Designs for practical SHE schemes based on Ring-LWR

Madalina Bolboceanu[1], Anamaria Costache[2*], Erin Hales[3*], Rachel Player[3*], Miruna Rosca[4], Radu Titiu[1]

[1]Bitdefender, Romania.
[2]NTNU, Norway.
[3]Royal Holloway, University of London, UK.
[4]Pi Squared Inc., USA.

*Corresponding author(s). E-mail(s): anamaria.costache@ntnu.no; Erin.Hales.2018@live.rhul.ac.uk; rachel.player@rhul.ac.uk;
Contributing authors: mbolboceanu@bitdefender.com; miruna.rosca@pi2.network; rtitiu@bitdefender.com;

Abstract

The Learning with Errors problem (LWE) and its variants are among the most popular assumptions underlying lattice-based cryptography. The Learning with Rounding problem (LWR) can be thought of as a deterministic variant of LWE. While lattice-based cryptography is known to enable many advanced constructions, constructing Fully Homomorphic Encryption schemes based on LWR remains an under-explored part of the literature. In this work, we present a thorough study of Somewhat Homomorphic Encryption schemes based on Ring-LWR that are the analogue of the Ring-LWE-based BFV scheme. Our main contribution is to present and analyse two new schemes, in the LPR and Regev paradigms. The Regev-type scheme can be seen as a generalisation of the only prior work in this direction (Costache-Smart, 2017). Both our schemes present several improvements compared to this prior work, and in particular we resolve the "tangled modulus" issue in the Costache-Smart scheme that led to unmanageable noise growth. Our schemes inherit the many benefits of being based on LWR, including ease of implementation, avoiding the need for expensive Gaussian sampling, improved resistance to side channels, suitability for hardware, and improved ciphertext size. Indeed, we give a detailed comparison showing that the LPR and Regev-type schemes marginally outperform the BFV scheme in terms of ciphertext size. Moreover, we show that both our schemes support RNS variants, which would make their practical performance competitive with BFV.

Keywords: homomorphic encryption, public-key encryption, Ring-LWR, RNS

MSC Classification: 11T71 , 94A60 , 68P25

1 Introduction

Homomorphic Encryption.

Fully Homomorphic Encryption (FHE) [1] is an advanced cryptographic primitive that supports computations on ciphertexts without revealing any information about the underlying plaintexts. All known constructions of FHE are based on augmenting a Somewhat Homomorphic Encryption (SHE) scheme, which supports computation of circuits up to a certain depth, with bootstrapping, a process which refreshes ciphertexts so as to enable further computations. In applications where the circuit to be homomorphically evaluated is known in advance, SHE can be sufficient and indeed more efficient, as bootstrapping can be a very costly operation.

Homomorphic encryption schemes can be divided into four generations. The first generation of schemes includes Gentry's original scheme [1, 2]. The second generation schemes includes the BFV and BGV schemes [3–6]. These schemes have been extensively optimised [7–9] and are widely used and implemented [10–13], most often in the SHE setting. These schemes tend to have a very slow bootstrapping operation, but have very good Single Instruction Multiple Data (SIMD) properties. The third generation of FHE schemes began with the GSW scheme [14], and continued with the line of work of [15, 16]. In contrast to the second generation schemes, these schemes support very fast bootstrapping, but have very low SIMD capabilities. Finally, the fourth generation of schemes includes the approximate scheme CKKS [17]. These schemes behave similarly to the second generation ones, with slow bootstrapping and very high SIMD capabilities, but they are approximate.

Learning with Errors in FHE.

The Learning with Errors problem (LWE) [18], and its structured variants, such as Ring-LWE [19, 20], are widely used hardness assumptions in lattice-based cryptography [14, 21–23]. The most widely used homomorphic encryption schemes, including those that are being considered for standardisation [10], all base their security on the Ring Learning with Errors problem (Ring-LWE).

Learning with Rounding - an alternative to Learning with Errors.

The Learning with Rounding (LWR) problem was introduced by Banerjee et al. [24] as a deterministic alternative to LWE and its variants. In this work, we focus on its ring variant, Ring-LWR [24]. We provide an informal definition below.

The *Decision Ring-LWR problem* for $q > p$ in the rings $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ and $R_p = \mathbb{Z}_p[x]/(x^n + 1)$, denoted Decision Ring-LWR$_{n,q,p}$, asks to distinguish uniformly random pairs of elements $(a, b) \in R_q \times R_p$ from pairs sampled from the distribution that outputs $(a, b = \lfloor a \cdot s \rceil_{q,p}) \in R_q \times R_p$ where $a \in R_q$ is uniformly random, $s \in R_q$

is a secret polynomial, and the rounding operation is defined (coefficient-wise) as $\lfloor x \rceil_{q,p} := \left\lfloor \frac{p}{q} x \right\rceil$.

The focus of this paper is to present new SHE schemes whose security is based on the hardness of Ring-LWR. Our schemes can be seen as analogues of the second-generation BFV scheme [6], and most optimisations that can be applied to BFV naturally extend to our schemes.

Why Learning with Rounding?

A key benefit of exploring LWR-based FHE is the potential applicability to hardware acceleration. Hardware acceleration has been an important recent direction towards making FHE practical [25–30], as it can improve the performance of computationally heavy operations such as bootstrapping [26, 29] and enable highly parallelised RNS computations [27, 30]. We expect LWR-based FHE schemes to be amenable to hardware acceleration, since they inherit many of the properties that have been shown to be for public-key encryption and key encapsulation acceleration [31].

A further benefit is improved bandwidth in LWR-based schemes compared to their LWE-based counterparts [32]. Bandwidth efficiency is critical in FHE applications such as Private Information Retrieval [33] or Private Set Intersection [34], which typically require the transmission of many ciphertexts to reduce computational cost.

More generally, LWR-based schemes can be preferred to their LWE-based analogues for several other reasons. For example, using rounding avoids the need for Gaussian noise sampling, which can be expensive [35], and vulnerable to side channel attacks [36–43]. In addition, the rounding function is easy to implement: for example, if q and p are both chosen as powers of 2, as in [44–46], then rounding corresponds to simply dropping the least significant bits.

Several candidates in the NIST post-quantum standardisation process used LWR for their underlying hardness assumption, including Lizard [45], Round 5 [44], as well as the third-round finalist Saber [46]. While LWE enables many advanced constructions, including for example attribute-based encryption [47], counterparts based on LWR or its variants have not been as well explored. To the best of our knowledge, apart from homomorphic encryption [35, 48, 49], only identity-based encryption [50, 51] has been studied.

The open problem.

Costache and Smart gave the first proposal of an SHE construction based on an LWR assumption [35]. As observed in [48, 49], the proposal suffers from the so-called "tangled modulus" problem which leads to unmanageably large noise growth.

To describe the main issue with the construction, we note that the Costache-Smart scheme is BFV-like and thus introduces the parameter $\Delta_p := \left\lfloor \frac{p}{t} \right\rfloor = \frac{p}{t} - \epsilon_p$, where $0 \leq \epsilon_p < 1$, p is one of the ciphertext moduli, and t is the plaintext modulus. The parameter Δ_p is used in encryption to put the message in the high-order bits. Then, ciphertexts of the Costache-Smart scheme are of the form:

$$(\mathtt{ct}_0, \mathtt{ct}_1) = \left(\sum_{k=1}^{\ell} r_k \cdot v_k, \Delta_p \cdot m + \sum_{k=1}^{\ell} r_k \cdot w_k \right) \in R_q \times R_p,$$

where $m \in R_t$ is a message, the r_k's are uniform random bits from $\{0, 1\}$, and the public key is given by the ℓ pairs $(v_k, w_k) \in R_q \times R_p$. The issue arises in homomorphic multiplication, which is also analogous to BFV multiplication. Namely, when multiplying two ciphertexts $(\mathtt{ct}_0, \mathtt{ct}_1)$ and $(\mathtt{ct}'_0, \mathtt{ct}'_1)$, we obtain an intermediate ciphertext of the form $(\mathtt{ct}_0\mathtt{ct}'_0, \mathtt{ct}_0\mathtt{ct}'_1 + \mathtt{ct}'_0\mathtt{ct}_1, \mathtt{ct}_1\mathtt{ct}'_1)$, scaled by an appropriate amount, which is chosen in [35] to be $1/\Delta_p$.

In [35], it is not explicitly stated to which ring any component of the intermediate ciphertext should correspond. Luo *et al.* [48] assert, for example, that the $\mathtt{ct}_0\mathtt{ct}'_0$ component should be interpreted modulo q. In this case, when calculating the noise growth in multiplication, we would lift to R and manipulate an object of the form $\mathtt{ct}_0\mathtt{ct}'_0 + kq$ where k is a polynomial with integer coefficients. This leads to noise terms including a factor of k. We cannot tightly bound k, and the worst-case bound is very large. This leads Luo et al. [48] to conclude that presenting a homomorphic encryption scheme from LWR assumptions that is analogous to BFV is not possible. In this work we are able to present such a scheme, resolving the previously open problem.

Our contribution.

In this work, we present a thorough study of BFV-like SHE schemes based on Ring-LWR. These can be seen as analogous of the second generation (R)LWE-based SHE schemes. Our main contributions are:

- We introduce the first practical SHE schemes based on Ring-LWR: an LPR-type [19] scheme and a Regev-type [18] scheme, that are comparable with BFV in terms of parameters. Indeed, both our LPR-type and Regev-type schemes have ciphertexts of the same form, and we show that the homomorphic operations and noise analyses are entirely analogous to those in BFV. Additionally, we give a thorough theoretical and concrete security analysis of both schemes, and a proof of concept implementation of the LPR-type scheme.
- We resolve open problems from prior work [48], demonstrating the tractability of a BFV-like HE scheme from LWR assumptions, and avoiding the "tangled modulus" issue. [35]
- We show that ciphertext sizes (and thus bandwidth in applications) are improved by either our LPR-type scheme, Regev-type scheme, or both (depending on the choice of plaintext modulus), by providing a comparison between our schemes and the BFV scheme in terms of ciphertext size (see Section 6). Both our schemes also inherit the other benefits of using LWR, including amenability to hardware acceleration, and avoiding the need for Gaussian sampling that is expensive and vulnerable to side channel attacks.
- We demonstrate, perhaps surprisingly, that both of our schemes support RNS variants, which would make their practical performance comparable with existing implementations of BFV. Using an RNS variant incurs one bit of additional

noise compared to using the original schemes, exactly in analogue to the RNS-BFV scheme given in [52].

1.1 Overview of technical contributions

1. The LPR-type scheme.

We make use of four moduli $r > q > p > t$, where t is the plaintext modulus. The public key consists of one Ring-LWR$_{n,r,q}$ sample, $(a, b) = (a, \lfloor a \cdot s \rceil_{r,q}) \in R_r \times R_q$. To encrypt a message $m \in R_t$, we encode it in the high order bits and add it to a rounded randomised public key: $\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1) = (\lfloor a \cdot u \rceil_{r,q}, \lfloor b \cdot u \rceil_{q,p} + \Delta_p \cdot m) \in R_q \times R_p$, where $u \in R$ is a polynomial with coefficients from $\{-1, 0, 1\}$. To decrypt, one can check that $\frac{t}{p}(\mathtt{ct}_1 - \frac{p}{q} \cdot s \cdot \mathtt{ct}_0) = m + N + t \cdot G \in \mathbb{Q}[X]/(X^n + 1)$, with G an integer polynomial and $\|N\| < 1/2$ in infinity norm. To recover $m \in R_t$ we round off the noise N and interpret the result modulo t.

Suppose that we are multiplying $(\mathtt{ct}_0, \mathtt{ct}_1)$ with $(\mathtt{ct}'_0, \mathtt{ct}'_1)$. If we try to mimic BFV ciphertext multiplication, we have to compute a "tensor product" of the form

$$(c_2, c_1, c_0) = (\mathtt{ct}_0\mathtt{ct}'_0, \mathtt{ct}_0\mathtt{ct}'_1 + \mathtt{ct}'_0\mathtt{ct}_1, \mathtt{ct}_1\mathtt{ct}'_1),$$

and scale it appropriately. We want the above to satisfy the following equation

$$\frac{t}{p}\left[\left(\frac{p}{q}\right)^2 c_2 s^2 - \frac{p}{q} c_1 s + c_0\right] = m_{\text{mult}} + N_{\text{mult}} + tG_{\text{mult}}\,,$$

with a small noise polynomial N_{mult} with rational coefficients. In this context, because ciphertexts components correspond to different moduli p, q it is not clear in which ring we should compute (c_2, c_1, c_0) to have a guaranteed small noise N_{mult}. To this end, we introduce explicit moduli for the intermediate computation of (c_2, c_1, c_0) and we give a careful analysis of the noise. More concretely, we show that if the c_2 component is interpreted modulo q^2/p, the c_1 component is interpreted modulo q, and the c_0 component is interpreted modulo p, then there is no uncontrollable noise growth in multiplication. This approach thus addresses the "tangled modulus" [49] issue present in [35].

Another technical issue when adapting from Ring-LWE to Ring-LWR arises in the security proof. The IND-CPA security argument in the original LPR scheme [19] can be made through a series of hybrid games. First, the public-key is indistinguishable from uniformly random ring elements by the decision Ring-LWE assumption. Then, assuming the public-key is uniform, ciphertexts are indistinguishable from random ring elements, by again invoking the decisional Ring-LWE assumption. When trying to apply the same strategy for proving the IND-CPA security for our Ring-LWR based LPR-type scheme (Theorem 2), we can easily do the first transition. We can assume that the public key (a, b) is uniformly random in $R_r \times R_q$ by the Ring-LWR$_{n,r,q}$ assumption. To prove that ciphertexts are indistinguishable from random, in the second part of the argument, it is sufficient to argue that $(\lfloor a \cdot u \rceil_{r,q}, \lfloor b \cdot u \rceil_{q,p})$ is uniformly random in the eyes of an adversary that knows the public key. Informally, we would

like tuples $(a, b, \lfloor a \cdot u \rceil_{r,q}, \lfloor b \cdot u \rceil_{q,p})$, where $a \leftarrow R_q$, $b \leftarrow R_p$ and u is a uniform polynomial with coefficients in $\{-1, 0, 1\}$, to be computationally indistinguishable from the uniform distribution on $R_r \times R_q \times R_q \times R_p$. This is what we call the 3-moduli Ring-LWR$_{n,r,q,p}$ problem. In Theorem 1, we give a tight reduction from the Ring-LWR$_{n,r,q}$ problem to the 3-moduli Ring-LWR$_{n,r,q,p}$ problem under the constraints $q|r$ and $pr = q^2$. Hence we prove IND-CPA security for our scheme relying solely on the decisional Ring-LWR$_{n,r,q}$ assumption. Considering that powers of two moduli are preferred for rounding efficiently, these constraints for the security proof are not a concern.

2. The Regev-type scheme.

This can be seen as a generalisation of the Costache-Smart proposal [35], where the encryption randomness r_k can be chosen from any finite set of polynomials $X \subseteq R = \mathbb{Z}[x]/(x^n + 1)$, with coefficients within some predetermined bound, not necessarily restricted to $X = \{0, 1\}$. The scheme uses three moduli $q > p > t$, where again t is the plaintext modulus. The public-key consists of ℓ pairs of Ring-LWR$_{n,q,p}$ samples $\{(v_k, w_k = \lfloor v_k \cdot s \rceil_{q,p})\}_{k=1}^{\ell}$, constructed under the same secret s. To encrypt a message $m \in R_t$, we sample ℓ random ring elements $r_k \leftarrow X$, encode the message in the high order bits and compute: $\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1)$, $\mathtt{ct}_0 = \sum_{k=1}^{\ell} r_k \cdot v_k \in R_q$ and $\mathtt{ct}_1 = \sum_{k=1}^{\ell} r_k \cdot w_k + \Delta_p \cdot m \in R_p$. Decryption is the same as in the LPR-type scheme above.

We provide an IND-CPA security proof that is based solely on the decisional Ring-LWR$_{n,q,p}$ assumption for our Regev-type scheme (Corollary 2). We proceed as in any Regev-type scheme: first, we replace the public-key pairs with ℓ uniformly random elements from $R_q \times R_p$, by invoking the decisional Ring-LWR$_{n,q,p}$ assumption. Next, we use a Leftover Hash Lemma (LHL) [53] argument to prove that the ciphertext is uniformly random.

The reason that we use an enlarged randomness set X is that it allows us to apply a variant of LHL for much smaller values of $\ell \approx \log(pq)$, (see Corollary 2) compared to the case when $X = \{0, 1\}$, which requires $\ell \approx n \log(pq)$ (see Corollary 1). To this end we develop a suitable LHL approach in Theorem 3. The improved bounds on ℓ are a result of adapting [54, Lemma 4] to our two-ring setting and using [55, Corollary 1.2]. In particular, the enlarged set X allows us to have provable security based solely on Ring-LWR$_{n,q,p}$, for smaller values of ℓ, which translates to smaller public key size.

For the Regev-type scheme, the proof of Theorem 4 involves the two-ring Decision Knapsack Problem (2DKS) that we introduce in Definition 9. The 2DKS problem is inspired by the single-ring knapsack problem considered in [54], and in fact we can show these problems are equivalent (see Appendix E). In Theorem 3 we show that 2DKS is statistically indistinguishable from uniform under certain parameter constraints.

3. RNS variants of our schemes.

In practice, the most performant implementations of BFV use an RNS variant [52, 56, 57] that enables to avoid handling very large ciphertext polynomial coefficients (of size hundreds of bits). In RNS-BFV, the ciphertext modulus q is chosen to be a product of smaller coprime moduli $q = \prod_{i \in I} q_i$, where each q_i can for example

be chosen to be word-size to enable more efficient operations. In our case, we have two ciphertext moduli, and so we would accordingly wish to take $q = \prod_{i \in I} q_i$ and $p = \prod_j p_{j \in J}$ for pairwise coprime q_i and p_j. Importantly, and perhaps surprisingly, we show that it is not necessary for p and q themselves to be coprime. Indeed, we present RNS variants for both our LPR-type and Regev-type scheme in parameter settings that meet the requirements for our security proofs: namely, for $p = \prod_j p_j$ for pairwise coprime p_j, we can choose $q = 13p$, respectively $q = 16p$, for the Regev-type scheme (c.f. Corollary 2), respectively the LPR-type scheme (c.f. Theorem 1. For these parameters, we show that the techniques of [52] can either be directly applied, or slightly modified, to achieve RNS variants of our schemes. Our RNS variants have the same one bit of noise overhead compared to their non-RNS counterparts as the RNS-BFV of [52] does compared to the original BFV scheme, and should achieve the corresponding performance benefit.

4. Additional contributions.

In our schemes, we further improve upon [35] by adapting the scaling in decryption and multiplication from $1/\Delta_p$ to t/p. This removes unnecessary noise terms coming from rounding errors and leads us to define an invariant noise for our schemes, analogous to the definition proposed in [13, 58]. The invariant noise N for the ciphertext $(\mathtt{ct}_0, \mathtt{ct}_1)$ that encrypts the message m modulo t is the minimal polynomial such that

$$\frac{t}{p}\left(-\frac{p}{q}\mathtt{ct}_0 s + \mathtt{ct}_1\right) = m + N + tG,$$

for some integer polynomial G. Using this definition of noise, we show that our noise growth is completely analogous to the BFV formulas.

We not only look at provable security, but also consider the concrete security of the underlying assumptions under the best known attacks. Thus we can either choose parameters that instantiate the schemes in a provably secure parameter setting, or we can choose parameters according to concrete cryptanalysis for better performance. In particular, we consider 2DKS in a parameter range that is outside the constraints required for statistical hardness. Our study of concrete security enables us to suggest example parameter sets for our schemes that target 128-bit security.

As an additional contribution, we compare our schemes against the BFV scheme. We have implemented our LPR-type scheme, but our implementation is only a proof-of-concept, so we choose not to compare runtime performance. Instead, as in [58, 59], we chose to compare the schemes by their ciphertext size. This is because a large ciphertext size will incur the largest overhead, both in terms of memory and latency. Our comparison follows the methodology in [58, 59]. We pick a circuit of $\zeta = 8$ additions and L multiplications, and look at the smallest parameter set required for each of the schemes to correctly evaluate this circuit for a given plaintext modulus. We present our results for different choices of plaintext modulus t in Tables 6, 7, 8, 9, 10, 11, and 12. The results show that the LPR and Regev-type schemes marginally outperform the BFV scheme in terms of ciphertext size.

We believe that the new problems 2DKS and 3-moduli Ring-LWR that we introduce will be of independent interest. For example, our LPR-type scheme is reminiscent

of recent LWR-based schemes, including Saber, and our techniques may be applicable for these schemes. The original security proof of Saber [46, Theorem 3] relies on two separate Module-LWR assumptions, which could perhaps be simplified to only one, via a 3-moduli Module-LWR assumption (as for our Theorem 2). We briefly investigated this for Saber, and found that our approach would either imply worse parameters than Saber uses, or we would not be able to reduce the introduced 3-moduli assumption into a standard LWR assumption (as for our Theorem 1), so we did not pursue this further. Nevertheless, it would be interesting to investigate the applicability of 3-moduli LWR assumptions in other contexts.

1.2 Related work

Aside from the work of Costache and Smart [35], two other prior works [48, 49] have proposed homomorphic encryption schemes based on LWR assumptions, neither of which target similarity to BFV or BGV. An LWR-based scheme in the style of GSW was given in [49]. In [48], Luo *et al.* proposed a Ring-LWR-based scheme, which is based on Dual Regev encryption [60]. In Table 1 we present a comparison of our schemes with the BFV scheme [6] and the LWWC scheme [48]. It can be seen that ciphertexts in our scheme are slightly smaller than in BFV, while those in LWWC are asymptotically much larger. The relinearization keys in our LPR-type scheme are also smaller than in BFV and LWWC. Bootstrapping techniques for BGV/BFV have been presented in [61–65]. We defer the exploration of bootstrapping of our schemes to future work.

2 Preliminaries

2.1 Notation

For a finite set X, we write $x \leftarrow X$ to mean sampling x uniformly at random over X. For a vector $\mathbf{x} \in \mathbb{C}^n$, its infinity norm $\|\cdot\|$ is defined as $\|\mathbf{x}\| = \max_i |\mathbf{x}_i|$.

2.2 Parameters

Our Ring-LWR-based schemes are parameterised by n, t, p, q, and (optionally) ω. The dimension n is a chosen to be power of two. The dimension n, the plaintext modulus t and the ciphertext moduli $q > p$ parameterise the underlying plaintext and ciphertext rings. The plaintext space is $R_t = \mathbb{Z}_t[x]/(x^n + 1)$. The ciphertext space is given by $R_q \times R_p$ where $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ and $R_p = \mathbb{Z}_p[x]/(x^n + 1)$. The schemes support the variant of relinearization that decomposes a ciphertext component with respect to a base ω.

We refer to our first scheme as LPR-type as it follows the public-key encryption (PKE) scheme of [19]. The LPR-type scheme requires an additional modulus $r > q$ to define the public key. We refer to our second scheme as Regev-type as it follows the PKE scheme of [18]. The public key in the Regev-type scheme is composed of ℓ encryptions of zero. The secret key in both schemes is typically taken to be small, e.g. with uniform ternary coefficients. Encryption in the Regev-type scheme is done by taking a random subset sum of the public key, where the randomness is chosen from

a finite subset $X \subseteq R = \mathbb{Z}[x]/(x^n+1)$. For a positive integer B, we will be interested in the set $X = S_{B/2}$, where $S_{B/2} := \{-B/2, \ldots, B/2\}$ is the set of integer scalars bounded by $B/2$; and the set $X = P_{B/2}$, denoting the set of polynomials modulo x^n+1 with integer coefficients bounded by $B/2$.

Both schemes are BFV-like [6] and the parameter $\Delta_p := \lfloor \frac{p}{t} \rfloor = \frac{p}{t} - \epsilon_p$, such that $0 \leq \epsilon_p < 1$, is used in encryption to put the message in the high order bits. We define the rounding operation that maps from R_q to R_p as $\lfloor x \rceil_{q,p} := \lfloor \frac{p}{q} \cdot x \rceil = \frac{p}{q}x + \epsilon$ where the coefficients of $\epsilon \in R$ are in $\left(-\frac{1}{2}, \frac{1}{2}\right]$.

2.3 Problem definitions

The Learning with Errors (LWE) problem was introduced by Regev [18].

Definition 1 (LWE distribution)**.** *Let n and q be positive integers, χ be a probability distribution on $\mathbb{Z}$, and $\boldsymbol{s}$ be a secret vector in $\mathbb{Z}_q^n$. The LWE distribution with parameters n, q, χ (denoted by $LWE_{n,\chi,q}$) is the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing $\boldsymbol{a} \in \mathbb{Z}_q^n$ uniformly at random, sampling e from χ and considering it modulo q, and returning $(\boldsymbol{a}, b) = (\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.*

Definition 2 (LWE problem)**.** *Decision $LWE_{n,\chi,q}$ is the problem of deciding whether pairs $(\boldsymbol{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ are sampled from the $LWE_{n,\chi,q}$ distribution for a fixed $\boldsymbol{s}$ or the uniform distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$. Search $LWE_{n,\chi,q}$ is the problem of recovering $\boldsymbol{s}$ from samples $(\boldsymbol{a}, b)$ sampled from the $LWE_{n,\chi,q}$ distribution.*

The Learning with Rounding (LWR) problem was introduced by Banerjee *et al.* [24] as a derandomised version of the LWE problem. Ring variants of these problems, known as Ring-LWE [19, 20] and Ring-LWR [24], can also be defined. We give a definition for Ring-LWR in the power-of-two cyclotomic setting.

Definition 3 (Ring-LWR distribution)**.** *Let n be a power of two and $q \geq p$ be integers. For $s \in R_q$, define the Ring-LWR distribution with parameters n, q, p (denoted by Ring-$LWR_{n,q,p}$) as the distribution over $R_q \times R_p$ obtained by choosing $a \in R_q$ uniformly at random and returning $(a, b = \lfloor a \cdot s \rceil_{q,p})$.*

Definition 4 (Ring-LWR problem)**.** *Decision Ring-$LWR_{n,q,p}$ is the problem of deciding whether pairs $(a, b) \in R_q \times R_p$ are sampled according to the Ring-$LWR_{n,q,p}$ distribution for a fixed $s \in R_q$ or the uniform distribution on $R_q \times R_p$. Search Ring-$LWR_{n,q,p}$ is the problem of recovering s from samples (a, b) sampled from the Ring-$LWR_{n,q,p}$ distribution.*

2.4 Leftover Hash Lemma

We will use the Leftover Hash Lemma [53].

Definition 5 (**Family of universal hash functions**)**.** *A finite family $\mathcal{H}$ of functions $h : X \to Y$ is universal if $Pr_{h \leftarrow \mathcal{H}}[h(x_1) = h(x_2)] = \frac{1}{|Y|}$, for any $x_1 \neq x_2 \in X$.*

Lemma 1 (**Leftover Hash Lemma [53, Lem 2.1]**)**.** *Let X, Y denote finite sets. Let H be a universal family of functions $h : X \to Y$. Then for $h \leftarrow H$, $x \leftarrow X$, and $y \leftarrow Y$, we have $\Delta\Big((h, h(x)), (h, y)\Big) \leq \frac{1}{2}\sqrt{|Y|/|X|}$.*

2.5 Canonical embedding norm

Following previous works [8, 9, 58, 59, 66], we will present heuristic bounds for the noise growth behaviour of our schemes with respect to the canonical embedding norm $\|\cdot\|^{\text{can}}$, using the methodology of Iliashenko [66]. Throughout this work, for a polynomial $a \in R$, the notation $\|a\|$ refers to the infinity norm of (the coefficient vector of) a, while $\|a\|^{\text{can}}$ refers to the canonical embedding norm. The canonical embedding norm of an element a is defined to be the infinity norm of the canonical embedding[1] $\sigma(a)$ of a, so $\|a\|^{\text{can}} = \|\sigma(a)\|$.

We will use the following properties of the canonical embedding norm. For any polynomial $a \in R$ we have $\|a\| \leq c_{2n} \|a\|^{\text{can}}$, where c_{2n} is a constant known as the ring expansion factor [68]. We have $c_{2n} = 1$ when the dimension n is a power of two [68]. In this case, it suffices for correctness to ensure that $\|v\|^{\text{can}}$ is less than the maximal value of $\|v\|$ such that decryption succeeds. For any polynomials $a, b \in R$ we have $\|ab\|^{\text{can}} \leq \|a\|^{\text{can}} \|b\|^{\text{can}}$.

Let $R = \mathbb{Z}[x]/(x^n + 1)$ and let ζ be a primitive $2n^{\text{th}}$ root of unity (by definition of the canonical embedding norm, it does not matter which one). Let $a \in R$ be a polynomial for which the variance of each coefficient is V_a. Then, the variance of the random variable $a(\zeta)$ is nV_a [9, 59, 66]. We use the fact that $\text{erfc}(6) \approx 2^{-55}$ to obtain the following bound $\|a\|^{\text{can}} \leq 6\sqrt{n}\sqrt{V_a}$.

We also use the following facts. Let V_a and V_b be the variances of the coefficients of two polynomials $a \in R$ and $b \in R$ chosen from zero-mean distributions, and let γ be a constant. The variance of the coefficients of the polynomial $a + b$ is $V_{a+b} = V_a + V_b$. The variance of the coefficients of the polynomial γa is $V_{\gamma a} = \gamma^2 V_a$ for a fixed scalar γ. In particular, $V_{\epsilon_p m} = \epsilon_p^2 V_m$. The variance of the coefficients of the polynomial ab is $V_{ab} = nV_aV_b$ (see [66] for a proof), assuming that the polynomials a and b are independent, and that their coefficients are independently distributed.

We also make use of the following facts. The coefficients of a polynomial f that are distributed uniformly in $\{-\frac{k}{2}, \ldots, \frac{k}{2}\}$ have variance $V_f \approx \frac{k^2}{12}$. The coefficients of a polynomial ϵ that are distributed uniformly in $(-\frac{1}{2}, \frac{1}{2}]$ have variance $V_f = \frac{1}{12}$. The coefficients of a polynomial s that are drawn from the uniform distribution on the ternary set $\{-1, 0, 1\}$ have variance $V_s = \frac{2}{3}$.

3 An LPR-type SHE scheme based on Ring-LWR

We first define an LPR-type [19] SHE scheme based on Ring-LWR, as in the BFV scheme itself [6]. The security proof of our scheme assumes the hardness of the Ring-LWR problem, as well as the hardness of a variant of the Ring-LWR problem that we call 3-moduli Ring-LWR. A 3-moduli Ring-LWR looks like two Ring-LWR samples that share the same secret, but are computed in different rings that make use of three different moduli. We show in Section 3.5 that the 3-moduli Ring-LWR problem is at least as hard as the Ring-LWR problem.

In our presentation of the scheme, we omit levelled notation for clarity of exposition. The scheme still permits a modulus switching operation for efficiency, but we delay the presentation and analysis to Section D.

[1] For a definition of the canonical embedding and other algebraic background, see [67].

Our LPR-type SHE scheme is similar to the LWR-based PKE schemes submitted to the NIST post-quantum standardisation process (see e.g. [44–46]). The scheme is parameterised by four moduli $r > q > p > t$, where r and q will be the moduli in the public key Ring-LWR instance, p and q will be the ciphertext moduli, and t will be the plaintext modulus. In particular, the plaintext space is R_t, where we take the coefficient representatives as being in the set $\{-\frac{t}{2}, \ldots, \frac{t}{2}\}$. We recall the notation $\lfloor x \rceil_{q,p} := \left\lfloor \frac{p}{q} x \right\rceil$, and note that we can rewrite this as $\lfloor x \rceil_{q,p} = \frac{p}{q} x + \epsilon$. In this section, all equations are implicitly modulo the polynomial $x^n + 1$.

3.1 Public-key encryption scheme

Key Generation.

The secret key $\mathtt{sk} := s \leftarrow R_r$ is chosen to be small, for example uniform ternary. The public key $\mathtt{pk} := (\mathtt{pk}_0, \mathtt{pk}_1) \in R_r \times R_q$ is a Ring-LWR sample formed as follows. Sample $a \leftarrow R_r$ uniformly at random, and set $(\mathtt{pk}_0, \mathtt{pk}_1) = (a, \lfloor a \cdot s \rceil_{r,q})$.

Encryption.

For $m \in R_t$ and $\mathtt{pk} = (\mathtt{pk}_0, \mathtt{pk}_1) \in R_r \times R_q$. Let $\Delta := \lfloor \frac{p}{t} \rfloor = \frac{p}{t} - \epsilon_p$ where $0 \leq \epsilon < 1$. The ciphertext $\mathtt{ct} := (\mathtt{ct}_0, \mathtt{ct}_1) \in R_q \times R_p$ is obtained as follows. Sample $u \leftarrow R_r$ from the secret distribution. Set $\mathtt{ct}_0 = \lfloor \mathtt{pk}_0 u \rceil_{r,q}$ and $\mathtt{ct}_1 = \lfloor \mathtt{pk}_1 u \rceil_{q,p} + \Delta m$.

Decryption.

For $\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1) \in R_q \times R_p$ and secret key s, output

$$m' = \left\lfloor \frac{t}{p} \left(-\frac{p}{q} \mathtt{ct}_0 s + \mathtt{ct}_1 \right) \right\rceil \pmod{t}.$$

3.2 Correctness

Consider the decryption of a fresh ciphertext:

$$\begin{aligned}
m' &= \left\lfloor \frac{t}{p} \left(-\frac{p}{q} \mathtt{ct}_0 s + \mathtt{ct}_1 \right) \right\rceil \pmod{t} \\
&= \left\lfloor -\frac{t}{q} s \lfloor \mathtt{pk}_0 u \rceil_{r,q} + \frac{t}{p} \lfloor \mathtt{pk}_1 u \rceil_{q,p} + \frac{t}{p} \Delta m \right\rceil \pmod{t} \\
&= \left\lfloor -\frac{t}{r} (\mathtt{pk}_0) u s - \frac{t}{q} \epsilon_0 s + \frac{t}{q} (\mathtt{pk}_1) u + \frac{t}{p} \epsilon_1 + \frac{t}{p} \Delta m \right\rceil \pmod{t} \\
&= \left\lfloor -\frac{t}{r} a u s - \frac{t}{q} \epsilon_0 s + \frac{t}{q} \left(\frac{q}{r} a s + \epsilon \right) u + \frac{t}{p} \epsilon_1 + \frac{t}{p} \left(\frac{p}{t} - \epsilon_p \right) m \right\rceil \pmod{t} \\
&= m + \left\lfloor -\frac{t}{q} \epsilon_0 s + \frac{t}{q} \epsilon u + \frac{t}{p} \epsilon_1 - \frac{t}{p} \epsilon_p m \right\rceil \pmod{t}.
\end{aligned}$$

where ϵ_0 and ϵ_1 are the errors arising from the rounding. Thus decryption outputs $m' = m \pmod{t}$ if $\lfloor -\frac{t}{q} \epsilon_0 s + \frac{t}{q} \epsilon u + \frac{t}{p} \epsilon_1 - \frac{t}{p} \epsilon_p m \rceil = 0$.

Definition of noise.

The structure of decryption above motivates the definition of noise N in a ciphertext $\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1)$ as the polynomial of minimal infinity norm among all the polynomials for which there exists an integer polynomial G such that

$$\frac{t}{p}\left(-\frac{p}{q}\cdot \mathtt{ct}_0 \cdot s + \mathtt{ct}_1\right) = m + N + tG\,.$$

For correctness, we always require $\|N\| < 1/2$. This definition is analogous to the invariant noise definition for BFV as in [13, 58].

3.3 Homomorphic operations

Noise in a fresh ciphertext.

The argument of Section 3.2 shows that the noise N_{fresh} in a fresh ciphertext is given by

$$N_{\text{fresh}} = -\frac{t}{q}\epsilon_0 s + \frac{t}{q}\epsilon u + \frac{t}{p}\epsilon_1 - \frac{t}{p}\epsilon_p m\,.$$

In the noise expression, s and u are uniform with coefficients in $\{-1, 0, 1\}$ and we model ϵ, ϵ_0, and ϵ_1 with coefficients as uniform over $(-\frac{1}{2}, \frac{1}{2}]$. We also model the plaintext m as having coefficients uniform over $\{-\frac{t}{2}, \ldots, \frac{t}{2}\}$. Let V_{fresh} denote the coefficient variance of N_{fresh}. Using Section 2.5, we can bound $\|N_{\text{fresh}}\|^{\text{can}} \le 6\sqrt{nV_{\text{fresh}}}$, so that

$$\|N_{\text{fresh}}\|^{\text{can}} \le t \cdot \sqrt{\frac{4n^2}{q^2} + \frac{3n}{p^2}\cdot\left(1 + t^2\epsilon_p^2\right)}.$$

Addition.

Let $\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1)$ and $\mathtt{ct}' = (\mathtt{ct}'_0, \mathtt{ct}'_1)$ be ciphertexts encrypting m and m' with noises N and N' respectively. Define the output of homomorphic addition as $\mathtt{ct}_{\text{add}} := (\mathtt{ct}_{0,\text{add}}, \mathtt{ct}_{1,\text{add}})$ where $\mathtt{ct}_{0,\text{add}} = \mathtt{ct}_0 + \mathtt{ct}'_0 \pmod q$ and $\mathtt{ct}_{1,\text{add}} = \mathtt{ct}_1 + \mathtt{ct}'_1 \pmod p$. Then $\mathtt{ct}_{\text{add}}$ encrypts $m + m' \pmod t$ with noise $N_{\text{add}} := N + N'$. To see this, we note that for some integer polynomials A and B, and for the integer polynomial $C := B - As$, we have

$$\begin{aligned}\frac{t}{p}\left(-\frac{p}{q}\mathtt{ct}_{0,\text{add}}s + \mathtt{ct}_{1,\text{add}}\right) &= \frac{t}{p}\left(-\frac{p}{q}\mathtt{ct}_0 s + \mathtt{ct}_1\right) + \frac{t}{p}\left(-\frac{p}{q}\mathtt{ct}'_0 s + \mathtt{ct}'_1\right) + tC \\ &= m + m' + (N + N') + t(G + G' + C)\,.\end{aligned}$$

Multiplication.

Let $\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1)$ and $\mathtt{ct}' = (\mathtt{ct}'_0, \mathtt{ct}'_1)$ be two ciphertexts which encrypt plaintexts m and m' with noise N and N' respectively. We define the output of the multiplication of $\mathtt{ct}$ and $\mathtt{ct}'$ as:

$$(c_2, c_1, c_0) = \left(\left[\left\lfloor\frac{t}{p}\mathtt{ct}_0\mathtt{ct}'_0\right\rceil\right]_{q^2/p}, \left[\left\lfloor\frac{t}{p}(\mathtt{ct}_0\mathtt{ct}'_1 + \mathtt{ct}'_0\mathtt{ct}_1)\right\rceil\right]_q, \left[\left\lfloor\frac{t}{p}\mathtt{ct}_1\mathtt{ct}'_1\right\rceil\right]_p\right).$$

In this expression, we give an explicit modular reduction for each component c_i. In particular, this enables us to tightly bound the size of each component. While a definition is possible without these explicit modular reductions, it may lead to larger intermediate terms, which would lead to a larger evaluation key.

Abusing notation, we can treat this intermediate (c_2, c_1, c_0) as a ciphertext that encrypts $m_{\text{mult}} := m \cdot m' \pmod t$. In particular, we can define a decryption operation for (c_2, c_1, c_0) as follows:

$$\widetilde{m} = \left\lfloor \frac{t}{p} \left[\left(\frac{p}{q}\right)^2 c_2 s^2 - \frac{p}{q} c_1 s + c_0 \right] \right\rceil ,$$

where s^2 is the square of the secret key. We can hence define the noise N_{mult} in (c_2, c_1, c_0) as the polynomial of minimal infinity norm for which there exists some integer polynomial G_{mult} such that,

$$\frac{t}{p} \left[\left(\frac{p}{q}\right)^2 c_2 s^2 - \frac{p}{q} c_1 s + c_0 \right] = m_{\text{mult}} + N_{\text{mult}} + tG_{\text{mult}} .$$

It can be seen from the form of the decryption expression that the specified moduli for each of the components c_2, c_1, c_0 is the natural choice, in the sense that if we make the modular reduction explicit in each component when analysing the noise, the arising terms will result as summands in the expression for G_{mult} and disappear modulo t. In Section B we will show that

$$N_{\text{mult}} = NN' + (m' + tG')N + (m + tG)N' + \frac{tp}{q^2}\epsilon_2 s^2 - \frac{t}{q}\epsilon_1 s + \frac{t}{p}\epsilon_0$$

and that this can be bounded as

$$\begin{aligned} \|N_{\text{mult}}\|^{\text{can}} \leq \|N\|^{\text{can}} \cdot \|N'\|^{\text{can}} + t \cdot \sqrt{2n^2 + 3n} \cdot \left(\|N\|^{\text{can}} + \|N'\|^{\text{can}}\right) \\ + \frac{tp}{q^2} \cdot 4\sqrt{3} \cdot n^{3/2} + t \cdot \sqrt{\frac{2n^2}{q^2} + \frac{3n}{p^2}} . \end{aligned}$$

Relinearization (decomposition).

The goal of relinearization is to turn a three-element intermediate (c_2, c_1, c_0) encrypting a message m that is output from multiplication into a two-element ciphertext $\mathtt{ct}_{\text{relin}} = (\mathtt{ct}_{0,\text{relin}}, \mathtt{ct}_{1,\text{relin}})$ encrypting the same message. Let ω be a base, and write c_2 in base ω as $c_2 = \sum_{j=0}^{k} c_2^{(j)} \cdot \omega^j$, for some integer k and for some integer polynomials $c_2^{(j)}$ with coefficients in $\{-\frac{\omega}{2}, \ldots, \frac{\omega}{2}\}$. Since c_2 is taken modulo q^2/p, we have that $k = \left\lceil \log_\omega \frac{q^2}{2p} \right\rceil$. The evaluation key $\mathtt{evk}$ at level (q, p) is given by $k+1$ pairs (a_j, b_j) for

$0 \leq j \leq k$, where a_j is chosen uniformly at random from R_q and $b_j \in R_p$ is given by

$$b_j = \left\lceil a_j s \right\rfloor_{q,p} + \left\lceil \frac{p^2}{q^2} \omega^j s^2 \right\rfloor \pmod{p}.$$

We then define $\mathtt{ct}_{\text{relin}} = (\mathtt{ct}_{0,\text{relin}}, \mathtt{ct}_{1,\text{relin}})$ as $\mathtt{ct}_{0,\text{relin}} = c_1 + \sum_{j=0}^{k} c_2^{(j)} a_j \pmod{q}$ and $\mathtt{ct}_{1,\text{relin}} = c_0 + \sum_{j=0}^{k} c_2^{(j)} b_j \pmod{p}$.

Let relinearization be applied to an intermediate (c_2, c_1, c_0) encrypting m with noise N. Then we show in Section C that $\mathtt{ct}_{\text{relin}}$ encrypts m with noise $N_{\text{relin}} = N + \frac{t}{p} \sum_{j=0}^{k} c_2^{(j)} (\epsilon_1 + \epsilon_2)$; where ϵ_1 and ϵ_2 are rounding errors arising from the two instances of rounding in the evaluation key. Moreover, the noise N_{relin} can be bounded as

$$\|N_{\text{relin}}\|^{\text{can}} \leq \|N\|^{\text{can}} + \frac{t}{p} \cdot 6n \sqrt{(k+1) \cdot \frac{1}{6} \cdot \frac{\omega^2}{12}}.$$

Modulus switching.

This technique is typically used in homomorphic encryption to optimise the execution of homomorphic operations and to reduce parameter sizes [9]. In Section D we present details on modulus switching for our scheme.

3.4 The 3-moduli Ring Learning with Rounding Problem

The security proof of the underlying PKE scheme of Section 3.1 relies on the hardness of Ring-LWR. However, the proof invokes a second problem, that we call the 3-moduli Ring-LWR problem. We define it in this section. A sample from the 3-moduli Ring-LWR distribution looks similar to two Ring-LWR samples under the same small secret, but computed in different rings that share a common modulus. Theorem 1 establishes the hardness of 3-moduli Ring-LWR by giving an efficient reduction from Ring-LWR when certain constraints on the moduli are satisfied.

Let p, q, r be integers such that $p < q < r$. Recall that $R_p = \mathbb{Z}_p[x]/(x^n + 1)$, $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ and $R_r = \mathbb{Z}_r[x]/(x^n + 1)$.

Definition 6 (3-moduli Ring-LWR distribution)**.** *The 3-moduli Ring Learning with Rounding Problem distribution with parameters n, p, q, r (denoted by 3-moduli Ring-LWR$_{n,r,q,p}$) is the distribution over $R_r \times R_q \times R_q \times R_p$ which consists of samples $(a_1, \lfloor a_1 \cdot s \rceil_{r,q}, a_2, \lfloor a_2 \cdot s \rceil_{q,p})$ where a_1 is sampled uniformly from R_r, a_2 is sampled uniformly from R_q, and the secret $s \in R$ has coefficients sampled uniformly from $[-\gamma, \gamma]$, for some $\gamma < p$.*

Definition 7 (3-moduli Ring-LWR problem)**.** *The Decision 3-moduli Ring Learning with Rounding Problem with parameters n, p, q, r (denoted by Decision 3-moduli Ring-LWR$_{n,r,p,q}$) is defined as follows: given a sample (a_1, b_1, a_2, b_2) from $R_r \times R_q \times R_q \times R_p$, decide whether it comes from the 3-moduli Ring-LWR$_{n,r,q,p}$ distribution or it is uniformly random in $R_r \times R_q \times R_q \times R_p$.*

The proof of the hardness of 3-moduli Ring-LWR relies on the following lemma.

Lemma 2. *Let α be positive integer. Then the map $\pi : R_{\alpha q} \times R_{\alpha p} \to R_q \times R_p$ given by $(x, y) \mapsto (x \bmod q, y \bmod p)$ maps Ring-LWR$_{n,\alpha q,\alpha p}$ samples to Ring-LWR$_{n,q,p}$ samples and the uniform distribution to the uniform distribution.*

Proof. A proof is given in Section A. □

Theorem 1. *Let p, q, r be integers such that $q|r$ and $pr = q^2$. If there is an efficient algorithm for the 3-moduli Ring-LWR$_{n,r,q,p}$ problem that distinguishes between the two distributions with non-negligible probability ν, then there is an efficient solver for the Ring-LWR$_{n,r,q}$ problem that distinguishes with the same probability ν.*

Proof. It is enough to give an efficient transformation that takes two Ring-LWR$_{n,r,q}$ samples $(a_1, b_1, a_2, b_2) \in R_r \times R_q \times R_r \times R_q$ and outputs a 3-moduli Ring-LWR$_{n,r,q,p}$ sample of the form $(\bar{a}_1, \bar{b}_1, \bar{a}_2, \bar{b}_2) \in R_r \times R_q \times R_q \times R_p$. We must also ask that this transformation maps the uniform distribution to the uniform distribution. The transformation is given by the following map:

$$(\bar{a}_1, \bar{b}_1, \bar{a}_2, \bar{b}_2) := (a_1, b_1, a_2 \bmod q, b_2 \bmod p) \in R_r \times R_q \times R_q \times R_p$$

To finish the proof, we apply Lemma 2. We interpret $r = q \cdot \frac{r}{q}$ and $q = p \cdot \frac{r}{q}$ in the transformation $R_r \times R_q \to R_q \times R_p$ given by $(x, y) \mapsto (x \bmod q, y \bmod p)$ that is applied on the last two components of the two Ring-LWR samples. □

3.5 Security proof

In this subsection, we prove in Theorem 2 the IND-CPA security of the PKE scheme underlying our LPR-type scheme via a series of hybrids, assuming the hardness of Decision Ring-LWR and using Theorem 1.

Theorem 2. *If p, q, r are integers such that $q|r$ and $pr = q^2$, then the LPR-like Ring-LWR based public key encryption scheme described in Section 3.1 is IND-CPA secure assuming the hardness of Decision Ring-LWR$_{n,r,q}$.*

Proof. The proof is via the following series of hybrids:

- **Hybrid 1**: This is the real IND-CPA game where the ciphertext `ct` encrypting a message m is the output of the encryption algorithm run on a public key that is output from an honest execution of the key generation algorithm.
- **Hybrid 2**: This is the same as Hybrid 1, except that the public key `pk` is chosen uniformly at random from the public key space $R_r \times R_q$ rather than as an honest output of the key generation algorithm. The ciphertext `ct` is honestly generated by encrypting m under this uniform public key.
- **Hybrid 3**: This is the same as Hybrid 2 except that the ciphertext `ct` is chosen uniformly at random from the ciphertext space $R_q \times R_p$ rather than as an honest output of the encryption algorithm.

We first observe that the advantage of the adversary $\mathcal{A}$ in Hybrid 3 is 0 since all the information that the adversary has is uniformly random and independent of the message m. We proceed by arguing that the distinguishing advantages of an adversary between Hybrid 2 and Hybrid 1 and between Hybrid 3 and Hybrid 2 are negligible.

We can then conclude that the adversary's advantage in Hybrid 1, the real IND-CPA game, is negligible.

We now show that the distinguishing advantage between Hybrid 2 and Hybrid 1 is negligible assuming that Decision Ring-LWR$_{n,r,q}$ is hard. Consider an algorithm $\mathcal{A}$ that can distinguish between the two hybrids. We can transform the algorithm $\mathcal{A}$ to an algorithm $\mathcal{D}$ that distinguishes Ring-LWR$_{n,r,q}$ samples from uniformly random ones. The algorithm $\mathcal{D}$ forms a public key $\mathtt{pk}$ with its input and encrypts a message m under this public key to obtain a ciphertext $\mathtt{ct}$. The algorithm then invokes $\mathcal{A}$ on $(\mathtt{pk}, \mathtt{ct})$. The algorithm $\mathcal{D}$ then forwards the output of $\mathcal{A}$. If the input samples are Ring-LWR$_{n,r,q}$ samples, then $\mathcal{D}$ perfectly simulates Hybrid 1. Similarly, if the input samples are uniform, then $\mathcal{D}$ perfectly simulates Hybrid 2. The advantage of the algorithm $\mathcal{A}$ is less than the advantage of the algorithm $\mathcal{D}$. Hence, the advantage of $\mathcal{A}$ is negligible assuming that Decision Ring-LWR$_{n,r,q}$ is hard.

It remains to show that the distinguishing advantage between Hybrid 3 and Hybrid 2 is negligible. Consider an algorithm $\mathcal{A}$ that can distinguish between the two hybrids. We can transform the algorithm $\mathcal{A}$ to an algorithm $\mathcal{D}$ that distinguishes 3-moduli Ring-LWR$_{n,r,q,p}$ samples from uniformly random samples in $R_r \times R_q \times R_q \times R_p$. The algorithm $\mathcal{D}$ receives an input (a_1, b_1, a_2, b_2). It invokes $\mathcal{A}$ with $\mathtt{pk} = (a_1, a_2)$ and $\mathtt{ct} = (b_1, b_2 + \Delta m)$. The algorithm $\mathcal{D}$ then forwards the output of $\mathcal{A}$. If the input samples are 3-moduli Ring-LWR$_{n,r,q,p}$ samples, then $\mathcal{D}$ perfectly simulates Hybrid 2. Similarly, if the input samples are uniform, then $\mathcal{D}$ perfectly simulates Hybrid 3. The advantage of the algorithm $\mathcal{A}$ is less than the advantage of the algorithm $\mathcal{D}$. Moreover, the advantage of $\mathcal{D}$ is negligible, by the choice of the moduli and assuming that Decision Ring-LWR$_{n,r,q}$ is hard, by Theorem 1. Hence, the advantage of $\mathcal{A}$ is negligible assuming that Decision Ring-LWR$_{n,r,q}$ is hard. □

4 A Regev-type SHE scheme based on Ring-LWR

In this section, we introduce a Regev-type [18] SHE scheme whose security is based on the hardness of the Ring-LWR problem. Our starting point is a proposal of Costache and Smart [35]. We improve upon their construction in several important aspects. Firstly, we make explicit the modular reduction in each ciphertext component, including the output from homomorphic multiplication. Moreover, we alter the scaling factor to t/p, rather than $1/\Delta_p$. Our choice of scaling makes for a cleaner handling of the implicit modular reduction. These adaptations enable us to resolve the "tangled modulus" problem of [35] to obtain a scheme that can be instantiated. In fact, we show that the noise growth in multiplication in our scheme is completely analogous to the noise growth in BFV multiplication.

4.1 Public-key encryption scheme

All equations written in this section are implicitly taken modulo the polynomial x^n+1. Let X be a finite subset of $R = \mathbb{Z}[x]/(x^n + 1)$.

Key Generation.

The secret key $\mathtt{sk} := s \in R_q$ is chosen to have small coefficients, for example uniform ternary. The public key $\mathtt{pk} := \{(v_1, w_1), \ldots (v_\ell, w_\ell)\}$ consists of ℓ encryptions of zero, where each pair (v_k, w_k) is formed as follows: $v_k \leftarrow R_q$ and $w_k = \lfloor v_k \cdot s \rceil_{q,p}$. In particular, this means that for any $k \in \{1, \ldots, l\}$, $w_k = \frac{p}{q} \cdot v_k \cdot s + e_k + pA_k$ for some integer polynomials A_k and some polynomials e_k with coefficients in $\left(-\frac{1}{2}, \frac{1}{2}\right]$.

Encryption.

Let $\Delta_p := \lfloor \frac{p}{t} \rfloor = \frac{p}{t} - \epsilon_p$ where $0 \leq \epsilon_p < 1$. To encrypt a message $m \in R_t$, first choose r_k uniformly from the set X for any $k \in \{1, \ldots, l\}$. Then output the ciphertext $\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1)$, where $\mathtt{ct}_0 = \sum_{k=1}^{\ell} r_k v_k \pmod q$ and $\mathtt{ct}_1 = \Delta_p m + \sum_{k=1}^{\ell} r_k w_k \pmod p$. This means that we have $\mathtt{ct}_0 = \sum_{k=1}^{\ell} r_k v_k + qB$ and $\mathtt{ct}_1 = \Delta_p m + \sum_{k=1}^{\ell} r_k w_k + pC$ for some integer polynomials B and C.

Decryption.

Decryption is the same as for the LPR-type scheme. Given a ciphertext $\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1)$, output

$$m' = \left\lceil \frac{t}{p} \left(-\frac{p}{q} \cdot \mathtt{ct}_0 \cdot s + \mathtt{ct}_1 \right) \right\rfloor \pmod t .$$

4.2 Correctness

Given a ciphertext $\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1)$ corresponding to a message m, in decryption we compute

$$\begin{aligned} m' &= \left\lceil \frac{t}{p} \left(-\frac{p}{q} \cdot \mathtt{ct}_0 \cdot s + \mathtt{ct}_1 \right) \right\rfloor \\ &= \left\lceil \frac{t}{p} \left(-\frac{p}{q} \left(s \sum_{k=1}^{\ell} r_k v_k + qs \cdot B \right) + \Delta_p m + pC + \sum_{k=1}^{\ell} r_k w_k \right) \right\rfloor \\ &= \left\lceil \frac{t}{p} \left(\Delta_p m + \sum_{k=1}^{\ell} r_k e_k \right) + t \left(-s \cdot B + C + \sum_{k=1}^{\ell} r_k A_k \right) \right\rfloor \\ &= m + \left\lceil \frac{t}{p} \left(\epsilon_p m + \sum_{k=1}^{\ell} r_k e_k \right) \right\rfloor + t \left(-s \cdot B + C + \sum_{k=1}^{\ell} r_k A_k \right) . \end{aligned}$$

The output $m' = m \pmod t$ if and only if $\left\lceil \frac{t}{p} \left(-\epsilon_p m + \sum_{k=1}^{\ell} r_k e_k \right) \right\rfloor = 0$.

Noise.

As the structure of decryption is the same as for the LPR-type scheme of Section 3, we use the same definition of noise. That is, the noise N in a Regev-type ciphertext

$\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1)$ is defined as the polynomial of minimal infinity norm among all the polynomials for which there exists an integer polynomial G such that

$$\frac{t}{p}\left(-\frac{p}{q}\cdot \mathtt{ct}_0 \cdot s + \mathtt{ct}_1\right) = m + N + tG\,.$$

4.3 Homomorphic operations

The scheme depends on the choice of a finite set X. We analyse the size of the noise in a fresh ciphertext for the particular choices $X = S_{B/2}$ and $X = P_{B/2}$, as defined in Section 2.2.

Noise in a fresh ciphertext for $X = S_{B/2}$.
As shown above, the noise in a fresh ciphertext satisfies $N_{\text{fresh}} = \frac{t}{p}\left(-\epsilon_p m + \sum_{k=1}^{\ell} r_k e_k\right)$, where the r_k's are uniform scalars in $\{-B/2, \dots, B/2\}$ and the coefficients of the e_k's are uniform in $\left(-\frac{1}{2}, \frac{1}{2}\right]$. With high probability, using Section 2.5, we can write the bound $\|N_{\text{fresh}}\|^{\text{can}} \leq 6\sqrt{nV_{\text{fresh}}}$, where

$$V_{\text{fresh}} = \frac{t^2}{p^2}\left(\epsilon_p^2 V_m + \ell V_r V_{e_k}\right) = \frac{t^2}{p^2}\left(\frac{\epsilon_p^2 t^2}{12} + \ell \cdot \frac{B^2}{12}\cdot\frac{1}{12}\right).$$

Hence we have the following bound on the noise in a fresh ciphertext

$$\|N_{\text{fresh}}\|^{\text{can}} \leq 6\frac{t}{p}\cdot\sqrt{\frac{\epsilon_p^2 t^2 n}{12} + \frac{\ell n B^2}{144}} = \frac{t}{p}\cdot\sqrt{3\epsilon_p^2 t^2 n + \frac{\ell n B^2}{4}}.$$

Noise in a fresh ciphertext for $X = P_{B/2}$.
As shown above, the noise in a fresh ciphertext satisfies $N_{\text{fresh}} = \frac{t}{p}\left(-\epsilon_p m + \sum_{k=1}^{\ell} r_k e_k\right)$, where the r_k's are uniform polynomials with coefficients in $\{-B/2, \dots, B/2\}$ and the coefficients of the e_k's are uniform in $\left(-\frac{1}{2}, \frac{1}{2}\right]$. With high probability, using Section 2.5, we can write the bound $\|N_{\text{fresh}}\|^{\text{can}} \leq 6\sqrt{nV_{\text{fresh}}}$, where

$$V_{\text{fresh}} = \frac{t^2}{p^2}\left(\epsilon_p^2 V_m + \ell \cdot n \cdot V_r V_{e_k}\right) = \frac{t^2}{p^2}\left(\frac{\epsilon_p^2 t^2}{12} + \ell \cdot n \cdot \frac{B^2}{12}\cdot\frac{1}{12}\right).$$

Hence we have the following bound on the noise in a fresh ciphertext

$$\|N_{\text{fresh}}\|^{\text{can}} \leq 6\frac{t}{p}\cdot\sqrt{\frac{\epsilon_p^2 t^2 n}{12} + \frac{\ell n^2 B^2}{144}} = \frac{t}{p}\cdot\sqrt{3\epsilon_p^2 t^2 n + \frac{\ell n^2 B^2}{4}}\,.$$

Other homomorphic operations.

Let $\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1)$ and $\mathtt{ct}' = (\mathtt{ct}'_0, \mathtt{ct}'_1)$ be input ciphertexts to homomorphic operations. The outputs of homomorphic addition, homomorphic multiplication, relinearization, and modulus switching are defined and analysed in the same way as in Section 3.3 for the LPR-type scheme.

4.4 The Decisional Knapsack problem

The security proof of the Regev-type scheme relies on the hardness of the Ring-LWR problem. However, an intermediate problem is encountered in the proof, namely a two-ring decisional knapsack problem that we define in this section. We will also show in Theorem 3 that under well chosen parameters, this problem is statistically intractable.

The two-ring Decisional Knapsack problem.

Let p and q two integers and n a power of two. Let X be a finite subset of the ring $R = \mathbb{Z}[x]/(x^n + 1)$. Recall that $R_p = \mathbb{Z}_p[x]/(x^n + 1)$ and $R_q = \mathbb{Z}_q[x]/(x^n + 1)$.

Definition 8 (2DKS distribution)**.** *The two-ring Decisional Knapsack Problem distribution with parameters n, p, q, ℓ, X (denoted by the $2DKS_{n,p,q,\ell,X}$ distribution) is the distribution over $R_p^\ell \times R_q^\ell \times R_p \times R_q$ which consists of samples $(w_1, \ldots, w_\ell, v_1, \ldots, v_\ell, w, v)$ formed as follows. The $v_1, \ldots, v_\ell$ are independent and uniformly random in R_q, the $w_1, \ldots, w_\ell$ are independent and uniformly random in R_p, $v = \sum_{k=1}^{\ell} r_k v_k$ and $w = \sum_{k=1}^{\ell} r_k w_k$, for some independently chosen, uniformly random elements r_k from X.*

Definition 9 (2DKS problem)**.** *The two-ring Decisional Knapsack Problem with parameters n, p, q, ℓ, X, denoted by $2DKS_{n,p,q,\ell,X}$, is defined as follows: given samples $(w_1, \ldots, w_\ell, v_1, \ldots, v_\ell, w, v)$, decide whether they are sampled from the $2DKS_{n,p,q,\ell,X}$ distribution or are uniformly random over $R_p^\ell \times R_q^\ell \times R_p \times R_q$.*

In order for the security proof to hold, the set X from which the randomness is sampled uniformly at random needs to have a special property.

Definition 10 (Exceptional set)**.** *Let $p \geq 2$ and $q \geq 2$ be two integers. We say that a finite subset $E_{p,q}$ of $R = \mathbb{Z}[x]/(x^n + 1)$ is exceptional if, for any $r \neq r'$ in $E_{p,q}$, $r - r'$ is invertible both mod p and mod q.*

We prove that our scheme is secure if the set X is exceptional, and look at the particular examples $X = S_{B/2}$ and $P_{B/2}$, for a suitable choice of positive integer B.

Knapsack-type problems have been previously studied in the literature [54, 69, 70]. A knapsack problem in a single ring was considered by Baum et al. [54]. Our problem slightly differs from those in prior works, as it is defined over two rings, and we allow for the r_k values to be polynomials. However, we show in Section E that 2DKS is equivalent to the single-ring problem of [54, Def. 1]. Moreover, the latter problem can be shown to be equivalent to LWE [70], and we discuss this further in Section F.

Hardness of the Decisional Knapsack Problem.

We now show that the 2DKS problem that we have introduced is indeed hard under well chosen parameters for the case of exceptional sets $E_{p,q}$. In order to set the parameters for which this problem is indeed hard, we need to compute the statistical

distance

$$\Delta((v_1,\ldots,v_\ell,w_1,\ldots,w_\ell,v,w),(v_1,\ldots,v_\ell,w_1,\ldots,w_\ell,\sum_{k=1}^{\ell}r_kv_k,\sum_{k=1}^{\ell}r_kw_k)),$$

where $v_1,\ldots,v_\ell,v \leftarrow R_q$, $w_1,\ldots,w_\ell,w \leftarrow R_p$, and $r_1,\ldots,r_\ell \leftarrow E_{p,q}$.

Lemma 3. *Let p and q be two integers and $E_{p,q}$ be an exceptional set of $R = \mathbb{Z}[x]/(x^n+1)$. Given ℓ pairs of polynomials $(v_k,w_k) \in R_q \times R_p$, we define the following map $h_{(v_k,w_k)_k} : E_{p,q}^{\ell} \to R_q \times R_p$ as follows: $h_{(v_k,w_k)_k}(r_1,\ldots,r_\ell) = \left(\sum_{k=1}^{\ell} r_kv_k, \sum_{k=1}^{\ell} r_kw_k\right)$. Then the family $\mathcal{H}$ of functions $h_{(v_k,w_k)_k}$ is universal.*

Proof. A proof is given in Section A. □

The following result can be used to argue the hardness of the 2DKS problem. In the case when $E_{p,q}$ is a set of polynomials with bounded coefficients, this theorem generalises the result from [54, Lemma 4] to the two-ring case.

Theorem 3. *Let p, q and ℓ be positive integers, and let $E_{p,q}$ be an exceptional set of the ring $R = \mathbb{Z}[x]/(x^n+1)$. If ℓ satisfies the following inequality*

$$\ell \geq \frac{1}{\log|E_{p,q}|} \cdot (n\log pq + 2\lambda - 2),$$

then for any elements $v_k, v \leftarrow R_q$, $w_k, w \leftarrow R_p$, and $r_k \leftarrow E_{p,q}$ for any $k \in \{1,\ldots,\ell\}$, we have that

$$\Delta((v_1,\ldots,v_\ell,w_1,\ldots,w_\ell,v,w),(v_1,\ldots,v_\ell,w_1,\ldots,w_\ell,\sum_{k=1}^{\ell}r_kv_k,\sum_{k=1}^{\ell}r_kw_k)) \leq \frac{1}{2^\lambda}.$$

Proof. According to Lemma 3, the family of functions $h_{(v_k,w_k)_k} : E_{p,q}^{\ell} \to R_q \times R_p$ defined as follows: $h_{(v_k,w_k)_k}(r_1,\ldots,r_\ell) = (\sum_{k=1}^{\ell} r_kv_k, \sum_{k=1}^{\ell} r_kw_k)$ is universal. By Lemma 1, it follows that for $h \leftarrow \mathcal{H}$ (or equivalently, $(v_k,w_k)_k \leftarrow R_q^{\ell} \times R_p^{\ell}$), $r_1,\ldots,r_\ell \leftarrow E_{p,q}$ and $(v,w) \leftarrow R_q \times R_p$:

$$\Delta((h,h(r_1,\ldots,r_\ell),(h,(v,w))) \leq \frac{1}{2}\sqrt{\frac{|R_q\times R_p|}{|E_{p,q}|^\ell}} = \frac{1}{2}\sqrt{\frac{(pq)^n}{|E_{p,q}|^\ell}} \leq \frac{1}{2^\lambda},$$

by the choice of ℓ. □

4.5 Security proof

In this section we prove that the public-key encryption scheme underlying our Regev-type SHE scheme is IND-CPA secure assuming the hardness of the Decision Ring-LWR Problem, under well chosen parameters.

Theorem 4. *Let p, q and ℓ be positive integers. If the set X from which the randomness is sampled in the encryption process is an exceptional set $E_{p,q}$ of the ring $R = \mathbb{Z}[x]/(x^n + 1)$ and ℓ is chosen such that*

$$\ell \geq \frac{1}{\log |E_{p,q}|} \cdot (n \log pq + 2\lambda - 2),$$

then the public key encryption scheme underlying the Regev-type homomorphic encryption scheme described in Section 4.1 is IND-CPA secure assuming the hardness of Decision Ring-LWR$_{n,q,p}$.

Proof. The proof works via the following series of hybrids:

- **Hybrid** 1: This is the real IND-CPA game, where the ciphertext $\mathtt{ct}$ encrypting a message m is the output of the encryption algorithm run on a genuine public key $\mathtt{pk} := \{(v_1, u_1), \ldots (v_\ell, u_\ell)\}$ that is output from an honest execution of the key generation algorithm.
- **Hybrid** 2: This is the same as Hybrid 1, except that the public key $\mathtt{pk}$ is chosen uniformly at random from the public key space $R_q^\ell \times R_p^\ell$, rather than as an honest output of the key generation algorithm. The ciphertext $\mathtt{ct}$ is honestly generated by encrypting m under this uniform public key.
- **Hybrid** 3: This is the same as Hybrid 2 except that the ciphertext $\mathtt{ct}$ is chosen uniformly at random from the ciphertext space $R_q \times R_p$ rather than as an honest output of the encryption algorithm.

We first observe that the advantage of the adversary $\mathcal{A}$ in Hybrid 3 is 0 since all the information that the adversary has is uniformly random and independent of the message m. We proceed by arguing that the distinguishing advantages of an adversary between Hybrid 2 and Hybrid 1 and between Hybrid 3 and Hybrid 2 are negligible. We can then conclude that the adversary's advantage in Hybrid 1, the real IND-CPA game, is negligible.

We now show that the distinguishing advantage between Hybrid 2 and Hybrid 1 is negligible, assuming that the Decision Ring-LWR problem is hard. Consider an algorithm $\mathcal{A}$ that can distinguish between these two hybrids. We can transform the algorithm $\mathcal{A}$ to an algorithm $\mathcal{D}$ that distinguishes Ring-LWR$_{n,q,p}$ samples from uniformly random ones. The algorithm $\mathcal{D}$ collects ℓ input pairs and forms a public key, encrypts a message m under the public key, and invokes $\mathcal{A}$ on $(\mathtt{pk}, \mathtt{ct})$. The algorithm $\mathcal{D}$ then forwards the output of $\mathcal{A}$. If the input samples are Ring-LWR$_{n,q,p}$ samples, then $\mathcal{D}$ perfectly simulates Hybrid 1. Similarly, if the input samples are uniform, then $\mathcal{D}$ perfectly simulates Hybrid 2. The advantage of the algorithm $\mathcal{A}$ is less than the advantage of the algorithm $\mathcal{D}$. Hence, the advantage of $\mathcal{A}$ is negligible assuming that Decision Ring-LWR$_{n,q,p}$ is hard.

We now show that the distinguishing advantage between Hybrid 2 and Hybrid 3 is negligible, assuming that the two-ring Decisional Knapsack Problem is hard. Consider an algorithm $\mathcal{A}$ that can distinguish between the two hybrids. We can transform the algorithm $\mathcal{A}$ to an algorithm $\mathcal{D}$ that solves 2DKS. Let $(v_1, \ldots, v_\ell, w_1, \ldots, w_\ell, v, w)$ be an input to $\mathcal{D}$. The algorithm $\mathcal{D}$ invokes $\mathcal{A}$ for public key $(v_1, \ldots, v_\ell, w_1, \ldots, w_\ell)$ and

ciphertext $(v, w+\Delta_p m)$. The algorithm $\mathcal{D}$ then outputs the output of $\mathcal{A}$. If the input of $\mathcal{D}$ comes from the 2DKS distribution, then $\mathcal{D}$ perfectly simulates Hybrid 2. Similarly, if the input comes from a uniform distribution, then $\mathcal{D}$ perfectly simulates Hybrid 3. The advantage of the algorithm $\mathcal{A}$ is less than the advantage of the algorithm $\mathcal{D}$. Moreover, the advantage of the algorithm $\mathcal{D}$ is negligible, by the choice of parameters and by Theorem 3. Hence, the advantage of $\mathcal{A}$ is negligible. □

Instantiating X with the sets $S_{B/2}$ and $P_{B/2}$, we get the following statements by invoking Theorem 3, Theorem 4 and [55, Corollary 1.2]. Their proofs are given in Section A.

Corollary 1. *Let $p = p_1 \cdot \ldots \cdot p_s$ and $q = q_1 \cdot \ldots \cdot q_t$, for distinct primes $p_1, \ldots, p_s$ and distinct primes $q_1, \ldots, q_t$, and B a positive integer, $B \leq \frac{1}{2}\min\{p_i, q_j\}_{1\leq i\leq s, 1\leq j\leq t}$. Let $E_{p,q}$ be the subset $S_{B/2}$ of integer scalars bounded by $B/2$ of the ring R, from which the randomness in the encryption process is sampled. Then, $E_{p,q}$ is exceptional and if ℓ is chosen as*

$$\ell \geq \frac{1}{\log(B+1)} \cdot (n \log(pq) + 2\lambda - 2),$$

then the public-key encryption scheme is IND-CPA secure, assuming the hardness of the Decision Ring-LWR$_{n,q,p}$ problem.

Corollary 2. *Let $1 < d_1, \ldots, d_s, d'_1, \ldots, d'_t \leq n$ be powers of 2 and distinct prime integers $p_1, \ldots, p_s$ and distinct prime integers $q_1, \ldots, q_t$ prime integers such that $p_i \equiv 2d_i + 1 \pmod{4d_i}$ and $q_j \equiv 2d'_j + 1 \pmod{4d'_j}$, for any $1 \leq i \leq s$ and $1 \leq j \leq t$. Let $p = p_1 \cdot \ldots \cdot p_s$, $q = q_1 \cdot \ldots \cdot q_t$ and B be a positive integer such that $B + 1 \leq \min\{\frac{1}{\sqrt{d_i}} \cdot p_i^{1/d_i}, \frac{1}{\sqrt{d'_j}} \cdot q_j^{1/d'_j}\}_{1\leq i\leq s, 1\leq j\leq t}$. Let $E_{p,q}$ be the set $P_{B/2}$ of polynomials with integer coefficients in $\{-B/2, \ldots, B/2\}$, from which the randomness in the encryption process is sampled. Then, $E_{p,q}$ is exceptional and if ℓ is chosen as*

$$\ell \geq \frac{1}{\log(B+1)} \cdot \left(\log(pq) + \frac{2\lambda - 2}{n}\right),$$

the public-key encryption scheme is IND-CPA secure, assuming the hardness of the Decision Ring-LWR$_{n,q,p}$ problem.

5 Concrete security against best known attacks

In this section we analyse the concrete security of the LPR-type scheme and of the Regev-type scheme, by considering key recovery and plaintext attacks. We begin by discussing the concrete security of their underlying hard problems, the Ring Learning with Rounding (Ring-LWR) problem and the two-ring Decisional Knapsack problem (2DKS).

5.1 Concrete security of Ring-LWR

The concrete security of a (Ring)-LWR instance can be estimated by interpreting it as an LWE instance, since there are no known attacks exploiting the LWR (or ring)

structure. An LWR instance $\left(\mathbf{a}, b := \left\lceil \frac{p}{q}\langle \mathbf{a}, \mathbf{s}\rangle \right\rfloor\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$ can be mapped to an LWE instance $\left(\mathbf{a}, b' := \frac{q}{p} \cdot b\right)$, where $b' = \langle \mathbf{a}, \mathbf{s}\rangle + e$ and e is chosen from a uniform distribution on the set $\{-\frac{q}{2p} + 1, \ldots, \frac{q}{2p}\}$. This enables us to use a tool such as the Lattice Estimator [71] to estimate the concrete security of an LWR parameter set. We model the implied LWE error distribution as a Gaussian with standard deviation $\sigma = \sqrt{((q/p)^2 - 1)/12}$, as done e.g. in [72].[2]

To check a parameter set meets a 128-bit security target, we verify that the dual_hybrid, primal_usvp, and primal_bdd algorithms are estimated to cost at least 2^{128} rop[3]. Note that these are expected to be the most performant algorithms for FHE parameter sets [73]. Where n is small enough, we also verified the parameter sets using the top level estimate() function.

5.2 Concrete security of the 2DKS problem

The security proof of our Regev-type scheme (Theorem 4) shows a regime for the parameter ℓ that assures the statistical intractability of the two-ring decisional knapsack problem, $\text{2DKS}_{n,p,q,\ell,X}$ (Definition 9). However, we can also consider the cryptanalysis of this problem, which may enable us to choose a smaller parameter ℓ such that algorithms for solving $\text{2DKS}_{n,p,q,\ell,X}$ are expected to be computationally inefficient. We focus on this point of view in this subsection.

By Theorem 5, the $\text{2DKS}_{n,p,q,\ell,X}$ problem is equivalent to the decisional knapsack problem in a single ring, $\text{DKS}_{n,pq,\ell,X}$. We now summarise algorithms for solving this decisional knapsack problem. This discussion will enable us to justify a concrete choice for the parameter ℓ for the public key.

Brute force.

Clearly, if, for example, $r_k \in X = \{0, 1\}$ and ℓ is very small, a knapsack sample $\sum_{k=1}^{\ell} r_k a_k = a \bmod pq$, with a_k uniform over R_{pq}, is easy to distinguish from a uniform element $a \leftarrow R_{pq}$, since we can enumerate all choices for r_k and hence all possible elements of the form $\sum_{k=1}^{\ell} r_k v_k$. The choice of $\ell = 80$ was proposed in [35] to make such a brute force approach infeasible. We show in Remark 1 that this choice of ℓ is vulnerable to a linear algebra attack.

Remark 1. *The matrix form of the $DKS_{n,pq,\ell,X}$ problem in the case that X consists of constant polynomials (e.g. $X = \{0, 1\}$ or $X = S_{B/2}$) implies that we must choose $\ell \geq n$ to avoid a linear algebra attack. Indeed, given a sample of its distribution, $\sum_{k=1}^{\ell} r_k a_k = a \bmod pq$, for $r_k \in X$, let $\boldsymbol{A}$ be the horizontal concatenation of vector of coefficients of a_k, and let $\boldsymbol{a}$ be the vector of coefficients of a. Let also $\boldsymbol{r}$ be the vector of coefficients of r_k, such that $\boldsymbol{A}\boldsymbol{r} = \boldsymbol{a} \bmod pq$, with $\boldsymbol{A} \leftarrow \mathbb{Z}_{pq}^{n\times\ell}$, $\boldsymbol{r} \leftarrow X^{\ell}$, and $\boldsymbol{a} \in \mathbb{Z}_{pq}^{n}$. If, on the contrary, $\ell < n$, we can consider the matrix $\boldsymbol{A}$ as a vertical concatenation of two row blocks, $\boldsymbol{A}_1 \in \mathbb{Z}_{pq}^{\ell\times\ell}$ and $\boldsymbol{A}_2 \in \mathbb{Z}_{pq}^{n-\ell\times\ell}$. We can also see the vector $\boldsymbol{a}$ as a vertical concatenation of two vectors, $\boldsymbol{a}_1 \in \mathbb{Z}_{pq}^{\ell}$ and $\boldsymbol{a}_2 \in \mathbb{Z}_{pq}^{n-\ell}$. As $\boldsymbol{A}$ is random, it has full rank ℓ with high probability, and further, by a permutation of rows, we can*

[2] That is, the standard deviation σ is set to be equal to the standard deviation of the uniform distribution over $\{-\frac{q}{2p} + 1, \ldots, \frac{q}{2p}\}$.

[3] We will make available our estimation scripts upon acceptance of the paper.

assume, without loss of generality, that $\boldsymbol{A}_1$ *satisfies this rank. Using this notation, we derive two equations, namely* $\boldsymbol{A}_1\boldsymbol{r} = \boldsymbol{a}_1 \bmod pq$ *and* $\boldsymbol{A}_2\boldsymbol{r} = \boldsymbol{a}_2 \bmod pq$*. As* $\boldsymbol{A}_1$ *is invertible, we can easily recover* $\boldsymbol{r}$ *from the first equation and check if this satisfies the second equation.*

Combinatorial algorithms.

The best classical and quantum algorithms for the related problem of binary subset sum (without modular reduction) that we are aware of were presented in [74]. The idea is to generalise meet-in-the-middle approaches by constructing lists of *representations* which can be merged to form solutions. Meet-in-the-middle algorithms for LWE with ternary secret using representation techniques were presented in [75], and more generally for secrets chosen from a specified distribution over a small range $[-\eta, \eta]$, for a positive integer $\eta \leq 3$ in [76]. Since the decisional knapsack problem can be reduced to LWE (as we will detail below), such algorithms may apply in our context. However, the complexity of these approaches must be calculated as a numerical optimisation, and there is no theory that states when the complexity will converge to the optimal value [77]. Moreover, the work [76] notes that their asymptotic running times are slightly worse than lattice reduction. For this reason, we do not consider combinatorial approaches further.

Reduction to Ring Inhomogeneous SIS.

The argument of Remark 1 shows how to express the knapsack problem as a problem over vectors and matrices when X is a set of constant polynomials. We can also express the knapsack problem as a problem over vectors and matrices when for a general subset $X \subseteq R$ as follows. Given a $\mathrm{DKS}_{n,pq,\ell,X}$ sample $\sum_{k=1}^{\ell} r_k a_k = a \bmod pq$, for $r_k \in X$, write each random polynomial r_k as its vector of coefficients $\mathbf{r}_k$ and stack these vectors to make a vector $\mathbf{r}$. Consider $\mathbf{a}$ as the vector of coefficients of the polynomial a. Write the rotation matrix of multiplication by each a_k as $Rot(a_k) \in \mathbb{Z}_{pq}^{n\times n}$. Then form $\mathbf{A}$ as the horizontal concatenation of the $Rot(a_k)$ so that $\mathbf{Ar} = \mathbf{a}$, where $\mathbf{A} \leftarrow \mathbb{Z}_{pq}^{n\times \ell n}, \mathbf{a} \in \mathbb{Z}_{pq}^{n}$ and $\mathbf{r} \in \mathbb{Z}_{pq}^{\ell n}$.

In particular, if $X = S_{B/2}$ or $X = P_{B/2}$ then the resulting vector $\mathbf{r}$ has bounded coefficients. This formulation of the problem then resembles a (Ring) Inhomogeneous SIS instance [78, 79]. Algorithms for Inhomogeneous SIS may thus be applicable to the decisional knapsack problem in this case. Such algorithms can be combinatorial [79] or lattice based [78, 80]. A full analysis of algorithms for solving the knapsack problem is beyond the scope of this work.

Reduction to LWE.

We will justify our choice of ℓ by viewing the DKS problem as LWE in Hermite Normal Form (HNF). This was also done for a similar scheme in [80]. Moreover, as will we show in Section 5.4, it leads to a choice $\ell = O(n)$ that is asymptotically optimal, when comparing to the lower bound of $\ell \geq n$ shown in Remark 1. Thus, we believe this approach for justifying the choice of ℓ is reasonable.

We present this attack of [80] for solving the knapsack problem $\mathrm{DKS}_{n,pq,\ell,X}$ for the specific sets X that we will encounter in our Regev-type scheme. If $X = P_{B/2}$ then,

as shown above, we can express the DKS sample as $\mathbf{Ar} = \mathbf{a}$, where the matrix $\mathbf{A}$ can be split as $\begin{pmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{pmatrix}$, with $\mathbf{A}_1 \in \mathbb{Z}_{pq}^{n \times \ell n - n}$ and $\mathbf{A}_2 \in \mathbb{Z}_{pq}^{n \times n}$. By ignoring the algebraic structure, we can consider the matrix $\mathbf{A}$ as uniform over $\mathbb{Z}_{pq}^{n \times \ell n}$, as the polynomials a_i are uniform over R_{pq}. Hence, with high probability, it has full rank equal to n. By an eventual permutation of the rows, we can further assume $\mathbf{A}_2$ has full rank n, and so it admits an inverse. We also split the vector $\mathbf{r}$ as $\mathbf{r} = \begin{pmatrix}\mathbf{r}_1 & \mathbf{r}_2\end{pmatrix}$, with $\mathbf{r}_1 \in \mathbb{Z}_{pq}^{\ell n - n}$ and $\mathbf{r}_2 \in \mathbb{Z}_{pq}^{n}$. The equation now reads as: $\mathbf{A}_1\mathbf{r}_1 + \mathbf{A}_2\mathbf{r}_2 = \mathbf{a} \bmod pq$. If we multiply it by the inverse of $\mathbf{A}_2$, we get $\mathbf{A}_2^{-1}\mathbf{A}_1\mathbf{r}_1 + \mathbf{r}_2 = \mathbf{A}_2^{-1}\mathbf{a} \bmod pq$. We can see that this equation resembles an LWE instance in HNF form, with modulus pq, secret dimension $\ell n - n$ and both secret and error following the uniform distribution over $\{-B/2, \ldots, B/2\}$. Remark 1 shows that a similar formulation can be given if $X = S_{B/2}$, and the above split of $\mathbf{A}$ is also possible, as we must choose $\ell \geq n$. In particular, we get a LWE instance in HNF form, with modulus pq, secret dimension $\ell - n$ and both secret and error following the uniform distribution over $\{-B/2, \ldots, B/2\}$.

For the sake of completeness, we note that we can also view the $\mathrm{DKS}_{n,pq,\ell,X}$ problem as an equivalent LWE problem via the work of [70]. We present in Supplementary Material Section F more details on this equivalence and the choices of ℓ that can be justified via this equivalence. Whether using the approach of [80] or [70], we estimate the security of a $\mathrm{DKS}_{n,pq,\ell,X}$ instance seen as an LWE instance using the Lattice Estimator.

5.3 Parameters for the LPR-type scheme

In this section, we present in Table 2 some concrete parameters for our LPR-type scheme and justify their security. As the public key is of the form $\mathtt{pk} = (a, \lfloor a \cdot s \rceil_{r,q})$, we can see that a key recovery attack corresponds to solving the search Ring-LWR instance given by ring dimension n, moduli $r > q$ and a uniform ternary secret s. As a ciphertext is of the form

$$\mathtt{ct} = (\lfloor \mathtt{pk}_0 u \rceil_{r,q}, \Delta_p m + \lfloor \mathtt{pk}_1 u \rceil_{q,p}) \in R_q \times R_p \,,$$

we can also see (from the $\mathtt{ct}_1$ component) that a plaintext recovery attack corresponds to solving a Ring-LWR instance parameterised by ring dimension n, moduli $q > p$ and a uniform ternary ephemeral secret u. From the security proof (Theorem 2), we can see that the hardness of the scheme can be reduced to the hardness of $\text{Ring-LWR}_{n,r,q}$ when $q|r$ and $pr = q^2$.

We illustrate some example parameter sets that target 128-bit security and that satisfy these constraints on the moduli. We choose power-of-two p, q and r for performance reasons. We then choose the ratios between the moduli to correspond to a standard deviation as close as possible to $\sigma = 3.2$, since this is the choice for the Gaussian error most widely used in Ring-LWE-based homomorphic encryption schemes [10]. We choose to fix the ratios $r/q = q/p = 16$. For this choice of r, q, and p our (Ring-)LWR instances can be modelled as LWE instances with standard deviation $\sigma \approx 4.61$, as described in Section 5.1. We note that this choice of the ratio of moduli is similar to choices for LWR-based PKEs. For example, Round5 [44] suggests $q/p = 8$ for parameter sets targetting 128-bit security.

In line with the Homomorphic Encryption standard [10], we target 128-bit security according to the Lattice Estimator, and we present parameters for ring dimensions n with $\log n \in \{10, \ldots, 15\}$ and uniform ternary secret distribution. We choose the modulus r to be equivalent in bitsize to the modulus that would be chosen in [10] in the Ring-LWE context, so that our parameters can be easily compared to Ring-LWE-based schemes.

This discussion justifies the possible concrete parameter sets for our LPR-type scheme presented in Table 2. The Lattice Estimator[4] of [71] was used to verify that the parameter sets in Table 2 are estimated to have 128-bit security. In particular, Table 2 presents the maximal moduli r, q, and p that satisfy the security constraint for fixed ratios $r/q = q/p = 16$ and fixed ring dimension n.

5.4 Parameters for the Regev-type schemes

In this section, we present in Tables 3 and 4 some concrete parameters for our Regev-type scheme and justify their security. Recall that this scheme is parametrised by a finite subset X of $R = \mathbb{Z}[x]/(x^n + 1)$, for sampling the randomness in encryption. The discussion in this section will be split depending on the choice of X.

We first consider a key recovery attack. As the public key is of the form $\texttt{pk} := \{(v_1, w_1 = \lfloor v_1 \cdot s \rceil_{q,p}), \ldots (v_\ell, w_\ell = \lfloor v_\ell \cdot s \rceil_{q,p})\}$, with v_i's drawn from uniform distribution over R_q and s as a uniform ternary secret, we can see that a key recovery attack corresponds to solving the search Ring-LWR instance given by ring dimension n, moduli $q > p$, a uniform ternary secret s, and ℓ samples. As the hardness of LWE is essentially independent of the number of samples [81], we can ignore the dependence on ℓ.

Depending on the choice of X, we need to consider further constraints on the moduli p and q. For $X = S_{B/2}$, Corollary 1 requires p and q to be products of distinct primes. For $X = P_{B/2}$, Corollary 2 considers p and q to be products of distinct primes satisfying $p \equiv 2d+1 \pmod{4d}$ and $q \equiv 2d'+1 \pmod{4d'}$, for some powers of two d and d'. This also suggests that setting d and d' as small as possible allows a larger value of B and hence, sampling random polynomials from a larger set. Therefore, for this X, we set p and q as products of distinct primes congruent to 5 mod 8. In both cases, we choose the modulus q to have identical bitsize as the modulus r chosen in Section 5.3. In particular, in both cases we set $q = 13p$ for prime p. This choice allows that the ratio q/p implies a standard deviation close to $\sigma = 3.2$ when modelling this Ring-LWR instance as a Ring-LWE instance, and satisfies the constraints on the form of q and p, noting that $13 = 5$ modulo 8. For readability, rather than presenting the explicit q and p in the manuscript, we will make available code that generates the q and p we have chosen that meet these constraints for ring dimensions n with $\log n \in \{10, \ldots, 15\}$. Assuming a uniform ternary secret distribution, the LWE instance implied by a key recovery attack with these parameters costs at least 2^{130} ring operations according to the Lattice Estimator.

[4] https://bitbucket.org/malb/lwe-estimator, commit cf36315

We next consider a plaintext recovery attack. Recall that a ciphertext is as follows:

$$\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1) = \left(\sum_{k=1}^{\ell} r_k v_k, \Delta_p m + \sum_{k=1}^{\ell} r_k w_k \right) \in R_q \times R_p,$$

where the random polynomials r_k are either from $X = S_{B/2}$ or from $X = P_{B/2}$. Given $v_1, \ldots, v_\ell$ in R_q as part of the public key, it suffices to recover the random r_k's, from the first component of the ciphertext, $\mathtt{ct}_0$, i.e., to solve the knapsack problem implied by the first ciphertext component.

Section 5.2 shows that the knapsack instance with $X = S_{B/2}$ can be seen as an LWE instance of modulus q, secret dimension $\ell - n$, number of samples n and secret and error distributions as uniform distributions over $\{-B/2, \ldots, B/2\}$ and that the knapsack instance with $X = P_{B/2}$ can be seen as an LWE instance of modulus q, secret dimension $\ell n - n$, number of samples n, and secret and error distributions as uniform distributions over $\{-B/2, \ldots, B/2\}$. In Tables 3 and 4 we present choices for fixed $B \in \{2, 4, 6\}$ of $\ell \approx 2n$ for $X = S_{B/2}$ and $\ell \in \{2, 3\}$ for $X = P_{B/2}$; such that the respective implied LWE instance is estimated to be 128-bit secure following Section 5.1. These example choices of $B \in \{2, 4, 6\}$ are motivated by similar choices in other lattice-based schemes [82, 83], but we note that a suitable ℓ could be chosen for any desired value of B.

6 Implementation and comparison with BFV

The goal of this section is to compare our LPR-type and Regev-type schemes with the BFV scheme [6]. This comparison is relevant since we consider the Ring-LWR-based LPR-type scheme (Section 3) to be a natural adaptation of the BFV scheme to the setting where rounding is used instead of Gaussian sampling. Moreover, in all the schemes from this work, the multiplication error grows similarly to the one of the scale-invariant BFV scheme.

Implementation.

To verify the practicality of our schemes we developed a proof-of-concept implementation of our LPR-type scheme in Python.[5] We present running time figures of our implementation in Table 5. The most costly operations are relinearization key generation and relinearization, and they roughly have the same running time. This is due to the increasing size of the relinearization key. The next most expensive operations is `Encrypt`. The ratio between `RelinKeyGen` and `Encrypt` is approximately 30 for the smallest parameter set, and approximately 257 for the largest parameter set.

Comparison of ciphertext sizes.

Since our implementation is only a proof-of-concept, comparing the running time with state-of-art BFV implementations such as [13] is not relevant. Because of this, we compare the performance of the schemes based on ciphertext size as in [58, 59]. This is

[5] We will publish the source code upon acceptance of the paper.

an important metric as the computational overhead when doing homomorphic evaluations heavily depends on the size of the ciphertext. Ciphertexts in each of our schemes and in the BFV scheme are comparable as in each case they consist of two ring elements. Moreover, homomorphic addition and multiplication in all the schemes follow nearly identical operations on ring elements. Hence, a smaller ciphertext size means smaller ring moduli and ring dimension, which translate into better performance.

Methodology.

We consider the same tree-shaped arithmetic circuit as in [59] and [58] that is parametrised by $\zeta = 8$ and depth L and takes input $(2\zeta)^L$ fresh ciphertexts. Each gate of this circuit performs ζ additions followed by one multiplication. For each scheme, our goal is to find a parameter set that simultaneously minimises the ciphertext size, is estimated to have at least 128 bits of security, and supports correct decryption of the output ciphertext of the above circuit. We find such parameter sets for each scheme for various choices of plaintext modulus $t \in \{3, 256, 257, 2^{16}, 2^{16}+1, 2^{32}, 2^{32}+1\}$ and circuits of depth $L \in \{2, 4, 6, \ldots, 30\}$.

To find such parameter sets, we recursively compute a bound on the noise of the output ciphertext of this circuit. This bound allows us to decide if this final ciphertext correctly decrypts. More concretely, for each scheme, we start with the noise bound of a fresh ciphertext N_0, and then compute a recurrence of the form $N_{i+1} = \texttt{multiplicative\_noise}(\texttt{additive\_noise}(N_i, \zeta))$ for $i \in \{0, \ldots, L-1\}$. The functions used in the recurrence are explicitly computed by using the noise bounds on fresh noise, addition, multiplication and relinearization presented in Section 3 for the LPR-type scheme, in 4 for the Regev-type scheme, and in [66] for the BFV scheme.

For a fixed scheme and choice (t, L), we iterate $n \in \{2^{11}, 2^{12}, 2^{13}, 2^{14}, 2^{15}\}$ and look for the smallest bit size of ciphertext modulus for which the final ciphertext correctly decrypts and that the underlying Ring-LWR or Ring-LWE assumption is estimated to be at least 128-bit secure. In particular, the final ciphertext will decrypt correctly if $N_L < 1/2$. Once we find such a minimum modulus (r for BFV or q for our schemes), we compute the ciphertext size as $2n\log(r)$ for the BFV scheme or $n(\log q + \log p)$ for our schemes. We will publish the script that generates the results tables upon acceptance of the paper.

We set the other parameters as follows. For BFV we set $\sigma = 3.2$ as the standard deviation for the Gaussian error. For the LPR-type scheme we set p, q, r as powers of two such that $q/p = 16$ and $r/q = 16$, according to Section 5.3. For the Regev-type scheme, denoted in the results tables as Regev($\ell = 3$), we sampled the randomness as polynomials with coefficients from $\{-1, 0, 1\}$ and set $\ell = 3$, according to Section 5.4. We also set the prime $p = 1 \bmod t$, which implies that $\epsilon_p = 1/t$ is as small as possible. Moreover, we set the primes p and q such that $q/p \approx 11.13$ according to Section 5.4.

We also considered a variant of the Regev-scheme, denoted in the results tables as Regev (LHL). In this variant, we again sampled the randomness as polynomials with coefficients in $\{-1, 0, 1\}$ and set the primes p and q such that $q/p \approx 11.13$. We chose ℓ according to Corollary 2, which corresponds to a setting for which the security proof holds. We set $p = 5 \bmod t$ when t is a power of two, so we are consistent with

the condition $p = 5 \bmod 8$, as also required for the security proof. This implies that $\epsilon_p = 5/t \in [0, 1)$.

Results.

Our results are presented in Tables 6, 7, 8, 9, 10, 11, and 12. When the plaintext modulus is either small, such $t = o(\sqrt{n})$, or when $p \bmod t$ is small in relation to t[6], the results show that the LPR and Regev-type schemes are marginally better than the BFV scheme in terms of ciphertext size for almost all circuit depths L. This is illustrated in Figure 1 which plots the results of Table 6 for which $t = 3$. When the plaintext modulus is large, the results show that in almost all cases, the LPR-type scheme and the BFV scheme perform comparably, while the Regev scheme marginally outperforms the two. This is illustrated in Figure 2 which plots the results of Table 12 for which $t = 2^{32} + 1$.

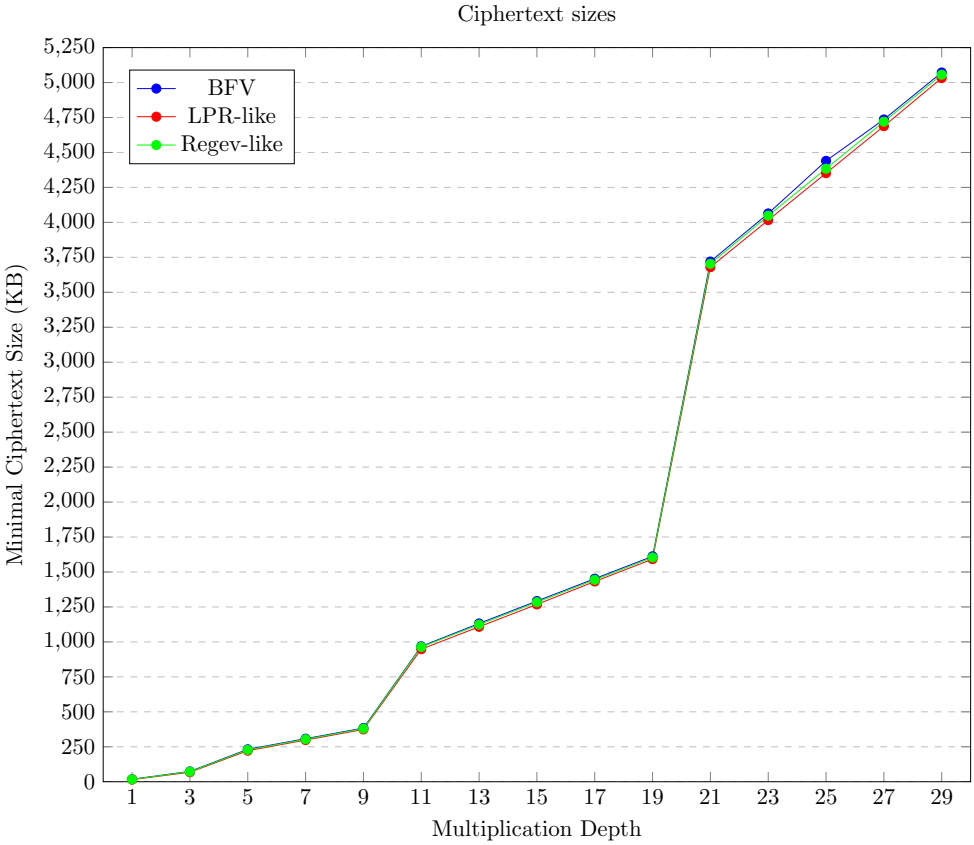

Figure 1 Minimal ciphertext size in kilobytes (kB) for which decryption is guaranteed to work for the given multiplicative depth, when fixing plaintext modulus $t = 3$.

[6]This is equivalent to $\epsilon_p = O(1/t)$, where $\epsilon_p = (p \bmod t)/t \in [0, 1)$

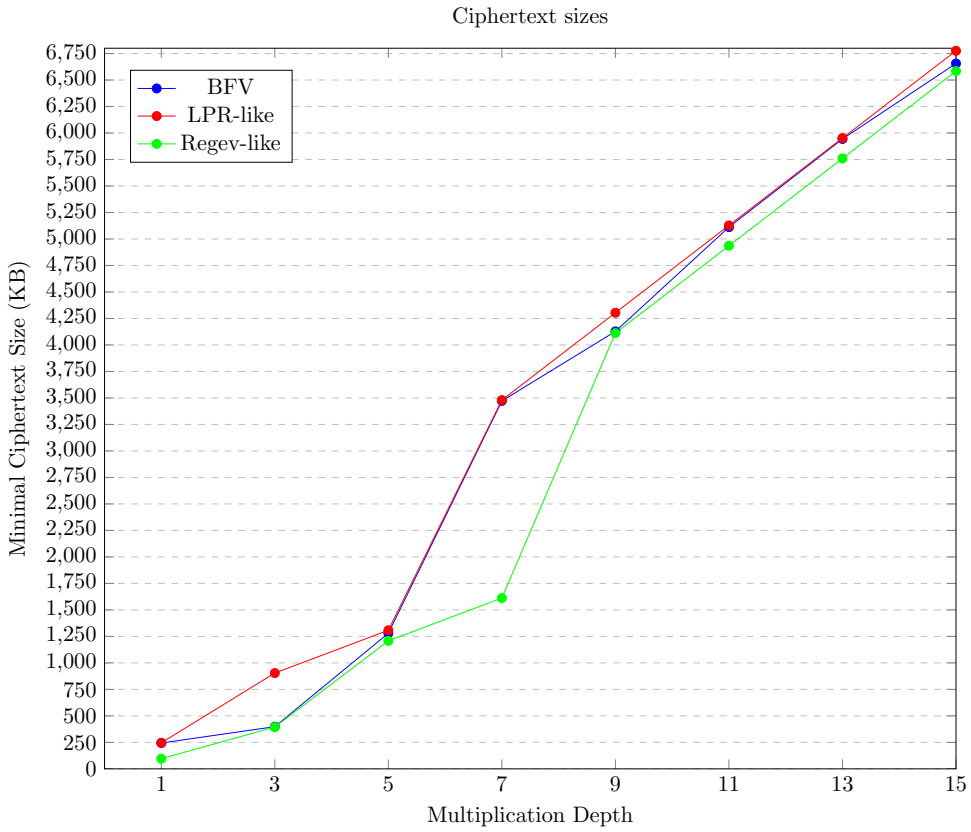

Figure 2 Minimal ciphertext size in kilobytes (kB) for which decryption is guaranteed to work for the given multiplicative depth, when fixing plaintext modulus $t = 2^{32} + 1$.

Both Figure 1 and Figure 2 show a step-wise behaviour [7]. This is expected: as the depth L increases, it will become necessary to switch to progressively larger power-of-two cyclotomic rings (i.e. larger n) in order to ensure that the ciphertext moduli are both large enough for correctness and small enough for the underlying Ring-LWR or Ring-LWE instance to be secure.

In the case of Figure 1, for all choices of depth L, all three schemes perform comparably, in the sense that the same choice of ring dimension n can support the computation for each of BFV, the LPR-type scheme, and the Regev-type scheme. In this case, we have that the sizes of the ciphertext moduli for the Regev-type and LPR-type scheme are slightly smaller than the ciphertext modulus for BFV, and so the overall ciphertext sizes are marginally smaller. To better illustrate the differences, we plot zoomed-in versions of Figure 1 on restricted ranges of multiplication depth in Appendix G.

[7] Plots for the other results tables would be similar, and so are omitted.

In the case of Figure 2, we see a couple of examples where the 'jump' to larger dimension may occur at different depths. For example, for $L = 3$, it is necessary to use a larger ring dimension for the LPR-type scheme than for BFV or the Regev-type scheme. On the other hand, for $L = 7$, it is necessary to use a larger ring dimension for the LPR-type scheme and BFV than for the Regev-type scheme. In other cases, the same ring dimension will support the computation for all three schemes. Overall, Figure 2 illustrates that for larger plaintext modulus t, the LPR-type scheme and BFV perform comparably, while the Regev-type scheme offers the smallest ciphertext size in each case.

The observation that the LPR and Regev-type schemes outperform the BFV scheme when the plaintext modulus is small can be explained when considering the size of the fresh noise of each scheme. For example, let us try to explain the difference in ciphertext size for the BFV scheme and the LPR-type scheme. Ignoring smaller terms in the noise recurrence, we notice that $N_{i+1} \approx t\sqrt{2n^2+3n} \cdot (\zeta \cdot N_i)$ (see Section 3 and [66]). The final noise is then approximated by $N_L \approx (t\zeta\sqrt{2n^2+3n})^{L-1} \cdot N_0$. Moreover, the final noise scaled by q should be close to $q/2$ in both cases, or else we could have chosen a smaller modulus. We can thus approximate $(t\zeta\sqrt{2n^2+3n})^{L-1} \cdot N_0^{BFV} q_{BFV} \approx (t\zeta\sqrt{2n^2+3n})^{L-1} \cdot N_0^{LPR} q_{LPR}$, since n, ζ, L, and t are fixed for both schemes. This simplifies to $q_{BFV}/q_{LPR} \approx N_0^{LPR}/N_0^{BFV}$. Using the fresh noise bounds of Section 3 and [66], we can thus approximate $q_{LPR}/q_{BFV} \approx \sqrt{\frac{4n+3(q/p)^2 \cdot (1+(t\epsilon_p)^2)}{48\sigma^2 n + 36\sigma^2 + 3(t\epsilon_q)^2}}$. When $\{\epsilon_p, \epsilon_q\} \in O(1/t)$, or when t is small, and when n is fairly large this can be simplified to $q_{LPR}/q_{BFV} \approx \frac{1}{2\sqrt{3}\sigma} \leq \frac{1}{16}$. This is consistent with our observation that the bitsizes of q_{BFV} that we obtained were larger than those of q_{LPR} by at most 4 bits.

Comparison of relinearization keys.

In addition to comparing the ciphertext sizes of our schemes with BFV, we can also compare the relinearization key sizes. We do now, assuming that relinearization uses a base-2 decomposition. Recall that in the BFV scheme [6], the relinearization key is made of $k+1$ pairs from $R_q \times R_q$, where $k = \lfloor \log_2(q) \rfloor$. Therefore, the relinearization key size is $\approx 2 \cdot n \cdot (\log_2(q) + 1) \cdot \log_2(q)$. According to Section 3.3, in the LPR-type scheme and in the Regev-type scheme, the relinearization key is made of $k+1$ pairs from $R_q \times R_p$, where $k = \lfloor \log_2(\frac{q^2}{p}) \rfloor$. By the choice $r/q = q/p = 16$ made in Section 5.3, we have $k = \lfloor \log_2(r) \rfloor$, and hence the relinearization key size in the LPR-type scheme is $\approx 2 \cdot n \cdot (\log_2(r) + 1) \cdot (\log_2(r) - 6)$. Given the fact that the BFV modulus q has the same bit size as the LPR modulus r, it follows that the LPR relinearization key is $12n \cdot (\log_2(r) + 1)$ bits less than the BFV relinearization key. By the choice $q/p = 13$ made in Section 5.4, we have $k = \lfloor \log_2(13q) \rfloor$, and hence the relinearization key size in the Regev-type scheme is $\approx n \cdot (\log_2(q) + \log_2(13) + 1) \cdot (\log_2(q) + \log_2(p))$. Given the fact that the BFV modulus q has the same bit size as the Regev-type scheme modulus q, it follows that the Regev relinearization key is $(\log(p) - 1) \cdot n \log(13)$ bits longer than the BFV relinearization key.

7 RNS variants of our schemes

When carrying out operations in our LWR-based schemes we are manipulating elements in large cyclotomic rings, using large moduli. Similarly to equivalent LWE-based schemes, we can use the Residue Number System (RNS) to speed up calculations with these large integers. In this section, we show how both of our schemes can be adapted to support RNS variants, similarly to RNS variants of BFV [52, 56, 57]. In order to use the RNS representation, we choose the moduli p and q to be products of smaller numbers, $q = \prod_{i \in I} q_i$ and $p = \prod_j p_{j \in J}$ for pairwise coprime q_i and p_j. We can then use the Chinese Remainder theorem to represent an integer $x \in \mathbb{Z}_q$ as $\{x_i = x \mod q_i \in \mathbb{Z}_i\}_i$. Operations on $x \in \mathbb{Z}_q$ can be carried out via applying the same operation to the $x_i \in \mathbb{Z}_{q_i}$. Note that p and q themselves do not need to be coprime, and in fact we will require that $p|q^2$ in order to support multiplication. Indeed, a particular permissible instance is $q = 13p$ for $p = \prod_j p_j$ for pairwise coprime p_j, which is a provably secure parameter setting for the Regev-type scheme (c.f. Corollary 2) or $q = 16p$ for the LPR-type scheme (c.f. Theorem 1, Section 5.3). A slight generalisation of this setting is $p = a$ and $q = ak$ for coprime a and k, which are each a product of distinct primes meeting the conditions of Corollary 2).

Many operations in the Regev-type and LPR-type schemes are the same (evaluation key generation, decryption, addition, multiplication, relinearization, and modulus switching). We focus on the details of the operations that are in common, and which are of most relevance to performant RNS computations. We address key generation and encryption in Appendix I. Addition is straightforward to implement RNS-component-wise so is omitted. In common with prior work on RNS-BFV [52, 56], we do not address modulus switching.

RNS background.

Our approach relies heavily on that of [52][8], and particularly we use their basis extension and simple scaling techniques. Since we mainly use these results directly, we defer the details of these techniques to Appendix H.

Decryption.

Decryption can be implemented with an adaptation of simple scaling [52, Section 2.3]. Suppose that the plaintext modulus t is a machine sized-integer (the approach can be adapted componentwise modulo an RNS representation of t if it is larger). Assume we have a ciphertext $\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1) \in R_q \times R_p$, then decryption is equivalent to computing:

$$m' := \left\lfloor \frac{t}{p} \left(-\frac{p}{q}(\mathtt{ct}_0 \cdot s) + \mathtt{ct}_1 \right) \right\rceil \bmod t$$

If the ciphertext is given in the RNS representation then we can compute $\mathtt{ct}_0 \cdot s$ in the RNS representation corresponding to q. So, for decryption it is sufficient to be able to compute the following. Given two ring elements $x \in R_q$ and $y \in R_p$

[8] While later work on RNS-BFV e.g. [57] has been presented that further improves on [52], we are not aiming to present fully optimised RNS variants in this work: our goal is rather to show that RNS variants of our schemes are possible to achieve.

(concretely, $x := -\mathtt{ct}_0 \cdot s \bmod q$ and $y = \mathtt{ct}_1 \bmod p$), with their corresponding RNS representations $x \equiv (x_i)_{i \in I}$ and $y \equiv (y_j)_{j \in J}$ we want to efficiently compute:

$$m' := \left\lfloor \frac{t}{q} \cdot x + \frac{t}{p} \cdot y \right\rceil \bmod t$$

Recall that the elements $x \in R_q$ and $y \in R_p$ can be reconstructed from their RNS components as $x = \sum_{i \in I} x_i \cdot q_i^\star \cdot \tilde{q}_i - v_x \cdot q$ and $y = \sum_{j \in J} y_j \cdot p_j^\star \cdot \tilde{p}_j - v_y \cdot p$, for some integers $v_x, v_y \in \mathbb{Z}$ (c.f. [52, Equation (3)]). Hence:

$$m' = \left\lfloor \sum_{i \in I} x_i \cdot \frac{t\tilde{q}_i}{q_i} - t v_x + \sum_{j \in J} y_j \cdot \frac{t\tilde{p}_j}{p_j} - t v_y \right\rceil \bmod t$$

which is equal to

$$m' = \left\lfloor \sum_{i \in I} x_i \cdot \frac{t\tilde{q}_i}{q_i} + \sum_{j \in J} y_j \cdot \frac{t\tilde{p}_j}{p_j} \right\rceil \bmod t$$

We can precompute the values $\frac{t\tilde{q}_i}{q_i} = \omega_{i,q} + \theta_{i,q}$ and $\frac{t\tilde{p}_j}{p_j} = \omega_{j,p} + \theta_{j,p}$ with $\omega_{i,q}, \omega_{j,p} \in \mathbb{Z}$ and $\theta_{i,q}, \theta_{j,p} \in [-1/2, 1/2)$. The computational cost of this approach is $|I| + |J|$ machine-sized integer multiplications, $|I| + |J|$ floating-point multiplications, and 2 floating-point roundings.

Noise.

The noise N in an RNS ciphertext, considered in its 'recombined' form $(\mathtt{ct}_0, \mathtt{ct}_1)$, is the polynomial of minimal infinity norm among all the polynomials for which there exists an integer polynomial G such that

$$\frac{t}{p} \left(-\frac{p}{q} \cdot \mathtt{ct}_0 \cdot s + \mathtt{ct}_1 \right) = m + N + tG \,.$$

That is, we define noise exactly the same as in the non-RNS case.

Correctness.

The correctness of decryption can be analysed using similar techniques as in [52]. In particular, we need to ensure the ciphertext noise is at most $1/4$, so that our RNS variants incur one bit of additional noise compared to our non-RNS schemes. Full details are given in Appendix I. For other operations, we directly apply results from [52] and so omit a detailed analysis.

Multiplication.

Multiplication is possible with an adaptation of the approach outlined in [52, Section 4.1]. Given RNS representatives of $\mathtt{ct}$ and $\mathtt{ct}'$ we want to compute an RNS

representation of:

$$(c_2, c_1, c_0) = \left(\left[\left\lfloor \frac{t}{p} \mathtt{ct}_0 \mathtt{ct}_0' \right\rceil \right]_{q^2/p}, \left[\left\lfloor \frac{t}{p} (\mathtt{ct}_0 \mathtt{ct}_1' + \mathtt{ct}_1 \mathtt{ct}_0') \right\rceil \right]_q, \left[\left\lfloor \frac{t}{p} \mathtt{ct}_1 \mathtt{ct}_1' \right\rceil \right]_p \right).$$

We can compute the CRT representatives of $c_0 = \left[\left\lfloor \frac{t}{p} x_0 \right\rceil \right]_p$ using the complex scaling procedure of [52, Section 2.4] as follows. Let $x_0 := \mathtt{ct}_1 \mathtt{ct}_1'$. Let $P = \prod_\iota P_\iota$ be coprime with p, q, $Q = q^2/p$, and t be such that $x_0 \in \mathbb{Z} \cap [-pP/2t, pP/2t) \subseteq \mathbb{Z}_{pP}$. We take the inputs $\mathtt{ct}_1$ and $\mathtt{ct}_1'$ with respect to base p, then do a basis extension to obtain the representatives $[\mathtt{ct}_1]_{pP}$ and $[\mathtt{ct}_1']_{pP}$. We then compute x_0 componentwise with respect to base pP. Then, the simple scaling procedure [52, Section 2.3] would be applied sufficiently many times using $q' = pP$ and $t' = tP$. This gives us $\left[\left\lfloor \frac{tP}{pP} x_0 \right\rceil \right]_{tP}$ which also gives us $\left[\left\lfloor \frac{t}{p} x_0 \right\rceil \right]_P$ (when discarding the mod t components). Moreover, by choosing P such that $x_0 \in \mathbb{Z} \cap [-pP/2t, pP/2t)$, we have $\left\lfloor \frac{t}{p} x_0 \right\rceil \in [-P/2, P/2)$, so we have computed the representatives mod P of $\left\lfloor \frac{t}{p} x_0 \right\rceil$ without modular reduction. Hence, we can apply basis extension [52, Section 2.2] to recover the mod-p_j representatives and thus obtain the representatives of $\left[\left\lfloor \frac{t}{p} x_0 \right\rceil \right]_p$ as required. For example, the choice $P > nt\frac{q^2}{p}$ can be used for computing each of c_2, c_1 and c_0, where this choice is analogous to how P is chosen in [52].

Let $x_1 := \mathtt{ct}_0 \mathtt{ct}_1' + \mathtt{ct}_1 \mathtt{ct}_0'$. Let $M = \mathrm{lcm}(p, q)$. Let $P > nt\frac{q^2}{p}$ be coprime with p, q, $Q = q^2/p$, and t such that $x_1 \in \mathbb{Z} \cap [-pP/2t, pP/2t) \subseteq \mathbb{Z}_{pP} \subseteq \mathbb{Z}_{MP}$. We are going to basis extend to mod MP. Given as input the RNS decompositions of $[\mathtt{ct}_0]_q$, $[\mathtt{ct}_1]_p$, $[\mathtt{ct}_0']_q$, $[\mathtt{ct}_1']_p$ we basis extend to obtain $[\mathtt{ct}_0]_{MP}$, $[\mathtt{ct}_1]_{MP}$, $[\mathtt{ct}_0']_{MP}$, $[\mathtt{ct}_1']_{MP}$. We can then compute x_1 RNS-componentwise with respect to base MP. Then, the simple scaling procedure [52, Section 2.3] would be applied sufficiently many times using $q' = MP$ and $t' = (M/p)tP$. This gives us $\left[\left\lfloor \frac{(M/p)tP}{MP} x_1 \right\rceil \right]_{(M/p)tP} = \left[\left\lfloor \frac{t}{p} x_1 \right\rceil \right]_{(M/p)tP}$ which also gives us $\left[\left\lfloor \frac{t}{p} x_1 \right\rceil \right]_P$ when discarding the mod t and mod (M/p) components. Moreover, by the choice of P, we have $\left\lfloor \frac{t}{p} x_1 \right\rceil \in [-P/2, P/2)$, so we have computed the representatives mod P of $\left\lfloor \frac{t}{p} x_1 \right\rceil$ without modular reduction. Hence, we can apply basis extension to recover the mod-q_j representatives and thus obtain the representatives of $\left[\left\lfloor \frac{t}{p} x_1 \right\rceil \right]_q$ as required.

Let $x_2 := \mathtt{ct}_0 \mathtt{ct}_0'$. We can again take $P > nt\frac{q^2}{p}$ coprime with p, q, $Q = q^2/p$, and t such that $x_2 \in \mathbb{Z} \cap [-pP/2t, pP/2t) \subseteq \mathbb{Z}_{pP}$. We take the inputs $\mathtt{ct}_0$ and $\mathtt{ct}_0'$ with respect to base q, then do a basis extension to obtain the representatives $[\mathtt{ct}_1']_{pP}$ and $[\mathtt{ct}_1']_{pP}$ (by extending to MP for $M = \mathrm{lcm}(p, q)$ and ignoring the representatives that are only factors of q). Then, the simple scaling procedure [52, Section 2.3] would be applied sufficiently many times using $q' = pP$ and $t' = tP$. This gives us $\left[\left\lfloor \frac{tP}{pP} x_2 \right\rceil \right]_{tP}$

which also gives us $\left[\left\lfloor \frac{t}{p} x_2 \right\rceil\right]_P$ (when discarding the mod t components). Moreover by the choice of P this is actually the mod P representatives of $\left\lfloor \frac{t}{p} x_2 \right\rceil$ without modular reduction. We can then apply a basis extension to obtain the mod q^2/p representatives of $\left[\left\lfloor \frac{t}{p} x_2 \right\rceil\right]_{q^2/p}$ as required.

We can define the noise in (c_2, c_1, c_0) exactly as for the non-RNS case, namely as the polynomial of minimal infinity norm among all the polynomials for which there exists an integer polynomial G_{mult} such that

$$\frac{t}{p}\left[\left(\frac{p}{q}\right)^2 c_2 s^2 - \frac{p}{q} c_1 s + c_0\right] = m_{\text{mult}} + N_{\text{mult}} + tG_{\text{mult}}\,.$$

The noise analysis is then exactly the same as for the non-RNS case, as presented in Appendix B.

Relinearisation.

Let us express the parameters as $q = \prod_{\iota \in I} q_i$, $p = \prod_{j \in J} p_j$, and let $Q = \prod_{h \in H} Q_h$ where we define $Q := q^2/p$. Notice that $J \subseteq I$ and $J \subseteq H$. Let $Q_h^* = Q/Q_h \in \mathbb{Z}$ and $\widetilde{Q_h} = [Q_h^{*-1}]_{Q_h} \in \mathbb{Z}_{Q_h}$. The input we have is c_2, with respect to its mod Q components Q_h, for $h \in H$, c_1 with respect to its mod q components q_i, for $i \in I$, and c_0 with respect to its mod p components p_j, for $j \in J$. In particular we can write $c_{2,h} = [c_2]_{Q_h}$ and so on.

The relinearisation keys are given by $(\alpha_h, \beta_h) \in R_q \times R_p$ where $\alpha_h \leftarrow R_q$ is chosen uniformly at random and $\beta_h = \left\lceil \alpha_h s \right\rfloor_{q,p} + \left\lceil \frac{p^2}{q^2} \widetilde{Q_h} Q_h^* s^2 \right\rfloor$. Using these keys, we want to compute $\widetilde{\mathtt{ct}_0}$ to be the RNS representations of $\sum_{h \in H} \beta_h c_{2,h} \pmod p$ and $\widetilde{\mathtt{ct}_1}$ to be the RNS reps of $\sum_{h \in H} \alpha_h c_{2,h} \pmod q$. We would then output $(\mathtt{ct}_0, \mathtt{ct}_1) = (\widetilde{\mathtt{ct}_1} + c_1, \widetilde{\mathtt{ct}_0} + c_0)$ where these additions can be implemented RNS-component-wise, mod q and mod p respectively. In Appendix I, we will show that if the ciphertext input to relinearisation has noise N, then this output ciphertext has noise

$$N_{\text{relin}} = N + \frac{t}{p} \sum_{h \in H} (\epsilon_{1,h} + \epsilon_{2,h}) c_{2,h}\,,$$

where $\epsilon_{1,h}$ and $\epsilon_{2,h}$ come from the two roundings in β_h. This noise growth can be seen to be completely analogous to the non-RNS case (c.f. Appendix C).

The remaining details are to show that we can express the relinearisation keys with respect to suitable RNS representation, and use these together with a suitable representation of c_2 in order to compute $\widetilde{\mathtt{ct}_0}$ and $\widetilde{\mathtt{ct}_1}$ RNS-component-wise. Indeed, we can compute $\widetilde{\mathtt{ct}_0}$ RNS-component-wise mod p by computing $\widetilde{\mathtt{ct}_{0,j}} = \sum_{h \in H} [\beta_h]_{p_j} \cdot [c_{2,h}]_{p_j} \pmod{p_j}$, and we can compute $\widetilde{\mathtt{ct}_1}$ RNS-component-wise mod q by computing $\widetilde{\mathtt{ct}_{1,i}} = \sum_{h \in H} [\alpha_h]_{q_i} \cdot [c_{2,h}]_{q_i} \pmod{q_i}$.

Firstly, let us consider the generation of each relinearisation key (α_h, β_h). We can directly sample each α_h in an RNS representation mod q, i.e. it gives us $[\alpha_h]_{q_i}$ for all

h. We can compute $\alpha_h \cdot s$ modulo q RNS-component-wise. We can then use simple scaling [52, Section 2.3] to obtain $\left\lceil \alpha_h s \right\rfloor_{q,p}$ in an RNS representation mod p. The object $\widetilde{Q_h} Q_h^* s^2$ can be considered mod Q and so given an RNS representation of this object mod Q, an RNS representation mod p of $\left\lceil \frac{p^2}{q^2} \widetilde{Q_h} Q_h^* s^2 \right\rfloor$ can be obtained via simple scaling with $\mathcal{T} = p$ and $\mathcal{Q} = q^2/p$. These terms can be added to give an RNS representation of β_h mod p, i.e. it gives us $[\beta_h]_{p_j}$ for all j. For each object $c_{2,h} = [c_2]_{Q_h}$ we need to obtain $[c_{2,h}]_{p_j}$ for all j and $[c_{2,h}]_{q_i}$ for all i in order to compute $\widetilde{\mathtt{ct}}_{0,j}$ and $\widetilde{\mathtt{ct}}_{1,i}$ respectively.

To see how this can be done, let us specialise to the case where $p = a$ and $q = ak$ for coprime a and k, each a product of distinct primes. In this case $Q = ak^2$. Considering an RNS basis for Q, the set $L = J \cup \{0\}$ and the representatives would be $Q_0 = k^2$ and $Q_j = p_j$, for each $j \in J$. Let us consider each component of c_2 mod Q. For $h \in H$, $c_{2,h} = c_2 \mod Q_h = p_h$ is in the range $[-p_h/2, p_h/2)$ and so $c_{2,h} = [c_{2,h}]_{p_h}$. Since $p_j \neq p_h$ for $j \neq h$, we can basis extend $[c_{2,h}]_{p_h}$ by each p_j for $j \neq h$ to obtain the RNS representation of $[c_{2,j}]$ mod p. Similarly, since k is also coprime to each p_j, we can basis extend $[c_{2,h}]_{p_h}$ by each p_j for $j \neq h$ and by k to obtain the RNS representation of $[c_{2,h}]$ mod q. For $h = 0$ we have $c_{2,0} = c_2 \mod Q_0 = k^2$ is in the range $[-k^2/2, k^2/2)$ and so $c_{2,0} = [c_{2,0}]_{k^2}$. Since k^2 is coprime to each p_ι, we can basis extend $[c_{2,0}]_{k^2}$ by each p_j for $j \in J$ and then discard the k^2 component to obtain the RNS representation of $c_{2,0}$ mod p. Moreover, since $k < p$, we have $Q_0 < q$, so $c_{2,0}$ is already reduced mod q. This means we can take its representatives mod q directly, in particular, we can obtain the mod p representatives as before and we can obtain the mod k representatives by directly computing $c_{2,0} \mod k$.

Acknowledgements. Anamaria Costache and Rachel Player were partially supported by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701). Erin Hales was supported by the EPSRC as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/P009301/1). Most of this work was done while Miruna Rosca was working for Bitdefender. We thank Yixin Shen for helpful discussions about knapsack problems.

Declarations

The authors declare that they have no competing financial or non-financial interests.

	Regev-type scheme	LPR-type scheme	LWWC [48]	BFV [6]
Size of `pk`	$\ell n \log(PQ)$	$n \log(rq)$	$(\ell'+1)n \log(q')$	$2n \log(q'')$
Size of `evk`	$n(K+1) \log(PQ)$	$n(k+1) \log(pq)$	$(\ell'+1)nk' \log(p')$	$2n(k''+1) \log(q'')$
Size of `ct`	$n \log(PQ)$	$n \log(pq)$	$(\ell'+1)n \log(p')$	$2n \log(q'')$
Security	Ring-LWR$_{n,Q,P}$	Ring-LWR$_{n,r,q}$	Ring-LWR$_{n,q',p'}$	Ring-LWE$_{n,\chi,q''}$
Constraints	P, Q products of primes; $P \vert Q^2$	$q \mid r$, $pr = q^2$	$p' \mid q'$	N/A

Table 1 A comparison of our Regev-type scheme and our LPR-type scheme with the prior schemes LWWC [48] and BFV [6]. The parameter constraints specified are required for provable security. We may assume $\log(q'') = \log(Q) = \log(r)$, while $q' > p'$ and p' is a polynomial factor larger than q''. If the encryption randomness in the Regev-type scheme is sampled from the set of scalars $\{-B/2, \ldots B/2\}$ then $\ell \geq 1/\log(B+1)(n \log(PQ) + 2\lambda - 2)$, while $\ell' > \log(q') + 2\lambda$. In the `evk` we have $K = \lfloor \log(\frac{Q^2}{P}) \rfloor$, $k = \lfloor \log(\frac{q^2}{p}) \rfloor$, $k' = (\ell'+1)(\ell'+2)(\lfloor \log(p') \rfloor + 1)/2$, and $k'' = \lfloor \log(q'') \rfloor$.

n	r	q	p
2^{15}	2^{856}	2^{852}	2^{848}
2^{14}	2^{425}	2^{421}	2^{417}
2^{13}	2^{211}	2^{207}	2^{203}
2^{12}	2^{105}	2^{101}	2^{97}
2^{11}	2^{52}	2^{48}	2^{44}
2^{10}	2^{26}	2^{22}	2^{18}

Table 2 Possible parameter sets n, r, q, p, for our LPR-type scheme, assuming a uniform ternary secret distribution, and targeted at 128-bit security.

n	ℓ $B = 2$	ℓ $B = 4$	ℓ $B = 6$
2^{15}	65606	65541	65506
2^{14}	32838	32773	32738
2^{13}	16444	16379	16344
2^{12}	8252	8187	8152
2^{11}	4146	4081	4046
2^{10}	2103	2033	1998

Table 3 Possible parameter sets n, ℓ, B, for our Regev-type scheme with $X = S_{B/2}$, assuming a uniform ternary secret. The moduli q and p are chosen as described in the text of Section 5.4.

n	ℓ $B = 2$	ℓ $B = 4$	ℓ $B = 6$
2^{15}	3	2	2
2^{14}	3	3	2
2^{13}	3	2	2
2^{12}	3	2	2
2^{11}	3	2	2
2^{10}	3	2	2

Table 4 Possible parameter sets n, ℓ, B, for our Regev-type scheme with $X = P_{B/2}$, assuming a uniform ternary secret. The moduli q and p are chosen as described in the text of Section 5.4.

Algorithm	Parameters $(\log n, \log q)$			
	$(11, 48)$	$(12, 101)$	$(13, 207)$	$(14, 421)$
`KeyGen`	0.04	0.09	0.21	0.72
`RelinKeyGen`	1.21	6.02	33.98	267.62
`Encrypt`	0.04	0.09	0.26	1.04
`Decrypt`	0.01	0.03	0.11	0.48
`Add`	0.00	0.00	0.00	0.01
`Mult`	1.09	6.85	46.61	409.89

Table 5 Running time in seconds for operations in our Python implementation of our LPR-type scheme. We set plaintext modulus $t = 3$. We set $r/q = q/p = 16$.

Scheme	Multiplication depth														
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
BFV [58]	18	74	232	308	384	968	1132	1292	1452	1612	3720	4064	4400	4736	5072
LPR-like	15	68	222	298	374	948	1108	1268	1432	1592	3680	4016	4352	4688	5032
Regev($\ell = 3$)	17	72	228	304	380	964	1124	1284	1444	1604	3704	4048	4384	4720	5056
Regev(LHL)	18	74	234	310	388	976	1136	1296	1460	1620	3736	4072	4416	4752	5088

Table 6 Minimal ciphertext size in kilobytes (kB) for plaintext modulus $t = 3$ and multiplicative depth for which decryption is guaranteed to work.

Scheme	Multiplicative depth														
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
BFV [58]	24	99	308	410	1064	1276	1488	1700	3968	4408	4848	5288	5728	6168	6608
LPR-like	50	99	306	408	1060	1272	1484	3512	3952	4392	4832	5272	5712	6152	6592
Regev($\ell = 3$)	23.5	97	304	406	1056	1268	1480	1692	3952	4392	4832	5272	5712	6152	6592
Regev(LHL)	24.5	100	312	414	1072	1284	1496	3544	3984	4424	4864	5312	5752	6192	6632

Table 7 Minimal ciphertext size in kilobytes (kB) for plaintext modulus $t = 256$ and multiplicative depth for which decryption is guaranteed to work.

Scheme	Multiplicative depth														
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
BFV [58]	24	99	310	412	1068	1280	1492	3536	3976	4416	4856	5296	5736	6176	6616
LPR-like	50	99	306	408	1060	1272	1484	3512	3952	4392	4832	5272	5712	6152	6592
Regev($\ell = 3$)	23.5	98	306	408	1060	1272	1484	3520	3960	4400	4840	5280	5720	6160	6600
Regev(LHL)	24.5	100	312	414	1072	1284	1496	3544	3984	4432	4872	5312	5752	6192	6632

Table 8 Minimal ciphertext size in kilobytes (kB) for plaintext modulus $t = 257$ and multiplicative depth for which decryption is guaranteed to work.

Scheme	Multiplicative depth														
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
BFV [58]	66	270	404	1108	1384	1660	3984	4552	5120	5688	6256	-	-	-	-
LPR-like	74	284	860	1136	1412	3464	4032	4600	5168	5736	6304	-	-	-	-
Regev($\ell = 3$)	64	266	400	1100	1376	1652	3968	4536	5104	5672	6240	-	-	-	-
Regev(LHL)	67	272	408	1116	1392	1668	4000	4568	5144	5712	6280	-	-	-	-

Table 9 Minimal ciphertext size in kilobytes (kB) for plaintext modulus $t = 2^{16}$ and multiplicative depth for which decryption is guaranteed to work.

Scheme	Multiplicative depth														
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
BFV [58]	66	270	406	1132	1404	1664	4024	4592	5120	5688	6296	-	-	-	-
LPR-like	74	284	860	1136	1412	3464	4032	4600	5168	5736	6304	-	-	-	-
Regev($\ell = 3$)	64	266	400	1100	1376	1652	3968	4536	5104	5672	6240	-	-	-	-
Regev(LHL)	67	272	408	1116	1392	1668	4000	4568	5144	5712	6280	-	-	-	-

Table 10 Minimal ciphertext size in kilobytes (kB) for plaintext modulus $t = 2^{16} + 1$ and multiplicative depth for which decryption is guaranteed to work.

Scheme	Multiplicative depth														
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
BFV [58]	98	398	1216	1620	4128	4952	5776	6600	-	-	-	-	-	-	-
LPR-like	246	904	1308	3480	4304	5128	5952	6776	-	-	-	-	-	-	-
Regev($\ell = 3$)	96	394	1208	1612	4112	4936	5760	6584	-	-	-	-	-	-	-
Regev(LHL)	99	402	1224	1628	4144	4976	5800	6624	-	-	-	-	-	-	-

Table 11 Minimal ciphertext size in kilobytes (kB) for plaintext modulus $t = 2^{32}$ and multiplicative depth for which decryption is guaranteed to work.

Scheme	Multiplicative depth														
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
BFV [58]	244	398	1280	3472	4128	5112	5944	6656	-	-	-	-	-	-	-
LPR-like	246	904	1308	3480	4304	5128	5952	6776	-	-	-	-	-	-	-
Regev($\ell = 3$)	96	394	1208	1612	4112	4936	5760	6584	-	-	-	-	-	-	-
Regev(LHL)	99	402	1224	1628	4144	4976	5800	6624	-	-	-	-	-	-	-

Table 12 Minimal ciphertext size in kilobytes (kB) for plaintext modulus $t = 2^{32} + 1$ and multiplicative depth for which decryption is guaranteed to work.

Appendix A Omitted proofs

Proof of Lemma 2.

Proof. It is clear that the uniform distribution on $R_{\alpha q} \times R_{\alpha p}$ is mapped to the uniform distribution on $R_q \times R_p$. Next, given a Ring-LWR$_{n,\alpha q,\alpha p}$ sample (a, b), with $b := \lfloor a \cdot s \rceil_{\alpha q,\alpha p}$ and $(\bar{a}, \bar{b}) := \pi((a, b))$, we need to show that $\bar{b} = \lfloor \bar{a} \cdot s \rceil_{q,p}$, as in a Ring-LWR$_{n,q,p}$ sample. Notice that $b = \lfloor as \cdot \alpha p/\alpha q \rceil \bmod \alpha p = \lfloor as \cdot p/q \rceil + i \cdot \alpha p$ and $\lfloor \bar{a} \cdot s \rceil_{q,p} = \lfloor (a + j \cdot q) \cdot s \cdot p/q \rceil \bmod p$ which is equal to $\lfloor as \cdot p/q \rceil + j \cdot s \cdot p$. Therefore $\lfloor \bar{a} \cdot s \rceil_{q,p} = b \bmod p$, which is the same as $\bar{b}$. □

Proof of Lemma 3.

Proof. Let $(r_1, \ldots, r_\ell) \neq (r'_1, \ldots, r'_\ell) \in E^\ell_{p,q}$. Notice that there must exist $i \in \{1, \ldots, \ell\}$ such that $r_i \neq r'_i$. Since the set $E_{p,q}$ is exceptional and $r_i - r'_i$ is nonzero, $r_i - r'_i$ is invertible both modulo p and q. Using this, we have that

$$
\begin{aligned}
P &:= Pr_{h \leftarrow \mathcal{H}}[h(r_1, \ldots, r_k) = h(r'_1, \ldots, r'_k)] \\
&= Pr_{(v_k, w_k)_{k \in [\ell]} \leftarrow R_q^\ell \times R_p^\ell}[(\sum_{k=1}^{\ell} r_k v_k, \sum_{k=1}^{\ell} r_k w_k) = (\sum_{k=1}^{\ell} r'_k v_k, \sum_{k=1}^{\ell} r'_k w_k)] \\
&= Pr_{(v_k, w_k)_{k \in [\ell]} \leftarrow R_q^\ell \times R_p^\ell}[\sum_{k=1}^{\ell} (r_k - r'_k) v_k = 0 \wedge \sum_{k=1}^{\ell} (r_k - r'_k) w_k = 0] \\
&= Pr_{(v_k, w_k)_{k \in [\ell]} \leftarrow R_q^\ell \times R_p^\ell}[v_i = -(r_i - r'_i)^{-1} \sum_{k=1, k \neq i}^{\ell} (r_k - r'_k) v_k \\
&\wedge w_i = -(r_i - r'_i)^{-1} \sum_{k=1, k \neq i}^{\ell} (r_k - r'_k) w_k] \\
&= \frac{1}{|R_q \times R_p|}.
\end{aligned}
$$

□

Proof of Corollary 1.

Proof. Take r and r' two distinct elements from $E_{p,q}$. Notice that $r - r'$ is therefore bounded by B, and by the choice of B, taking its representatives via Chinese Remainder Theorem does not change it. By the primality of p_1,..., p_s, the difference $r - r'$ is invertible mod p_1,..., p_s. Therefore, by Chinese Remainder Theorem, it is also invertible mod p. Similarly, by the primality of q_1,..., q_t, the difference $r - r'$ is invertible mod q. Therefore, the difference $r - r'$ is invertible both mod p and mod q. The conclusion now follows by applying Theorem 3 and Theorem 4. □

Proof of Corollary 2.

Proof. Take r and r' two distinct elements from $E_{p,q}$. Notice that $r - r'$ is a nonzero polynomial with coefficients bounded (in absolute value) by B, and since B is of course

less than minimum of all the primes, taking its representatives via Chinese Remainder Theorem does not change it. By the choice of B and by [55, Corollary 1.2], it follows that $r - r'$ is invertible mod p_i and mod q_j, for any $1 \leq i \leq s$ and $1 \leq j \leq t$. Therefore, by Chinese Remainder Theorem, $r - r'$ is invertible both mod p and mod q. The conclusion now follows by applying Theorem 3 and Theorem 4 and the fact that $|E_{p,q}| = (B+1)^n$. □

Appendix B Bound for multiplication noise growth

In this section we analyse the noise N_{mult} from the multiplication operation, for both the LPR-type and Regev-type schemes. To determine the noise N_{mult}, we first note that by definition of noise in the input ciphertexts $\mathtt{ct}$ and $\mathtt{ct}'$ we have:

$$\frac{t}{p}\left(-\frac{p}{q}\cdot \mathtt{ct}_0 \cdot s + \mathtt{ct}_1\right) = m + N + tG\,,$$
$$\frac{t}{p}\left(-\frac{p}{q}\cdot \mathtt{ct}'_0 \cdot s + \mathtt{ct}'_1\right) = m' + N' + tG'\,.$$

Multiplying these two equations, and expanding the left hand side, we obtain:

$$\frac{t^2}{p^2}\left(-\frac{p}{q}\mathtt{ct}_0 s + \mathtt{ct}_1\right)\left(-\frac{p}{q}\mathtt{ct}'_0 s + \mathtt{ct}'_1\right) = (m+N+tG)(m'+N'+tG')$$
$$\frac{t^2}{q^2}\mathtt{ct}_0\mathtt{ct}'_0 s^2 - \frac{t^2}{pq}(\mathtt{ct}_0\mathtt{ct}'_1 + \mathtt{ct}'_0\mathtt{ct}_1)s + \frac{t^2}{p^2}\mathtt{ct}_1\mathtt{ct}'_1 = (m+N+tG)(m'+N'+tG')\,.$$

Moreover, we note that, for some $H = Gm' + G'm + tGG'$,

$$(m+N+tG)(m'+N'+tG') = mm' + Nm' + N'm + NN' + tGN' + tG'N + tH\,.$$

Next, we make explicit the modular reduction in the intermediate (c_2, c_1, c_0) as follows, where A_0, A_1 and A_2 are integer polynomials.

$$c_2 = \frac{t}{p}\mathtt{ct}_0\mathtt{ct}'_0 + \epsilon_2 + \frac{q^2}{p}A_2$$
$$c_1 = \frac{t}{p}(\mathtt{ct}_0\mathtt{ct}'_1 + \mathtt{ct}'_0\mathtt{ct}_1) + \epsilon_1 + qA_1$$
$$c_0 = \frac{t}{p}\mathtt{ct}_1\mathtt{ct}'_1 + \epsilon_0 + pA_0\,.$$

By the definition of noise in (c_2, c_1, c_0), we see that, for $A = A_2 s^2 - A_1 s + A_0$,

$$\frac{t}{p}\left(\left(\frac{p}{q}\right)^2 c_2 s^2 - \frac{p}{q}c_1 s + c_0\right) = \frac{t}{p}\left(\frac{p}{q}\right)^2\left(\frac{t}{p}\mathtt{ct}_0\mathtt{ct}'_0 + \epsilon_2 + \frac{q^2}{p}A_2\right)s^2$$

$$
\begin{aligned}
&- \frac{t}{p}\frac{p}{q}\left(\frac{t}{p}(\mathtt{ct}_0\mathtt{ct}_1' + \mathtt{ct}_0'\mathtt{ct}_1) + \epsilon_1 + qA_1\right)s \\
&+ \frac{t}{p}\left(\frac{t}{p}\mathtt{ct}_1\mathtt{ct}_1' + \epsilon_0 + pA_0\right) \\
&= \frac{t^2}{q^2}\mathtt{ct}_0\mathtt{ct}_0's^2 - \frac{t^2}{pq}(\mathtt{ct}_0\mathtt{ct}_1' + \mathtt{ct}_0'\mathtt{ct}_1)s + \frac{t^2}{p^2}\mathtt{ct}_1\mathtt{ct}_1' \\
&+ \frac{tp}{q^2}\epsilon_2 s^2 - \frac{t}{q}\epsilon_1 s + \frac{t}{p}\epsilon_0 + tA \\
&= mm' + Nm' + N'm + NN' + \frac{tp}{q^2}\epsilon_2 s^2 \\
&- \frac{t}{q}\epsilon_1 s + \frac{t}{p}\epsilon_0 + tGN' + tG'N + t(A+H)\,,
\end{aligned}
$$

so that

$$
N_{\text{mult}} = NN' + (m' + tG')N + (m + tG)N' + \frac{tp}{q^2}\epsilon_2 s^2 - \frac{t}{q}\epsilon_1 s + \frac{t}{p}\epsilon_0\,.
$$

In particular, the resulting expression for N_{mult} is the same for both schemes. Moreover, we can compare this with the corresponding invariant noise expression presented in [84] for the multiplication noise in the BFV scheme. It is clear that the noise growth in multiplication of either the LPR-type or Regev-type scheme is entirely analogous to the noise growth in BFV.

We show that with high probability, the canonical norm of N_{mult} can be bounded by:

$$
\begin{aligned}
\|N_{\text{mult}}\|^{\text{can}} \leq \|N\|^{\text{can}} \cdot \|N'\|^{\text{can}} + t \cdot \sqrt{2n^2 + 3n} \cdot \left(\|N\|^{\text{can}} + \|N'\|^{\text{can}}\right) \\
+ \frac{tp}{q^2} \cdot 4\sqrt{3} \cdot n^{3/2} + t \cdot \sqrt{\frac{2n^2}{q^2} + \frac{3n}{p^2}}\,.
\end{aligned}
$$

To see this, we rewrite the expression of N_{mult} as follows

$$
\begin{aligned}
N_{\text{mult}} &= NN' + (m' + tG')N + (m + tG)N' + \frac{tp}{q^2}\epsilon_2 s^2 - \frac{t}{q}\epsilon_1 s + \frac{t}{p}\epsilon_0 \\
&= NN' + (-\frac{t}{q}\mathtt{ct}_0's + \frac{t}{p}\mathtt{ct}_1' - N')N + (-\frac{t}{q}\mathtt{ct}_0 s + \frac{t}{p}\mathtt{ct}_1 - N)N' + \frac{tp}{q^2}\epsilon_2 s^2 - \frac{t}{q}\epsilon_1 s + \frac{t}{p}\epsilon_0 \\
&= -NN' + (-\frac{t}{q}\mathtt{ct}_0's + \frac{t}{p}\mathtt{ct}_1')N + (-\frac{t}{q}\mathtt{ct}_0 s + \frac{t}{p}\mathtt{ct}_1)N' + \frac{tp}{q^2}\epsilon_2 s^2 - \frac{t}{q}\epsilon_1 s + \frac{t}{p}\epsilon_0,
\end{aligned}
$$

which implies that:

$$
\|N_{\text{mult}}\|^{\text{can}} \leq \|N\|^{\text{can}}\,\|N'\|^{\text{can}} + \left\|-\frac{t}{q}\mathtt{ct}_0's + \frac{t}{p}\mathtt{ct}_1'\right\|^{\text{can}} \|N\|^{\text{can}} + \left\|-\frac{t}{q}\mathtt{ct}_0 s + \frac{t}{p}\mathtt{ct}_1\right\|^{\text{can}} \|N'\|^{\text{can}}
$$

$$+ \left\| \frac{tp}{q^2}\epsilon_2 s^2 \right\|^{\text{can}} + \left\| -\frac{t}{q}\epsilon_1 s + \frac{t}{p}\epsilon_0 \right\|^{\text{can}} .$$

We consider the terms in turn, and develop bounds using Section 2.5. We have $\left\| \frac{tp}{q^2}\epsilon_2 s^2 \right\|^{\text{can}} \leq \left\| \frac{tp}{q^2}\epsilon_2 s \right\|^{\text{can}} \cdot \|s\|^{\text{can}}$ and with high probability, this can be further bounded by

$$6\sqrt{n}\frac{tp}{q^2}\sqrt{nV_{\epsilon_2}V_s} \cdot 6\sqrt{n}\sqrt{V_s} = \frac{tpn}{q^2} \cdot 6\sqrt{\frac{1}{12}\cdot\frac{2}{3}} \cdot 6\sqrt{n}\sqrt{\frac{2}{3}} = \frac{4\sqrt{3}n^{3/2}tp}{q^2}.$$

Let $A := -\frac{t}{q}\epsilon_1 s + \frac{t}{p}\epsilon_0$. Then we can bound $\left\| -\frac{t}{q}\epsilon_1 s + \frac{t}{p}\epsilon_0 \right\|^{\text{can}} \leq 6\sqrt{n}\sqrt{V_A}$, where

$$V_A = \frac{t^2}{q^2}nV_\epsilon V_s + \frac{t^2}{p^2}V_\epsilon = t^2\left(\frac{n}{18q^2} + \frac{1}{12p^2}\right).$$

Let $B := -\frac{t}{q}\mathtt{ct}_0 s + \frac{t}{p}\mathtt{ct}_1$. Then we can bound $\left\| -\frac{t}{q}\mathtt{ct}_0 s + \frac{t}{p}\mathtt{ct}_1 \right\|^{\text{can}} \leq 6\sqrt{n}\sqrt{V_B}$, where

$$V_B = \frac{t^2}{q^2}nV_{\mathtt{ct}_0}V_s + \frac{t^2}{p^2}V_{\mathtt{ct}_1} = \frac{t^2}{q^2}n \cdot \frac{q^2}{12}\cdot\frac{2}{3} + \frac{t^2}{p^2}\cdot\frac{p^2}{12} = t^2\left(\frac{n}{18} + \frac{1}{12}\right).$$

By symmetry, the term $\left\| -\frac{t}{q}\mathtt{ct}'_0 s + \frac{t}{p}\mathtt{ct}'_1 \right\|^{\text{can}}$ can be bounded in the same way. By substituting the established bounds and rearranging the terms, we obtain the desired bound.

Appendix C Relinearization noise growth

In this section we analyse the noise N_{relin} from the relinearization operation, for both the LPR-type and Regev-type schemes. Let (c_2, c_1, c_0) be an intermediate obtained from multiplication that encrypts m with noise N. Let $(\mathtt{ct}_{0,\text{relin}}, \mathtt{ct}_{1,\text{relin}})$ defined either as in Section 3. Then we have

$$\begin{aligned}
&\frac{t}{p}\left(-\frac{p}{q}\cdot\mathtt{ct}_{0,\text{relin}}s + \mathtt{ct}_{1,\text{relin}}\right) \\
&\quad= \frac{t}{p}\left(-\frac{p}{q}\left(c_1 + \sum_{j=0}^{k} c_2^{(j)}a_j + qA\right)s\right) + \frac{t}{p}\left(c_0 + \sum_{j=0}^{k} c_2^{(j)}b_j + pB\right) \\
&\quad= -\frac{t}{q}\sum_{j=0}^{k} c_2^{(j)}a_j s + \frac{t}{p}\sum_{j=0}^{k} c_2^{(j)}b_j - \frac{t}{q}c_1 s + \frac{t}{p}c_0 + t(B - As) \\
&\quad= -\frac{t}{q}\sum_{j=0}^{k} c_2^{(j)}a_j s + \frac{t}{p}\sum_{j=0}^{k} c_2^{(j)}\left(\left\lceil a_j s \right\rfloor_{q,p} + \left\lceil \frac{p^2}{q^2}\omega^j s^2 \right\rfloor\right)
\end{aligned}$$

$$
\begin{aligned}
&\quad -\frac{t}{q}c_1 s + \frac{t}{p}c_0 + t(B - As) \\
&= -\frac{t}{q}\sum_{j=0}^{k} c_2^{(j)} a_j s + \frac{t}{p}\sum_{j=0}^{k} c_2^{(j)} \left(\frac{p}{q}(a_j s) + \epsilon_1 + \frac{p^2}{q^2}\omega^j s^2 + \epsilon_2\right) \\
&\quad -\frac{t}{q}c_1 s + \frac{t}{p}c_0 + t(B - As) \\
&= \frac{t}{p}\frac{p^2}{q^2}s^2 \sum_{j=0}^{k} c_2^{(j)}\omega^j - \frac{t}{q}c_1 s + \frac{t}{p}c_0 + \frac{t}{p}\sum_{j=0}^{k} c_2^{(j)}(\epsilon_1 + \epsilon_2) + t(B - As) \\
&= \frac{t}{p}\left(\frac{p^2}{q^2}c_2 s^2 - \frac{p}{q}c_1 s + c_0\right) + \frac{t}{p}\sum_{j=0}^{k} c_2^{(j)}(\epsilon_1 + \epsilon_2) + t(B - As) \\
&= m + N + \frac{t}{p}\sum_{j=0}^{k} c_2^{(j)}(\epsilon_1 + \epsilon_2) + t(B - As + G)\,.
\end{aligned}
$$

Thus $\mathtt{ct}_{\text{relin}}$ encrypts m with noise $N_{\text{relin}} = N + \frac{t}{p}\sum_{j=0}^{k} c_2^{(j)}(\epsilon_1 + \epsilon_2)$, with ϵ_1, ϵ_2 polynomials with coefficients in $(-\frac{1}{2}, \frac{1}{2}]$ and $c_2^{(j)}$ polynomials with coefficients in $\{-\frac{\omega}{2}, \ldots, \frac{\omega}{2}\}$. Using Section 2.5, the variance of the additive term is

$$
\begin{aligned}
V_{\text{relin}} &= \frac{t^2}{p^2} \cdot (k+1) \cdot n \cdot V_{c_2^{(j)}} \cdot (V_{\epsilon_1} + V_{\epsilon_2}) \\
&= \frac{t^2}{p^2} \cdot (k+1) \cdot n \cdot \frac{\omega^2}{12} \cdot \frac{1}{6}
\end{aligned}
$$

With high probability, using Section 2.5, we have the bound $\|N_{\text{relin}}\|^{\text{can}} \leq \|N\|^{\text{can}} + \frac{tn\omega}{p} \cdot \sqrt{\frac{k+1}{2}}$, which can be seen as follows:

$$
\|N_{\text{relin}}\|^{\text{can}} \leq \|N\|^{\text{can}} + \frac{t}{p}\left\|\sum_{j=0}^{k} c_2^{(j)}(\epsilon_1 + \epsilon_2)\right\|^{\text{can}} \leq \|N\|^{\text{can}} + \frac{t}{p} \cdot 6n\sqrt{(k+1) \cdot \frac{\omega^2}{12} \cdot \frac{1}{6}}\,.
$$

Notice that in the case $\omega = 2$, the polynomials $c_2^{(j)}$ from the decomposition have coefficients 0 or 1, and hence by using Section 2.5, with high probability $\|N_{\text{relin}}\|^{\text{can}} \leq \|N\|^{\text{can}} + \frac{tn}{p} \cdot \sqrt{\frac{3(k+1)}{2}}$, as

$$
\|N_{\text{relin}}\|^{\text{can}} \leq \|N\|^{\text{can}} + \frac{t}{p}\left\|\sum_{j=0}^{k} c_2^{(j)}(\epsilon_1 + \epsilon_2)\right\|^{\text{can}} \leq \|N\|^{\text{can}} + \frac{t}{p} \cdot 6n\sqrt{(k+1) \cdot \frac{1}{4} \cdot \frac{1}{6}}\,.
$$

Appendix D Modulus switching

Defining a modulus switch method is more challenging for LWR-based schemes such as ours due to the two ciphertext moduli q and p at each level, as opposed to the single ciphertext modulus q of comparable LWE-based schemes such as BGV [3] and BFV [6]. In this section, we show that it can be done. We present a modulus switching algorithm that is compatible with either the LPR-type or Regev-type scheme. This is possible as the schemes share the same decryption equation and ciphertext space $R_q \times R_p$. We describe modulus switching from a ciphertext $(\mathtt{ct}_0, \mathtt{ct}_1)$, that encrypts m with noise N at level i to a ciphertext $\mathtt{ct}_{\mathrm{mod}} = (\mathtt{ct}_{0,\mathrm{mod}}, \mathtt{ct}_{1,\mathrm{mod}})$ at level $i-1$. The approach is analogous to the BFV modulus switching presented in [58]. We define $\mathtt{ct}_{0,\mathrm{mod}} = \left\lfloor \frac{q_{i-1}}{q_i} \mathtt{ct}_0 \right\rceil \pmod{q_{i-1}}$ and $\mathtt{ct}_{1,\mathrm{mod}} = \left\lfloor \frac{p_{i-1}}{p_i} \mathtt{ct}_1 \right\rceil \pmod{p_{i-1}}$. Then, the ciphertext $(\mathtt{ct}_{0,\mathrm{mod}}, \mathtt{ct}_{1,\mathrm{mod}})$ at level $i-1$ encrypts m with noise $N_{\mathrm{mod}} = N - \frac{t}{q_{i-1}}\epsilon_1 s + \frac{t}{p_{i-1}}\epsilon_0$. To see this, we make the modular reduction in $\mathtt{ct}_{0,\mathrm{mod}}$ and $\mathtt{ct}_{1,\mathrm{mod}}$ explicit and denote the rounding by the addition of some ϵ_i whose coefficients are uniform in $(-\frac{1}{2}, \frac{1}{2}]$, so that, for some integer polynomials A and B, we have

$$\begin{aligned}
\mathtt{ct}_{0,\mathrm{mod}} &= \frac{q_{i-1}}{q_i}\mathtt{ct}_0 + \epsilon_0 + q_{i-1}A \\
\mathtt{ct}_{1,\mathrm{mod}} &= \frac{p_{i-1}}{p_i}\mathtt{ct}_1 + \epsilon_1 + p_{i-1}B\,.
\end{aligned}$$

We then have, for some polynomial $C = B - As$:

$$\begin{aligned}
&\frac{t}{p_{i-1}}\left(-\frac{p_{i-1}}{q_{i-1}}\mathtt{ct}_{0,\mathrm{mod}}s + \mathtt{ct}_{1,\mathrm{mod}}\right) \\
&\quad= -\frac{t}{q_i}\mathtt{ct}_0 s - \frac{t}{q_{i-1}}\epsilon_0 s - Ats + \frac{t}{p_i}\mathtt{ct}_1 + \frac{t}{p_{i-1}}\epsilon_1 + tB \\
&\quad= \frac{t}{p_i}\left(-\frac{p_i}{q_i}\mathtt{ct}_0 s + \mathtt{ct}_1\right) - \frac{t}{q_{i-1}}\epsilon_1 s + \frac{t}{p_{i-1}}\epsilon_0 + tC \\
&\quad= m + N - \frac{t}{q_{i-1}}\epsilon_0 s + \frac{t}{p_{i-1}}\epsilon_1 + t(G + C)\,.
\end{aligned}$$

We can bound N_{mod} with high probability as follows:

$$\begin{aligned}
\|N_{\mathrm{mod}}\|^{\mathrm{can}} &\le \|N\|^{\mathrm{can}} + \left\|-\frac{t}{q_{i-1}}\epsilon_0 s + \frac{t}{p_{i-1}}\epsilon_1\right\|^{\mathrm{can}} \\
&\le \|N\|^{\mathrm{can}} + 6t\sqrt{n}\sqrt{\frac{n}{18q_{i-1}^2} + \frac{1}{12p_{i-1}^2}} \\
&\le \|N\|^{\mathrm{can}} + t \cdot \sqrt{\frac{2n^2}{q_{i-1}^2} + \frac{3n}{p_{i-1}^2}}\,.
\end{aligned}$$

Appendix E Equivalence between 2DKS and DKS

In this section we give the definition of the Decisional Knapsack Problem in a single ring [54, Definition 1], and show that our two-ring version of the problem is equivalent with this problem. Let p be an integer and n a power of two. Let X be a finite subset of the ring $R = \mathbb{Z}[x]/(x^n + 1)$. Recall that $R_p = \mathbb{Z}_p[x]/(x^n + 1)$.

Definition 11 (DKS distribution)**.** *The Decisional Knapsack Problem distribution with parameters n, p, ℓ, X (denoted by the $DKS_{n,p,\ell,X}$ distribution) is the distribution over $R_p^\ell \times R_p$ which outputs a sample $(a_1, \ldots, a_\ell, a)$, where $a_1, \ldots, a_\ell$ are independent and uniformly random in R_p and $a = \sum_{k=1}^{\ell} r_k a_k$, for some independently chosen, uniformly random elements r_k from X.*

Definition 12 (DKS problem [54])**.** *The Decisional Knapsack Problem with parameters n, p, ℓ, X, denoted by $DKS_{n,p,\ell,X}$, is defined as follows: given samples $(a_1, \ldots, a_\ell, a)$ from $R_p^\ell \times R_p$, decide whether they are sampled from the $DKS_{n,p,\ell,X}$ distribution or are uniformly random in $R_p^\ell \times R_p$.*

Theorem 5. *Let p and q be two coprime integers, let ℓ be a positive integer, and let X be a finite subset of R. Then the two-ring Decisional Knapsack Problem $2DKS_{n,p,q,\ell,X}$ and the Decisional Knapsack Problem $DKS_{n,pq,\ell,X}$ are equivalent.*

Proof. Recall the notation $R_m = \mathbb{Z}_m[x]/(x^n + 1)$ for any integer m. Both reductions will make use of an application of the Chinese Remainder Theorem. This theorem states that, due to the choice of p and q, we have an R module isomorphism map φ:

$$R_{pq} \xrightarrow{\varphi} R_p \times R_q,$$

defined as $\varphi(x) = (x \pmod p, x \pmod q)$.

Assume that we are given an efficient algorithm $\mathcal{A}$ for solving 2DKS. We want to construct an efficient algorithm $\mathcal{B}$ for solving DKS, meaning that, given an instance $(a_1, \ldots, a_\ell, a)$, for $a_1, \ldots, a_\ell \leftarrow R_{pq}$, the algorithm $\mathcal{B}$ decides if $a \leftarrow R_{pq}$ or if $a = \sum_{k=1}^{\ell} r_k a_k$, for some $r_k \leftarrow X$. The algorithm $\mathcal{B}$ sets $(w_k, v_k) = \varphi(a_k)$ and $(w, v) = \varphi(a)$. Notice that the tuples (w_k, v_k) are uniform over $R_p \times R_q$, as φ is an isomorphism and a_k's are uniform over R_{pq}. Hence, $\mathcal{B}$ can run $\mathcal{A}$ on input $((w_k, v_k)_k, w, v)$ and return as output whatever $\mathcal{A}$ outputs. We argue now that if $(a_1, \ldots, a_\ell, a)$ is a DKS instance, then $(v_k, w_k)_k, w, v)$ is a 2DKS instance. Indeed, as $a = \sum_{k=1}^{\ell} r_k a_k$, for some $r_k \leftarrow X$, since φ is an R-module map and $X \subseteq R$, we have

$$\varphi(a) = \varphi(\sum_{k=1}^{\ell} r_k a_k) = \sum_{k=1}^{\ell} r_k \varphi(a_k) = \sum_{k=1}^{\ell} r_k (w_k, v_k) = (\sum_{k=1}^{\ell} r_k w_k, \sum_{k=1}^{\ell} r_k v_k),$$

and this proves the claim. If a is uniform over R_{pq}, then $(w, v) = \varphi(a)$ is also uniform over $R_q \times R_p$, as φ is an isomorphism.

Now assume that we are given an efficient algorithm $\mathcal{A}$ for solving DKS. We want to construct an efficient algorithm $\mathcal{B}$ for solving 2DKS, meaning that, given an instance $(w_1, \ldots, w_\ell, v_1, \ldots, v_\ell, w, v)$, for $w_1, \ldots, w_\ell \leftarrow R_p, v_1, \ldots, v_\ell \leftarrow R_q$, to tell if $w \leftarrow R_p$ and $v \leftarrow R_q$ or if $v = \sum_{k=1}^{\ell} r_k v_k$ and $w = \sum_{k=1}^{\ell} r_k w_k$, for some $r_k \leftarrow X$. The algorithm

$\mathcal{B}$ sets $a_k = \varphi^{-1}(w_k, v_k)$ and $a = \varphi^{-1}(w, v)$. Notice that the a_k's are uniform over R_{pq}, as φ is an isomorphism and the tuples (w_k, v_k) are uniform over $R_q \times R_p$. Hence, $\mathcal{B}$ can run $\mathcal{A}$ on input $(a_1, \ldots, a_\ell, a)$ and return as output whatever $\mathcal{A}$ outputs. We argue now that if $((w_k, v_k)_k, w, v)$ is a 2DKS instance, then $(a_1, \ldots, a_\ell, a)$ is a DKS instance. Indeed, as $w = \sum_{k=1}^{\ell} r_k w_k$ and $v = \sum_{k=1}^{\ell} r_k v_k$ for some $r_k \leftarrow X$, since φ is an R-module map and $X \subseteq R$, we have

$$a = \varphi^{-1}(w, v) = \varphi^{-1}(\sum_{k=1}^{\ell} r_k w_k, \sum_{k=1}^{\ell} r_k v_k) = \sum_{k=1}^{\ell} r_k \varphi^{-1}(w_k, v_k) = \sum_{k=1}^{\ell} r_k a_k,$$

and this proves the claim. If (w, v) is uniform over $R_q \times R_p$, then $a = \varphi^{-1}(w, v)$ is also uniform over R_{pq}, as φ is an isomorphism. □

Appendix F On the DKS-LWE equivalence [70]

Theorem 5 shows that our two-ring decisional knapsack problem $\text{2DKS}_{n,p,q,\ell,X}$ (Definition 9) is equivalent to a decisional knapsack problem $\text{DKS}_{n,pq,\ell,X}$ in a single ring, for a finite set X of polynomials from $R = \mathbb{Z}[x]/(x^n + 1)$. This latter problem can be further expressed in terms of linear algebra as in Section 5.2, which resonates with the knapsack terminology of [70]. In fact, it can be shown that the one-ring decisional knapsack problem is equivalent to a corresponding decision LWE problem.

More precisely, when $X = S_{B/2}$, the set of scalars bounded (in absolute value) by $B/2$, we can consider the equivalent knapsack function in the notation of [70] as the function $f_{\mathbf{g}}$ with input distribution the uniform distribution over $\{-B/2, \ldots, B/2\}^\ell$, indexed by $\mathbf{g} \in (\mathbb{Z}_{pq}^n)^\ell$ and defined as $f_{\mathbf{g}}(\mathbf{r}) = \langle \mathbf{g}, \mathbf{r} \rangle$. When $X = P_{B/2}$, the set of polynomials with coefficients bounded (in absolute value) by $B/2$, we can consider the equivalent knapsack function in the notation of [70] as the function $f_{\mathbf{g}}$ with input distribution the uniform distribution over $\{-B/2, \ldots, B/2\}^{\ell n}$, indexed by $\mathbf{g} \in (\mathbb{Z}_{pq}^n)^{\ell n}$ and defined as $f_{\mathbf{g}}(\mathbf{r}) = \langle \mathbf{g}, \mathbf{r} \rangle$. It is shown in [70, Lemmas 4.8, 4.9] that, for such a knapsack function, inverting its output, respectively distinguishing it from uniform, is equivalent to solving the search variant, respectively the decision variant, of the LWE problem. Here we give the full statements, for the sake of completeness. We denote by $U(X)$ the uniform distribution over a set X.

Corollary 3 (Adapted from [70])**.** *For $X = S_{B/2}$, the $\text{DKS}_{n,pq,\ell,X}$ problem is equivalent to the $\text{LWE}_{\ell,\ell-n,U(\{-B/2,\ldots,B/2\}^\ell),q}$ problem.*

Corollary 4 (Adapted from [70])**.** *For $X = P_{B/2}$, the $\text{DKS}_{n,pq,\ell,X}$ problem is equivalent to the $\text{LWE}_{\ell n,\ell n-n,U(\{-B/2,\ldots,B/2\}^{\ell n}),pq}$ problem.*

Corollary 3 shows that for $X = S_{B/2}$ the decisional knapsack problem is equivalent to an LWE instance of modulus pq, secret dimension $\ell - n$, number of samples ℓ, secret distribution as uniform over $\mathbb{Z}_{pq}$ and error distribution as uniform distribution over $\{-B/2, \ldots, B/2\}$. Corollary 4 shows that for $X = P_{B/2}$ the decisional knapsack problem is equivalent to an LWE instance of secret dimension $\ell n - n$, number of samples ℓn, and the same secret and error distribution as before. Hence, for a specific choice of ℓ and B, we can estimate the security of the $\text{DKS}_{n,pq,\ell,X}$ instance by estimating the security of the implied LWE instance, using the Lattice Estimator.

	n	ℓ $B=2$	ℓ $B=4$	ℓ $B=6$
$X = S_{B/2}$	2^{15}	98164	98104	98064
	2^{14}	49022	48962	48922
	2^{13}	24456	24396	24356
	2^{12}	12178	12118	12078
	2^{11}	6044	5984	5944
	2^{10}	2982	2922	2882
$X = P_{B/2}$	2^{15}	3	3	3
	2^{14}	3	3	3
	2^{13}	3	3	3
	2^{12}	3	3	3
	2^{11}	3	3	3
	2^{10}	3	3	3

Table F1 Suggested values of ℓ for our Regev-type scheme, chosen according to the DKS-LWE equivalence [70]. The randomness is set as $X = S_{B/2}$ or $X = P_{B/2}$, for $B \in \{2, 4, 6\}$. The parameters are targeted at 128-bit security.

When attempting to call the Estimator on such LWE instances, we found that setting the secret distribution as uniform modulo pq failed due to the modulus being too large. Hence, we modelled the implied LWE instances in Hermite Normal Form (i.e. with the secret distribution following the error distribution), at the cost of n samples, as in [85]. Table F1 shows for $X = S_{B/2}$ and for fixed choices of $B \in \{2, 4, 6\}$, a choice of $\ell \approx 3n$ that, with n, p and q as in Table 3, leads to an implied LWE instance that is estimated to be 128-bit secure following Section 5.1. Table F1 further shows for $X = P_{B/2}$ and for fixed choices of $B \in \{2, 4, 6\}$, the choice of $\ell = 3$ with n, p and q as in Tables 4 leads to an implied LWE instance that is estimated to be 128-bit secure following Section 5.1.

Appendix G Zooming in on Figure 1

In this section we present zoomed in versions of Figure 1, which illustrates the results of Table 6. Figure G1 presents the results of Table 6 when restricting the multiplicative depth considered to between 1 and 9. Figure G2 presents the results of Table 6 when restricting the multiplicative depth considered to between 11 and 19. Figure G3 presents the results of Table 6 when restricting the multiplicative depth considered to between 21 and 29.

Appendix H Background for RNS

In this section we include details from [52], on RNS building blocks, namely on basis extension and simple scaling. We also discuss their complexity analysis. We show that these tools may actually require fewer floating-point operations than claimed in [52], whenever the input and output RNS bases of the operations have some common elements.

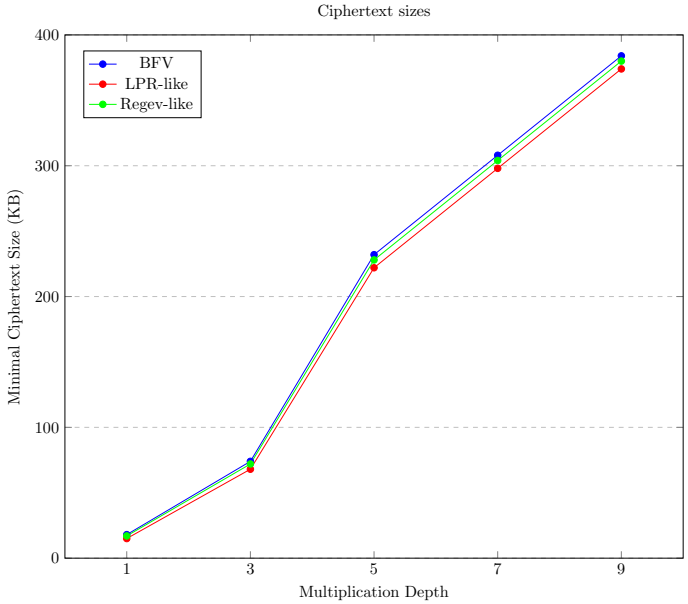

Figure G1 Partial plot of Figure 1. Minimal ciphertext size in kilobytes (kB) for which decryption is guaranteed to work for depth $L \in \{1, 3, 5, 7, 9\}$, when fixing plaintext modulus $t = 3$.

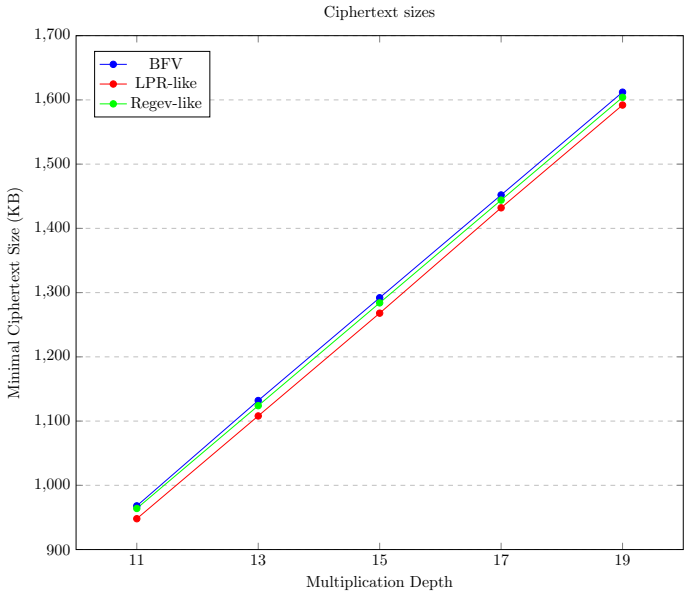

Figure G2 Partial plot of Figure 1. Minimal ciphertext size in kilobytes (kB) for which decryption is guaranteed to work for depth $L \in \{11, 13, 15, 17, 19\}$, when fixing plaintext modulus $t = 3$.

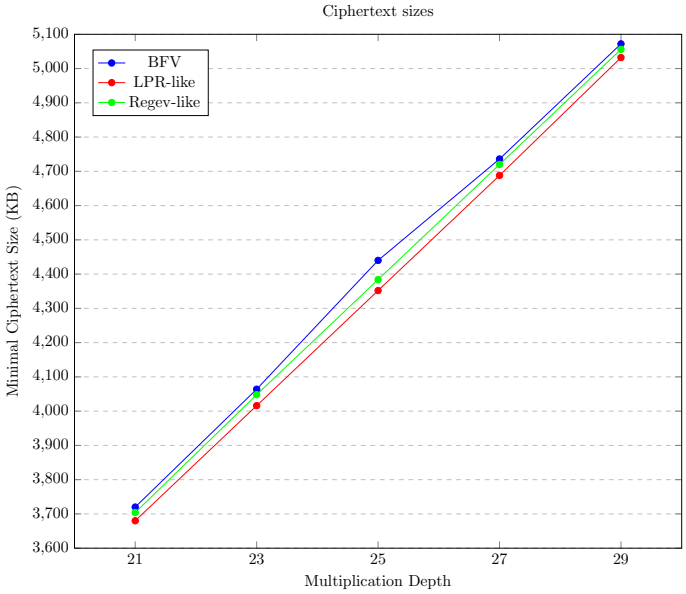

Figure G3 Partial plot of Figure 1. Minimal ciphertext size in kilobytes (kB) for which decryption is guaranteed to work for depth $L \in \{21, 23, 25, 27, 29\}$, when fixing plaintext modulus $t = 3$.

H.1 Basis extension as in [52, Section 2.2]

Suppose we have an integer x modulo q, i.e. $x \in \left[-\frac{q}{2}, \frac{q}{2}\right)$, that is given in the RNS representation $\{x_i\}_{i \in I}$ with respect to the RNS basis $\{q_i\}_{i \in I}$. We want to efficiently compute $[x]_p$, for some modulus p that is coprime with all the q_i's. Denote by $q_i^\star := q/q_i$, where $q = \prod_{i \in I} q_i$ and $\tilde{q}_i := (q_i^\star)^{-1} \bmod q_i$. By CRT we have:

$$x = \left(\sum_{i \in I} [\tilde{q}_i \cdot x_i]_{q_i} \cdot q_i^\star \right) - v \cdot q, \text{ for some } v \in \mathbb{Z}.$$

From this we can easily deduce that $v = \left\lceil \sum_{i \in I} \frac{[x_i \cdot \tilde{q}_i]_{q_i}}{q_i} \right\rfloor$. So we can compute $[x]_p$ as $[x]_p = \left[\sum_{i \in I} y_i \cdot [q_i^\star]_p - v \cdot [q]_p \right]_p$, where $y_i := [x_i \cdot \tilde{q}_i]_{q_i}$.

Complexity analysis.

The values $[q]_p, [q_i^\star]$ and $\tilde{q}_i$ are all pre-computed. Computing v requires $|I|$ single-precision integer multiplications (to compute any y_i) and $|I|$ floating-point divisions followed by one floating-point rounding. Finally, another $|I|+1$ integer multiplications (mod p) are required. In total there are $2|I| + 1$ integer multiplications followed by $|I| + 1$ floating-point operations.

H.2 Simple scaling as in [52, Section 2.3]

Again suppose that we have an integer $x \in \left[-\frac{q}{2}, \frac{q}{2}\right)$, given in the RNS representation $\{x_i\}_{i\in I}$ with respect to the RNS basis $\{q_i\}_{i\in I}$. We want to efficiently compute $\left\lceil \frac{t}{q} \cdot x \right\rfloor \pmod t$ with the result represented in the RNS basis $\{t_l\}_{l\in L}$ of t. Using the notation of $q_i^\star$ and of $\tilde{q}_i$ as above, we have by CRT the following:

$$x = \left(\sum_{i\in I} [\tilde{q}_i \cdot x_i]_{q_i} \cdot q_i^\star \right) - v \cdot q, \text{ for some } v \in \mathbb{Z},$$

which leads to

$$\left\lfloor \frac{t}{q} \cdot x \right\rceil = \left\lfloor \sum_{i\in I} x_i \cdot \frac{t\tilde{q}_i}{q_i} - tv_x \right\rceil.$$

Therefore,

$$\left\lfloor \frac{t}{q} \cdot x \right\rceil = \left\lfloor \sum_{i\in I} x_i \cdot \frac{t\tilde{q}_i}{q_i} \right\rceil \bmod t$$

The rational values $\frac{t\tilde{q}_i}{q_i}$, are known in advance, so they are pre-computed as integral and fractional parts: $\frac{t\tilde{q}_i}{q_i} = \omega_i + \theta_i$, with $\omega_i \in \mathbb{Z}$, $\theta_i \in \left[-\frac{1}{2}, \frac{1}{2}\right)$. Notice that in the case when q_i is a prime of t, the rational value becomes an integer one, so $\theta_i = 0$. The pre-computed values that are stored are $\{[\omega_i]_{t_j}, \theta_i\}_{i,j}$. In order to compute $\left[\left\lceil \frac{t}{q} \cdot x \right\rfloor\right]_{t_j}$, we start with $v := \left[\left\lceil \sum_{i\in I} x_i \cdot \theta_i \right\rfloor\right]$ and $w_j := \left[\sum_{i\in I} x_i \cdot [\omega_i]_{t_j}\right]_{t_j}$, then output $[v + w_j]_{t_j}$.

Complexity analysis.

Denote by C the subset of I, $\{i \in I : q_i | t\}$. As for the case when q_i is a prime of t, i.e. $i \in C$, $\frac{t\tilde{q}_i}{q_i} = \omega_i \in \mathbb{Z}$. Therefore, computing v requires only $|I \setminus C|$ floating-point multiplications and one rounding, in total $|I \setminus C| + 1$ floating point operations. Moreover,

$$\begin{aligned} w_j :&= \left[\sum_{i\in I} x_i \cdot [\omega_i]_{t_j} \right]_{t_j} \\ &= \left[\sum_{i\in C} x_i \cdot [\omega_i]_{t_j} \right]_{t_j} + \left[\sum_{i\notin C} x_i \cdot [\omega_i]_{t_j} \right]_{t_j} \\ &= \left[\sum_{i\in C : q_i = t_j} x_i \cdot [\omega_i]_{t_j} \right]_{t_j} + \left[\sum_{i\notin C} x_i \cdot [\omega_i]_{t_j} \right]_{t_j}, \end{aligned}$$

as if $q_i | t$ and $q_i \neq t_j$, then $t_j | \frac{t\tilde{q}_i}{q_i}$, so $[\omega_i]_{t_j} = 0$. Therefore, the first sum contains only one term and hence, computing w_j requires $|I \setminus C| + 1$ single precision integer

multiplications. The total cost is therefore $|J| \cdot (|I \setminus C| + 1)$ single precision integer multiplications.

Appendix I Further RNS details

In this section, we present details of RNS variants of our Regev-type and LPR-type schemes that were omitted from Section 7.

Regev-type secret/public key generation and encryption.

The secret key (with small coefficients) can be interpreted directly mod each q_i. The public key can have each v_k sampled directly mod each q_i, then each $w_k = \lfloor v_k \cdot s \rceil_{q,p}$ can be computed modulo the q_j using simple scaling. For encryption, if the r_k are small enough then $r_k \mod q_i = r_k \mod p_j$ for all i and j. The RNS components of the scaled message can be computed as $\Delta_p m \mod p_j$. Encryption can then be implemented RNS-component-wise as it is simply a linear operation in each ciphertext part: $\mathtt{ct}_0 = \sum_{k=1}^{\ell} r_k v_k \pmod q$ and $\mathtt{ct}_1 = \Delta_p m + \sum_{k=1}^{\ell} r_k w_k \pmod p$.

LPR-type secret/public key generation and encryption.

The secret key (with small coefficients) can be interpreted directly mod each r_ι, where r_ι's are the components of $r = q^2/p$ (c.f. Theorem 1, Section 5.3). For the public key, a can be sampled directly mod each r_ι, then $\lfloor a \cdot s \rceil_{r,q}$ can be computed modulo the q_j using simple scaling. For encryption, the random element u (with small coefficients) can be interpreted directly mod each r_ι and q_i, for all ι and i. Interpreting u as mod r, we can compute $\mathtt{pk}_0 u$ RNS-componentwise mod r and then compute $\mathtt{ct}_0 = \lfloor \mathtt{pk}_0 u \rceil_{r,q} \pmod q$ using simple scaling. Interpreting u as mod q, we can compute $\mathtt{pk}_1 u$ RNS-componentwise mod q and then compute $\lfloor \mathtt{pk}_1 u \rceil_{q,p} \pmod p$ using simple scaling. The RNS components of the scaled message can be computed as $\Delta_p m \mod p_j$. The final addition operation to obtain $\mathtt{ct}_1 = \Delta_p m + \lfloor \mathtt{pk}_1 u \rceil_{q,p} \pmod p$ can be computed RNS-component-wise mod p as it is simply a linear operation.

Correctness of decryption.

The only source for possible errors comes from using floating-point numbers. Because all the values $\theta_{i,q}$ and $\theta_{j,p}$ live in $(-1/2, 1/2]$, for $i \in I$ and $j \in J$, we can approximate them as $\theta_{i,q} = \tilde{\theta}_{i,q} + \epsilon_{i,q}$ and $\theta_{j,p} = \tilde{\theta}_{j,p} + \epsilon_{j,p}$, where the approximation errors given by the "double float" IEEE 754 standard satisfy $|\epsilon_{i,q}| < 2^{-53}$ and $|\epsilon_{j,p}| < 2^{-53}$. Besides the usual noise, there is some extra error that might affect decryption, which can be bounded as $\left|\sum_{i \in I} x_i \cdot \epsilon_{i,q} + \sum_{j \in J} y_j \cdot \epsilon_{j,p}\right| \leq 2^{-53} \cdot \left(\sum_{i \in I} q_i/2 + \sum_{j \in J} p_j/2\right)$. For $q = q_0 \cdot p$, for $I = J \cup \{0\}$, where in this work $q_0 = k \leq 16$ and assuming each prime is at most α bits long, an upper bound on the error term is given by $2^{-53} \cdot (|J| \cdot 2^{\alpha} + 8)$. For $|J| = 32$ and $\alpha = 46$ (i.e. $p_j \leq 2^{46}$) we are guaranteed that this extra error is smaller than $1/4$. By making sure that the ciphertext we decrypt has a noise value smaller than $1/4$, decryption always works in RNS.

Relinearisation noise growth.
Consider decrypting the output of relinearisation $(\mathtt{ct}_0, \mathtt{ct}_1)$:

$$
\begin{aligned}
&\left\lceil \frac{t}{p}\left(-\frac{p}{q}\cdot \mathtt{ct}_0 \cdot s + \mathtt{ct}_1\right)\right\rfloor \pmod t \\
&= \left\lceil \frac{t}{p}\left(-\left(\frac{p}{q}\cdot(\widetilde{\mathtt{ct}_1} + c_1)\right)\cdot s + \widetilde{\mathtt{ct}_0} + c_0\right)\right\rfloor \pmod t \\
&= \left\lceil \frac{t}{p}\left(-\frac{p}{q}c_1 s + c_0 - \frac{p}{q}\widetilde{\mathtt{ct}_1} s + \widetilde{\mathtt{ct}_0}\right)\right\rfloor \pmod t \\
&= \left\lceil \frac{t}{p}\left(-\frac{p}{q}c_1 s + c_0 - \frac{p}{q}\sum_{h\in H}\alpha_h c_{2,h} s + \sum_{h\in H}\beta_h c_{2,h}\right)\right\rfloor \pmod t \\
&= \left\lceil \frac{t}{p}\left(-\frac{p}{q}c_1 s + c_0 - \frac{p}{q}\sum_{h\in H}\alpha_h c_{2,h} s + \sum_{h\in H}\left(\left\lceil \alpha_h s\right\rfloor_{q,p} + \left\lceil \frac{p^2}{q^2}\widetilde{Q_h}Q_h^* s^2\right\rfloor\right)c_{2,h}\right)\right\rfloor \pmod t \\
&= \left\lceil \frac{t}{p}\left(-\frac{p}{q}c_1 s + c_0 + \frac{p^2}{q^2}s^2\sum_{h\in H}\widetilde{Q_h}Q_h^* c_{2,h} + \sum_{h\in H}(\epsilon_{1,h}+\epsilon_{2,h})c_{2,h}\right)\right\rfloor \pmod t \\
&= \left\lceil \frac{t}{p}\left(\frac{p^2}{q^2}s^2 c_2 - \frac{p}{q}c_1 s + c_0 + \sum_{h\in H}(\epsilon_{1,h}+\epsilon_{2,h})c_{2,h}\right)\right\rfloor \pmod t\,,
\end{aligned}
$$

where we used that $c_2 = \sum_{h\in H}\widetilde{Q_h}Q_h^* c_{2,h} \mod Q$. Thus, if the ciphertext input to relinearisation has noise N, then the output ciphertext has noise

$$N_{\text{relin}} = N + \frac{t}{p}\sum_{h\in H}(\epsilon_{1,h}+\epsilon_{2,h})c_{2,h}\,.$$

References

[1] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009, pp. 169–178. ACM, ??? (2009). https://doi.org/10.1145/1536414.1536440 . https://doi.org/10.1145/1536414.1536440

[2] Gentry, C., Halevi, S.: Implementing gentry's fully-homomorphic encryption scheme. In: Paterson, K.G. (ed.) Advances in Cryptology – EUROCRYPT 2011, pp. 129–148. Springer, Berlin, Heidelberg (2011)

[3] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. In: Goldwasser, S. (ed.) Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012, pp. 309–325. ACM, ??? (2012). https://doi.org/10.1145/2090236.2090262 . https://doi.org/10.1145/2090236.2090262

[4] Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Ostrovsky, R. (ed.) IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011, pp. 97–106. IEEE Computer Society, ??? (2011). https://doi.org/10.1109/FOCS.2011.12 . https://doi.org/10.1109/FOCS.2011.12

[5] Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-lwe and security for key dependent messages. In: Rogaway, P. (ed.) Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6841, pp. 505–524. Springer, ??? (2011). https://doi.org/10.1007/978-3-642-22792-9_29 . https://doi.org/10.1007/978-3-642-22792-9_29

[6] Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. IACR Cryptol. ePrint Arch., 144 (2012)

[7] Gentry, C., Halevi, S., Smart, N.P.: Better bootstrapping in fully homomorphic encryption. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7293, pp. 1–16. Springer, ??? (2012). https://doi.org/10.1007/978-3-642-30057-8_1 . https://doi.org/10.1007/978-3-642-30057-8_1

[8] Gentry, C., Halevi, S., Smart, N.P.: Fully homomorphic encryption with polylog overhead. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7237, pp. 465–482. Springer, ??? (2012). https://doi.org/10.1007/978-3-642-29011-4_28 . https://doi.org/10.1007/978-3-642-29011-4_28

[9] Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluation of the AES circuit. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7417, pp. 850–867. Springer, ??? (2012). https://doi.org/10.1007/978-3-642-32009-5_49 . https://doi.org/10.1007/978-3-642-32009-5_49

[10] Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., Lokam, S., Micciancio, D., Moody, D., Morrison, T., Sahai, A., Vaikuntanathan, V.: Homomorphic encryption security standard. Technical report, HomomorphicEncryption.org (2018)

[11] HElib. https://github.com/shaih/HElib (2019)

[12] PALISADE v1.0. https://git.njit.edu/palisade/PALISADE. New Jersey Institute

of Technology (NJIT) (2017)

[13] Microsoft SEAL (release 4.1). https://github.com/Microsoft/SEAL. Microsoft Research, Redmond, WA. (2023)

[14] Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8042, pp. 75–92. Springer, ??? (2013). https://doi.org/10.1007/978-3-642-40041-4_5 . https://doi.org/10.1007/978-3-642-40041-4_5

[15] Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 3–33 (2016). https://doi.org/10.1007/978-3-662-53887-6_1 . https://doi.org/10.1007/978-3-662-53887-6_1

[16] Ducas, L., Micciancio, D.: FHEW: bootstrapping homomorphic encryption in less than a second. In: Oswald, E., Fischlin, M. (eds.) 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. Lecture Notes in Computer Science, vol. 9056, pp. 617–640. Springer, ??? (2015). https://doi.org/10.1007/978-3-662-46800-5_24 . https://doi.org/10.1007/978-3-662-46800-5_24

[17] Cheon, J.H., Kim, A., Kim, M., Song, Y.S.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10624, pp. 409–437. Springer, ??? (2017). https://doi.org/10.1007/978-3-319-70694-8_15 . https://doi.org/10.1007/978-3-319-70694-8_15

[18] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005, pp. 84–93. ACM, ??? (2005). https://doi.org/10.1145/1060590.1060603 . https://doi.org/10.1145/1060590.1060603

[19] Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of

Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6110, pp. 1–23. Springer, ??? (2010). https://doi.org/10.1007/978-3-642-13190-5_1 . https://doi.org/10.1007/978-3-642-13190-5_1

[20] Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: Matsui, M. (ed.) Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5912, pp. 617–635. Springer, ??? (2009). https://doi.org/10.1007/978-3-642-10366-7_36 . https://doi.org/10.1007/978-3-642-10366-7_36

[21] Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6110, pp. 553–572. Springer, ??? (2010). https://doi.org/10.1007/978-3-642-13190-5_28 . https://doi.org/10.1007/978-3-642-13190-5_28

[22] Agrawal, S., Boyen, X., Vaikuntanathan, V., Voulgaris, P., Wee, H.: Functional encryption for threshold functions (or fuzzy IBE) from lattices. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7293, pp. 280–297. Springer, ??? (2012). https://doi.org/10.1007/978-3-642-30057-8_17 . https://doi.org/10.1007/978-3-642-30057-8_17

[23] Boyen, X.: Attribute-based functional encryption on lattices. In: Sahai, A. (ed.) Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7785, pp. 122–142. Springer, ??? (2013). https://doi.org/10.1007/978-3-642-36594-2_8 . https://doi.org/10.1007/978-3-642-36594-2_8

[24] Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7237, pp. 719–737. Springer, ??? (2012). https://doi.org/10.1007/978-3-642-29011-4_42 . https://doi.org/10.1007/978-3-642-29011-4_42

[25] Samardzic, N., Feldmann, A., Krastev, A., Devadas, S., Dreslinski, R., Peikert, C., Sanchez, D.: F1: A fast and programmable accelerator for fully homomorphic encryption. In: MICRO-54: 54th Annual IEEE/ACM International Symposium

on Microarchitecture. MICRO '21, pp. 238–252. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3466752.3480070 . https://doi.org/10.1145/3466752.3480070

[26] Kim, S., Kim, J., Kim, M.J., Jung, W., Kim, J., Rhu, M., Ahn, J.H.: BTS: An accelerator for bootstrappable fully homomorphic encryption. In: Proceedings of the 49th Annual International Symposium on Computer Architecture. ISCA '22, pp. 711–725. Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3470496.3527415 . https://doi.org/10.1145/3470496.3527415

[27] Mert, A.C., Aikata, Kwon, S., Shin, Y., Yoo, D., Lee, Y., Roy, S.S.: Medha: Microcoded hardware accelerator for computing on encrypted data. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2023**(1), 463–500 (2023) https://doi.org/10.46586/tches.v2023.i1.463-500

[28] Geelen, R., Van Beirendonck, M., L. Pereira, H.V., Huffman, B., McAuley, T., Selfridge, B., Wagner, D., Dimou, G., Verbauwhede, I., Vercauteren, F., al.: Basalisc: Programmable hardware accelerator for bgv fully homomorphic encryption. IACR Transactions on Cryptographic Hardware and Embedded Systems **2023**(4), 32–57 (2023) https://doi.org/10.46586/tches.v2023.i4.32-57

[29] Bertels, J., Beirendonck, M.V., Turan, F., Verbauwhede, I.: Hardware acceleration of FHEW. In: Jenihhin, M., Kubátová, H., Metens, N., Raik, J., Ahmed, F., Belohoubek, J. (eds.) 26th International Symposium on Design and Diagnostics of Electronic Circuits and Systems, DDECS 2023, Tallinn, Estonia, May 3-5, 2023, pp. 57–60. IEEE, ??? (2023). https://doi.org/10.1109/DDECS57882.2023.10139347 . https://doi.org/10.1109/DDECS57882.2023.10139347

[30] Pont, D.D., Bertels, J., Turan, F., Beirendonck, M.V., Verbauwhede, I.: Hardware Acceleration of the Prime-Factor and Rader NTT for BGV Fully Homomorphic Encryption. Cryptology ePrint Archive, Paper 2024/217. https://eprint.iacr.org/2024/217 (2024). https://eprint.iacr.org/2024/217

[31] Mera, J.M.B., Karmakar, A., Kundu, S., Verbauwhede, I.: Scabbard: a suite of efficient learning with rounding key-encapsulation mechanisms. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2021**(4), 474–509 (2021) https://doi.org/10.46586/tches.v2021.i4.474-509

[32] Bhattacharya, S., García-Morchón, Ó., Rietman, R., Tolhuizen, L.: spkex: An optimized lattice-based key exchange. IACR Cryptol. ePrint Arch., 709 (2017)

[33] Gentry, C., Halevi, S.: Compressible FHE with applications to PIR. In: Hofheinz, D., Rosen, A. (eds.) Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II. Lecture Notes in Computer Science, vol. 11892, pp. 438–464. Springer, ??? (2019). https://doi.org/10.1007/978-3-030-36033-7_17 . https://doi.org/10.

1007/978-3-030-36033-7_17

[34] Chen, H., Laine, K., Rindal, P.: Fast private set intersection from homomorphic encryption. In: Thuraisingham, B., Evans, D., Malkin, T., Xu, D. (eds.) Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, pp. 1243–1255. ACM, ??? (2017). https://doi.org/10.1145/3133956.3134061 . https://doi.org/10.1145/3133956.3134061

[35] Costache, A., Smart, N.P.: Homomorphic encryption without gaussian noise. IACR Cryptol. ePrint Arch., 163 (2017)

[36] Roy, S.S., Vercauteren, F., Verbauwhede, I.: High precision discrete gaussian sampling on fpgas. In: Lange, T., Lauter, K.E., Lisonek, P. (eds.) Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers. Lecture Notes in Computer Science, vol. 8282, pp. 383–401. Springer, ??? (2013). https://doi.org/10.1007/978-3-662-43414-7_19 . https://doi.org/10.1007/978-3-662-43414-7_19

[37] Bruinderink, L.G., Hülsing, A., Lange, T., Yarom, Y.: Flush, gauss, and reload - A cache attack on the BLISS lattice-based signature scheme. In: Gierlichs, B., Poschmann, A.Y. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9813, pp. 323–345. Springer, ??? (2016). https://doi.org/10.1007/978-3-662-53140-2_16 . https://doi.org/10.1007/978-3-662-53140-2_16

[38] Pessl, P.: Analyzing the shuffling side-channel countermeasure for lattice-based signatures. In: Dunkelman, O., Sanadhya, S.K. (eds.) Progress in Cryptology - INDOCRYPT 2016 - 17th International Conference on Cryptology in India, Kolkata, India, December 11-14, 2016, Proceedings. Lecture Notes in Computer Science, vol. 10095, pp. 153–170 (2016). https://doi.org/10.1007/978-3-319-49890-4_9 . https://doi.org/10.1007/978-3-319-49890-4_9

[39] Espitau, T., Fouque, P., Gérard, B., Tibouchi, M.: Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongswan and electromagnetic emanations in microcontrollers. In: Thuraisingham, B., Evans, D., Malkin, T., Xu, D. (eds.) Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, pp. 1857–1874. ACM, ??? (2017). https://doi.org/10.1145/3133956.3134028 . https://doi.org/10.1145/3133956.3134028

[40] Pessl, P., Bruinderink, L.G., Yarom, Y.: To BLISS-B or not to be: Attacking strongswan's implementation of post-quantum signatures. In: Thuraisingham, B., Evans, D., Malkin, T., Xu, D. (eds.) Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, pp. 1843–1855. ACM, ??? (2017). https:

//doi.org/10.1145/3133956.3134023 . https://doi.org/10.1145/3133956.3134023

[41] Kim, S., Hong, S.: Single trace analysis on constant time cdt sampler and its countermeasure. Appl. Sci. **1809** (2018)

[42] Marzougui, S., Wisiol, N., Gersch, P., Krämer, J., Seifert, J.-P.: Machine-learning side-channel attacks on the galactics constant-time implementation of BLISS (2022)

[43] Zhang, S., Lin, X., Yu, Y., Wang, W.: Improved power analysis attacks on falcon. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 14007, pp. 565–595. Springer, ??? (2023). https://doi.org/10.1007/978-3-031-30634-1_19 . https://doi.org/10.1007/978-3-031-30634-1_19

[44] Baan, H., Bhattacharya, S., Cheon, J.H., Fluhrer, S., Garcia-Morchon, O., Gorissen, P., Laarhoven, T., Player, R., Rietman, R., Saarinen, M.-J.O., Son, Y., Tolhuizen, L., Torre-Arce, J.L., Zhang, Z.: Round5: KEM and PKE based on (Ring) Learning with Rounding. round5.org (2020)

[45] Cheon, J.H., Kim, D., Lee, J., Song, Y.: Lizard: Cut off the tail! A practical post-quantum public-key encryption from LWE and LWR. In: Catalano, D., Prisco, R.D. (eds.) Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11035, pp. 160–177. Springer, ??? (2018). https://doi.org/10.1007/978-3-319-98113-0_9 . https://doi.org/10.1007/978-3-319-98113-0_9

[46] D'Anvers, J., Karmakar, A., Roy, S.S., Vercauteren, F.: Saber: Module-lwr based key exchange, cpa-secure encryption and cca-secure KEM. In: Joux, A., Nitaj, A., Rachidi, T. (eds.) Progress in Cryptology - AFRICACRYPT 2018 - 10th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 7-9, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10831, pp. 282–305. Springer, ??? (2018). https://doi.org/10.1007/978-3-319-89339-6_16 . https://doi.org/10.1007/978-3-319-89339-6_16

[47] Tsabary, R.: Fully secure attribute-based encryption for t-cnf from LWE. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11692, pp. 62–85. Springer, ??? (2019). https://doi.org/10.1007/978-3-030-26948-7_3 . https://doi.org/10.1007/978-3-030-26948-7_3

[48] Luo, F., Wang, F., Wang, K., Chen, K.: Fully homomorphic encryption based on the ring learning with rounding problem. IET Information Security **13**(6), 639–648 (2019)

[49] Luo, F., Wang, F., Wang, K., Li, J., Chen, K.: Lwr-based fully homomorphic encryption, revisited. Security and Communication Networks **2018** (2018)

[50] Fang, F., Li, B., Lu, X., Liu, Y., Jia, D., Xue, H.: (deterministic) hierarchical identity-based encryption from learning with rounding over small modulus. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. ASIA CCS '16, pp. 907–912. Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2897845.2897922 . https://doi.org/10.1145/2897845.2897922

[51] Cheon, J.H., Cho, H., Jung, J., Lee, J., Lee, K.: Efficient identity-based encryption from lwr. In: Seo, J.H. (ed.) Information Security and Cryptology – ICISC 2019, pp. 225–241. Springer, Cham (2020)

[52] Halevi, S., Polyakov, Y., Shoup, V.: An improved RNS variant of the BFV homomorphic encryption scheme. In: Matsui, M. (ed.) Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11405, pp. 83–105. Springer, ??? (2019). https://doi.org/10.1007/978-3-030-12612-4_5 . https://doi.org/10.1007/978-3-030-12612-4_5

[53] Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. IACR Cryptol. ePrint Arch., 235 (2003)

[54] Baum, C., Damgård, I., Lyubashevsky, V., Oechsner, S., Peikert, C.: More efficient commitments from structured lattice assumptions. In: Catalano, D., Prisco, R.D. (eds.) Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11035, pp. 368–385. Springer, ??? (2018). https://doi.org/10.1007/978-3-319-98113-0_20 . https://doi.org/10.1007/978-3-319-98113-0_20

[55] Lyubashevsky, V., Seiler, G.: Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I. Lecture Notes in Computer Science, vol. 10820, pp. 204–224. Springer, ??? (2018). https://doi.org/10.1007/978-3-319-78381-9_8 . https://doi.org/10.1007/978-3-319-78381-9_8

[56] Bajard, J., Eynard, J., Hasan, M.A., Zucca, V.: A full RNS variant of FV like somewhat homomorphic encryption schemes. In: Avanzi, R., Heys, H.M. (eds.) Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10532, pp. 423–442. Springer, ??? (2016). https://doi.org/10.1007/978-3-319-69453-5_23 . https://doi.org/10.

1007/978-3-319-69453-5_23

[57] Kim, A., Polyakov, Y., Zucca, V.: Revisiting homomorphic encryption schemes for finite fields. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part III. Lecture Notes in Computer Science, vol. 13092, pp. 608–639. Springer, ??? (2021). https://doi.org/10.1007/978-3-030-92078-4_21 . https://doi.org/10.1007/978-3-030-92078-4_21

[58] Costache, A., Laine, K., Player, R.: Evaluating the effectiveness of heuristic worst-case noise analysis in FHE. In: Chen, L., Li, N., Liang, K., Schneider, S.A. (eds.) Computer Security - ESORICS 2020 - 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14-18, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12309, pp. 546–565. Springer, ??? (2020). https://doi.org/10.1007/978-3-030-59013-0_27 . https://doi.org/10.1007/978-3-030-59013-0_27

[59] Costache, A., Smart, N.P.: Which ring based somewhat homomorphic encryption scheme is best? In: Sako, K. (ed.) Topics in Cryptology - CT-RSA 2016 - The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9610, pp. 325–340. Springer, ??? (2016). https://doi.org/10.1007/978-3-319-29485-8_19 . https://doi.org/10.1007/978-3-319-29485-8_19

[60] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008, pp. 197–206. ACM, ??? (2008). https://doi.org/10.1145/1374376.1374407 . https://doi.org/10.1145/1374376.1374407

[61] Chen, H., Han, K.: Homomorphic lower digits removal and improved FHE bootstrapping. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I. Lecture Notes in Computer Science, vol. 10820, pp. 315–337. Springer, ??? (2018). https://doi.org/10.1007/978-3-319-78381-9_12 . https://doi.org/10.1007/978-3-319-78381-9_12

[62] Halevi, S., Shoup, V.: Bootstrapping for helib. J. Cryptol. **34**(1), 7 (2021) https://doi.org/10.1007/s00145-020-09368-7

[63] Geelen, R., Vercauteren, F.: Bootstrapping for BGV and BFV revisited. J. Cryptol. **36**(2), 12 (2023) https://doi.org/10.1007/s00145-023-09454-6

[64] Geelen, R., Iliashenko, I., Kang, J., Vercauteren, F.: On polynomial functions modulo p^e and faster bootstrapping for homomorphic encryption. In: Hazay, C.,

Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part III. Lecture Notes in Computer Science, vol. 14006, pp. 257–286. Springer, ??? (2023). https://doi.org/10.1007/978-3-031-30620-4_9 . https://doi.org/10.1007/978-3-031-30620-4_9

[65] Okada, H., Player, R., Pohmann, S.: Homomorphic polynomial evaluation using galois structure and applications to BFV bootstrapping. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023, pp. 69–100. Springer, Singapore (2023)

[66] Iliashenko, I.: Optimisations of fully homomorphic encryption. PhD thesis, KU Leuven (2019)

[67] Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-lwe cryptography. IACR Cryptol. ePrint Arch., 293 (2013)

[68] Damgård, I., Pastro, V., Smart, N.P., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7417, pp. 643–662. Springer, ??? (2012). https://doi.org/10.1007/978-3-642-32009-5_38 . https://doi.org/10.1007/978-3-642-32009-5_38

[69] Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In: 43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings, pp. 356–365. IEEE Computer Society, ??? (2002). https://doi.org/10.1109/SFCS.2002.1181960 . https://doi.org/10.1109/SFCS.2002.1181960

[70] Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: Rogaway, P. (ed.) Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6841, pp. 465–484. Springer, ??? (2011). https://doi.org/10.1007/978-3-642-22792-9_26 . https://doi.org/10.1007/978-3-642-22792-9_26

[71] Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. J. Mathematical Cryptology **9**(3), 169–203 (2015)

[72] Albrecht, M.R., Curtis, B.R., Deo, A., Davidson, A., Player, R., Postlethwaite, E.W., Virdia, F., Wunderer, T.: Estimate all the {LWE, NTRU} schemes! In: Catalano, D., Prisco, R.D. (eds.) Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11035, pp. 351–367. Springer, ??? (2018). https://doi.org/10.1007/978-3-319-98113-0_19 . https://doi.org/10.

1007/978-3-319-98113-0_19

[73] Curtis, B.R., Player, R.: On the feasibility and impact of standardising sparse-secret LWE parameter sets for homomorphic encryption. In: Brenner, M., Lepoint, T., Rohloff, K. (eds.) Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography, WAHC@CCS 2019, London, UK, November 11-15, 2019, pp. 1–10. ACM, ??? (2019). https://doi.org/10.1145/3338469.3358940 . https://doi.org/10.1145/3338469.3358940

[74] Bonnetain, X., Bricout, R., Schrottenloher, A., Shen, Y.: Improved classical and quantum algorithms for subset-sum. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12492, pp. 633–666. Springer, ??? (2020). https://doi.org/10.1007/978-3-030-64834-3_22 . https://doi.org/10.1007/978-3-030-64834-3_22

[75] May, A.: How to meet ternary LWE keys. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12826, pp. 701–731. Springer, ??? (2021). https://doi.org/10.1007/978-3-030-84245-1_24 . https://doi.org/10.1007/978-3-030-84245-1_24

[76] Glaser, T., May, A.: How to enumerate LWE keys as narrow as in kyber/dilithium. In: Deng, J., Kolesnikov, V., Schwarzmann, A.A. (eds.) Cryptology and Network Security - 22nd International Conference, CANS 2023, Augusta, GA, USA, October 31 - November 2, 2023, Proceedings. Lecture Notes in Computer Science, vol. 14342, pp. 75–100. Springer, ??? (2023). https://doi.org/10.1007/978-981-99-7563-1_4 . https://doi.org/10.1007/978-981-99-7563-1_4

[77] Shen, Y.: Personal communication (2023)

[78] Lyubashevsky, V.: Digital signatures based on the hardness of ideal lattice problems in all rings. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10032, pp. 196–214 (2016). https://doi.org/10.1007/978-3-662-53890-6_7 . https://doi.org/10.1007/978-3-662-53890-6_7

[79] Bai, S., Galbraith, S.D., Li, L., Sheffield, D.: Improved combinatorial algorithms for the inhomogeneous short integer solution problem. J. Cryptol. **32**(1), 35–83 (2019) https://doi.org/10.1007/s00145-018-9304-1

[80] Boudgoust, K., Sakzad, A., Steinfeld, R.: Vandermonde meets regev: public key encryption schemes based on partial vandermonde problems. Des. Codes

Cryptogr. **90**(8), 1899–1936 (2022) https://doi.org/10.1007/s10623-022-01083-7

[81] Regev, O.: The learning with errors problem (invited survey). In: 2010 IEEE 25th Annual Conference on Computational Complexity, pp. 191–204 (2010). https://doi.org/10.1109/CCC.2010.26

[82] Bos, J.W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In: 2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018, pp. 353–367. IEEE, ??? (2018). https://doi.org/10.1109/EUROSP.2018.00032 . https://doi.org/10.1109/EuroSP.2018.00032

[83] Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2018**(1), 238–268 (2018) https://doi.org/10.13154/TCHES.V2018.I1.238-268

[84] Player, R.: Parameter selection in lattice-based cryptography. PhD thesis, Royal Holloway, University of London (2018)

[85] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5677, pp. 595–618. Springer, ??? (2009). https://doi.org/10.1007/978-3-642-03356-8_35 . https://doi.org/10.1007/978-3-642-03356-8_35