Homomorphic Encryption without Gaussian Noise

Anamaria Costache and Nigel P. Smart

Dept. Computer Science,
University of Bristol,
United Kingdom.

Abstract. We propose a Somewhat Homomorphic Encryption (SHE) scheme based on the Learning With Rounding (LWR) problem. The LWR problem is somewhat similar to the more classical Learning With Errors (LWE) and was proposed as a deterministic variant of it and setting up an LWR instance does not require the generation of gaussian noise. Thus our SHE scheme can be instantiated without the need for expensive Gaussian noise sampling. Our initial scheme provides lower ciphertext sizes for small plaintext spaces than existing leading schemes such as BGV.

1 Introduction

Fully Homomorphic Encryption (FHE) was initially introduced as a concept shortly after the development of the RSA cryptosystem, by Rivest et al. [30]. Although long sought after, the first functional scheme was only proposed over thirty years later by Gentry [18, 19] in 2009. The same blueprint to construct FHE has been followed in all subsequent work. First a scheme is constructed which can evaluate arithmetic circuits of a limited depth, a so-called Somewhat Homomorphic Encryption (SHE) scheme. If the complexity of the circuits which the SHE scheme can evaluate is slightly more than the complexity of the decryption circuit for the SHE scheme, then (by placing a SHE encryption of the scheme's private key inside the public key) one can bootstrap the SHE scheme into a FHE scheme. This bootstrapping operation is obtained by homomorphically evaluating the decryption circuit on input of the ciphertext to be bootstrapped and the encryption of the secret key.

So far, there have been roughly three generations of SHE schemes. The first generation consisted of Gentry's original scheme, which was based on having two representations of a basis of an ideal of a number field, one easy basis and one hard basis. Gentry's original scheme was simplified and implemented in [20, 32], where the ideal was chosen to be principal, with the easy basis being the principal generator and the hard basis being the standard two element representation of this ideal. A second family in the first generation of schemes was based on the approximate-GCD problem, and consisted of so-called "integer based" schemes [15]. The first family in the initial generation schemes is now considered insecure due to work of Cramer et al [13], who extended the work of Campbell et al [9] to solve the problem of finding small generators of principal ideals in cyclotomic number fields. The second family, despite having numerous optimizations applied to it - such as [10, 11] - is still not considered competitive compared to the second generation schemes.

The second generation schemes were all based on the Learning With Errors (LWE) problem, and its generalisation to rings (the Ring-LWE problem) [6–8]. These schemes, generally referred to as BGV, were extensively optimized and implemented in a series of works by Gentry et al [21–24], with an implementation (HELib) being given in [26]. A variant of BGV, called FV, was presented in [17] which embeds the message into the upper bits of the underlying ring. The second generation systems also include those based on the NTRU assumption [5, 28], although the security of these has since been called into question [1].

A third generation of schemes, based on standard LWE and encoding messages via matrix eigenvalues, was presented in [25]. These schemes have an interesting property of asymmetric noise growth; and as such have given rise to some interesting theoretical applications and a fast method to perform boostrapping [16]. However, they are particularly focused on bit-encryption and hence evaluation of binary circuits on encrypted data; thus in practice their efficiency does not match that of the second generation schemes.

We therefore focus our attention on the second generation scheme based on the Learning With Errors problem (LWE). The LWE problem consists of distinguishing between the distribution of uniformly random pairs $(\mathbf{a}, b)$ of elements in $\mathbb{Z}_q^n \times \mathbb{Z}_q$ and the distribution of pairs of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e \pmod q)$. Here $\mathbf{s}$ is a secret vector and e some error vector drawn from a given distribution χ (usually a discrete Gaussian of small variance). The search version of the LWE problem is to recover the secret $\mathbf{s}$. There is a ring version of this problem, called Ring-LWE, where one is given a polyomial ring R_q modulo q and the task is to distinguish uniformly random pairs of elements $(a, b) \in R_q \times R_q$ from the distribution of pairs of elements $(a, a \cdot s + e)$ where $a, s \in R_q$ with s a fixed secret polynomial and e is a polynomial chosen from a distribution χ (which is usually the distribution of polynomials with coefficients selected from a discrete Gaussian of small variance).

As remarked above, the LWE problem has been used to construct a number of SHE schemes. In this paper we examine basing SHE schemes on a different, but closely related, problem; the Learning With Rounding (LWR) problem. The LWR problem was initially introduced by Banerjee et al. [3] as a deterministic alternative to Learning With Errors (LWE). The decision variant of the problem concerns the hardness of distinguishing samples selected from the distribution

$$\left(\mathbf{x}, \left\lceil \langle \mathbf{x}, \mathbf{s} \rangle \right\rfloor_{q_1, q_2} \right) \in \mathbb{Z}_{q_1}^n \times \mathbb{Z}_{q_2}, \tag{1}$$

from samples from the uniform distribution over $\mathbb{Z}_{q_1}^n \times \mathbb{Z}_{q_2}$. The function $\left\lceil \cdot \right\rfloor_{q_1, q_2}$ represents the scaled rounding function:

$$\begin{aligned} \left\lceil \cdot \right\rfloor_{q_1, q_2} : \mathbb{Z}_{q_1} &\to \mathbb{Z}_{q_2} \\ x &\mapsto \left\lceil \frac{q_2}{q_1} \cdot x \right\rfloor. \end{aligned}$$

The search problem is, as usual, the problem of recovering the secret $\mathbf{s}$ given samples of the form (1). Note that in LWR we have an implicit interaction between two moduli $q_1 > q_2$, whereas in LWE there is only a single modulus q.

The generalisation to rings for LWR is immediate. We extend the function $\left\lceil \cdot \right\rfloor_{q_1,q_2}$ to apply to coefficients of polynomials, and then the task is to distinguish uniformly random elements in $R_{q_1} \times R_{q_2}$, from pairs selected via

$$\left(a, \left\lceil a \cdot s \right\rfloor_{q_1,q_2}\right),$$

where s is a uniformly random secret polynomial in $R_{q_{1,i}}$.

To encrypt messages in both schemes we use a constant $\Delta_{q_2} = \left\lfloor \frac{q_2}{p} \right\rfloor$, where p is the plaintext modulus. We write $\Delta_{q_2} = \frac{q_2}{p} - \epsilon_{q_2}$ where $0 \leq \epsilon_{q_2} < 1$. An LWE encryption of a message $\mathbf{m} \in \mathbb{Z}_p$ will look like an element of the form

$$(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e + \Delta_q \cdot \mathbf{m} \pmod q),$$

whereas a LWR encryption will look like an element of the form

$$\left(\mathbf{x}, \left\lceil \langle \mathbf{x}, \mathbf{s} \rangle \right\rfloor_{q_1,q_2} + \Delta_{q_2} \cdot \mathbf{m} \pmod{q_2}\right).$$

It is clear how to generalise both of these encryptions via Ring-LWE (resp. Ring-LWR) to enable encryptions of arbitrary elements in R_p. The ring versions of the two problems we will denote by RLWE and RLWR.

In this paper we explain how the above LWR/RLWR based encryption scheme can be used to build a SHE scheme which is similar to the FV scheme based on RLWE [17]. In particular the scheme uses the above embedding of the plaintext into the upper bits of the ring modulo q_2, as opposed to the embedding into the lower bits used in schemes such as BGV [6]. We describe in this paper the RLWR version, the extension to the LWR version is immediate. Our main focus is developing a SHE scheme. The extension via bootstrapping to a FHE scheme is immediate, since it will be clear that the same bootstrapping blueprint used in other schemes will apply to our methodology. Indeed, the decryption "circuit" of our scheme is very similar to that of BGV or FV; thus all the "standard" methodologies used in these other schemes will apply to our scheme.

So one should ask what are the benefits of basing a scheme on the LWR problem as opposed to the LWE problem? Firstly, we can dispense with the costly Gaussian noise sampling required in the LWE based schemes. This is often a large computational bottleneck in RLWE based schemes. Secondly, when comparing to the BGV scheme, we find that our parameters result in slightly smaller ciphertexts for small plaintext modulus.

2 Preliminaries

In this section we formally introduce the LWR problem, introduced by Banerjee, Peikert and Rosen [3] and discuss some of the properties of this problem and its relation to LWE. As discussed in the introduction, both problems entail distinguishing between pairs of carefully constructed elements and uniformly random elements. In the LWE problem we hide an inner product by adding a small noise perturbation, whilst in the LWR problem we use rounding to hide the inner product.

2.1 Problem Definitions

We first introduce the LWE problem.

Definition 1. *Let $n \geq 1$ and q be an integer, and define the* LWE *function as follows: For a fixed vector $\mathbf{s} \in \mathbb{Z}_q^n$, we let $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ be chosen uniformly at random, and output*

$$(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^{n+1},$$

where e is chosen from some error distribution χ over $\mathbb{Z}_q$; usually a discrete Gaussian of small standard deviation σ. The decision-$\mathsf{LWE}_{n,q,\chi}$ *problem is to distinguish (with non-negligible advantage) between independent samples drawn according to the* LWE *function and uniform and independent samples drawn from $\mathbb{Z}_q^{n+1}$. The search problem is to recover $\mathbf{s}$ given independent samples drawn according to the* LWE *function.*

The LWR problem is formally given by the following analogous definition[1].

Definition 2. *Let $n \geq 1$ and $q_1 \geq q_2$ be integers, and define the* LWR *function as follows: For a fixed vector $\mathbf{s} \in \mathbb{Z}_{q_1}^n$, we let $\mathbf{a} \leftarrow \mathbb{Z}_{q_1}^n$ be chosen uniformly at random, and output*

$$(\mathbf{a}, \left\lceil \langle \mathbf{a}, \mathbf{s} \rangle \right\rfloor_{q_1,q_2}) \in \mathbb{Z}_{q_1}^n \times \mathbb{Z}_{q_2}.$$

The decision-LWR_{n,q_1,q_2} *problem is to distinguish (with non-negligible advantage) between independent samples drawn according to the* LWR *function and uniform and independent samples drawn from $\mathbb{Z}_{q_1}^n \times \mathbb{Z}_{q_2}$. The search problem is to recover $\mathbf{s}$ given independent samples drawn according to the* LWR *function.*

We can also define ring vairants of the LWE and LWR problems; which we call the RLWE and RLWR problems. As usual in the literature we work in a cyclotomic ring $R = \mathbb{Z}[x]/(\Phi_m(x))$ of degree $\phi(m)$. For an integer q, define $R_q := \mathbb{Z}[x]/(\Phi_m(x), q)$, i.e. the ring R modulo q. We give the definition of RLWR, with the analogous definition for RLWE being immediate.

Definition 3. *Let $q_1 \geq q_2$ be integers, and define the* RLWR *function as follows: For an element $s \in R_{q_1}$, we let $a \leftarrow R_{q_1}$ be chosen uniformly at random, and output*

$$(a, \left\lceil a \cdot s \right\rfloor_{q_1,q_2}) \in R_{q_1} \times R_{q_2}.$$

The decision-RLWR_{q_1,q_2} *problem is to distinguish (with non-negligible advantage) between independent samples drawn according to the* RLWR *function and uniform and independent samples drawn from $R_{q_1} \times R_{q_2}$. The search problem is to recover s given independent samples drawn according to the* RLWR *function.*

[1] We use (q_1, q_2) rather than the traditional (q, p) as we want to reserve the letter p for our plaintext modulus.

2.2 Relationships Between LWR and LWE

The LWE problem is well studied in cryptography, but the LWR one less so. In order to have confidence in schemes built on top of LWR, various authors have presented results which link the two problems. The reductions enable us to link the hardness of $\mathsf{LWE}_{n,q_1,\chi}$ to the hardness of $\mathsf{RLWR}_{n,q_1,q_2}$. Many of the reductions call a distribution B-bounded if it takes values over the integer interval $\{-B, \cdots, B\}$, $B \leq \frac{q-1}{2}$. A B-bounded distribution χ is said to be balanced if $\mathsf{Pr}[\chi \leq 0] \geq \frac{1}{2}]$ and $\mathsf{Pr}[\chi \geq 0] \geq \frac{1}{2}]$. We can take $B = 6 \cdot \sigma$ in the case of discrete Gaussians of standard deviation σ in practice.

In their original paper [3] Bannerjee et al showed that if one can solve LWR with advantage ϵ then one can solve LWE with advantage $\epsilon - O(m \cdot B \cdot q_2/q_1)$, where χ selects values from a B-bounded distribution, and m is the number of samples. However this result only holds when q_1 is an exponential function of the security parameter. In [2], another reduction of LWR from LWE is given which removes the exponential restriction on q_1, but replaces it with other number theoretic conditions.

This is improved upon in [4] which gives the following hardness result, with a similar result relating Ring-LWR and Ring-LWE being given. The main reduction [4, Theorem 1] shows that any algorithm which recovers $\mathbf{s}$ from t independent random LWR samples with probability ϵ can also be used to recover the secret $\mathbf{s}$ from m independent random samples of the form $\left(\mathbf{x}, \left\lceil \langle \mathbf{x}, \mathbf{s} \rangle + e \right\rfloor_{q_1,q_2}\right)$ with probability at least $\epsilon^2/(1 + 2 \cdot B \cdot q_2/q_1)^t$, as long as $q_1 \geq 2 \cdot B \cdot q_2$. Thus we can solve LWE modulo q_1 by embedding the LWE instance in a LWR problem via $\left(\mathbf{x}, \left\lceil \langle \mathbf{x}, \mathbf{s} \rangle + e \right\rfloor_{q_1,q_2}\right)$ and then applying our existing algorithm for LWR. Assuming $q_1 \geq 2 \cdot B \cdot q_2$ the search problem for LWE (modulo q_1) must therefore be no harder than the search problem for LWR (modulo (q_1, q_2)). More formally,

Theorem 1. *For q_1, q_2, n, t and B integers such that $q_1 > 2 \cdot q_2 \cdot B$, for every algorithm* Learn*,*

$$\mathsf{Pr}_{\mathbf{A},\mathbf{s}}[\mathsf{Learn}(\mathbf{A}, \left\lceil \mathbf{A}\mathbf{s} + \mathbf{e} \right\rfloor_{q_1,q_2}) = \mathbf{s}] \geq \frac{1}{(1 + 2 \cdot q_2 \cdot B/q_1)^t} \cdot \mathsf{Pr}_{\mathbf{A},\mathbf{s}}[\mathsf{Learn}(\mathbf{A}, \left\lceil \mathbf{A} \cdot \mathbf{s} \right\rfloor_{q_1,q_2}) = \mathbf{s}]^2,$$

where $\mathbf{A} \leftarrow \mathbb{Z}_{q_1}^{t \times n}$, e is B-bounded and balanced and $\mathbf{s}$ is chosen from $(\mathbb{Z}_{q_1}^*)^n$.

The condition on $\mathbf{s}$ will be satisified with overwhelming probability if q_1 is prime. The ring variant of this result is similar except the right hand side is divided by $(1 + 2 \cdot q_2 \cdot B/q_1)^{t \cdot \phi(m)}$, and s is restricted to be chosen from any subset of units in R_q. When q is prime this implies that the secret key *can be* selected to have small Hamming weight, as is sometimes done in homomorphic encryption schemes [24, 12].

The authors of [4] also give a hardness result for the decision version of the LWR problem. In particular they show that any distinguisher Dist for LWR can be used to solve the search version of LWE, assuming the secret is selected from $\{0, 1\}^n$. More formally,

Theorem 2. *For every $\epsilon > 0, n, t, q_1 > 2 \cdot q_2 \cdot B$ and an algorithm* Dist *such that*

$$|\mathsf{Pr}_{\mathbf{A},\mathbf{s}}[\mathsf{Dist}(\mathbf{A}, \left\lceil \mathbf{A} \cdot \mathbf{s} \right\rfloor_{q_1,q_2}) = 1] - \mathsf{Pr}_{\mathbf{A},\mathbf{u}}[\mathsf{Dist}(\mathbf{A}, \left\lceil \mathbf{u} \right\rfloor_{q_1,q_2}) = 1]| \geq \epsilon,$$

where $\mathbf{A} \leftarrow \mathbb{Z}_{q_1}^{t \times n}$, $\mathbf{s} \leftarrow \{0,1\}^n$ and $\mathbf{u} \leftarrow \mathbb{Z}_{q_1}^t$, there exists an algorithm Learn *that runs in polynomial time in n, t, the number of divisors of q and the running time of* Dist *such that*

$$\mathsf{Pr}_{\mathbf{A},\mathbf{s}}[\mathsf{Learn}(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) = \mathbf{s}] \geq \big(\frac{\epsilon}{4 \cdot q_1 \cdot t} - \frac{2^n}{q_2^t}\big)^2 \cdot \frac{1}{(1 + 2 \cdot B \cdot q_2/q_1)^t}.$$

Alas no RLWR variant of this result is currently known.

3 A Ring-LWR Based Somewhat Homomorphic Encryption Scheme

In this section we define our Somewhat Homomorphic Encryption (SHE) scheme based on RLWR. As earlier, we let R denote the ring $R = \mathbb{Z}[x]/(\Phi_m(x))$, for the m-th cyclotomic polynomial $\Phi_m(x)$; this ring has degree $N = \phi(m)$. As we define each operation we analyse it for both correctness and for the associated "noise" growth. We will use a levelled ciphertext space and thus, from here on, denote our moduli by $q_{1,i}$ and $q_{2,i}$. These represent the products

$$q_{1,L-1} = q_1 \cdot \prod_{j=0}^{L-1} p_j \quad \text{and} \quad q_{2,L-1} = q_2 \cdot \prod_{j=0}^{L-1} p_j.$$

The index i will be used to denote the level we are at. The p_i are selected to be primes such that $p_i \equiv 1 \pmod{p}$ for all i. We also require that $q_1 \equiv q_2 \equiv 1 \pmod{p}$. This is because we will define a modulus switch operation, as in [6], which is described in Section 3.3.

Following [29], we use the canonical embedding norm as a way to measure polynomials. We define the canonical embedding of a polynomial $a \in R$ into the complex sapce (for R a polynomial ring) as the $\phi(m)$-vector $\sigma(a) = (a(\zeta_m^i))_i$. ζ_m is a complex primitive m-th root of unity and the indexes i range over all of $(\mathbb{Z}/m\mathbb{Z})^*$. We write

$$\Big\|a\Big\|_\infty^{\text{can}} = \big\|\sigma(a)\big\|_\infty.$$

We will need to bound the canonical embedding norm of polynomials that are produced by the what are essentially randomly chosen polynomials, well as products of such polynomials. Following the work in [24] we use a heuristic approach, which we now recap. Let $a \in R$ be a polynomial chosen by independently selecting all the coefficients in a from the same distribution. For a complex primitive m-th root of unity ζ_m, the evaluation $a(\zeta_m)$ is the inner product between the coefficient vector of a and the fixed vector $\mathbf{z}_m = (1, \zeta_m, \zeta_m^2, \ldots)$, whose Euclidean norm is exactly $\sqrt{\phi(m)}$. Hence the random variable $a(\zeta_m)$ has variance $V = \sigma^2\phi(m)$, where σ^2 is the variance of each coefficient

of a. Specifically, when a is chosen uniformly with coefficients in $[-q/2, \ldots, q/2]$ then each coefficient has variance $(q-1)^2/12 \approx q^2/12$, and so we obtain a variance $V_U = q^2 \cdot \phi(m)/12$. When choosing a with coefficients from $\{-1, 0, 1\}$ with Hamming weight h we obtain a variance of $V_H = h$ (but not $\phi(m)$, since a has only h nonzero coefficients).

Just as in [12] we model all canonical embedding norms as if from a random distribution. In particular we shall assume that messages from the ring R_p, and similar quantities, behave as if selected uniformly at random and hence estimate $\|m\|_\infty^{\text{can}} \leq 6 \cdot p \cdot \sqrt{\phi(m)/12} = p \cdot \sqrt{3 \cdot \phi(m)}$.

Moreover, the random variable $a(\zeta_m)$ is a sum of many independent identically distributed random variables, hence by the law of large numbers it is distributed similarly to a complex Gaussian random variable of the specified variance.We therefore use $6\sqrt{V}$ (i.e. six standard deviations) as a high-probability bound on the size of $a(\zeta_m)$. Since the evaluation of a at all the roots of unity obeys the same bound, we use six standard deviations as our bound on the canonical embedding norm of a. We chose six standard deviations since $\text{erfc}(6) \approx 2^{-55}$, which is good enough for us even when using the union bound and multiplying it by $\phi(m) \approx 2^{16}$.

In many cases, we need to bound the canonical embedding norm of a product of two or more such "random polynomials". In this case our task is to bound the magnitude of the product of two random variables, both of which are distributed close to Gaussians, with variances σ_a^2, σ_b^2, respectively. For this case, we use $16 \cdot \sigma_a \cdot \sigma_b$ as our bound, since $\text{erfc}(4) \approx 2^{-25}$, so the probability that both variables exceed their standard deviation by more than a factor of four is roughly 2^{-50}. For a product of three variables we use $40 \cdot \sigma_a \cdot \sigma_b \cdot \sigma_c$, since $\text{erfc}(3.4) \approx 2^{-19}$, and $3.4^3 \approx 40$; and for a product of four variables we use $70 \cdot \sigma_a \cdot \sigma_b \cdot \sigma_c \cdot \sigma_d$ since $\text{erfc}(2.9) \approx 2^{-14}$, and $2.9^4 \approx 70$.

3.1 The Basic Encryption Scheme

Key Generation: We encrypt at the top level $L-1$ and let $s \in R_{q_{1,L-1}}$ be our secret key $\mathbf{sk} := s$. We select s to have coefficients in $\{-1, 0, 1\}$, with Hamming weight h. This will be small enough so that we can think of s as being at any level i. This choice of 'sparse' s is to keep the noise growth below small, and is comparable with similar choices in [12, 24] made for other SHE schemes. The public key is made up of ℓ encryptions of zero; where ℓ is a security parameter (say $\ell = 80$). To generate the encryptions of zero we select $v_k \leftarrow R_{q_{1,L-1}}$ and then set

$$u_k \leftarrow \left\lceil v_k \cdot s \right\rfloor_{q_{1,L-1}, q_{2,L-1}}.$$

The final public key is the set $\mathbf{pk} := \{(v_1, u_1), \ldots, (v_\ell, u_\ell)\}$. For later use we write

$$e_k = u_k - \frac{q_{2,L-1}}{q_{1,L-1}} \cdot v_k \cdot s = \left\lceil \frac{q_{2,L-1}}{q_{1,L-1}} \cdot v_k \cdot s \right\rfloor - \frac{q_{2,L-1}}{q_{1,L-1}} \cdot v_k \cdot s,$$

where we think of v_k and u_k as elements of R. The polynomials e_k we can thus assume are distributed uniformly with coefficients in $[-1/2, \ldots, 1/2]$ and hence we can assume $\|e_k\|_\infty^{\text{can}} \leq 6 \cdot \sqrt{\phi(m)/12} = \sqrt{3 \cdot \phi(m)}$.

Encryption: To encrypt a message $\mathbf{m} \in R_p$ at level $L-1$ we first select ℓ random bits $r_k \in \{0,1\}$ for $k = 1, \ldots, \ell$ and then we set

$$\begin{aligned}\mathrm{Enc}(\mathbf{m}, \mathbf{pk}) := \mathbf{ct} &= (v, w) \\ &= \left(\sum_{k=1}^{\ell} r_k \cdot v_k \pmod{q_{1,L-1}}, \right. \\ &\qquad \left. \Delta_{q_{2,L-1}} \cdot \mathbf{m} + \sum_{k=1}^{\ell} r_k \cdot u_k \pmod{q_{2,L-1}} \right),\end{aligned}$$

where we think of $\mathbf{m}$ as an element in $R_{q_{2,L-1}}$. Thus we obtain an encryption of $\mathbf{m}$ by simply adding on a random subset of our collection of encryptions of zero. To a ciphertext $\mathbf{ct} = (v, w)$ encrypting a message $\mathbf{m}$ we associate the following "noise" value:

$$e = w - \frac{q_{2,L-1}}{q_{1,L-1}} \cdot v \cdot s - \Delta_{q_{2,L-1}} \cdot \mathbf{m},$$

where we intepret all the component polynomials as being lifted in a trivial manner to R, i.e. $e \in R \otimes_{\mathbb{Z}} \mathbb{Q}$. For a fresh ciphertext we note that we have (interpreting all the polynomials in R)

$$\begin{aligned}e &= \left(\Delta_{q_{2,L-1}} \cdot \mathbf{m} + \sum_{k=1}^{\ell} r_k \cdot u_k \right) - \frac{q_{2,L-1}}{q_{1,L-1}} \cdot \left(\sum_{k=1}^{\ell} r_k \cdot v_k \right) \cdot s - \Delta_{q_{2,L-1}} \cdot \mathbf{m} \\ &= \sum_{k=1}^{\ell} (r_k \cdot u_k - \frac{q_{2,L-1}}{q_{1,L-1}} \cdot r_k \cdot v_k \cdot s) = \sum_{k=1}^{\ell} r_k \cdot e_k.\end{aligned}$$

Thus for a fresh ciphertext we have that we expect the error is bounded by $\|e\|_{\infty}^{\mathrm{can}} \leq \ell \cdot \sqrt{3 \cdot \phi(m)}/2 = B_0$.

Decryption: Decryption, given a ciphertext $\mathbf{ct} = (v, w)$ at level i proceeds as follows:

$$\mathrm{Dec}(\mathbf{ct}, \mathbf{sk}) = \left\lceil \frac{1}{\Delta_{q_{2,i}}} \left(-\frac{q_{2,i}}{q_{1,i}} \cdot v \cdot s + w \right) \right\rfloor \mod p,$$

where all operations within the rounding function are performed in the field $R \otimes_{\mathbb{Z}} \mathbb{Q}$. To check correctness we notice that, where e is the noise value associated to the ciphertext (v, w) defined above,

$$\begin{aligned}\mathrm{Dec}(\mathbf{ct}, \mathbf{sk}) &= \left\lceil \frac{1}{\Delta_{q_{2,i}}} \left(-\frac{q_{2,i}}{q_{1,i}} \cdot v \cdot s + w \right) \right\rfloor \mod p \\ &= \left\lceil \frac{1}{\Delta_{q_{2,i}}} \left(-\frac{q_{2,i}}{q_{1,i}} \cdot v \cdot s + \frac{q_{2,i}}{q_{1,i}} \cdot v \cdot s + \Delta_{q_{2,i}} \cdot \mathbf{m} + e \right) \right\rfloor \mod p \\ &= \left\lceil \frac{1}{\Delta_{q_{2,i}}} \left(\Delta_{q_{2,i}} \cdot \mathbf{m} + e \right) \right\rfloor \mod p.\end{aligned}$$

$$= \mathbf{m} + \left\lceil \frac{e}{\Delta_{q_{2,i}}} \right\rfloor.$$

Thus $\mathbf{ct}$ correctly decrypts to m if (and only if) $\left\lceil \frac{e}{\Delta_{q_{2,i}}} \right\rfloor = 0$. This is guaranteed to happen if the error term e has coefficients bounded by $\Delta_{q_{2,i}}/2$, i.e. the ∞-norm of e in the coefficient embedding is less than $q_{2,i}/2$. Since our error analysis will be in terms of the canonical norm of e, this means we need to ensure that

$$\|e\|_{\infty}^{\mathrm{can}} \leq \Delta_{q_{2,i}}/(2 \cdot c_m), \tag{2}$$

where c_m is the "ring-constant" associated to R; see [14] for a discussion of c_m. Asymptotically c_m can grow super-polynomially but for "small" values of m used in any scheme the size of c_m is relatively small.

Theorem 3. *The encryption scheme* $(\mathrm{Enc}, \mathrm{Dec})$ *is IND-CPA secure assuming the (decision)* RLWR *problem is hard.*

Proof. A ciphertext (v, w) masks the value $\Delta_{q_{2,i}} \cdot \mathbf{m} \in R_{q_{2,i}}$ via the value of the form $\left\lceil v \cdot s \right\rfloor_{q_{1,i}, q_{2,i}}$. Assuming the RLWR problem is hard this latter value is indistinguishable from a random value in $R_{q_{2,i}}$, and thus $\Delta_{q_{2,i}} \cdot \mathbf{m}$ is essentially one-time pad encrypted. Thus, by a standard hybrid argument, one can bound the advantage of an adversary which breaks the IND-CPA security of the encryption scheme, by twice the advantage of an adversary in breaking decision RLWR. The details we leave to the reader.

3.2 Homomorphic Operations

Having defined our basic encryption scheme we now turn to defining the homomorphic operations which it supports.

Addition: Suppose we are given two ciphertexts at the same level i, $\mathbf{ct} = (v, w)$ and $\mathbf{ct}' = (v', w')$ encrypting $\mathbf{m}$ and $\mathbf{m}'$ with respective noise e and e', i.e. we have that

$$\begin{aligned} w &= \frac{q_{2,i}}{q_{1,i}} \cdot v \cdot s + \Delta_{q_{2,i}} \cdot \mathbf{m} + e \\ w' &= \frac{q_{2,i}}{q_{1,i}} \cdot v' \cdot s + \Delta_{q_{2,i}} \cdot \mathbf{m}' + e'. \end{aligned}$$

Then we define the ciphertext produced by the homomorphic addition operation as simply

$$\mathbf{ct}_{\mathrm{add}} = (v + v', w + w') = (v_{\mathrm{add}}, w_{\mathrm{add}}),$$

where we will write $\mathbf{m}_{\mathrm{add}} = \mathbf{m} + \mathbf{m}' \mod p = \mathbf{m} + \mathbf{m}' + r_a \cdot p$, since the messages may wrap around modulo p.

$$\mathrm{Dec}(\mathbf{ct}, \mathbf{sk}) = \left\lceil \frac{1}{\Delta_{q_{2,i}}} \left(w_{\mathrm{add}} - \frac{q_{2,i}}{q_{1,i}} \cdot v_{\mathrm{add}} \cdot s \right) \right\rfloor \mod p$$

$$
\begin{aligned}
&= \left\lceil \frac{1}{\Delta_{q_{2,i}}} \left(\frac{q_{2,i}}{q_{1,i}} \cdot (v + v') \cdot s + \Delta_{q_{2,i}} \cdot (\mathbf{m} + \mathbf{m}') + e + e' \right. \right. \\
&\qquad \left. \left. - \frac{q_{2,i}}{q_{1,i}} \cdot (v + v') \cdot s \right) \right\rfloor \mod p \\
&= \left\lceil \frac{1}{\Delta_{q_{2,i}}} \left(\Delta_{q_{2,i}} \cdot (m_{\text{add}} - r_a \cdot p) + e + e' \right) \right\rfloor \mod p \\
&= \left\lceil \frac{1}{\Delta_{q_{2,i}}} \left(\Delta_{q_{2,i}} \cdot \mathbf{m}_{\text{add}} - \Delta_{q_{2,i}} \cdot r_a \cdot p + e + e' \right) \right\rfloor \mod p \\
&= \mathbf{m}_{\text{add}} + \left\lceil \frac{e + e}{\Delta_{q_{2,i}}} \right\rfloor - r_a \cdot p \mod p, \\
&= \mathbf{m}_{\text{add}} + \left\lceil \frac{e + e}{\Delta_{q_{2,i}}} \right\rfloor.
\end{aligned}
$$

We conclude that correct decryption occurs if and only if $\left\lceil \frac{e+e}{\Delta_{q_{2,i}}} \right\rfloor = 0$. The above analysis also allows us to conclude that $e_{\text{add}} = e' + e$, i.e. that noise grows additively.

Multiplication: Just as in many FHE schemes, multiplication is a less straightforward operation than addition. Here, this is worsened by the fact that the interplay between the two moduli $q_{1,i}$ and $q_{2,i}$ - and their corresponding rings - means we need to pay close attention to the domain of each operation. We follow the blueprint of most second generation homomorphic encryption schemes and present multiplication as a four step process. In the first two steps we form a tensor product of the two input ciphertexts - both at the same level i - which will decrypt under the tensor of the secret key. These first two steps are performed in the ring R, and result in a potential growth of the coefficient sizes as well as the production of a ciphertext of dimension three. We address both these issues in the final two steps. In the third step we perform reduction modulu $q_{1,i}$ and $q_{2,i}$ so as to deal with coefficient growth. Then, in a final relinearisation step, we reduce the ciphertext form and secret key back to the original form, essentially by performing a key switching operation. Since our message is embedded in the upper bits of the modulo $q_{2,i}$ space, like other scale invariant schemes such as FV [17] and YASHE [5], we also need to scale by a factor of $1/\Delta_{q_{2,i}}$. We will only present the methodology in the main body of the paper. See Appendix A for proofs of correctness, and analysis of the noise growth.

Multiplication Step 1: Tensoring: We can consider the standard decryption operation in the following form: take a ciphertext $\mathbf{ct} = (v, w)$ and a secret key in the form $\mathbf{sk} = (-\frac{q_{2,i}}{q_{1,i}} \cdot s, 1)$. We consider these and the computations in this step in the general ring R. We evaluate

$$\left\lceil \frac{1}{\Delta_{q_{2,i}}} (\mathbf{ct} \otimes \mathbf{sk}) \right\rfloor \pmod p,$$

where $\otimes$ is the tensor product. For multiplication let $\mathbf{ct} = (v, w), \mathbf{ct}' = (v', w')$ be our two input ciphertexts, with associated noise e and e', respectively. We first form the

tensor product ciphertext

$$\mathbf{ct} \otimes \mathbf{ct}' = (v \cdot v',\ \ v \cdot w',\ \ v' \cdot w,\ \ w \cdot w') \in R^4,$$

which will decrypt by tensoring with the tensor secret key

$$\mathbf{sk} \otimes \mathbf{sk} = \left(\frac{q_{2,i}}{q_{1,i}}^2 \cdot s^2,\ \ -\frac{q_{2,i}}{q_{1,i}} \cdot s,\ \ -\frac{q_{2,i}}{q_{1,i}} \cdot s,\ \ 1 \right) \in (R \otimes_{\mathbb{Z}} \mathbb{Q})^4.$$

Simplifying, by combining the two middle terms in each quadruple we form the three element ciphertext

$$\mathbf{ct}^0_{\mathrm{mult}} = (v \cdot v', w \cdot v' + w' \cdot v, w \cdot w') = (d'_0, d'_1, d'_2) \in R^3,$$

which will decrypt via the equation

$$\left\lceil \frac{1}{\Delta^2_{q_{2,i}}} \left(d'_2 - \frac{q_{2,i}}{q_{1,i}} \cdot d'_1 \cdot s + \left(\frac{q_{2,i}}{q_{1,i}} \right)^2 \cdot d'_0 \cdot s^2 \right) \right\rfloor.$$

In what follows we will write $\mathbf{m}_{\mathrm{mult}} = \mathbf{m} \cdot \mathbf{m}' \mod p = \mathbf{m} \cdot \mathbf{m}' + r_b \cdot p$.

Multiplication Step 2: Reduction by $\Delta_{q_{2,i}}$: Instead of the tensor ciphertext (d'_0, d'_1, d'_2) decrypting via the above equation, we would rather have a three element ciphertext (d_0, d_1, d_2) which decrypts via the equation

$$\left\lceil \frac{1}{\Delta_{q_{2,i}}} \left(d_2 - \frac{q_{2,i}}{q_{1,i}} \cdot d_1 \cdot s + \left(\frac{q_{2,i}}{q_{1,i}} \right)^2 \cdot d_0 \cdot s^2 \right) \right\rfloor. \tag{3}$$

As in [17], we multiply each d'_k by $\frac{p}{q_{2,i}}$ and then round to produce

$$d_k = \left\lceil \frac{p \cdot d'_k}{q_{2,i}} \right\rfloor \in R.$$

This operation, performed globally, gives a decryption equation of,

$$d_2 - \frac{q_{2,i}}{q_{1,i}} \cdot d_1 \cdot s + \left(\frac{q_{2,i}}{q_{1,i}} \right)^2 \cdot d_0 \cdot s^2 = \mathbf{m} \cdot \mathbf{m}' \cdot \Delta_{q_{2,i}} + e_{\mathrm{mult}}.$$

For the analysis of the term e_{mult} see Appendix A. If B (resp. B') is an upper bound on the canonical norm of e (resp. e') then the canonomical norm of e_{mult} grows essentially as $p \cdot B \cdot B'/q_{2,i}$. Note that the noise term for multiplication for the FV scheme described in [12] also grows essentially by the same amount. We bound the canonical norm of e_{mult} by the function $F(.,.)$

$$\|e_{\mathrm{mult}}\|^{\mathrm{can}}_{\infty} \leq F(B, B'),$$

which we discuss in Appendix A. Thus, given upper bounds on the canonical norm of the input noise to this step of the multiplication operation, we can bound the output noise as well. So, at the end of this step, we have a ciphertext $\mathbf{ct}^1_{\mathrm{mult}} = (d_0, d_1, d_2)$ which decrypts via equation (3), and whose associated noise term is bounded by $F(B, B')$.

Multiplication Step 3: Modular Reduction: As mentioned earlier, this step is used to ensure that computations are performed appropriately. Taking the previous $\mathbf{ct}^1_{\text{mult}} = (d_0, d_1, d_2)$, we set

$$\begin{aligned} f_0 = d_0 \pmod{q_{1,i}^2} &= d_0 + \epsilon_0 \cdot q_{1,i}^2, \\ f_1 = d_1 \pmod{q_{1,i}} &= d_1 + \epsilon_1 \cdot q_{1,i}, \\ f_2 = d_2 \pmod{q_{2,i}} &= d_2 + \epsilon_2 \cdot q_{2,i}, \end{aligned}$$

and output $\mathbf{ct}^2_{\text{mult}} = (f_0, f_1, f_2)$. This will now decrypt via

$$\left\lceil \frac{1}{\Delta_{q_{2,i}}} \left(f_2 - \frac{q_{2,i}}{q_{1,i}} \cdot f_1 \cdot s + \left(\frac{q_{2,i}}{q_{1,i}}\right)^2 \cdot f_0 \cdot s^2 \right) \right\rfloor. \tag{4}$$

Note that - as with general decryption - the inner bracket is computer *globally*. The above gives

$$\left\lceil \frac{1}{\Delta_{q_{2,i}}} \left(f_2 - \frac{q_{2,i}}{q_{1,i}} \cdot f_1 \cdot s + \left(\frac{q_{2,i}}{q_{1,i}}\right)^2 \cdot f_0 \cdot s^2 \right) \right\rfloor = \mathbf{m} \cdot \mathbf{m}' \cdot \Delta_{q_{2,i}} + e_{\text{mult}} + \epsilon_{\text{mult}}.$$

We bound the canonical norm of ϵ_{mult} by B_ϵ:

$$\|\epsilon_{\text{mult}}\|^{\text{can}}_\infty \leq B_\epsilon.$$

We give the analysis of the size of B_ϵ in Appendix A. Therefore, at the end of this step, we have a ciphertext $\mathbf{ct}^2_{\text{mult}} = (f_0, f_1, f_2)$ which decrypts via equation (4), and whose associated noise term is bounded by $F(B, B') + B_\epsilon$.

Multiplication Step 4: Relinearization: We now want to rework our output ciphertext into the form $\mathbf{ct}^3_{\text{mult}} = (c_0, c_1)$, with an associated decryption equation of

$$\left\lceil \frac{1}{\Delta_{q_{2,i}}} \left(-\frac{q_{2,i}}{q_{1,i}} \cdot c_0 \cdot s + c_1 \right) \right\rfloor \mod p.$$

This process is called *relinearization*. This is accomplished by adding so-called *key switching* matrices into the public key. We pick a modulus T and sample pairs (a_j, b_j) from $j = 0, \ldots, 2 \cdot \lceil \log_T q_{1,i} \rceil$ such that a_j is selected uniformly at random from $R_{q_{1,i}}$ and

$$\begin{aligned} b_j &= \left\lceil \frac{q_{2,i}}{q_{1,i}} \cdot a_j \cdot s + \left(\frac{q_{2,i}}{q_{1,i}}\right)^2 \cdot T^j \cdot s^2 \right\rfloor \\ &= \frac{q_{2,i}}{q_{1,i}} \cdot a_j \cdot s + \left(\frac{q_{2,i}}{q_{1,i}}\right)^2 \cdot T^j \cdot s^2 + \tau_j \in R_{q_{2,i}}, \end{aligned}$$

where τ_j is the rounding error, i.e. a polynomial with coefficients in $[-1/2, \ldots, 1/2]$. Note that the larger T, is the smaller the number of key switching matrices are required.

Thus the size of T has an effect on the size of the public key, although it has little effect on the size of other parameters.

In relinearization we replace the ciphertext $(f_0, f_1, f_2) \in R_{q_{1,i}^2} \times R_{q_{1,i}} \times R_{q_{2,i}}$ by a ciphertext $(c_0, c_1) \in R_{q_{1,i}} \times R_{q_{2,i}}$. The value f_2 gets mapped to c_1 and the value f_1 gets mapped to c_0; where the respective moduli sizes match up. This leaves us with the processing of terms coming from f_0 which need to be added into both c_0 and c_1 using the key switching matrices above. The value f_0 is given modulo $q_{1,i}^2$; note in practice the size of f_0 will be much smaller than this, we just leave this bound here for ease of analysis. Tighter bounds can be on the size of f_0 and hence the level of decomposition below can be found if desired.

To relinearize, we first take the element f_0 and expand it into its base-T representation

$$f_0 = \sum_{j=0}^{2\cdot\lceil \log_T(q_{1,i})\rceil} f_{0,j} \cdot T^j,$$

where each $f_{0,j}$ is a polynomial with coefficients in the range $[-T/2, \ldots, T/2]$. We then write

$$c_0 = f_1 + \sum_{j=0}^{2\cdot\lceil \log_T(q_{1,i})\rceil} f_{0,j} \cdot a_j, \quad (\mathrm{mod}\ q_{1,i})$$
$$c_1 = f_2 + \sum_{j=0}^{2\cdot\lceil \log_T(q_{1,i})\rceil} f_{0,j} \cdot b_j \quad (\mathrm{mod}\ q_{2,i}).$$

Putting all four steps together we obtain a multiplication operation which increases the noise inherent in the output ciphertext by a value bounded, in the canonical norm, by

$$\left\| e_{\mathrm{mult}} + \epsilon_{\mathrm{mult}} + \sum_{j=0}^{2\cdot\lceil \log_T(q_{1,i})\rceil} f_{0,j} \cdot \tau_j \right\|_\infty^{\mathrm{can}} \leq F(B, B') + B_\epsilon + 2 \cdot \lceil \log_T(q_{1,i})\rceil \cdot \phi(m) \cdot T/3 =: G(B, B').$$

We refer to Appendix A for an analysis of the terms $F(B, B')$, B_ϵ and $G(B, B')$.

3.3 Modulus Switch

In [6] the authors introduce a method called modulus switching, which allows the ciphertext modulus to be successively reduced. In practical implementations of the BGV scheme, see e.g. [24], this is found to produce not only an optimization during execution of homomorphic operations, but it also aids in significantly reducing the parameter sizes. It turns out, see for example [12], that a similar benefit in parameter reduction can also be applied to the scale-invariant schemes such as FV.

We recall that to a ciphertext $\mathbf{ct} = (v, w)$ encrypting a message $\mathbf{m}$ at level i we associate the following "noise" value:

$$e = w - \frac{q_{2,i}}{q_{1,i}} \cdot v \cdot s - \Delta_{q_{2,i}} \cdot \mathbf{m},$$

$$= \langle \mathbf{ct}, \mathbf{s} \rangle \pmod{q_{2,i}},$$

where we interpret all the component polynomials as being lifted to R and the secret key being given by $\mathbf{s} = (-\frac{q_{2,i}}{q_{1,,i}}\mathbf{sk}, 1)$.

We define a Modulus Switch operation. The tricky aspect here is the interplay between our two ciphertext moduli $q_{1,i}$ and $q_{2,i}$, as opposed to the single ciphertext modulus of other schemes such as BGV and FV.

Definition 4. *For an integer vector* $\mathbf{x}$ *over* $R_{q_{2,i+1}}$ *and integer moduli* $q_{2,i+1} > q_{2,i} > p$*, define*

$$\mathbf{x}' \leftarrow \mathsf{Scale}(\mathbf{x}, q_{2,i+1}, q_{2,i}, p)$$

to be the integer vector in $R_{q_{2,i}}$ *closest to* $\frac{1}{p_{i+1}} \cdot \mathbf{x}$ *that satisfies* $\mathbf{x} \equiv \mathbf{x}' \pmod{p}$.

So let $\mathbf{ct}' \leftarrow \mathsf{Scale}(\mathbf{ct}, q_{2,i+1}, q_{2,i}, p)$ and recall that for all $j \in \{1, 2\}, i \in \{1, \cdots L - 1\}$, we have $q_{j,i} \equiv 1 \pmod{p}$. In particular, this means that

$$q_{2,i+1} \equiv q_{2,i} \pmod{p} \tag{5}$$

We obtain the following theorem.

Theorem 4. *Let* $\mathbf{ct}$ *and* $\mathbf{ct}'$ *be ciphertexts in* $R_{q_{2,i+1}}$ *and* $R_{q_{2,i}}$ *respectively, such that* $\mathbf{ct}' \leftarrow \mathsf{Scale}(\mathbf{ct}, q_{2,i+1}, q_{2,i}, p)$*. Then*

$$(\langle \mathbf{ct}', \mathbf{s} \rangle \pmod{q_{2,i}}) \pmod{p} \equiv (\langle \mathbf{ct}, \mathbf{s} \rangle \pmod{q_{2,i+1}}) \pmod{p}).$$

Further, if the input ciphertext has noise ν*, then the output ciphertext* $\mathbf{ct}'$ *will have noise* $B_{\text{scale}} + \frac{\nu}{p_{i+1}}$*, where*

$$B_{\text{scale}} := 8p \cdot \sqrt{h \cdot \phi(m)/3}.$$

Proof. Re-write the noise error e as

$$\begin{aligned} e_{q_{2,i+1}} &= \langle \mathbf{ct}, \mathbf{s} \rangle \pmod{q_{2,i+1}} \\ &= \langle \mathbf{ct}, \mathbf{s} \rangle + k \cdot q_{2,i+1} \\ &= \nu. \end{aligned}$$

We also let δ be the following error term.

$$\delta = \mathbf{ct}' - \frac{1}{p_{i+1}} \cdot \mathbf{ct}.$$

Notice this has coefficients in the interval $[-p/2, p/2)$. For some integer k, we then have that

$$\begin{aligned} e_{q_{2,i}} &\equiv (\langle \mathbf{ct}', \mathbf{s} \rangle \pmod{q_{2,i}}) \pmod{p} \\ &\equiv \langle \mathbf{ct}', \mathbf{s} \rangle - k \cdot q_{2,i} \pmod{p} \\ &\equiv \langle \mathbf{ct}, \mathbf{s} \rangle - k \cdot q_{2,i+i} \pmod{p} \\ &\equiv (\langle \mathbf{ct}, \mathbf{s} \rangle \pmod{q_{2,i+1}}) \pmod{p}, \end{aligned}$$

which follows from (5). It follows that

$$(\langle \mathbf{ct}', \mathbf{s}\rangle \pmod{q_{2,i}}) \pmod{p} \equiv (\langle \mathbf{ct}, \mathbf{s}\rangle \pmod{q_{2,i+1}}) \pmod{p}.$$

To simplify our noise analysis, write the polynomial δ as $\delta = (\delta_0, \delta_1)$. Both its components have coefficients in the interval $[-p/2, p/2)$. Recall $\mathbf{s} = (-\frac{q_{2,i}}{q_{1,i}} \cdot \mathbf{sk}, 1)$.

$$\begin{aligned}
\|e_{q_{2,i}}\|_\infty^{\text{can}} &= \|\langle \mathbf{ct}', \mathbf{s}\rangle - k \cdot q_{2,i}\|_\infty^{\text{can}} \\
&= \|\langle \mathbf{ct}', \mathbf{s}\rangle - k \cdot q_{2,i} - \langle \frac{1}{p_{i+1}}\mathbf{ct}, \mathbf{s}\rangle + \langle \frac{1}{p_{i+1}}\mathbf{ct}, \mathbf{s}\rangle\|_\infty^{\text{can}} \\
&= \|\langle \mathbf{ct}' - \frac{1}{p_{i+1}}\mathbf{ct}, \mathbf{s}\rangle + \langle \frac{1}{p_{i+1}}\mathbf{ct}, \mathbf{s}\rangle - k \cdot q_{2,i}\|_\infty^{\text{can}} \\
&\leq \|\langle \delta, \mathbf{s}\rangle\|_\infty^{\text{can}} + \frac{1}{p_{i+1}}\|e_{q_{2,i+1}}\|_\infty^{\text{can}} \\
&\leq \|\delta_1 - \frac{q_{2,i}}{q_{1,i}} \cdot \mathbf{sk} \cdot \delta_0\|_\infty^{\text{can}} + \frac{1}{p_{i+1}}\|e_{q_{2,i+1}}\|_\infty^{\text{can}} \\
&\leq \|\delta_1\|_\infty^{\text{can}} + \frac{q_{2,i}}{q_{1,i}}\|\mathbf{sk} \cdot \delta_0\|_\infty^{\text{can}} + \frac{1}{p_{i+1}}\|e_{q_{2,i+1}}\|_\infty^{\text{can}} \\
&\leq p\sqrt{3\phi(m)} + \frac{8p \cdot q_{2,i}}{q_{1,i}} \cdot \sqrt{h \cdot \phi(m)/3} + \frac{\nu}{p_{i+1}} \\
&:= B_{\text{scale}} + \frac{\nu}{p_{i+1}}.
\end{aligned}$$

4 Parameter Analysis

There are two methodologies to select parameters in cryptography; either to use parameters which arise from experimental cryptanalysis, or to use parameters which arise from the tightness of cryptographic reductions. Since RLWR is a less studied problem, we use a combination of these approaches. For our 'conservative' approach we will use (the ring version of) Theorem 1 to relate the hardness of search RLWR to that of search RLWE. In our 'reckless' approach we simply equate the security of RLWR to that of RLWE and ignore the loss in tightness inherent in Theorem 1[2]. In both cases, we then use the analysis of Lindner and Peikert [27] to derive estimates for the hardness of RLWE in this situation. We could have used more accurate estimates for the hardness of RLWE, however we use the analysis in [27] so that our parameter size estimates for our Somewhat Homomorphic Encryption scheme based on RLWR can be compared to the parameters in other works such as [12, 24].

To compare our parameters with those of other schemes we use as our baseline the parameter sizes published in [12] for the FV and BGV schemes and use the same analysis (see Section 4). For that reason we also select secret keys with Hamming weight $h = 64$, ring constant $c_m = 1.3$ and security parameter $k = 80$. We consider the simple case where we are performing a balanced tree of multiplicative depth L, with $\zeta = 8$ addition at each level.

[2] After all this is exactly what happens in deployed cryptosystems in practice.

Conservative Parameters: For the conservative parameters we obtain the following equations from the analysis in Section 4, where we take $\sigma = 3.2$,

$$\begin{aligned} \epsilon_{\mathsf{RLWR}} &\leq \left(1 + 38.4 \cdot q_{2,L-1}/q_{1,L-1}\right)^{t \cdot \phi(m)/2} \cdot \sqrt{\epsilon_{\mathsf{RLWE}}} < 2^{-k}, \\ q_{1,L-1} &> 38.4 \cdot q_{2,L-1}, \\ \phi(m) &\geq \frac{\log(q_{1,L-1}/\sigma) \cdot (-(\log_2 \epsilon_{\mathsf{RLWE}}) + 110)}{7.2} \end{aligned} \tag{6}$$

where t is the number of RLWR samples which we give out. In our scheme this is the number of elements in the public key which is given by

$$t = \ell + \lceil \log_T q_{1,L-1} \rceil.$$

This then gives us a parameter space which we need to search to find suitable parameter values.

Reckless Parameters: For the reckless parameters we obtain the following equations, where again we take $\sigma = 3.2$,

$$\begin{aligned} q_{1,L-1} &> 38.4 \cdot q_{2,L-1}, \\ \phi(m) &\geq \frac{\log(q_{1,L-1}/\sigma) \cdot (k + 110)}{7.2} \end{aligned} \tag{7}$$

We are looking for parameters which satisfy the equations above and yield the smallest possible ciphertext size. To do this we iterate through a list of possible values for $\log_2 q_{L-1}$. We then determine $\phi(m)$, as the smallest value which satisfies equation (6) or (7).

We proceed to examine the noise bounds at each level: at level zero we have noise $B_0 = \sqrt{3} \cdot \ell \cdot \phi(m)/2$, where we select $\ell = 80$ since we are assuming a security level of $k = 80$ bits. We then determine the size of p_{L-1}, as follows: after ζ additions and one multiplication we obtain a noise bound of

$$B_{L-1} = G(\zeta \cdot B_0, \zeta \cdot B_0).$$

We then reduce this via a Modulus Switch operation to a noise value of $B = 2 \cdot B_{\mathsf{Scale}}$, which will provide us with a lower bound on the prime p_{L-1}. Repeating this for all other levels we obtain

$$B_i = G(\zeta \cdot 2 \cdot B_{\mathsf{Scale}}, \zeta \cdot 2 \cdot B_{\mathsf{Scale}}).$$

This allows us, on reducing the noise back to an invariant of $2 \cdot B_{\mathsf{Scale}}$, to define p_i for all other i. Thus we obtain the size of $\log_2 q_{L-2}$, via

$$\log_2 q_{2,i} = \log_2 q_{2,i+1} - \log_2 p_{i+1}.$$

If we obtain $\log_2 q_{2,i} < 0$ then we abort and pass to the next of $\log_2 q_{L-1}$ value. To decrypt correctly we will finally require, by equation 2, that

$$2 \cdot B_{\mathsf{Scale}} \leq \Delta_{q_{2,0}}/(2 \cdot c_m),$$

which gives a lower bound on q_2.

In the following tables we present the obtained values of N, $q_{1,L-1}$ and $q_{2,L-1}$ for various levels L. We look at plaintext space sizes of $p = 2$ and $p = 2^{32}$; of course the same parameter sizes will hold for any plaintext p of the same order of magnitude as these. The value of T makes little difference to these main parameter values, so we simply selected it to be equal to two. In a real implementation a larger value of T may be preferred so as to reduce the size of the public key.

By way of comparison we also present in the table the equivalent values of $\phi(m)$ and q for the BGV system found in [12]; whose methodology for assessing security we have replicated in our analysis. We select BGV over the FV system for our analysis as the BGV system outperforms FV for large plaintext moduli. We use the values from [12] which correspond to our method of relinearization; a second method of relinearization is presented for BGV in [12, 24]. Recall a BGV ciphertext consists of two elements in R_q, thus the single q value takes the place of our values $q_{1,L-1}$ and $q_{2,L-1}$.

As a compact way of comparing our scheme with the BGV scheme we note that the ciphertext size in our scheme is given by $\phi(m)\cdot(\log_2 q_{1,L-1}+\log_2 q_{2,L-1})$, whereas for BGV it is given by $2\cdot\phi(m)\cdot\log_2 q$. Also recall that addition in R_q requires $\phi(m)\cdot\log_2 q$ bit operations, and multiplication in R_q requires $\phi(m)\cdot(\log_2 q)^2$ operations (assuming elements are held in a DCRT like representation [24]). Thus ciphertext size is a good proxy for computational efficiency. We gather our results in two tables, one representing the reckless parameters 4 and one representing the conservative ones 4. We add to this the ciphertex sizes (in Megabytes) for each set of parameters.

		Reckless				BGV Values		
p	L	$\phi(m)\approx$	$\log_2 q_{1,L-1}$	$\log_2 q_{2,L-1}$	$\lvert\mathbf{ct}\rvert$	$\phi(m)\approx$	$\log_2 q$	$\lvert\mathbf{ct}\rvert$
2	2	810	32	26	0.006	1000	45	0.01
2	5	1890	73	66	0.031	2000	105	0.05
2	10	3630	139	133	0.118	4000	215	0.20
2	20	7560	288	281	0.513	8000	430	0.82
2	30	11700	444	438	1.230	12000	665	1.90
2^{32}	2	3050	117	111	0.083	2000	110	0.05
2^{32}	5	6640	253	247	0.396	5000	265	0.31
2^{32}	10	12700	481	475	1.447	9500	530	1.20
2^{32}	20	25100	951	945	5.673	19500	1070	4.97
2^{32}	30	37200	1411	1405	12.488	29000	1598	11.04

		Conservative				BGV Values		
p	L	$\phi(m)\approx$	$\log_2 q_{1,L-1}$	$\log_2 q_{2,L-1}$	$\lvert\mathbf{ct}\rvert$	$\phi(m)\approx$	$\log_2 q$	$\lvert\mathbf{ct}\rvert$
2	2	1790	47	27	0.016	1000	45	0.01
2	5	3410	91	68	0.065	2000	105	0.05
2	10	6240	166	141	0.228	4000	215	0.20
2	20	12200	322	295	0.897	8000	430	0.82
2	30	18000	479	450	1.993	12000	665	1.90
2^{32}	2	5150	136	112	0.152	2000	110	0.05
2^{32}	5	10450	276	250	0.655	5000	265	0.31
2^{32}	10	18900	504	474	2.203	9500	530	1.20
2^{32}	20	37000	975	945	4.469	19500	1070	4.97
2^{32}	30	54000	1437	1404	18.288	29000	1598	11.04

We can see that selecting 'reckless' parameters nearly halves the ring dimension needed, as can be expected as we are ignoring the square-root security loss in the security reduction in this case. In comparing to BGV, we immediately notice that for small plaintext moduli the ciphertext sizes are relatively comparable. Indeed even our conservative parameters seem to be better when looking at parameters which support high depth evaluations for small plaintext moduli. The benefit is less clear for large plaintext moduli. Our main bottleneck seems to be the size of the public key and how to efficiently produce fresh randomness, an issue we discuss in Appendix B.

Acknowledgements

This work has been supported in part by ERC Advanced Grant ERC-2015-AdG-IMPaCT, by EPSRC via grant EP/N021940/1 and by the European Union's H2020 Programme under grant agreement number ICT-644209 (HEAT). The authors would like to thank Fucai Luo and Fre Vercauteren for helpful discussions whilst writing this paper.

References

1. Martin R. Albrecht, Shi Bai, and Léo Ducas. A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 153–178. Springer, 2016.
2. Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited - new reduction, properties and applications. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 57–74. Springer, 2013.
3. Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 719–737. Springer, 2012.
4. Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon. On the hardness of learning with rounding over small modulus. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 209–224. Springer, 2016.
5. Joppe W. Bos, Kristin E. Lauter, Jake Loftus, and Michael Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In Martijn Stam, editor, *Cryptography and Coding - 14th IMA International Conference, IMACC 2013, Oxford, UK, December 17-19, 2013. Proceedings*, volume 8308 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2013.
6. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. Fully homomorphic encryption without bootstrapping. In *Innovations in Theoretical Computer Science (ITCS'12)*, 2012.

7. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS'11*. IEEE Computer Society, 2011.
8. Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer, 2011.
9. P. Campbell, M. Groves, and D. Shepherd. SOLILOQUY: A cautionary tale. ETSI 2nd Quantum-Safe Crypto Workshop, 2014.
10. Jean-Sébastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 487–504. Springer, 2011.
11. Jean-Sébastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 446–464. Springer, 2012.
12. Ana Costache and Nigel P. Smart. Which ring based somewhat homomorphic encryption scheme is best? In Kazue Sako, editor, *Topics in Cryptology - CT-RSA 2016 - The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings*, volume 9610 of *Lecture Notes in Computer Science*, pages 325–340. Springer, 2016.
13. Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 559–585. Springer, 2016.
14. Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In Safavi-Naini and Canetti [31], pages 643–662.
15. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in Cryptology - EUROCRYPT'10*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2010.
16. Léo Ducas and Daniele Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 617–640. Springer, 2015.
17. Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.
18. Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `http://crypto.stanford.edu/craig`.
19. Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *STOC*, pages 169–178. ACM, 2009.
20. Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 129–148. Springer, 2011.
21. Craig Gentry, Shai Halevi, Chris Peikert, and Nigel P. Smart. Ring switching in bgv-style homomorphic encryption. In Ivan Visconti and Roberto De Prisco, editors, *Security and Cryptography for Networks - 8th International Conference, SCN 2012, Amalfi, Italy, September 5-7, 2012. Proceedings*, volume 7485 of *Lecture Notes in Computer Science*, pages 19–37. Springer, 2012.

22. Craig Gentry, Shai Halevi, and Nigel Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 465–482. Springer, 2012.
23. Craig Gentry, Shai Halevi, and Nigel P. Smart. Better bootstrapping in fully homomorphic encryption. In Marc Fischlin, Johannes A. Buchmann, and Mark Manulis, editors, *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, volume 7293 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2012.
24. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In Safavi-Naini and Canetti [31], pages 850–867.
25. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013, Part I*, pages 75–92. Springer, 2013.
26. Shai Halevi and Victor Shoup. Design and implementation of a homomorphic-encryption library. manuscript, available at `http://people.csail.mit.edu/shaih/pubs/he-library.pdf`, Accessed January 2015.
27. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for lwe-based encryption. In *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2011.
28. Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1219–1234. ACM, 2012.
29. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23, 2010.
30. Ron Rivest, Leonard Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. In *Foundations of Secure Computation*, pages 169–180, 1978.
31. Reihaneh Safavi-Naini and Ran Canetti, editors. *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*. Springer, 2012.
32. Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography - PKC'10*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer, 2010.

A Multiplication Noise Analysis

In this section, we present the noise and correctness analyses corresponding to the multiplication steps.

Multiplication Step 1: Tensoring: We first verify that the three-element ciphertext obtained after the first multiplication step does decrypt under s^2 and after scaling by $\frac{1}{\Delta^2_{q_{2,i}}}$. The three element ciphertext formed in Section (3.2)

$$\mathbf{ct}^0_{\text{mult}} = (v \cdot v', w \cdot v' + w' \cdot v, w \cdot w') = (d'_0, d'_1, d'_2),$$

will indeed decrypt via the equation

$$\begin{aligned}
&\Big\lceil \frac{1}{\Delta_{q_{2,i}}^2}\Big(d_2' - \frac{q_{2,i}}{q_{1,i}} \cdot d_1' \cdot s + \left(\frac{q_{2,i}}{q_{1,i}}\right)^2 \cdot d_0' \cdot s^2\Big)\Big\rfloor \\
&\qquad = \Big\lceil \frac{1}{\Delta_{q_{2,i}}^2}\Big(w \cdot w' - \frac{q_{2,i}}{q_{1,i}}(w \cdot v' + w' \cdot v) \cdot s + \left(\frac{q_{2,i}}{q_{1,i}}\right)^2 \cdot v \cdot v' \cdot s^2\Big)\Big\rfloor \\
&\qquad = \Big\lceil \frac{1}{\Delta_{q_{2,i}}^2} \cdot \left(w - \frac{q_{2,i}}{q_{1,i}} \cdot v \cdot s\right) \cdot \left(w' - \frac{q_{2,i}}{q_{1,i}} \cdot v' \cdot s\right)\Big\rfloor \\
&\qquad = \Big\lceil \frac{1}{\Delta_{q_{2,i}}^2} \cdot \left(\Delta_{q_{2,i}} \cdot \mathbf{m} + e\right) \cdot \left(\Delta_{q_{2,i}} \cdot \mathbf{m}' + e'\right)\Big\rfloor \\
&\qquad = \Big\lceil \frac{1}{\Delta_{q_{2,i}}^2} \cdot \Big(\Delta_{q_{2,i}}^2 \cdot \mathbf{m} \cdot \mathbf{m}' + \Delta_{q_{2,i}} \cdot (\mathbf{m} \cdot e' + \mathbf{m}' \cdot e) + e \cdot e'\Big)\Big\rfloor \\
&\qquad = \mathbf{m} \cdot \mathbf{m}' + \Big\lceil \frac{\mathbf{m} \cdot e' + \mathbf{m}' \cdot e}{\Delta_{q_{2,i}}} + \frac{e \cdot e'}{\Delta_{q_{2,i}}^2}\Big\rfloor.
\end{aligned}$$

Multiplication Step 2: Reduction by $\boldsymbol{\Delta_{q_{2,i}}}$: Recall we re-wrote the elements d_k' as

$$d_k = \Big\lceil \frac{p \cdot d_k'}{q_{2,i}} \Big\rfloor.$$

To analyse the noise growth in this step, we need to examine this rounding in more detail via the equation

$$d_k = \frac{p \cdot d_k'}{q_{2,i}} + \delta_k,$$

where δ_i is a polynomial with coefficients in the range $[-1/2, \ldots, 1/2]$. We write

$$\delta = \delta_2 - \frac{q_{2,i}}{q_{1,i}} \cdot \delta_1 \cdot s + \left(\frac{q_{2,i}}{q_{1,i}}\right)^2 \cdot \delta_0 \cdot s^2$$

and note that with high probability we expect that

$$\|\delta\|_\infty^{\mathrm{can}} \leq \sqrt{3 \cdot \phi(m)} + 8 \cdot \frac{q_{2,i}}{q_{1,i}} \cdot \sqrt{h \cdot \phi(m)/3} + 20 \cdot \left(\frac{q_{2,i}}{q_{1,i}}\right)^2 \cdot h \cdot \sqrt{\phi(m)/3} = B_\delta.$$

Recall we have $\Delta_{q_{2,i}} = \frac{q_{2,i}}{p} - \epsilon_{q_{2,i}}$ where $0 \leq \epsilon_{q_{2,i}} < 1$, which entails

$$\begin{aligned}
d_2 - \frac{q_{2,i}}{q_{1,i}} \cdot \;& d_1 \cdot s + \left(\frac{q_{2,i}}{q_{1,i}}\right)^2 \cdot d_0 \cdot s^2 \\
&= \frac{p}{q_{2,i}}\left(d_2' - \frac{q_{2,i}}{q_{1,i}} \cdot d_1' \cdot s + \left(\frac{q_{2,i}}{q_{1,i}}\right)^2 \cdot d_0' \cdot s^2\right) + \delta \\
&= \frac{p}{q_{2,i}} \cdot \Big(\Delta_{q_{2,i}}^2 \cdot \mathbf{m} \cdot \mathbf{m}' + \Delta_{q_{2,i}} \cdot (\mathbf{m} \cdot e' + \mathbf{m}' \cdot e) + e \cdot e'\Big) + \delta
\end{aligned}$$

$$
\begin{aligned}
&= \frac{p}{q_{2,i}} \cdot \Big(\Delta_{q_{2,i}} \cdot (\frac{q_{2,i}}{p} - \epsilon_{q_{2,i}}) \cdot \mathbf{m} \cdot \mathbf{m}' \\
&\qquad + \Delta_{q_{2,i}} \cdot (\mathbf{m} \cdot e' + \mathbf{m}' \cdot e) + e \cdot e' \Big) + \delta \\
&= \mathbf{m} \cdot \mathbf{m}' \cdot \Delta_{q_{2,i}} - \frac{p \cdot \Delta_{q_{2,i}} \cdot \epsilon_{q_{2,i}}}{q_{2,i}} \cdot \mathbf{m} \cdot \mathbf{m}' \\
&\qquad + \frac{p}{q_{2,i}} \cdot \Delta_{q_{2,i}} \cdot (\mathbf{m} \cdot e' + \mathbf{m}' \cdot e) + \frac{p \cdot e \cdot e'}{q_{2,i}} + \delta \\
&= \mathbf{m} \cdot \mathbf{m}' \cdot \Delta_{q_{2,i}} + e_{\mathrm{mult}}.
\end{aligned}
$$

We bound the canonical norm of e_{mult} as follows, using the fact that $|p \cdot \Delta_{q_{2,i}}/q_{2,i}| < 1$.

$$
\begin{aligned}
\|e_{\mathrm{mult}}\|_\infty^{\mathrm{can}} &\leq \|\epsilon_{q_{2,i}} \cdot \mathbf{m} \cdot \mathbf{m}'\|_\infty^{\mathrm{can}} + \|\mathbf{m} \cdot e'\|_\infty^{\mathrm{can}} + \|\mathbf{m}' \cdot e\|_\infty^{\mathrm{can}} \\
&\qquad + \frac{p \cdot \|e\|_\infty^{\mathrm{can}} \cdot \|e'\|_\infty^{\mathrm{can}}}{q_{2,i}} + \|\delta\|_\infty^{\mathrm{can}} \\
&\leq 4 \cdot p^2 \cdot \phi(m)/3 + 6 \cdot (B + B') \cdot p \cdot \sqrt{\phi(m)/12} + \frac{p \cdot B \cdot B'}{q_{2,i}} + B_\delta \\
&=: F(B, B').
\end{aligned}
$$

Multiplication Step 3: Modular Reduction: Recall that in this step we set

$$
\begin{aligned}
f_0 = d_0 \pmod{q_{1,i}^2} &= d_0 + \epsilon_0 \cdot q_{1,i}^2, \\
f_1 = d_1 \pmod{q_{1,i}} &= d_1 + \epsilon_1 \cdot q_{1,i}, \\
f_2 = d_2 \pmod{q_{2,i}} &= d_2 + \epsilon_2 \cdot q_{2,i}.
\end{aligned}
$$

In this form, $\mathbf{ct}_{\mathrm{mult}}^2 = (f_0, f_1, f_2)$ decrypts as

$$
\Big\lceil \frac{1}{\Delta_{q_{2,i}}} \left(f_2 - \frac{q_{2,i}}{q_{1,i}} \cdot f_1 \cdot s + \left(\frac{q_{2,i}}{q_{1,i}} \right)^2 \cdot f_0 \cdot s^2 \right) \Big\rfloor. \tag{8}
$$

To analyse the noise term in the above we expand the inner bracket as follows:

$$
\begin{aligned}
&f_2 - \frac{q_{2,i}}{q_{1,i}} \cdot f_1 \cdot s + \left(\frac{q_{2,i}}{q_{1,i}} \right)^2 \cdot f_0 \cdot s^2 \\
&= d_2 + \epsilon_2 \cdot q_{2,i} - \frac{q_{2,i}}{q_{1,i}} \cdot (d_1 + \epsilon_1 \cdot q_{1,i}) \cdot s + \left(\frac{q_{2,i}}{q_{1,i}} \right)^2 \cdot \left(d_0 + \epsilon_0 \cdot q_{1,i}^2 \right) \cdot s^2 \\
&= d_2 + \epsilon_2 \cdot q_{2,i} - \frac{q_{2,i}}{q_{1,i}} \cdot d_1 \cdot s - \frac{q_{2,i}}{q_{1,i}} \cdot \epsilon_1 \cdot q_{1,i} \cdot s \\
&\qquad + \left(\frac{q_{2,i}}{q_{1,i}} \right)^2 \cdot d_0 \cdot s^2 + \left(\frac{q_{2,i}}{q_{1,i}} \right)^2 \cdot \epsilon_0 \cdot q_{1,i}^2 \cdot s^2 \\
&= d_2 - \frac{q_{2,i}}{q_{1,i}} \cdot d_1 \cdot s + \left(\frac{q_{2,i}}{q_{1,i}} \right)^2 \cdot d_0 \cdot s^2 + q_{2,i} \cdot (\epsilon_2 - \epsilon_1 \cdot s + q_{2,i} \cdot \epsilon_0 \cdot s^2)
\end{aligned}
$$

$$= \mathbf{m} \cdot \mathbf{m}' \cdot \Delta_{q_{2,i}} + e_{\text{mult}} + \epsilon_{\text{mult}}.$$

We now proceed to analyse the new noise term ϵ_{mult}.

$$\begin{aligned}\left\|\epsilon_{\text{mult}}\right\|_\infty^{\text{can}} &\leq \left\|q_{2,i} \cdot (\epsilon_2 - \epsilon_1 \cdot s + q_{2,i} \cdot \epsilon_0 \cdot s^2)\right\|_\infty^{\text{can}} \\ &\leq q_{2,i} \cdot \left(\left\|\epsilon_2\right\|_\infty^{\text{can}} + \left\|\epsilon_1 \cdot s\right\|_\infty^{\text{can}} + q_{2,i} \cdot \left\|\epsilon_0 \cdot s^2\right\|_\infty^{\text{can}}\right)\end{aligned}$$

We now use the expansions $\epsilon_0 = (f_0 + d_0)/q_{1,i}^2$ etc, and in what follows we will assume that the f_0 (resp. f_1 and f_2) behave as random polyomials with coefficients modulo $q_{1,i}^2$ (resp. $q_{1,i}$ and $q_{2,i}$).

Please do this again, and expand every thing to make it easy to follow, and also to AVOID MISTAKES. You had TONS of mistakes below. So many it was not even worth me keeping what you had written, so I just deleted it and am starting again. Dont skip steps and try to do them in your head. You clearly cannot, as you make so many mistakes. Expand everything, and reduce each thing in turn. And do proper indentation! And make sure connecting sentences make sense and refer correctly to the stuff you are doing! I have done the first few lines. Please do the rest properly! Also check I have not made mistakes, as I am doing this in a rush at the airport. This leads to:

$$\begin{aligned}\left\|\epsilon_{\text{mult}}\right\|_\infty^{\text{can}} &\leq q_{2,i} \cdot \left(\left\|\epsilon_2\right\|_\infty^{\text{can}} + \left\|\epsilon_1 \cdot s\right\|_\infty^{\text{can}} + q_{2,i} \cdot \left\|\epsilon_0 \cdot s^2\right\|_\infty^{\text{can}}\right) \\ &\leq q_{2,i} \cdot \Big(\frac{\|f_2\|_\infty^{\text{can}}}{q_{2,i}} + \frac{\|d_2\|_\infty^{\text{can}}}{q_{2,i}} + \frac{\|f_1 \cdot s\|_\infty^{\text{can}}}{q_{1,i}} + \frac{\|d_1 \cdot s\|_\infty^{\text{can}}}{q_{1,i}} \\ &\qquad + q_{2,i} \cdot \frac{\|f_0 \cdot s^2\|_\infty^{\text{can}}}{q_{1,i}^2} + q_{2,i} \cdot \frac{\|d_0 \cdot s^2\|_\infty^{\text{can}}}{q_{1,i}^2}\Big) \\ &\leq \|d_2\|_\infty^{\text{can}} + \frac{q_{2,i} \cdot \|d_1 \cdot s\|_\infty^{\text{can}}}{q_{1,i}} + \frac{q_{2,i}^2 \cdot \|d_0 \cdot s^2\|_\infty^{\text{can}}}{q_{1,i}^2} \\ &\qquad + \|f_2\|_\infty^{\text{can}} + \frac{q_{2,i} \cdot \|f_1 \cdot s\|_\infty^{\text{can}}}{q_{1,i}} + \frac{q_{2,i}^2 \cdot \|f_0 \cdot s^2\|_\infty^{\text{can}}}{q_{1,i}^2} \\ &\leq \dots\end{aligned}$$

So now in the above work out the bounds on the terms using the f_i using the fact they are random. Note that f_1 is multiplied by s and f_0 is multiplied by s^2. You utterly failed to notice this in your analysis! We then use the fact that

$$d_k = \frac{p \cdot d_k'}{q_{2,i}} + \delta_k.$$

To simplify further Again EXPAND out things slowly so you dont make mistakes!

$$\begin{aligned}\left\|\epsilon_{\text{mult}}\right\|_\infty^{\text{can}} &\leq \dots \\ &=: B_\epsilon.\end{aligned}$$

Multiplication Step 4: Relinearization: Recall that in this step we set

$$c_0 = f_1 + \sum_{j=0}^{\lceil 2\cdot\log_T(q_{1,i})\rceil} f_{0,j}\cdot a_j,$$
$$c_1 = f_2 + \sum_{j=0}^{\lceil 2\cdot\log_T(q_{1,i})\rceil} f_{0,j}\cdot b_j.$$

Checking correctness,

$$\begin{aligned}
\frac{1}{\Delta_{q_{2,i}}}\left(-\frac{q_{2,i}}{q_{1,i}}\cdot c_0\cdot s + c_1\right) &= \frac{1}{\Delta_{q_{2,i}}}\Big(-\frac{q_{2,i}}{q_{1,i}}\cdot f_1\cdot s - \frac{q_{2,i}}{q_{1,i}}\cdot s\cdot \sum_{j=0}^{2\cdot\lceil\log_T(q_{1,i})\rceil} f_{0,j}\cdot a_j \\
&\quad + f_2 + \sum_{j=0}^{2\cdot\lceil\log_T(q_{1,i})\rceil} f_{0,j}\cdot b_j\Big) \\
&= \frac{1}{\Delta_{q_{2,i}}}\Big(f_2 - \frac{q_{2,i}}{q_{1,i}}\cdot f_1\cdot s - \frac{q_{2,i}}{q_{1,i}}\cdot s\cdot \sum_{j=0}^{2\cdot\lceil\log_T(q_{1,i})\rceil} f_{0,j}\cdot a_j \\
&\quad + \sum_{j=0}^{2\cdot\lceil\log_T(q_{1,i})\rceil} f_{0,j}\cdot\frac{q_{2,i}}{q_{1,i}}\cdot a_j\cdot s \\
&\quad + \sum_{j=0}^{2\cdot\lceil\log_T(q_{1,i})\rceil} f_{0,j}\cdot\left(\frac{q_{2,i}}{q_{1,i}}\right)^2\cdot T^j\cdot s^2 \\
&\quad + \sum_{j=0}^{2\cdot\lceil\log_T(q_{1,i})\rceil} f_{0,j}\cdot\tau_j\Big) \\
&= \frac{1}{\Delta_{q_{2,i}}}\Big(d_2 - \frac{q_{2,i}}{q_{1,i}}\cdot f_1\cdot s - \frac{q_{2,i}}{q_{1,i}}\cdot s\cdot \sum_{j=0}^{2\cdot\lceil\log_T(q_{1,i})\rceil} f_{0,j}\cdot a_j \\
&\quad + \frac{q_{2,i}}{q_{1,i}}\cdot s\cdot \sum_{j=0}^{2\cdot\lceil\log_T(q_{1,i})\rceil} f_{0,j}\cdot a_j \\
&\quad + \left(\frac{q_{2,i}}{q_{1,i}}\right)^2\cdot s^2\cdot \sum_{j=0}^{2\cdot\lceil\log_T(q_{1,i})\rceil} f_{0,j}\cdot T^j \\
&\quad + \sum_{j=0}^{2\cdot\lceil\log_T(q_{1,i})\rceil} f_{0,j}\cdot\tau_j\Big) \\
&= \frac{1}{\Delta_{q_{2,i}}}\Big(f_2 - \frac{q_{2,i}}{q_{1,i}}\cdot f_1\cdot s + \left(\frac{q_{2,i}}{q_{1,i}}\right)^2\cdot f_0\cdot s^2
\end{aligned}$$

$$+ \sum_{j=0}^{2\cdot\lceil \log_T(q_{1,i})\rceil} f_{0,j}\cdot\tau_j\Big)$$

$$= \frac{1}{\Delta_{q_{2,i}}}\Big(\mathbf{m}\cdot\mathbf{m}'\cdot\Delta_{q_{2,i}} + e_{\text{mult}} + \epsilon_{\text{mult}} + \sum_{j=0}^{2\cdot\lceil \log_T(q_{1,i})\rceil} f_{0,j}\cdot\tau_j\Big).$$

Which is exactly as required, bar an additional noise term we need to deal with. We bound it as

$$\Big\| \sum_{j=0}^{2\cdot\lceil \log_T(q_{1,i})\rceil} f_{0,j}\cdot\tau_j \Big\|_\infty^{\text{can}} \leq 2\cdot\lceil \log_T(q_{1,i})\rceil\cdot\phi(m)\cdot T/3.$$

Explain why $f_{0,j}\cdot\tau_j$ is bounded by $\phi(m)\cdot T/3$. Maybe call this bound B_4 for the bound from the fourth step, and B_ϵ you can call B_3. Then $F(.,.)$ could be $B_2(.,.)$. Might be clearer then

Putting all three steps together we obtain a multiplication operation which increases the noise inherent in the output ciphertext by a value bounded, in the canonical norm, by

$$\Big\| e_{\text{mult}} + \epsilon_{\text{mult}} + \sum_{j=0}^{2\cdot\lceil \log_T(q_{1,i})\rceil} f_{0,j}\cdot\tau_j \Big\|_\infty^{\text{can}} \leq F(B,B') + 2\cdot\lceil \log_T(q_{1,i})\rceil\cdot\phi(m)\cdot T/3 + B_\epsilon$$

$$=: G(B,B').$$

Tidy this up, it breaks a line!

B Compact Public Key

At its heart our scheme is a symmetric key homomorphic encryption scheme. We then use the standard naive method to convert it into a public key scheme, by placing a large number of encryptions of zero into the public key, leading to a large public key. The RLWE based schemes use a different methodology, namely they place a single encryption of zero into the public key, and then present an efficient methodology to randomize this into another encryption of zero for the encryption method. At present no such direct methodology (which does not pass via some RLWE assumption) is available for RLWR encryption schemes, meaning the public key is relatively large compared to schemes such as BGV. Replicating this procedure for RLWR is somewhat trickier. To the best of our knowledge, the only way to do so efficiently is to slightly modify the hardness problem we are using.

So consider the following. Pick a uniformly at random from $R_{q_{1,i}}$ as before, but instead of the usual pair $(a, \lfloor a\cdot s \rceil_{q_{1,i},q_{2,i}}) \in R_{q_{1,i}}\times R_{q_{2,i}}$ we consider the slightly modified RLWR pair

$$(\lfloor a \rceil_{q_{1,i},q_{2,i}}, \lfloor a\cdot s \rceil_{q_{1,i},q_{2,i}}) = (\widetilde{a},\widetilde{b}) \in R_{q_{2,i}}\times R_{q_{2,i}}.$$

Then consider a lift of the pair to the ring $R_{q_{1,i}}$.

$$a^{\uparrow} = \left\lceil \widetilde{a} \cdot \frac{q_{1,i}}{q_{2,i}} \right\rfloor + r_a$$
$$b^{\uparrow} = \left\lceil \widetilde{b} \cdot \frac{q_{1,i}}{q_{2,i}} \right\rfloor + r_b,$$

where we choose the r_i according to (say) a bounded uniform distribution. Picking a small $u \leftarrow R_{q_{1,i}}$ and bringing this back down to $R_{q_{2,i}}$, we obtain new ring elements

$$a' = \left\lceil a^{\uparrow} \cdot u \right\rfloor_{q_{1,i},q_{2,i}}$$
$$b' = \left\lceil b^{\uparrow} \cdot u \right\rfloor_{q_{1,i},q_{2,i}}.$$

In the above, because we have

$$s \cdot a' \approx b',$$

this can be thought of as a modified RLWE instance. The error distribution, however is neither the usual Gaussian nor uniform. Therefore, we cannot make a concrete statement about the security of this 'modified' RLWE. The problem of how this relates to regular RLWE remains open.

Writing out the result of the procedure, we end up with something of the form

$$a' \approx u \cdot \widetilde{a} + \left\lceil u + r_a \right\rfloor_{q_{1,i},q_{2,i}}$$

and similarly for b'. Since we have preserved the term $\widetilde{a}$, there is no need to lift that element and so we can simplify the procedure by simply sampling the elements r_a and u in $R_{q_{1,i}}$ and mapping down as described above.

This procedure has the disadvantage of modifying the underlying hardness problems. Indeed, we discussed the unknown distribution in using the 'modified' RLWE above. Further, we modify the RLWR sample

$$(a, \left\lceil a \cdot s \right\rfloor_{q_{1,i},q_{2,i}}) \in R_{q_{1,i}} \times R_{q_{2,i}}$$

and instead consider a pair of elements

$$(\widetilde{a}, \widetilde{b}) = (\left\lceil a \right\rfloor_{q_{1,i},q_{2,i}}, \left\lceil a \cdot s \right\rfloor_{q_{1,i},q_{2,i}}) \in R_{q_{2,i}} \times R_{q_{2,i}}.$$

The question of how the security of the sample

$$(\left\lceil a \right\rfloor_{q_{1,i},q_{2,i}}, \left\lceil a \cdot s \right\rfloor_{q_{1,i},q_{2,i}})$$

relates to that of

$$(a, \left\lceil a \cdot s \right\rfloor_{q_{1,i},q_{2,i}})$$

remains open.