. The construction of Luo *et al.* follows the approximate eigenvector approach proposed by Gentry *et al.* [8], and uses specific matrix multiplication to perform the homomorphic multiplication to avoid the moduli problem. Li *et al.* [23] also used the approximate eigenvector approach to construct LWR-based levelled FHE scheme. However, both constructions are limited to encrypting bits, a restriction inherited from the scheme presented in [8]. In contrast, in this work, our construction can encrypt polynomials.

Cheon *et al.* [24] proposed a novel PKE scheme, called Lizard, without relying on the leftover hash lemma [25] and Gaussian noise sampling in the encryption phase. In Lizard, the public key and secret key are chosen from the LWE instances as in the previous LWE-based PKEs (e.g. [10, 14, 20]), whereas the encryption procedure of Lizard removes several least significant bits of each component of the computed vector rather than adding an auxiliary error vector. This encryption approach does not need the leftover hash lemma and Gaussian noise sampling and thus speed up encryption to a large extent. Actually, they can avoid using the leftover hash lemma because they use the rounding function to tailor the computed vector containing the plaintext and then obtain the corresponding ciphertext, and the ciphertext is of smaller size and forms an LWR instance with the public key. Hence, the security of Lizard is based on the hardness assumptions of the LWE and LWR problems. In addition, they also constructed ring variant of Lizard and IND-CCA Lizard, and they claimed that the IND-CCA Lizard is comparable to NTRU (e.g. [26, 27]) in terms of both Enc/Dec speed and ciphertext size by showing the

experimental results: in their single-core implementation on a laptop, the encryption and decryption of Lizard with 256-bit plaintext space under 128-bit quantum security take 0.016 and 0.037 ms. As for the security, the Lizard has stronger security guarantee than NTRU, in the sense that the Lizard has provable security from the LWE and LWR problems which have reductions from the standard lattice problems (e.g. GapSVP, SIVP), but NTRU does not. Note that in Lizard they also made use of the sparse small secrets as we do in this work.

In the aspect of homomorphism, the IND-CPA Lizard supports the bounded number of homomorphic additions but does not support homomorphic multiplication. This is because if one performs the homomorphic multiplication on Lizard, he/she has to use relinearisation technique, but that will produce two non-erasable noise terms (caused by $p\boldsymbol{e}/q$, where $\boldsymbol{e}$ is the error vector in LWE instances), resulting in the failure of decryption. This problem is very similar to the moduli problem existed in Costache and Smart's [19] RLWR-based FHE scheme.

1.4 Roadmap

In Section 2, we list some notations throughout this work and introduce the RLWE and RLWR problems. In Section 3, we describe a basic RLWR-based encryption scheme and give its full security proof, and we also present our specific relinearisation technique. In Section 4, we present our RLWR-based FHE scheme and give the concrete analysis, which includes correctness, homomorphic properties, and parameters. Finally, we conclude the paper in Section 5.

2 Preliminaries

Here, we give some notations beforehand. Also, for completeness, we recall the `BitDecomp` and `Powersoftwo` techniques [3, 4] that will be used in our specific relinearisation technique.

Notations. Let PPT denote probabilistic polynomial-time. We define a ring of integers $R = \mathbb{Z}[X]/(\Phi_m(X))$, where $\Phi_m(X)$ is the mth cyclotomic polynomial of degree $N = \varphi(m)$, the Euler's totient of m. Moreover, we define $T = \mathbb{R}[X]/(\Phi_m(X))\mathrm{mod}1$, which will be used in the analysis of noise caused by the scaled rounding function. For the positive integers q and p, we define the quotient rings $R_q = R/qR = \mathbb{Z}_q[X]/(\Phi_m(X))$ and $R_p = R/pR = \mathbb{Z}_p[X]/(\Phi_m(X))$, where all coefficients of polynomials in R_q and R_p are in $(-q/2, q/2]$ and $(-p/2, p/2]$, respectively. In this study, all logarithms on q and p are base 2 and all arithmetics are performed over the ring R or T when division is used. For ease of use, we let $[n] \triangleq \{1, \ldots, n\}$. We say that a function $\mathrm{negl}(n)$ is negligible if there are not any polynomial fractions smaller than the $\mathrm{negl}(n)$ for sufficiently large n. All definitions that follow apply to R, R_q, and R_p.

We denote vectors in bold italic lowercase (e.g. $\boldsymbol{x}$) and matrices in bold italic uppercase (e.g. $\boldsymbol{A}$). Let $\boldsymbol{x}[i]$ refer to the ith entry of $\boldsymbol{x}$. For any $x \in R$, we denote by $\lfloor x \rceil$, $\lceil x \rceil$, and $\lceil x \rfloor$ the rounding of all coefficients of x down, up, and or to the nearest integers, these notations also apply to vectors and matrices, e.g. for $\boldsymbol{x} = (x_1, \ldots, x_\ell)$, $\lfloor \boldsymbol{x} \rceil = (\lfloor x_1 \rceil, \ldots, \lfloor x_\ell \rceil)$. Moreover, we let $x \leftarrow_r \mathcal{D}$ denote that x is randomly sampled from the distribution $\mathcal{D}$.

Definition 1: (B-bounded distributions [8, 28]). A distribution ensemble $\{\mathcal{S}_n\}_{n \in \mathbb{N}}$, supported over the integers, is called B-bounded if

$$\Pr_{e \leftarrow_r \mathcal{S}_n}[|e| > B] = \mathrm{negl}(n).$$

We say a B-bounded distribution e is balanced if $\Pr[e \geq 0] \geq \frac{1}{2}$ and $\Pr[e \leq 0] \geq \frac{1}{2}$.

Operations. The dot product of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ over R_p^n is denoted as $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_p = \sum_{i=1}^{n} \boldsymbol{x}[i] \cdot \boldsymbol{y}[i] \mathrm{mod} p$ and $\langle a \rangle_p$ refers to taking all coefficients of polynomial $a \in R$ modulo p. To compare with the FV scheme, in the following, we define the norm similar to that defined in the FV scheme; i.e. for ring element $x \in R$, we let $||x|| \in R$ refer to the Euclidean norm of x's coefficient vector, and $\delta_R = \max\{||xy||/||x||||y|| : x, y \in R\}$ refer to the expansion factor of R, where it holds that $\delta_R \leq \sqrt{N}$ (recall that $N = \varphi(m)$ is the degree of the cyclotomic polynomial for R) by Cauchy–Schwarz. We express the tensor product of two vectors in R^n as $\boldsymbol{x} \otimes \boldsymbol{y}$, which is n^2-dimensional vector containing all elements of the form $\boldsymbol{x}[i] \cdot \boldsymbol{y}[j]$. It is easy to verify that $\langle \boldsymbol{x} \otimes \boldsymbol{y}, \boldsymbol{v} \otimes \boldsymbol{w} \rangle = \langle \boldsymbol{x}, \boldsymbol{v} \rangle \cdot \langle \boldsymbol{y}, \boldsymbol{w} \rangle$.

$\mathrm{BitDecomp}_q(\boldsymbol{x} \in R_q^n)$: which decomposes $\boldsymbol{x} \in R_q^n$ into its bit representation, i.e. we have $\boldsymbol{x} = \sum_{i=0}^{\lceil \log q \rceil - 1} 2^i \boldsymbol{x}_i (\mathrm{mod} q)$, where all vectors $\boldsymbol{x}_i$ are in R_2^n. Output the vector

$$(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{\lceil \log q \rceil - 1}) \in R_2^{n\lceil \log q \rceil}.$$

$\mathbf{Powersoftwo}_q(\boldsymbol{y} \in R_q^n)$: which outputs the vector

$$\left(\boldsymbol{y}, 2\boldsymbol{y}, \ldots, 2^{\lceil \log q \rceil - 1}\boldsymbol{y}\right) \in R_q^{n\lceil \log q \rceil}.$$

Then, for the vectors $\boldsymbol{x}, \boldsymbol{y} \in R_q^n$, we have

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_q = \langle \mathrm{BitDecomp_q}(\boldsymbol{x}), \mathrm{Powersoftwo_q}(\boldsymbol{y}) \rangle_q.$$

In our security proof, we will use the following variant of the standard leftover hash lemma [25].

Lemma 1: Let λ, q and $\ell \geq \log q + 2\lambda$ be integers, and let $x \leftarrow_r R_q$. For vectors $\boldsymbol{a} \leftarrow_r R_q^\ell$ and $\boldsymbol{s} \leftarrow_r R_2^\ell$, we have

$$\Delta((\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s} \rangle_q), (\boldsymbol{a}, x)) \leq \frac{1}{2}\sqrt{\left(\frac{q}{2^\ell}\right)^N} < 2^{-\lambda N},$$

where $\Delta(X, Y)$ denotes the statistical distance between two distributions X, Y.

2.1 RLWE problem

The LWE problem was initially introduced by Regev [10], who also showed a quantum reduction [10] from certain worst-case lattice problems to the LWE problem. Subsequently, some classical reductions were presented in [12, 14]. The RLWE problem is a variant of the LWE problem and was firstly introduced by Lyubashevsky *et al*. [11], who also proved that the RLWE problem is at least as hard as the well-established worst-case GapSVP problem on ideal lattices. Here we consider a simplified version of the RLWE problem that is borrowed from [4].

Definition 2: For security parameter λ, let $q = q(\lambda) \geq 2$ be an integer. Let $R = \mathbb{Z}[X]/(\Phi_m(X))$, $R_q = R/qR$ be polynomial rings and $\chi = \chi(\lambda)$ be a distribution over R, where $\Phi_m(X)$ is a mth cyclotomic polynomial which has degree $N = \varphi(m)$, the totient of m. The $\mathrm{RLWE}_{N,q,\chi}$ problem is to distinguish the following two distributions: in the first distribution, one first draws $s \leftarrow_r R_q$ uniformly and then samples $(a_i, b_i) \in R_q^2$ by sampling $a_i \leftarrow_r R_q$, $e_i \leftarrow_r \chi$, and setting $b_i = a_i s + e_i$. In the second distribution, one samples (a_i, b_i) uniformly from R_q^2. The $\mathrm{RLWE}_{N,q,\chi}$ assumption is that the $\mathrm{RLWE}_{N,q,\chi}$ problem is infeasible.

2.2 RLWR problem

The LWR problem, which can be regarded as a deterministic variant of the LWE problem, was firstly introduced by Banerjee *et al.* [29]. Actually, the LWR problem can be seen as a deterministic variant of the LWE problem, for the implicit noise in LWR is deterministic, and thus derandomises the Gaussian noise in LWE. In particular, the implicit noise in LWR is <1/2, while the noise in LWE is B-bounded (e.g. it is required to set $B > 2\sqrt{n}$ due to

security reasons [10]). We firstly recall the scaled rounding function $\lceil \cdot \rfloor_p$ [29] as follows: for $p < q$

$$\begin{aligned} \lceil \cdot \rfloor_p : \quad \mathbb{Z}_q &\to \mathbb{Z}_p \\ a &\mapsto \lceil \frac{p}{q} a \rfloor . \end{aligned}$$

The scaled rounding function $\lceil \cdot \rfloor_p$ denotes the component-wise rounding if the entry is a vector or matrix, and coefficient-wise rounding when the entry is a ring element.

The RLWR problem [29], which is a variant of the LWR problem, can be defined in the same way as the RLWE problem.

Definition 3: For security parameter λ, let $q = q(\lambda) \geq 2, p = p(\lambda) \geq 2$ be integers, where $q > p$. Let $R = \mathbb{Z}[X]/(\Phi_m(X)), R_q = R/qR, R_p = R/pR$ be polynomial rings, where $\Phi_m(X)$ is a mth cyclotomic polynomial of degree $N = \varphi(m)$, the totient of m. The $\mathrm{RLWR}_{N,q,p}$ problem is to distinguish the following two distributions: in the first distribution, one first draws $s \leftarrow_r R_q$ uniformly and then samples $(a_i, b_i) \in R_q \times R_p$ by sampling $a_i \leftarrow_r R_q$ and setting $b_i = \lceil a_i s \rfloor_p$. In the second distribution, one samples (a_i, b_i) uniformly from $R_q \times R_p$. The $\mathrm{RLWR}_{N,q,p}$ assumption is that the $\mathrm{RLWR}_{N,q,p}$ problem is infeasible.

Lemma 2: If $p|q$, then the distribution $\lceil \boldsymbol{u} \rfloor_p$ is uniform over R_p^k for $\boldsymbol{u} \leftarrow_r R_q^k$ (i.e. the distribution $\lceil \boldsymbol{u} \rfloor_p$ is equivalent to R_p^k).

Proof: First, for any $\boldsymbol{u} \leftarrow_r R_q^k$, we have $\lceil \boldsymbol{u} \rfloor_p = \lceil (p/q)\boldsymbol{u} \rfloor \in R_p^k$. As $p|q$, then for any $\boldsymbol{v} \in R_p^k$, we can always find a vector $\boldsymbol{u}' = (q/p)\boldsymbol{v} \in R_q^k$, such that $\boldsymbol{v} = \lceil (p/q)\boldsymbol{u}' \rfloor \in \lceil \boldsymbol{u} \rfloor_p$. In brief, the distributions $\lceil \boldsymbol{u} \rfloor_p$ (for $\boldsymbol{u} \leftarrow_r R_q^k$) and R_p^k satisfy: $\lceil \boldsymbol{u} \rfloor_p \subseteq R_p^k, R_p^k \subseteq \lceil \boldsymbol{u} \rfloor_p$. Thus, we have $\lceil \boldsymbol{u} \rfloor_p = R_p^k$ for $\boldsymbol{u} \leftarrow_r R_q^k$. This completes the proof. □

2.3 Pseudorandomness of the LWR and RLWR problems

As for the hardness of the LWR and RLWR problems, Banerjee *et al*. [29] firstly showed a direct efficient reduction for modulus q that is of an exponential order of magnitude in the security parameter, and the one for the RLWR problem proceeds identically. Although Alwen *et al.* [30] gave an improved reduction that allows for a polynomial modulus q, their reduction does not apply to all values of the modulus q. For example, the reduction does not cover values of q that are powers of two. Besides the limitation in modulus, their reduction does not include treatment for the RLWR problem. In 2016, Bogdanov *et al.* [28] generalised the theorem of [30] by removing the number-theoretic restrictions on the modulus q and relaxed the condition from $q \geq 2nmBp$ to $q \geq 2mBp$. Technically, they used the Rényi divergence (rather than statistical distance) to fine-tune the statistical analysis. In particular, they gave a reduction from the search version of the RLWE problem to the search version of the RLWR problem. Nevertheless, they do not present any reduction in the decision version of the RLWR problem. Subsequently, Alperin-Sheriff and Apon [21] presented a dimension-preserving reduction from the LWE problem to the LWR problem with a polynomial-sized modulus, which immediately implies improvements in parameters (i.e. security and efficiency) for all known applications of the poly-modulus LWR problem. In particular, their reduction can be directly generalised to the ring setting. In this work, since our scheme is constructed in a ring setting, we only present the main theorem in terms of the M-LWR problem shown in [21].

Definition 4: For a number field K with ring of integers R, dimensional parameters $d, w \in \mathbb{Z}$, *modulus* $q \geq 1$, a set $\mathcal{S} \in K^d$, a parameter k denoting the number of hints received, and a distribution ψ over R, the $\mathrm{Ext-M-LWE}_{\mathrm{R,d,w,q},\psi,\mathcal{S},\mathrm{k}}$ problem is defined as follows: The algorithm gets to choose $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k \in \mathcal{S}$ and receives back a tuple: $(\boldsymbol{A}, \boldsymbol{b}, (\mathrm{Tr}(\langle \boldsymbol{e}, \boldsymbol{a}_i \rangle))_{i \in [k]}) \in R_q^{d \times w} \times R_q^w \times \mathbb{Q}_q^k$. The goal is to distinguish between two cases. In the first, which we refer to as the $\mathrm{Ext-M-LWE}$ distribution, $\boldsymbol{A}$ is chosen uniformly at random, $\boldsymbol{e} \leftarrow_r \psi^w$, and $\boldsymbol{b} = \boldsymbol{A}^t \boldsymbol{s} + \boldsymbol{e}$ for $\boldsymbol{s}$ chosen uniformly at random. The second case, which we refer to as the uniform distribution, is identical, except that $\boldsymbol{b}$ is chosen uniformly at random and independently of everything else.

The Ext-M-LWE problem was proved to be at least as hard as the M-LWE problem for suitable parameters [21]. As for the hardness of the M-LWE problem, Langlois and Stehlé [31] showed a (quantum) reduction from Mod-SIVP (i.e. SIVP restricted to module lattices) to the M-LWE problem in both its search and decision versions. Therefore, the hardness of the Ext-M-LWE problem can be reduced to some worst-case problem on module lattices. We remark that the parameter B should be set $B = \omega(\sqrt{N \log N})$ for quantum reduction [11].

Theorem 1: Let λ be a security parameter. Choose a ring of integers $R = \mathbb{Z}[X]/(\Phi_m(X))$ of degree N (recall our notational convention) and a distribution χ which is an arbitrary coordinate-wise B-bounded distribution over R. Let $q = q(\lambda)$, $p = p(\lambda)$ and $\ell = \ell(\lambda)$ such that $q \geq 4e\lambda BN\ell p$, where e is the base of the natural logarithm. If there exists a PPT algorithm $\mathcal{A}$ (adversary) succeeding with advantage $\epsilon(\lambda) \geq (\lambda)^{-c}$ for some constant $c \geq 1$ in distinguishing $\mathrm{M-LWR}_{\mathrm{N},\ell,\mathrm{n},\mathrm{q},\mathrm{p}}$ from uniform, then there exists a PPT algorithm $\mathcal{B}$ succeeding with advantage $\epsilon(\ell N)^{-c}/4 \geq (\ell N\lambda)^{-c}/4$ in distinguishing $\mathrm{Ext-M-LWE}_{\mathrm{N},\ell,\mathrm{n},\mathrm{q},\chi}$ from uniform.

Note that the $\mathrm{M-LWR}_{\mathrm{N},\ell,\mathrm{n},\mathrm{q},\mathrm{p}}$ distribution comprised samples $\left\{(\boldsymbol{A}, \lceil \boldsymbol{A}^t \boldsymbol{s} \rfloor_p) \in R_q^{n \times \ell} \times R_p^{\ell} \middle| \boldsymbol{A} \leftarrow_r R_q^{n \times \ell}\right\}$ for a fixed vector $\boldsymbol{s} \leftarrow_r R_q^n$, and then we can get $\mathrm{RLWR}_{\mathrm{N},\ell,\mathrm{q},\mathrm{p}}$ distribution by letting $n = 1$.

3 Building block

As a warm up, in this section, we will present a basic encryption construction based on the RLWR problem and give its security proof. Moreover, we will present a specific relinearisation technique that will be used to squash the expanded ciphertexts. Just like in the FV scheme, we will uniformly sample the secret key from R_2 (rather than R_q) to keep the initial noise small and speed up the encryption/decryption. Although this kind of secret key will create many attacks (e.g. [22, 32]), there are theoretical results showing that certain small secret LWE variants are as hard as those with $\boldsymbol{s} \leftarrow_r \chi$, if the dimension n is increased sufficiently [12].

3.1 Basic RLWR-based encryption scheme

The basic encryption scheme based on the RLWR problem is constructed as follows:

- **B.Setup**$\left(1^{\lambda}\right)$. Choose two moduli $q = q(\lambda)$ and $p = p(\lambda)$ satisfying $q > p$ and $p|q$. Also, choose a parameter $\ell = O(\log q) \geq \log q + 2\lambda$ and let $R = \mathbb{Z}[X]/(\Phi_m(X))$ of degree $N = \varphi(m)$. Let params $= (q, p, \ell, N)$.
- **B.KeyGen**(params). Choose uniformly at random a vector $\boldsymbol{a}' \leftarrow_r R_q^{\ell}$, and a private vector $\boldsymbol{s}' \leftarrow_r R_2^{\ell}$. Then compute $u = \langle \boldsymbol{a}', \boldsymbol{s}' \rangle_q \in R_q$. After that, assemble $\boldsymbol{a} = (\boldsymbol{a}', u) \in R_q^{\ell} \times R_q$. Output the public key $pk{:}(\boldsymbol{a})$, and the secret key $sk{:}\boldsymbol{s} = (-\boldsymbol{s}', 1) \in R_2^{\ell+1}$. Note that $\langle \boldsymbol{s}, \boldsymbol{a} \rangle = 0 \bmod q$.
- **B.Enc**(pk, μ). To encrypt a message $\mu \in R_2$, choose a random element $r \leftarrow_r R_2$, and then output

$$\boldsymbol{c} = (\boldsymbol{v}, w) = \left(\lceil \boldsymbol{a}' r \rfloor_p, \lceil ur \rfloor_p + \mu \lceil \frac{p}{2} \rfloor \right) \in R_p^{\ell} \times R_p .$$

- **B.Dec**(sk, $\boldsymbol{c}$). For the ciphertext $\boldsymbol{c} \in R_p^{\ell+1}$, output

$$\mu = \left| \lceil \frac{2}{p} \langle \boldsymbol{s}, \boldsymbol{c} \rangle_p \rfloor \right| .$$

Correctness. Let the implicit noise $\boldsymbol{e}' = (p/q)\boldsymbol{a}'r - \lceil \boldsymbol{a}'r \rfloor_p$, where vector $\boldsymbol{e}'$ is distributed uniformly over T^{ℓ}. Indeed, all coefficients of its entries (polynomials) $\{\boldsymbol{e}'[i]\}_{i\in[\ell]}$ are distributed uniformly in the interval [–1/2, 1/2]. Similarly, Let $e_u = (p/q)ur - \lceil ur \rfloor_p \in T$. Then we can get the following lemma.

Lemma 3: Let λ be a security parameter and let N, q, p and $\ell = O(\log q) \geq \log q + 2\lambda$ be parameters of the basic RLWR-based encryption scheme, satisfying $q \geq 4\lambda eBN\ell p$, where e is the base of the natural logarithm. For a public key $\mathrm{pk} \leftarrow \mathbf{B.KeyGen}(\mathrm{params})$ and a ciphertext $\boldsymbol{c} \leftarrow \mathbf{B.Enc}(\mathrm{pk}, \mu)$, where the corresponding secret key is $\boldsymbol{s} \in R_2^{\ell+1}$ and the message is $\mu \in R_2$. Then it holds that

$$\langle \boldsymbol{s}, \boldsymbol{c} \rangle = \mu \lceil \frac{p}{2} \rfloor + \tilde{e} \bmod p,$$

where $\left|\left|\tilde{e}\right|\right| < N^{3/2}\ell = N^{3/2}O(\log q)$.

Proof: It holds that

$$\begin{aligned}\langle \boldsymbol{s}, \boldsymbol{c} \rangle &= -\langle \boldsymbol{s}', \boldsymbol{v} \rangle + w = \mu \lceil \frac{p}{2} \rfloor - \langle \boldsymbol{s}', \lceil \boldsymbol{a}'r \rfloor_p \rangle + \lceil ur \rfloor_p \\ &= \mu \lceil \frac{p}{2} \rfloor - \frac{p}{q} r \langle \boldsymbol{s}', \boldsymbol{a}' \rangle + \frac{p}{q} ru + \langle \boldsymbol{s}', \boldsymbol{e}' \rangle - e_u \\ &= \mu \lceil \frac{p}{2} \rfloor + \langle \boldsymbol{s}', \boldsymbol{e}' \rangle - e_u \bmod p\,.\end{aligned}$$

Then so long as

$$\begin{aligned}\left|\left|\tilde{e}\right|\right| &\triangleq \left|\left|\langle \boldsymbol{s}', \boldsymbol{e}' \rangle - e_u\right|\right| \leq \delta_R \sum_{i=1}^{\ell} \left|\left|\boldsymbol{s}'[i]\right|\right| \left|\left|\boldsymbol{e}'[i]\right|\right| \\ &+ \left|\left|e_u\right|\right| \leq \frac{1}{2}\delta_R N\ell + \frac{1}{2}\sqrt{N} < N^{\frac{3}{2}}\ell \leq \frac{1}{2}\lceil \frac{p}{2} \rfloor\end{aligned}$$

($\delta_R \leq \sqrt{N}$ in Section 2), the above equality can recover the message μ correctly. □

3.2 Security

We argue that our basic encryption scheme is IND-CPA secure under Theorem 1 presented in Section 2.3. Specifically, the $\mathrm{RLWE}_{\mathrm{N,q},\chi}$ assumption implies $\mathrm{RLWR}_{\mathrm{N,q,p}}$ assumption, and thus the security of the basic encryption scheme can be guaranteed by the hardness assumption of the RLWE problem. For completeness, we give the security proof below.

Theorem 2: The basic RLWR-based encryption scheme presented above is IND-CPA secure under the $\mathrm{RLWR}_{\mathrm{N,q,p}}$ assumption.

Proof: We prove the theorem via a series of hybrid arguments. Let $\mathscr{A}$ be an IND-CPA adversary for our basic encryption scheme that runs in time t. We consider a series of hybrids where $\mathrm{Adv}_H[\mathscr{A}]$ denotes the success probability of $\mathscr{A}$ in hybrid H.

- Hybrid 1: This is the real system (which is identical to the IND-CPA game). By definition

$$\begin{aligned}\mathrm{Adv}_{H_1}[\mathscr{A}] \triangleq \big|&\Pr[\mathscr{A}(\mathrm{pk}, \mathrm{B}.\mathrm{Enc}(\mathrm{pk}, \mu_0)) = 1] \\ &- \Pr[\mathscr{A}(\mathrm{pk}, \mathrm{B}.\mathrm{Enc}(\mathrm{pk}, \mu_1)) = 1]\big|\,.\end{aligned}$$

- Hybrid 2: This is the same as the Hybrid 1, except that the public key $\mathrm{pk} = (\boldsymbol{a}', u)$ is generated by choosing uniformly from $R_q^{\ell} \times R_q$. Then the ciphertext $\boldsymbol{c}$ encrypted for $\mu \in R_2$ is generated using the public key, as per the encryption procedure in Hybrid 1. By definition, we have

$$\begin{aligned}\mathrm{Adv}_{H_2}[\mathscr{A}] \triangleq \big|&\Pr[\mathscr{A}(\mathrm{pk}, \mathrm{B}.\mathrm{Enc}(\mathrm{pk}, \mu_0)) = 1] \\ &- \Pr[\mathscr{A}(\mathrm{pk}, \mathrm{B}.\mathrm{Enc}(\mathrm{pk}, \mu_1)) = 1]\big|\,.\end{aligned}$$

- We now claim that

$$\left|\mathrm{Adv}_{H_1}[\mathscr{A}] - \mathrm{Adv}_{H_2}[\mathscr{A}]\right| \leq 2^{-\lambda N}\,.$$

- This follows from the fact that $\ell \geq \log q + 2\lambda$ is sufficiently large, and by our ring analogue of the standard leftover hash lemma (Lemma 1) in Section 2.
- Hybrid 3: This is the same as the Hybrid 2, except that the ciphertext $\boldsymbol{c}$ is chosen uniformly from $R_p^{\ell} \times R_p$ and independently of the public key. By definition, we have

$$\begin{aligned}\mathrm{Adv}_{H_3}[\mathscr{A}] \triangleq \big|&\Pr[\mathscr{A}(\mathrm{pk}, \mathrm{B}.\mathrm{Enc}(\mathrm{pk}, \mu_0)) = 1] \\ &- \Pr[\mathscr{A}(\mathrm{pk}, \mathrm{B}.\mathrm{Enc}(\mathrm{pk}, \mu_1)) = 1]\big|\,.\end{aligned}$$

We say that the Hybrid 3 is computationally indistinguishable from the Hybrid 2, under the $\mathrm{RLWR}_{\mathrm{N,q,p}}$ assumption. This is shown by a simple reduction: if any hypothetical adversary $\mathscr{A}$ can distinguish these two hybrids, then we can construct algorithm $\mathscr{B}$ to break the $\mathrm{RLWR}_{\mathrm{N,q,p}}$ assumption. Namely, there exists an adversary $\mathscr{B}$ that runs in time $t + \mathrm{poly}(\lambda)$ and whose advantage is

$$\mathrm{RLWR}_{\mathrm{N,q,p}}\mathrm{Adv}[\mathscr{B}] = 1/2 \cdot \left|\mathrm{Adv}_{H_2}[\mathscr{A}] - \mathrm{Adv}_{H_3}[\mathscr{A}]\right|\,.$$

Specifically, after receiving $\ell + 1$ challenged instances $\{(a_i, b_i)\}_{i\in[\ell+1]} \in R_q \times R_p$ drawn from either the $\mathrm{RLWR}_{N,\ell,q,p}$ distribution or uniform distribution, $\mathscr{B}$ assembles a vector $\boldsymbol{a} = (a_1, \ldots, a_{\ell+1}) \in R_q^{\ell+1}$ as the public key, and then computes the ciphertext

$$\begin{aligned}\boldsymbol{c} = (\boldsymbol{v}, w) &= (b_1, \ldots, b_\ell, b_{\ell+1} + \mu \lceil \frac{p}{2} \rfloor) \in R_p^{\ell} \times R_p \\ &= (\boldsymbol{b}', b_{\ell+1} + \mu \lceil \frac{p}{2} \rfloor)(\boldsymbol{b} \triangleq (b_1, \ldots, b_\ell), u \triangleq b_{\ell+1})\,.\end{aligned}$$

Note that if the $\ell + 1$ challenged instances $\{(a_i, b_i)\}_{i\in[\ell+1]} \in R_q \times R_p$ are drawn from the $\mathrm{RLWR}_{N,\ell,q,p}$ distribution, then $b_i = \lceil a_i s \rfloor_p$ for $i \in [\ell]$ and then the above ciphertext turns into

$$\begin{aligned}\boldsymbol{c} = (\boldsymbol{v}, w) &= \left(\lceil a_1 s \rfloor_p, \ldots, \lceil a_\ell s \rfloor_p, \lceil a_{\ell+1} s \rfloor_p + \mu \lceil \frac{p}{2} \rfloor\right) \in R_p^{\ell} \times R_t \\ &= \left(\lceil \boldsymbol{a}' s \rfloor_p, \lceil us \rfloor_p + \mu \lceil \frac{p}{2} \rfloor\right)(\boldsymbol{a}' \triangleq (a_1, \ldots, a_\ell), u \triangleq a_{\ell+1})\,.\end{aligned}$$

It is clear that the ciphertext $\boldsymbol{c}$ as above is valid in the view of $\mathscr{A}$. In other words, $\mathscr{B}$ perfectly simulates either the Hybrid 2 or Hybrid 3 depending on whether the challenged instances are RLWR samples or uniform over $R_p \times R_p$. More specifically, if the challenged instances are selected from the $\mathrm{RLWR}_{\mathrm{N},\ell,\mathrm{q,p}}$ distribution, then the ciphertext $\boldsymbol{c}$ is generated as per Hybrid 2. Otherwise, the ciphertext $\boldsymbol{c}$ is generated as per Hybrid 3. Therefore, if $\mathscr{A}$ declares that the experiment is Hybrid 2, $\mathscr{B}$ then declares that the challenged instances come from the $\mathrm{RLWR}_{\mathrm{N},\ell,\mathrm{q,p}}$ distribution; vice versa. Since $\mathscr{B}$'s advantage must be negligible by the $\mathrm{RLWR}_{\mathrm{N,q,p}}$ assumption, i.e. we have $\mathrm{RLWR}_{\mathrm{N,q,p}}\mathrm{Adv}[\mathscr{B}] = \mathrm{negl}(\lambda)$, thus we have

$$\left|\mathrm{Adv}_{H_2}[\mathscr{A}] - \mathrm{Adv}_{H_3}[\mathscr{A}]\right| = 2 \cdot \mathrm{negl}(\lambda)\,.$$

Note that in Hybrid 3, the public key and the ciphertext are uniformly random and independent of message μ, so we have $\mathrm{Adv}_{H_3}[\mathscr{A}] = 0$. Putting these together, we get that

$$\begin{aligned}\mathrm{Adv}_{H_1}[\mathscr{A}] &\leq \mathrm{Adv}_{H_2}[\mathscr{A}] + 2^{-\lambda N} \\ &\leq 2 \cdot \mathrm{negl}(\lambda) + \mathrm{Adv}_{H_3}[\mathscr{A}] \\ &\quad + 2^{-\lambda N} \leq 2 \cdot \mathrm{negl}(\lambda) + 2^{-\lambda N},\end{aligned}$$

which is negligible for sufficiently large security parameter λ. This completes the proof. □

3.3 Relinearisation technique

In our coming RLWR-based FHE scheme, the size of the ciphertext will expand with every homomorphic multiplication, so we use the following specific relinearisation technique that is adapted from [5, 7], to bring the expanded ciphertext back down to its previous size. Firstly, going back to Lemma 3, we can recover the messages μ_1, μ_2 from the ciphertexts $\boldsymbol{c}_1, \boldsymbol{c}_2$ by the decryption equation

$$\langle \boldsymbol{s}, \boldsymbol{c} \rangle = w - \langle \boldsymbol{s}', \boldsymbol{v} \rangle = w - \sum_{i=1}^{\ell} \boldsymbol{s}'[i] \cdot \boldsymbol{v}[i] = \mu \lceil \frac{p}{2} \rfloor + e \bmod p\,.$$

Then we get

$$\begin{aligned} \langle \boldsymbol{s}, \boldsymbol{c}_1 \rangle + \langle \boldsymbol{s}, \boldsymbol{c}_2 \rangle &= \overbrace{(w_1 - \sum_{i=1}^{\ell} \boldsymbol{s}'[i] \cdot \boldsymbol{v}_1[i]) + (w_2 - \sum_{j=1}^{\ell} \boldsymbol{s}'[j] \cdot \boldsymbol{v}_2[j])}^{\Delta_{\text{add}}} \\ &= \lceil \frac{p}{2} \rfloor \langle \mu_1 + \mu_2 \rangle_2 + e_{\text{add}} \bmod p \end{aligned} \tag{3}$$

and

$$\begin{aligned} \frac{2}{p} \langle \boldsymbol{s}, \boldsymbol{c}_1 \rangle \cdot \langle \boldsymbol{s}, \boldsymbol{c}_2 \rangle &= \overbrace{\frac{2}{p} \Big(w_1 - \sum_{i=1}^{\ell} \boldsymbol{s}'[i] \cdot \boldsymbol{v}_1[i] \Big) \cdot \Big(w_2 - \sum_{j=1}^{\ell} \boldsymbol{s}'[j] \cdot \boldsymbol{v}_2[j] \Big)}^{\Delta_{\text{mult}}} \\ &= \frac{2}{p} \Big(\mu_1 \lceil \frac{p}{2} \rfloor + e_1 + pr_1 \Big) \cdot \Big(\mu_2 \lceil \frac{p}{2} \rfloor + e_2 + pr_2 \Big) \\ &= \lceil \frac{p}{2} \rfloor \langle \mu_1 \mu_2 \rangle_2 + e_{\text{mult}} \bmod p\,. \end{aligned} \tag{4}$$

We write

$$\begin{aligned} \Delta_{\text{mult}} &= \Big(w_1 - \sum_{i=1}^{\ell} \boldsymbol{s}'[i] \cdot \boldsymbol{v}_1[i] \Big) \cdot \Big(w_2 - \sum_{j=1}^{\ell} \boldsymbol{s}'[j] \cdot \boldsymbol{v}_2[j] \Big) \\ &= a_0 + \sum_{i=1}^{\ell} a_i \boldsymbol{s}'[i] + \sum_{1 \le i \le j \le \ell} a_{i,j} \boldsymbol{s}'[i] \cdot \boldsymbol{s}'[j] \\ &= \sum_{0 \le i \le j \le \ell} a_{i,j} \boldsymbol{s}'[i] \cdot \boldsymbol{s}'[j] \quad \big(\boldsymbol{s}'[0] \triangleq 1, a_0 \triangleq a_{0,0}, a_j \triangleq a_{0,j} \big)\,. \end{aligned} \tag{5}$$

Note that in term Δ_{add} the secret key keeps a constant dimension, which corresponds to the ciphertext of constant dimension in (3); while in term Δ_{mult} it goes up from ℓ to (roughly) $\ell^2/2$, which corresponds to the (roughly) $\ell^2/2$-dimensional ciphertext in (4). This is because we have to know all the linear and quadratic terms $\{\boldsymbol{s}'[i] \cdot \boldsymbol{s}'[j]\}_{i,j \in [\ell]}$ for decryption, and this is where the relinearisation technique comes into play. Indeed, the main idea of relinearisation is to generate evaluation keys to hide the terms $\{\boldsymbol{s}'[i] \cdot \boldsymbol{s}'[j]\}_{i,j \in [\ell]}$. Next, we show how to generate these evaluation keys.

RelinearKeyGen$\big(\boldsymbol{s}_{k-1} \in R_2^{\ell}, \boldsymbol{s}_k \in R_2^{\ell}\big)$: On input $\boldsymbol{s}_{k-1}, \boldsymbol{s}_k \in R_2^{\ell}$. For all $0 \le i \le j \le \ell$ and $\tau \in \{0, \ldots, \lceil \log p \rceil - 1\}$, choose uniformly at random $\boldsymbol{v}_{k,i,j,\tau} \leftarrow_r R_q^{\ell}$, and then compute

$$\begin{aligned} w_{k,i,j,\tau} = &\lceil \langle \boldsymbol{v}_{k,i,j,\tau}, \boldsymbol{s}_k \rangle \rfloor_p \\ &+ 2^{\tau} \boldsymbol{s}_{k-1}[i] \cdot \boldsymbol{s}_{k-1}[j] \bmod p \in R_p, \end{aligned}$$

where $\boldsymbol{s}_{k-1}[0] \triangleq 1$ and $\bar{e} \triangleq (p/q) \langle \boldsymbol{v}_{k,i,j,\tau}, \boldsymbol{s}_k \rangle - \lceil \langle \boldsymbol{v}_{k,i,j,\tau}, \boldsymbol{s}_k \rangle \rfloor_p \in T$. Output

$$\rho_{k,i,j,\tau} := (\lceil \boldsymbol{v}_{k,i,j,\tau} \rfloor_p, w_{k,i,j,\tau}) \in R_p^{\ell} \times R_p,$$

where $\boldsymbol{e}_{k,i,j,\tau} \triangleq (p/q) \boldsymbol{v}_{k,i,j,\tau} - \lceil \boldsymbol{v}_{k,i,j,\tau} \rfloor_p \in T^{\ell}$.

Note that for all $0 \le i \le j \le \ell$, and $\tau \in \{0, \ldots, \lceil \log p \rceil - 1\}$, it holds that

$$\begin{aligned} \langle (-\boldsymbol{s}_k, 1), \rho_{k,i,j,\tau} \rangle &= \langle (-\boldsymbol{s}_k, 1), (\lceil \boldsymbol{v}_{k,i,j,\tau} \rfloor_p, w_{k,i,j,\tau}) \rangle \\ &= w_{k,i,j,\tau} - \frac{p}{q} \langle \boldsymbol{s}_k, \boldsymbol{v}_{k,i,j,\tau} \rangle + \langle \boldsymbol{s}_k, \boldsymbol{e}_{k,i,j,\tau} \rangle \\ &= \frac{p}{q} \langle \boldsymbol{s}_k, \boldsymbol{v}_{k,i,j,\tau} \rangle - \frac{p}{q} \langle \boldsymbol{s}_k, \boldsymbol{v}_{k,i,j,\tau} \rangle + \langle \boldsymbol{s}_k, \boldsymbol{e}_{k,i,j,\tau} \rangle \\ &\quad - \bar{e} + 2^{\tau} \boldsymbol{s}_{k-1}[i] \cdot \boldsymbol{s}_{k-1}[j] \\ &= \langle \boldsymbol{s}_k, \boldsymbol{e}_{k,i,j,\tau} \rangle - \bar{e} + 2^{\tau} \boldsymbol{s}_{k-1}[i] \cdot \boldsymbol{s}_{k-1}[j] \bmod p, \end{aligned}$$

where we have

$$\begin{aligned} \Big\| \langle \boldsymbol{s}_k, \boldsymbol{e}_{k,i,j,\tau} \rangle - \bar{e} \Big\| &\le \delta_R \sum_{i=1}^{\ell} \Big\| \boldsymbol{s}_k[i] \Big\| \Big\| \boldsymbol{e}_{k,i,j,\tau}[i] \Big\| \\ + \Big\| \bar{e} \Big\| &\le \frac{1}{2} \delta_R N \ell + \frac{1}{2} \sqrt{N} < N^{\frac{3}{2}} \ell\,. \end{aligned}$$

Then the security is guaranteed by the following lemma.

Lemma 4: Given vectors $\boldsymbol{s}_{k-1}, \boldsymbol{s}_k \in R_2^{\ell}$, generate $(\ell+1)^2 \lceil \log p \rceil$ vectors $\rho_{k,i,j,\tau} \leftarrow \textbf{RelinearKeyGen}(\boldsymbol{s}_{k-1}, \boldsymbol{s}_k)$. Then all vectors $\{\rho_{k,i,j,\tau}\}_{i,j,\tau}$ are computationally indistinguishable from uniform over $R_p^{\ell} \times R_p$ under the $\text{RLWR}_{\text{N,q,p}}$ assumption.

Proof: It is clear that every vector $\rho_{k,i,j,\tau} := (\lceil \boldsymbol{v}_{k,i,j,\tau} \rfloor_p, w_{k,i,j,\tau}) \in R_p^{\ell} \times R_p$ is 'pseudo' encryption of the term $2^{\tau} \boldsymbol{s}_{k-1}[i] \cdot \boldsymbol{s}_{k-1}[j]$. (The hidden messages cannot be recovered by the secret key.) Therefore, combining with Lemma 2, all vectors $\{\rho_{k,i,j,\tau}\}_{i,j,\tau}$ are computationally indistinguishable from uniform under the $\text{RLWR}_{\text{N,q,p}}$ assumption. □

4 RLWR-based FHE scheme

Combining the basic RLWR-based encryption scheme with our tailored relinearisation technique, we construct an efficient RLWR-based FHE scheme, which removes complex and time-consuming Gaussian noise sampling. In the BGV scheme, the modulus switching was used to control the noise growth (quadratical) so as to be able to evaluate arbitrary polynomial-depth circuits without 'bootstrapping', whereas our scheme does not need the modulus switching, as the noise only grows linearly rather than quadratically with every homomorphic multiplication.

- **RLWR.Setup**$\big(1^{\lambda}, 1^{L}\big)$: On input the security parameter λ, a polynomial L. Choose two moduli $q = q(\lambda, L), p = p(\lambda, L)$ that satisfy $q > p, p|q$. Also, let parameter $\ell = O(\log q) \ge \log q + 2\lambda$ and let $R = \mathbb{Z}[X]/(\Phi(X))$ of degree $N = \varphi(m)$. Output parameters params $= (q, p, \ell, N)$.
- **RLWR.KeyGen**(params): Run the algorithm `B.KeyGen`(params) to get $L+1$ private vectors $\boldsymbol{s}'_0, \ldots, \boldsymbol{s}'_L \in R_2^{\ell}$ and the corresponding public keys $pk: (\boldsymbol{a}_0, \ldots, \boldsymbol{a}_L)$. Then for all $k \in [L]$, $0 \le i \le j \le \ell$ and $\tau \in \{0, \ldots, \lceil \log p \rceil - 1\}$, compute

  $$\rho_{k,i,j,\tau} := (\lceil \boldsymbol{v}_{k,i,j,\tau} \rfloor_p, w_{k,i,j,\tau}) \leftarrow \text{RelinearKeyGen}(\boldsymbol{s}'_{k-1}, \boldsymbol{s}'_k)\,.$$

- Finally, output the public keys pk, the evaluation keys $Evk = \{\rho_{k,i,j,\tau}\}_{k,i,j,\tau}$ and the secret keys of different levels $sk = \{\boldsymbol{s}_{k-1}\}_{k \in [L+1]}$, where $\boldsymbol{s}_{k-1} = (-\boldsymbol{s}'_{k-1}, 1) \in R_2^{\ell+1}$.
- **RLWR.Enc**(pk, μ): Run the algorithm `B.Enc` to encrypt a message $\mu \in R_2$. Output $\boldsymbol{c} \leftarrow \textbf{B.Enc}(pk, \mu)$.

- **RLWR.Add**(Evk, $\boldsymbol{c}_1, \boldsymbol{c}_2$): On input two ciphertexts $\boldsymbol{c}_1, \boldsymbol{c}_2 \in R_p^{\ell} \times R_p$ decrypting to messages $\mu_1, \mu_2 \in R_2$ under the same secret key $\boldsymbol{s}_{k-1} \in R_2^{\ell+1}$. Output

$$\boldsymbol{c}_{\text{add}} := \boldsymbol{c}_1 + \boldsymbol{c}_2 \in R_p^{\ell} \times R_p .$$

- If they are not under the same secret key, (repeatedly) invoke the `RLWR.Refresh` (below) to make it so.
- **RLWR.Mult**(Evk, $\boldsymbol{c}_1, \boldsymbol{c}_2$). On input two ciphertexts $\boldsymbol{c}_1 = (\boldsymbol{v}_1, w_1)$, $\boldsymbol{c}_2 = (\boldsymbol{v}_2, w_2) \in R_p^{\ell} \times R_p$ decrypting to messages $\mu_1, \mu_2 \in R_2$ under the same secret key $\boldsymbol{s}_{k-1} \in R_2^{\ell+1}$. If they are not under the same secret key, (repeatedly) run the `RLWR.Refresh` (below) to make it so. Output

$$\boldsymbol{c}_{\text{mult}} \leftarrow \text{RLWR}.\text{Refresh}(\text{Evk}, \boldsymbol{c}_1, \boldsymbol{c}_2) \in \text{Rp}_{\ell} \times \text{Rp} .$$

- **RLWR.Refresh**(Evk, $\boldsymbol{c}, \widetilde{\boldsymbol{c}}$). On input two ciphertexts $\boldsymbol{c} = (\boldsymbol{v}, w)$, $\widetilde{\boldsymbol{c}} = (\widetilde{\boldsymbol{v}}, \widetilde{w}) \in R_p^{\ell} \times R_p$ decrypting to messages $\mu, \widetilde{\mu} \in R_2$ under the same secret key $\boldsymbol{s}_{k-1} \in R_2^{\ell+1}$, and the evaluation keys $\{\rho_{k,i,j,\tau}\}_{i,j,\tau} \leftarrow$ Evk. Compute a helper

$$\widetilde{\boldsymbol{h}} := (\boldsymbol{v} \otimes \widetilde{\boldsymbol{v}}, w\widetilde{\boldsymbol{v}} + \widetilde{w}\boldsymbol{v}, w\widetilde{w})$$

  for generating a sequence of elements $\{a_{i,j}\}_{i,j} \in R_p$, which satisfy (5) in Section 3.3, i.e. (see equation below).
- Next, decompose $a_{i,j}$ into $a_{i,j} = \sum_{\tau=0}^{\lceil \log p \rceil - 1} 2^{\tau} a_{i,j,\tau}$ where $a_{i,j,\tau} \in R_2$. Then the above equation can be expressed as $\langle \boldsymbol{s}_{k-1}, \boldsymbol{c} \rangle \cdot \langle \boldsymbol{s}_{k-1}, \widetilde{\boldsymbol{c}} \rangle = \sum_{i,j,\tau} a_{i,j,\tau} (2^{\tau} \boldsymbol{s}'_{k-1}[i] \cdot \boldsymbol{s}'_{k-1}[j])$. Finally, output a new ciphertext

$$\widehat{\boldsymbol{c}} = (\widehat{\boldsymbol{v}}, \widehat{w}) = \left(\lceil \frac{2}{p} \sum_{i,j,\tau} a_{i,j,\tau} \lceil \boldsymbol{v}_{k,i,j,\tau} \rfloor_p \rfloor, \lceil \frac{2}{p} \sum_{i,j,\tau} a_{i,j,\tau} w_{k,i,j,\tau} \rfloor \right) \in R_p^{\ell} \times R_p .$$

- **RLWR.Eval**(Evk, $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_t), f$). On input a function $f: \{0,1\}^t \to \{0,1\}$, and t ciphertexts $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_t$. Output

$$\boldsymbol{c}_f \leftarrow f_{\text{Evk}}(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_t) .$$

- Note that any such function can be achieved using homomorphic addition and multiplication.
- **RLWR.Dec**(sk, $\boldsymbol{c}$). Assume the ciphertext $\boldsymbol{c}$ is under the key $\boldsymbol{s}_k \in R_2^{\ell+1}$. Run the algorithm **B.Dec**($\boldsymbol{s}_k, \boldsymbol{c}$) to recover the hidden message.

The IND-CPA security of the above RLWR-based FHE scheme follows immediately from the security of the basic encryption scheme in Section 3, and by the fact that the evaluation keys Evk are computationally indistinguishable from uniform (see Lemma 4 in Section 3.3).

4.1 Correctness, homomorphic properties, and parameters

In this subsection, we will analyse our RLWR-based FHE scheme's correctness and homomorphic properties and then choose appropriate parameters. The correctness of our scheme is captured by the following lemma.

Lemma 5: Let λ be a security parameter. Let q, p, $\ell = O(\log q) \geq \log q + 2\lambda$ and N be parameters of the RLWR-based FHE scheme, such that $q \geq 4\lambda e B N \ell p$, where e is the base of the natural logarithm. Let $(\text{pk}, \text{Evk}, \text{sk}) \leftarrow$ **RLWR.KeyGen**(params). Assume that two valid ciphertexts $\boldsymbol{c}_1, \boldsymbol{c}_2 \in R_p^{\ell} \times R_p$ corresponding to messages $\mu_1, \mu_2 \in R_2$ are under the same secret key $\boldsymbol{s}_{k-1} \in R_2^{\ell+1}$, with the noises e_1, e_2, where $||e_1||, ||e_2|| < N^{3/2} O(\log q)$. We write $e' \triangleq \max\{e_1, e_2\}$. For the ciphertexts $\boldsymbol{c}_{\text{add}} \leftarrow$ **RLWR.Add**(Evk, $\boldsymbol{c}_1, \boldsymbol{c}_2$) and $\boldsymbol{c}_{\text{mult}} \leftarrow$ **RLWR.Mult**(Evk, $\boldsymbol{c}_1, \boldsymbol{c}_2$), we have

$$\langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_{\text{add}} \rangle = \langle \mu_1 + \mu_2 \rangle_2 \lceil \frac{p}{2} \rfloor + e_{\text{add}} \bmod p,$$

$$\langle \boldsymbol{s}_k, \boldsymbol{c}_{\text{mult}} \rangle = \langle \mu_1 \mu_2 \rangle_2 \lceil \frac{p}{2} \rfloor + e_{\text{mult}} \bmod p,$$

where $||e_{\text{add}}|| \leq 2||e'|| + \sqrt{N}$, $||e_{\text{mult}}|| \leq N^{3/2} O(\log q) ||e'||$.

In the following, we analyse the noise growth in homomorphic addition and multiplication, which in turn proves Lemma 5.

Homomorphic addition. For two valid ciphertexts $\boldsymbol{c}_1, \boldsymbol{c}_2 \in R_p^{\ell} \times R_p$ that are designed to encrypt messages $\mu_1, \mu_2 \in R_2$ under the same secret key $\boldsymbol{s}_{k-1} \in R_2^{\ell+1}$, we have

$$\boldsymbol{c}_{\text{add}} := \boldsymbol{c}_1 + \boldsymbol{c}_2 \in R_p^{\ell} \times R_p .$$

Then according to Lemma 3 and (3), it holds that

$$\begin{aligned} \langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_{\text{add}} \rangle &= \langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_1 + \boldsymbol{c}_2 \rangle = \langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_1 \rangle + \langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_1 \rangle \\ &= (\mu_1 \lceil \frac{p}{2} \rfloor + e_1) + (\mu_2 \lceil \frac{p}{2} \rfloor + e_2) \\ &= \lceil \frac{p}{2} \rfloor \langle \mu_1 + \mu_2 \rangle_2 + e_{\text{add}} \bmod p, \end{aligned}$$

where $||e_{\text{add}}|| = ||e_1 + e_2 + 2r_p \varphi_{\mu}|| \leq 2||e'|| + \sqrt{N}$, where $(p/2) - \lceil (p/2) \rfloor \triangleq r_p \in [-1/2, 1/2]$, $\varphi_{\mu} \in R_2$. In particular, we have $q \geq 4\lambda e B N \ell p$, $\ell = O(\log q)$ and $B = \omega(\sqrt{N \log N})$ according to Theorem 1 in Section 2.3, then it suffices to set $q = O(pN^4)$ (the security parameter $\lambda < N$). It follows that

$$||e_{\text{add}}|| \leq 2||e'|| + \sqrt{N} < N^{3/2} O(\log p + 4 \log N),$$

due to $||e'|| < N^{3/2} O(\log p + 4 \log N)$. It is obvious that the noise growth in homomorphic addition is linear (additive). Therefore, after L levels of homomorphic additions, the magnitude of the noise is at most $LN^{3/2} O(\log p + 4 \log N)$.

Homomorphic multiplication. For two valid ciphertexts $\boldsymbol{c}_1, \boldsymbol{c}_2 \in R_p^{\ell} \times R_p$ that are designed to encrypt messages $\mu_1, \mu_2 \in R_2$ under the same secret key $\boldsymbol{s}_{k-1} \in R_2^{\ell+1}$. Looking back to the **RLWR.Refresh** process, we have

$$\langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_1 \rangle \cdot \langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_2 \rangle = \sum_{i,j,\tau} a_{i,j,\tau} (2^{\tau} \boldsymbol{s}'_{k-1}[i] \cdot \boldsymbol{s}'_{k-1}[j])$$

and

$$\begin{aligned} \boldsymbol{c}_{\text{mult}} &= (\boldsymbol{v}_{\text{mult}}, w_{\text{mult}}) \\ &= \left(\lceil \frac{2}{p} \sum_{i,j,\tau} a_{i,j,\tau} \lceil \boldsymbol{v}_{k,i,j,\tau} \rfloor_p \rfloor, \lceil \frac{2}{p} \sum_{i,j,\tau} a_{i,j,\tau} w_{k,i,j,\tau} \rfloor \right) \\ &\quad \in R_p^{\ell} \times R_p . \end{aligned}$$

Then we get

$$\begin{aligned} \langle \boldsymbol{s}_{k-1}, \boldsymbol{c} \rangle \cdot \langle \boldsymbol{s}_{k-1}, \widetilde{\boldsymbol{c}} \rangle &= (w - \sum_{i=1}^{\ell} \boldsymbol{s}'_{k-1}[i] \cdot \boldsymbol{v}[i]) \cdot \left(\widetilde{w} - \sum_{j=1}^{\ell} \boldsymbol{s}'_{k-1}[j] \cdot \widetilde{\boldsymbol{v}}[j] \right) \\ &= \sum_{0 \leq i \leq j \leq \ell} a_{i,j} \boldsymbol{s}'_{k-1}[i] \cdot \boldsymbol{s}'_{k-1}[j] (\boldsymbol{s}'_{k-1}[0] \triangleq 1) . \end{aligned}$$

$$\begin{aligned}\langle \boldsymbol{s}_k, \boldsymbol{c}_{\text{mult}}\rangle &= w_{\text{mult}} - \boldsymbol{s'}_k \boldsymbol{v}_{\text{mult}}\\ &= \frac{2}{p}\sum_{i,j,\tau} a_{i,j,\tau} w_{k,i,j,\tau} - \frac{2}{p}\boldsymbol{s'}_k \sum_{i,j,\tau} a_{i,j,\tau}\lceil \boldsymbol{v}_{k,i,j,\tau}\rfloor_p - \langle \boldsymbol{s}_k, \boldsymbol{c}_r\rangle\\ &= \frac{2}{p}\sum_{i,j,\tau} a_{i,j,\tau}(w_{k,i,j,\tau} - \boldsymbol{s'}_k\lceil \boldsymbol{v}_{k,i,j,\tau}\rfloor_p) - \langle \boldsymbol{s}_k, \boldsymbol{c}_r\rangle\\ &= \frac{2}{p}\sum_{i,j,\tau} a_{i,j,\tau}(\langle \boldsymbol{s'}_k, \boldsymbol{e}_{k,i,j,\tau}\rangle - \bar{e}\\ &\quad + 2^{\tau}\boldsymbol{s'}_{k-1}[i]\cdot \boldsymbol{s'}_{k-1}[j]) - \langle \boldsymbol{s}_k, \boldsymbol{c}_r\rangle\\ &= \frac{2}{p}\sum_{i,j,\tau} a_{i,j,\tau}(\langle \boldsymbol{s'}_k, \boldsymbol{e}_{k,i,j,\tau}\rangle - \bar{e})\\ &\quad + \frac{2}{p}\sum_{i,j,\tau} a_{i,j,\tau} 2^{\tau}\boldsymbol{s'}_{k-1}[i]\cdot \boldsymbol{s'}_{k-1}[j] - \langle \boldsymbol{s}_k, \boldsymbol{c}_r\rangle\\ &= \frac{2}{p}\langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_1\rangle \cdot \langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_2\rangle + \tilde{e}_{\text{mult}},\end{aligned} \tag{6}$$

where

$$\begin{aligned}\boldsymbol{c}_r \triangleq &\left(\frac{2}{p}\sum_{i,j,\tau} a_{i,j,\tau}\lceil \boldsymbol{v}_{k,i,j,\tau}\rfloor_p, \frac{2}{p}\sum_{i,j,\tau} a_{i,j,\tau} w_{k,i,j,\tau}\right)\\ &-\left(\lceil\frac{2}{p}\sum_{i,j,\tau} a_{i,j,\tau}\lceil \boldsymbol{v}_{k,i,j,\tau}\rfloor_p\rfloor, \lceil\frac{2}{p}\sum_{i,j,\tau} a_{i,j,\tau} w_{k,i,j,\tau}\rfloor\right)\end{aligned}$$

and $\tilde{e}_{\text{mult}} \triangleq (2/p)\sum_{i,j,\tau} a_{i,j,\tau}(\langle \boldsymbol{s}'_k, \boldsymbol{e}_{k,i,j,\tau}\rangle - \bar{e}) - \langle \boldsymbol{s}_k, \boldsymbol{c}_r\rangle$. Since $||\boldsymbol{c}_r[i]|| \le (1/2)\sqrt{N}$, it holds that

$$\begin{aligned}\left|\left|\tilde{e}_{\text{mult}}\right|\right| &\le \frac{2}{p}\delta_R(\ell+1)^2\lceil\log p\rceil\left(\frac{1}{2}\delta_R N^{3/2}\ell + \frac{1}{2}N\right)\\ &\quad + \frac{1}{2}\delta_R(\ell+1)N \quad < N^{3/2}\ell,\end{aligned}$$

due to $2\delta_R(\ell+1)^2\lceil\log p\rceil\left(\frac{1}{2}\delta_R N^{3/2}\ell + \frac{1}{2}N\right) \ll p$ and $\delta_R \le \sqrt{N}$. We proceed by analysing the term $\frac{2}{p}\langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_1\rangle \cdot \langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_2\rangle$ in (6). As all arithmetics are performed over the ring R or T, here we only need to focus on the modular operations. Firstly, according to Lemma 3, we have

$$\begin{aligned}\langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_1\rangle &= \mu_1\lceil\frac{p}{2}\rfloor + e_1 + \text{pr}_1;\\ \langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_2\rangle &= \mu_2\lceil\frac{p}{2}\rfloor + e_2 + \text{pr}_2,\end{aligned} \tag{7}$$

where $r_1, r_2 \in R$. Just like in the FV scheme and the scheme in [3], we bound the magnitude of r_1 (the same bound also holds for r_2) as follows:

$$\begin{aligned}||r_1|| &= \frac{\left|\left|\langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_1\rangle - \mu_1\lceil\frac{p}{2}\rfloor + e_1\right|\right|}{p} \le \frac{||\langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_1\rangle||}{p} + \sqrt{N}\\ &\le \frac{\delta_R\sum_{j=1}^{\ell+1}\left|\left|\boldsymbol{s}_{k-1}[j]\right|\right|\left|\left|\boldsymbol{c}_1[j]\right|\right|}{p} + \sqrt{N}\\ &\le \delta_R(\ell+1)\left|\left|\boldsymbol{s}_{k-1}[j]\right|\right|\frac{||\boldsymbol{c}_1[j]||}{p} + \sqrt{N}\\ &\le \frac{1}{2}\delta_R(\ell+1)N + \sqrt{N} < N^{\frac{3}{2}}\ell(\delta_R \le \sqrt{N}).\end{aligned}$$

Plugging (7) into (6), we get

$$\begin{aligned}\langle \boldsymbol{s}_k, \boldsymbol{c}_{\text{mult}}\rangle &= \frac{2}{p}\langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_1\rangle \cdot \langle \boldsymbol{s}_{k-1}, \boldsymbol{c}_2\rangle + \tilde{e}_{\text{mult}}\\ &= \frac{2}{p}(\mu_1\lceil\frac{p}{2}\rfloor + e_1 + pr_1)\cdot(\mu_2\lceil\frac{p}{2}\rfloor + e_2 + pr_2) + \tilde{e}_{\text{mult}}\\ &= \lceil\frac{p}{2}\rfloor\langle\mu_1\mu_2\rangle_2 + \gamma + \tilde{e}_{\text{mult}} + p(\psi_\mu + \mu_1 r_2 + \mu_2 r_1 + 2r_1 r_2),\end{aligned}$$

where $\mu_1\mu_2 \triangleq \langle\mu_1\mu_2\rangle_2 + 2\psi_\mu(\psi_\mu \in R)$ and γ is defined as (see equation below).

Recall that $e' \triangleq \max\{e_1, e_2\}$, $r_p \in [-1/2, 1/2]$ and $\left|\left|r_1\right|\right|, \left|\left|r_2\right|\right| < N^{3/2}\ell$. In particular, we have $||e'|| < (1/2)\lceil(p/2)\rfloor$, then it always holds that $\left|\left|(e'^2/p)\right|\right| < (p/8)$. In other words, the magnitude of the quadratic noise term has considerably lessened, i.e. the product of two noise terms is now practically insignificant. This is the essential reason that the noise grows linearly ('quasi-additive') rather than quadratically with every homomorphic multiplication. It follows that

$$\begin{aligned}||\gamma|| &\le \frac{1}{2}(N^{1/2} + 2N^{3/2}) + \frac{1}{2p}(N^{1/2} + N^{3/2})\\ &\quad + 2N^{3/2}\ell(2\left|\left|e'\right|\right| + N^{1/2})\\ &\quad + \frac{2p+2}{p}\left|\left|e'\right|\right|N^{1/2} + \frac{2\left|\left|e'\right|\right|^2}{p} < 5(N^{3/2}\ell)^2,\end{aligned}$$

due to $\left|\left|e'\right|\right| < N^{3/2}\ell \ll p$. Finally, as $\left|\left|\tilde{e}_{\text{mult}}\right|\right| < N^{3/2}\ell$, we have

$$\langle \boldsymbol{s}_k, \boldsymbol{c}_{\text{mult}}\rangle = \lceil\frac{p}{2}\rfloor\langle\mu_1\mu_2\rangle_2 + e_{\text{mult}} \bmod p,$$

where $\left|\left|e_{\text{mult}}\right|\right| \triangleq \left|\left|\gamma + \tilde{e}_{\text{mult}}\right|\right| < 5(N^{3/2}\ell)^2 + N^{3/2}\ell < 6(N^{3/2}\ell)^2$.

Since $\ell = O(\log q)$ and $q = O(pN^4)$, of which the selection of modulus q is done in the same manner as it was in homomorphic addition, we have $6(N^{3/2}\ell)^2 = (N^{3/2}O(\log p + 4\log N))^2$. Thus, it is clear that the noise growth factor in homomorphic multiplication is $6N^{3/2}\ell$ (a small polynomial). It follows that after L levels of homomorphic multiplications, the magnitude of the noise is at most $(N^{3/2}O(\log p + 4\log N))^{L+1}$.

To sum up, we get the following theorem that captures the homomorphic properties of our RLWR-based FHE scheme.

Theorem 3: Given the parameters q, p, and N satisfying $q = O(pN^4)$, and parameter $\ell = O(\log q)$, for any polynomial L, if

$$p \ge (N^{3/2}O(\log p + 4\log N))^{L+O(1)},$$

$$\gamma \triangleq \begin{cases} 2r_1e_2 + 2r_2e_1 + e_1\mu_2 + e_2\mu_1 + \dfrac{2e_1e_2}{p}, & \text{if } p \text{ is even;}\\ r_p(2\psi_\mu + \mu_1\mu_2) + \dfrac{2r_p^2\mu_1\mu_2}{p} + r_1(2e_2 + \mu_2) & \\ + r_2(2e_1 + \mu_1) + \dfrac{p+1}{p}(e_1\mu_2 + e_2\mu_1) + \dfrac{2e_1e_2}{p}, & \text{if } p \text{ is odd.}\end{cases}$$

then our RLWR-based FHE scheme is L-homomorphism.

Full homomorphism. In order to use the 'bootstrapping' method to achieve full homomorphism, we use the specific relinearisation technique described in Section 3.3 to reduce the dimension of ciphertext from ℓ to N. Then, we bound the depth of the decryption circuit. Actually, this approach is very similar to that in [5] (the depth is $O(\log k + \log\log p)$), which utilises dimension-modulus reduction technique to lower the complexity of decryption. For completeness, we present the following theorem.

Theorem 4: For parameters q, p, and N satisfying $q = O(pN^4)$, and parameter $\ell = O(\log q)$. Assume that the RLWR-based FHE scheme is weakly circular secure. For polynomial $O(\log N + \log\log p)$, if

$$p \geq (N^{3/2}O(\log p + 4\log N))^{O(\log N + \log\log p)},$$

then our RLWR-based FHE scheme can be transformed into the one that is full homomorphism, using our specific relinearisation technique.

Remark 1: We remark that all optimisation methods (e.g. batching, truncating the least significant bit of ciphertext) used in BGV and FV schemes also apply to our scheme, as our ciphertext structure and decryption procedure are very similar to theirs.

Comparison with the FV scheme. The ciphertext is $(\ell + 1)$-dimensional vector over R_p in our scheme, whereas the ciphertext is two-dimensional vector over $R_{\hat{q}}$ in the FV scheme, so we compare our parameter p with the parameter $\hat{q}$ of the FV scheme. In our scheme, we have $q \geq 4\lambda eBN\ell p$ according to Theorem 1 in Section 2.3 and $\ell \geq \log q + 2\lambda$ according to our scheme, then we can choose $\ell = 2\log q$ and hence it is sufficient to set $q/B \geq 24pN^3$, where $B = \omega(\sqrt{N}\log N)$. Therefore, we have

$$\begin{cases} p \geq (6N^{3/2}\ell)^{L+O(1)} = (12N^{3/2}\log q)^{L+O(1)} \\ \quad = (N^{3/2}O(\log p + 4\log N))^{L+O(1)}; \\ (N^{3/2}O(\log p + 4\log N))^{L+O(1)} < (O(N^{5/2}))^{L+O(1)}. \end{cases}$$

due to $\log p < N$. By comparison, under the same ring R but a different quotient ring, in the FV scheme the noise growth factor is $6N^{3/2}$ (roughly), which is almost the same as ours. Then after L levels of homomorphic multiplications, the magnitude of the noise is at most $4B(6N^{3/2})^{L+1}$. Therefore, in order to evaluate circuits of depth L correctly, it requires that $\hat{q} \geq 4B(O(N^{3/2}))^{L+O(1)} = N(O(N^{3/2}))^{L+O(1)}$.

As remarked above, our parameter p is bigger than the parameter $\hat{q}$ of the FV scheme by a polynomial factor, but our scheme does not require Gaussian noise sampling. Overall, our scheme still has an advantage over the FV scheme.

5 Summary and future direction

Our RLWR-based FHE scheme does not require Gaussian noise sampling, so it can be seen as an alternative to the existing lattice-based FHE schemes. The security of our RLWR-based FHE scheme is based on the hardness assumption of the RLWE problem due to the reduction from RLWE to RLWR, and hence on the hardness of certain worst-case ideal lattice problems. However, the dimension of the ciphertext in our scheme is larger than that in the existing RLWE-based FHE schemes, which weakens the advantages of our scheme. Therefore, the subject of our future work is to reduce the dimension of the ciphertext and further improve efficiency.

6 Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (no. 2017YFB0802000), the National Nature Science Foundation of China (nos. 61672030 and U1705264), the Research Foundation of Hangzhou Normal University (no. 2017QDL002), and the Scientific Research Fund of Zhejiang Provincial Education Department (no. Y201737292).

7 References

[1] Gentry, C.: 'Fully homomorphic encryption using ideal lattices'. Proc. 41st Annual ACM Symp. on Theory of Computing, STOC 2009, Bethesda, MD, USA, 31 May–2 June 2009, pp. 169–178

[2] Alperin-Sheriff, J., Peikert, C.: 'Faster bootstrapping with polynomial error'. Advances in Cryptology – CRYPTO 2014–34th Annual Cryptology Conf. Proc., Part I, Santa Barbara, CA, USA, 17–21 August 2014, pp. 297–314

[3] Brakerski, Z.: 'Fully homomorphic encryption without modulus switching from classical GapSVP'. Advances in Cryptology – CRYPTO 2012–32nd Annual Cryptology Conf., Santa Barbara, CA, USA, 19–23 August 2012, pp. 868–886

[4] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: '(Leveled) fully homomorphic encryption without bootstrapping'. Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, 8–10 January 2012, pp. 309–325

[5] Brakerski, Z., Vaikuntanathan, V.: 'Efficient fully homomorphic encryption from (standard) LWE'. IEEE Computer Society IEEE 52nd Annual Symp. on Foundations of Computer Science, FOCS 2011, 2011, pp. 97–106

[6] Clear, M., McGoldrick, C.: 'Multi-identity and multi-key leveled FHE from learning with errors'. Advances in Cryptology – CRYPTO 2015–35th Annual Cryptology Conf. Proc., Part II, Santa Barbara, CA, USA, 16–20 August 2015, pp. 630–656

[7] Fan, J., Vercauteren, F.: 'Somewhat practical fully homomorphic encryption', IACR Cryptology ePrint Archive, 2012, p. 144

[8] Gentry, C., Sahai, A., Waters, B.: 'Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based'. Advances in Cryptology – CRYPTO 2013–33rd Annual Cryptology Conf. Proc., Part I, Santa Barbara, CA, USA, 18–22 August 2013, pp. 75–92

[9] Luo, F., Wang, F., Wang, K., *et al.*: 'LWR-based fully homomorphic encryption, revisited', *Sec. Commun. Netw.*, 2018, **2018**, pp. 5967635:1–5967635:12

[10] Regev, O.: 'On lattices, learning with errors, random linear codes, and cryptography'. Proc. 37th Annual ACM Symp. on Theory of Computing, Baltimore, MD, USA, 22–24 May 2005, pp. 84–93

[11] Lyubashevsky, V., Peikert, C., Regev, O.: 'On ideal lattices and learning with errors over rings'. Advances in Cryptology – EUROCRYPT 2010, 29th Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, Monaco/French Riviera, 30 May–3 June 2010, pp. 1–23

[12] Brakerski, Z., Langlois, A., Peikert, C., *et al.*: 'Classical hardness of learning with errors'. Proc. Forty-Fifth Annual ACM Symp. on Theory of Computing, 2013, pp. 575–584

[13] Lyubashevsky, V., Peikert, C., Regev, O.: 'On ideal lattices and learning with errors over rings', *J. ACM*, 2013, **60**, (6), pp. 43:1–43:35

[14] Peikert, C.: 'Public-key cryptosystems from the worst-case shortest vector problem'. Proc. Forty-First Annual ACM Symp. on Theory of Computing, 2009, pp. 333–342

[15] Brakerski, Z., Cash, D., Tsabary, R., *et al.*: 'Targeted homomorphic attribute-based encryption'. Theory of Cryptography – 14th Int. Conf., TCC 2016-B Proc., Part II, Beijing, China, 31 October–3 November 2016, pp. 330–360

[16] Bruinderink, L. G., Hülsing, A., Lange, T., *et al.*: 'Flush, Gauss, and reload–a cache attack on the bliss lattice-based signature scheme'. Int. Conf. on Cryptographic Hardware and Embedded Systems, 2016, pp. 323–345

[17] Pessl, P.: 'Analyzing the shuffling side-channel countermeasure for lattice-based signatures'. Progress in Cryptology–INDOCRYPT 2016: 17th Int. Conf. on Cryptology in India, Kolkata, India, 11–14 December 2016, pp. 153–170

[18] Micciancio, D., Walter, M.: 'Gaussian sampling over the integers: efficient, generic, constant-time'. Advances in Cryptology – CRYPTO 2017–37th Annual Int. Cryptology Conf., Proc., Part II, Santa Barbara, CA, USA, 20–24 August 2017, pp. 455–485

[19] Costache, A., Smart, N.P.: 'Homomorphic encryption without Gaussian noise', IACR Cryptology ePrint Archive, 2017, p. 163

[20] Gentry, C., Peikert, C., Vaikuntanathan, V.: 'Trapdoors for hard lattices and new cryptographic constructions'. Proc. 40th Annual ACM Symp. on Theory of Computing, Victoria, British Columbia, Canada, 17–20 May 2008, pp. 197–206

[21] Alperin-Sheriff, J., Apon, D.: 'Dimension-preserving reductions from LWE to LWR', IACR Cryptology ePrint Archive, 2016, p. 589

[22] Albrecht, M. R.: 'On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL'. Advances in Cryptology - EUROCRYPT 2017–36th Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques Proc., Part II, Paris, France, 30 April–4 May 2017, pp. 103–129

[23] Li, M., Liu, Z., Zhang, M.: 'Fully homomorphic encryption scheme without Gaussian noise', *J. Comput. Applic.*, 2017, **37**, (12), pp. 3430–3434

[24] Cheon, J.H., Kim, D., Lee, J., *et al.*: 'Lizard: cut off the tail! A practical post-quantum public-key encryption from LWE and LWR'. Security and Cryptography for Networks - 11th Int. Conf., SCN 2018, Amalfi, Italy, 5–7 September 2018, pp. 160–177, Proceedings

[25] Håstad, J., Impagliazzo, R., Levin, L.A., *et al.*: 'A pseudorandom generator from any one-way function', *SIAM J. Comput.*, 1999, **28**, (4), pp. 1364–1396

[26] Hoffstein, J., Pipher, J., Silverman, J.H.: 'NTRU: A ring-based public key cryptosystem'. Int. Algorithmic Number Theory Symp., 1998, pp. 267–288

[27] Howgrave-Graham, N., Silverman, J.H., Singer, A., *et al.*: 'NAEP: provable security in the presence of decryption failures', IACR Cryptology ePrint Archive, 2003, p. 172

[28] Bogdanov, A., Guo, S., Masny, D., *et al.*: 'On the hardness of learning with rounding over small modulus'. Theory of Cryptography – 13th Int. Conf., TCC 2016-A, Tel Aviv, Proc., Part I, Israel, 10–13 January 2016, pp. 209–224
[29] Banerjee, A., Peikert, C., Rosen, A.: 'Pseudorandom functions and lattices'. Advances in Cryptology-EUROCRYPT 2012–31st Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, 15–19 April 2012, pp. 719–737
[30] Alwen, J., Krenn, S., Pietrzak, K., *et al.*: 'Learning with rounding, revisited – new reduction, properties and applications'. Advances in Cryptology - CRYPTO 2013–33rd Annual Cryptology Conf. Proc., Part I, Santa Barbara, CA, USA, 18–22 August 2013, pp. 57–74
[31] Langlois, A., Stehlé, D.: 'Worst-case to average-case reductions for module lattices', *Des. Codes Cryptogr.*, 2015, **75**, (3), pp. 565–599
[32] Bai, S., Galbraith, S.D.: 'Lattice decoding attacks on binary LWE'. Australasian Conf. on Information Security and Privacy, 2014, pp. 322–337