On Algebraic Homomorphic Encryption and its Applications to Doubly-Efficient PIR

Hiroki Okada[2,3], Rachel Player[1], Simon Pohmann[1], and Christian Weinert[1]

[1] Royal Holloway, University of London, UK
[2] KDDI Research, Japan
[3] The University of Tokyo, Japan

Abstract. The Doubly-Efficient Private Information Retrieval (DEPIR) protocol of Lin, Mook, and Wichs (STOC'23) relies on a Homomorphic Encryption (HE) scheme that is *algebraic*, i.e., whose ciphertext space has a ring structure that matches the homomorphic operations. While early HE schemes had this property, modern schemes introduced techniques to manage noise growth. This made the resulting schemes much more efficient, but also destroyed the algebraic property.

In this work, we study algebraic HE with the goal of improving its performance and thereby also the performance of DEPIR. We first prove a lower bound of $2^{\Omega(2^d)}$ for the ciphertext ring size of algebraic HE schemes that can evaluate a circuit of multiplicative depth d, thus demonstrating a gap between optimal algebraic HE and the existing schemes, which have a ciphertext ring size of $2^{O(2^{2d})}$. As we are unable to bridge this gap directly, we instead slightly relax the notion of being algebraic. This allows us to construct a practically more efficient *relaxed-algebraic* HE scheme. We then show that this also leads to a more efficient instantiation and implementation of DEPIR.

We experimentally demonstrate run-time improvements of more than 4x and reduce memory queries by more than 8x compared to prior work. Notably, our relaxed-algebraic HE scheme relies on a new variant of the Ring Learning with Errors (RLWE) problem that we call $\{0,1\}$-CRT RLWE. We give a formal security reduction to standard RLWE, and estimate its concrete security. Both the $\{0,1\}$-CRT RLWE problem and the techniques used for the reduction may be of independent interest.

1 Introduction

Homomorphic Encryption refers to private or public key encryption schemes that additionally allow an untrusted entity to perform computations on the encrypted data – without gaining information about it. Depending on which computations can be performed, we distinguish (among other variants) Additive Homomorphic Encryption (AHE), Somewhat Homomorphic Encryption (SHE), and Fully Homomorphic Encryption (FHE).

In AHE, it is only possible to compute the sum and scaling by integers of encrypted messages. AHE schemes have been known for a long time. In fact, even textbook RSA or ElGamal encryption have additive homomorphic properties (with the caveat that they are not IND-CPA secure).

On the other end of the spectrum, FHE schemes allow arbitrary computations on encrypted data. The first FHE scheme was proposed by Gentry [Gen09], who introduced the bootstrapping technique and applied it to a novel lattice-based encryption scheme that directly allows the homomorphic evaluation of a limited number of additions and multiplications.

Even without bootstrapping, which can be very expensive, being able to perform a limited number of arithmetic operations is already very useful, and schemes that support this are referred to as SHE schemes. For most schemes, including Gentry's initial scheme [Gen09] as well as newer LWE-based schemes like [Bra12; BV11; BGV12; FV12; GSW13], the limit of the number of arithmetic operations stems from the fact that an error or noise term is included during encryption. Operations increase the size of this noise term, and once it exceeds a threshold, decryption will no longer give correct results. It turns out that the noise increase is much larger for multiplications than for additions, although modern schemes usually do much better than earlier ones.

Since it lies close to the core of our work, we want to stress that the arithmetic approach to SHE (namely, implementing "arbitrary computations" using additions and multiplications) is more or less ubiquitous in the literature. While there are important exceptions, in particular the FHEW/TFHE family of schemes [DM15; CGGI20] that (in their most basic form) expose only logic gates, even they internally rely on the GSW scheme [GSW13], which follows an arithmetic approach.

A ring structure on ciphertexts. It turns out that all of the schemes mentioned above originated from schemes that do not only have a ring structure on the plaintext space, but also a matching ring structure on the ciphertext space. "Matching" here means that the encryption and decryption behave somewhat like ring homomorphisms, i.e., except in cases of noise overflow, adding resp. multiplying ciphertexts results in a new ciphertext that encrypts the sum resp. product of the encrypted input plaintexts. Following [LMW23], we call such schemes *Algebraic* Somewhat Homomorphic Encryption (ASHE).

Such an algebraic structure is present in the first scheme by Gentry [Gen09], and also in other early schemes like [vGHV10; BV11]. However, all modern SHE schemes use additional techniques to reduce noise growth, which, as a side effect, destroy this direct arithmetic relationship between plain-and ciphertexts. For example, BGV/BFV-family of schemes [Bra12; BGV12; FV12] all build on [BV11], but add modulus switching (resp. rescaling) and relinearization to achieve better performance and noise growth. The GSW scheme [GSW13] can also be considered to be implicitly based on a scheme whose ciphertext space is a ring of square matrices, i.e., the scheme is algebraic, but only over a non-commutative ring.

Up to recently, ASHE has only been considered an intuitive way to start building SHE, without any intrinsic value. However, it was shown in [LMW23] that a ring structure on ciphertexts enables the use of further, very powerful mathematical techniques, which in turn lead to the first proposal of *Doubly-Efficient* Private Information Retrieval (DEPIR). This enabled solving a problem that was posed by [BIM00] and remained unsolved until then.

ASHE and Doubly-Efficient PIR. Private Information Retrieval (PIR) refers to protocols that allow clients to query a database held by a server without revealing to the server which entries they are interested in. In order to exclude trivial solutions (like asking the server to send the entire database), one usually requires PIR protocols to have server-client communication sublinear in the size of the database N.

Because of its practical importance, a huge range of PIR schemes exists in the literature, many of them being very practical (see, e.g., [MW22; ZPSZ23]). However, the asymptotic computational cost on the server side has remained a sore point. In fact, in a setting without preprocessing, it must be at least $O(N)$ as shown by [BIM00]. This is very undesirable, and so the literature has explored preprocessing variants of PIR. Great progress has been made in the client-dependent preprocessing setting, where the server performs a separate preprocessing phase for each client (see, e.g., [CK20; CHK22; ZLTS23]).

However, what we really would like to have is a client-independent preprocessing phase, which is only run once by the server and its output is then used for all clients. Schemes that achieve sublinear computational online cost in such a setting are called *Doubly-Efficient* PIR (DEPIR), and their existence was only shown recently [LMW23]. The construction of [LMW23] relies on using ASHE in combination with a special datastructure for polynomial evaluation that was discovered earlier by [KU11]. Unfortunately, the scheme of [LMW23], while having perfect asymptotic performance, is completely unpractical due to huge logarithmic factors in server storage and runtime requirements [OPPW24].

Closing the loop, the impractical performance of the DEPIR scheme by [LMW23] is, to a large extent, caused by the fact that no well-performing ASHE scheme is known, and so the authors had to use the somewhat outdated BV scheme [BV11]. It was concluded in [OPPW24] that finding a good ASHE scheme "could immediately lead to practical instantiations of DEPIR".

1.1 Our contributions

Motivated by the significance of ASHE for building DEPIR, and also by its historical role in the development of FHE, we try to find more efficient ASHE schemes and understand their performance limitations better. This lead us to the following five main contributions:

- We propose a framework of “Somewhat Homomorphic Functions” that generalizes ASHE as well as some other tools that are used in cryptography [CL22; CCXY18; EHL+23]. Using this framework, we show a fundamental lower bound on the performance of post-quantum ASHE schemes. The bound implies that they must be significantly slower than modern, non-algebraic SHE schemes. We also show that there remains an asymptotic performance gap between our proven lower bounds and the performance of previously known ASHE schemes.
- Since we are not able to find an ASHE scheme that matches our theoretical bounds, we instead consider relaxing the notion of ASHE. This enables us to construct a new scheme that, at least for practical purposes, comes very close to our theoretical lower bounds for ASHE. Moreover, we show that this new scheme enables us to build a more performant DEPIR. We believe that this indicates that the notion of ASHE is too strict, and it might be worth investigating whether other relaxations of ASHE are still sufficient to construct DEPIR.
- In order to construct this scheme, we rely on a new variant of the Ring-LWE problem [LPR10; SSTX09] that we call “$\{0,1\}$-CRT RLWE”. More concretely, we take the value b of RLWE samples (a,b) from a nonstandard distribution over the set of elements whose image under the double-CRT-isomorphism has coefficients in $\{0,1\}$. To gain confidence in the security of this new assumption, and hence of our new scheme, we provide a reduction from standard RLWE. We believe that this reduction and the novel underlying techniques may be of independent interest. Furthermore, we discuss the concrete security of $\{0,1\}$-CRT RLWE by exploring how known geometric, combinatorial, and algebraic attacks can be applied and improved for our setting. This leads us to believe that $\{0,1\}$-CRT RLWE is not significantly easier than RLWE and helps to derive concrete parameter recommendations.
- As an additional contribution, we show how the polynomial evaluation datastructure used in [LMW23; OPPW24] can be improved using a similar idea as in our modified ASHE scheme. This improves the amount of required storage by a factor of 2^m, where m is the number of variables and in practice $\in \{4,5,6\}$.
- Finally, to demonstrate that our techniques indeed lead to a more performant DEPIR, we provide an implementation that improves upon [OPPW24] by a factor of up to 13× in terms of runtime, while also slightly reducing memory requirements[4].

We now summarize our main technical ideas.

Somewhat Homomorphic Functions We want to capture the notion of a function between rings $R \to S$ that behaves like a ring homomorphism, but only up to a certain number of additions and multiplications. In particular, our definition should be satisfied for the decryption function of an ASHE scheme (when fixing a secret key).

Here we are guided by the notion of a reverse multiplication-friendly embedding, which refers to a pair of linear maps

$$f : S \to R, \ g : R \to S \quad \text{such that } g(f(x)f(y)) = xy$$

This was already generalized to a larger number of multiplications by both [CL22] and [EHL+23] in different ways. Our definition is similar, but we drop the requirement that the involved functions must be linear. In more detail, we say a function $\phi : R \to S$ is (d_+, d_*)-somewhat homomorphic, if there exists $\hat{\phi} : S \to R$ such that

$$\phi(\Gamma(\hat{\phi}(s_1), \ldots, \hat{\phi}(s_m))) = \Gamma(s_1, \ldots, s_m)$$

for all arithmetic circuits[5] $\Gamma(X_1, \ldots, X_m)$ of additive depth $\leq d_+$ and multiplicative depth $\leq d_*$. Using elementary additive combinatorics, we can then prove a lower bound on $\#R$ depending on (d_+, d_*). The formal theorem is given in Thm. 3.2.

We can apply these results on SHFs to ASHE by choosing $\phi = \mathrm{Dec}(\cdot, \mathrm{sk})$ and S, R to be the ciphertext and plaintext ring, respectively. We remark that it is shown in [AGKP14] that a post-quantum ASHE scheme will

[4] Our code is available at https://github.com/FeanorTheElf/ashe-depir

[5] An arithmetic circuit is a computational circuit built from (binary) addition, multiplication and (unary) negation gates. We do not allow constant-value gates, since that allows us to evaluate the circuit on inputs from any ring.

never give an actual ring homomorphism. Excluding "trivial SHFs", i.e., SHFs that are ring homomorphisms, we then get our lower bound (formally stated in Corollary 3.3).

Theorem 1.1 (ASHE lower bound (informal)). *Consider a post-quantum ASHE scheme that can evaluate all arithmetic circuits of additive depth* $2d$ *and multiplicative depth* d*. Then its ciphertext ring is of size* $2^{\Omega(2^d)}$*.*

We believe that the size of the ciphertext ring is a very good metric for performance. In particular, it directly relates to the size of a ciphertext, and also to the runtime of DEPIR when instantiated with the ASHE scheme in question.

The BV scheme [BV11], which was used by [LMW23; OPPW24], has a ciphertext ring with $2^{O(4^d)}$ elements, so is considerably larger than the lower bound of Corollary 3.3. Considering that evaluating a polynomial f requires a circuit of multiplicative depth $\log \deg(f)$, this means that the size of a ciphertext is $O(\deg(f)^2)$, while our bound is only linear $\Omega(\deg(f))$. This leads to the following question:

Open Question 1.2. *Does there exist an ASHE scheme that can evaluate circuits of multiplicative depth* d *and has ciphertext ring size* $2^{O(2^d)}$*?*

Generalizing ASHE It seems possible that Question 1.2 has an affirmative answer, but we are unfortunately unable to construct a suitable scheme. Instead, we give a scheme that (for practical purposes at least) comes close to the lower bound, but is not strictly algebraic since now evaluating a polynomial/circuit on plaintexts corresponds to evaluating a *different* polynomial/circuit on ciphertexts. However, we will show that this relaxation is sufficient to construct DEPIR.

Assume we want to evaluate a polynomial $f(X_1, \ldots, X_m)$ on encrypted plaintexts. Instead of directly evaluating f on the ciphertexts, the idea is to derive (in some way) new polynomials $g_1^{(f)}(X_1, \ldots, X_m), \ldots, g_k^{(f)}(X_1, \ldots, X_m)$, and then combine the evaluations of each $g_i^{(f)}$ on parts of the ciphertexts into the result ciphertext. This is sketched in Figure 1.

This comes in useful when we want to exploit a non-algebraic structure of input ciphertexts. For example, in the BV scheme, ciphertexts are polynomials $R_q[Y]$ over some ring R_q. To make $R_q[Y]$ a finite ring, we usually have to restrict it to polynomials of degree $< d$, say, by switching to $R[Y]/(Y^d - 1)$. Now $R_q[Y]/(Y^d - 1)$ is huge, but the output of $\mathrm{Enc}(\cdot)$ is always a polynomial of degree 1 – the degree only increases through multiplications. Instead of ignoring this fact and evaluating f on elements of $R_q[Y]/(Y^d - 1)$ as done in [LMW23; OPPW24], our first idea is to "decompose" the polynomial f as

$$f(b_1 + Ya_1, \ldots, b_m + Ya_m) = \sum_i Y^i f_i(a_1, \ldots, a_m, b_1, \ldots, b_m)$$

Now every evaluation of f_i only happens in R_q instead of in $R_q[Y]/(Y^d - 1)$. Despite decreasing the size of the ring, on its own, this technique harms the performance of the DEPIR construction. The reason is that now we have to evaluate polynomials f_i with $2m$ indeterminates instead of m, which has a huge impact on the amount preprocessed data that must be stored. One might notice that after this decomposition, f_i is homogeneous as polynomial in $a_1, \ldots, a_m$, so we can decrease the number of variables to $2m - 1$ by normalization.

However, we can modify the underlying scheme to reduce the number of indeterminates. As already proposed in [OPPW24], we can choose the parameters so that we get the double-CRT isomorphism

$$\iota : R_q \xrightarrow{\sim} \bigoplus_{j=1}^{n} \bigoplus_{p \mid q} \mathbb{F}_p \tag{1}$$

Hence, we only need to perform the evaluation $f_i(a_1, \ldots, a_m, b_1, \ldots, b_m)$ for $a_i, b_i \in \mathbb{F}_p$. Our main technical idea is to further modify the BV scheme to end up with $b_i \in \{0, 1\}$, which we will prove can be done securely. With this modification, it is enough to do the DEPIR precomputation once for each $(b_1, \ldots, b_m) \in \{0, 1\}^m$ on the polynomial $f_{i,b_1,\ldots,b_m}(a_1, \ldots, a_m)$ with the b_i already "plugged in". The number of variables is now m (or even $m - 1$, using homogeneity), while still having ciphertext ring R_q.

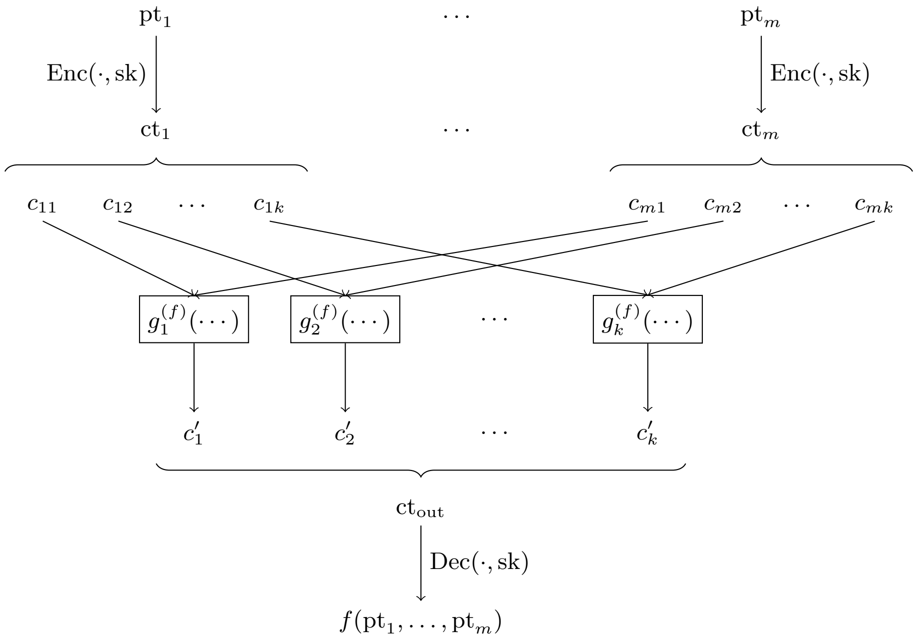

Fig. 1. The idea underlying the evaluation operation of our generalized ASHE scheme. Instead of evaluating f on the ciphertexts, we evaluate polynomials $g_i^{(f)}$ that are derived from f on parts of the ciphertexts.

We remark that we can additionally apply a very similar decomposition during the algorithm underlying [LMW23; OPPW24], which yields an additional improvement in terms of storage requirements of a factor of roughly 2^m.

$\{0, 1\}$-CRT RLWE We have just seen that we are able to switch from a polynomial evaluation in $R_q[Y]/(Y^d - 1)$ to one in R_q, assuming that the BV scheme [BV11] does not lose security when we choose the constant coefficients b of ciphertexts such that in the decomposition Eq. (1), $\iota(b)$ has only components in $\{0, 1\}$. These constant coefficient in BV ciphertexts relate directly to the bs in the underlying RLWE samples. Thus, we are interested in the security of RLWE when restricting RLWE samples (a, b) to ones with $b \in S$ for

$$\begin{aligned} S :&= \iota^{-1}\left(\{0,1\}^{n\times \mathrm{divisors}(q)}\right) \\ &= \left\{ b \in R_q \;\middle|\; b \bmod \ (p, X - \zeta^j) \in \{0,1\} \text{ for all } p \mid q,\ j \in \ (\mathbb{Z}/2n\mathbb{Z})^* \right\} \end{aligned}$$

where n is a power of two, and $R = \mathbb{Z}[X]/(X^n + 1)$ is a power-of-two cyclotomic number ring. We call this problem "$\{0, 1\}$-CRT RLWE". We want to mention that this problem can be thought of as lying between standard RLWE and NTRU, as described in Remark 2.7. As such, it may be of independent interest in the study of the hardness of NTRU.

We first observe it is easy to sample from the $\{0, 1\}$-CRT RLWE distribution. Restricting the distribution of b may not appear intuitive at first glance, since usually in RLWE, b is chosen as $b = as + e$ for a randomly chosen a and some noise/error e. However, there is no reason why we cannot start by sampling b (and perhaps doing so from a "weird" distribution) and then set $a = s^{-1}(b - e)$. In other words, we can easily create $\{0, 1\}$-CRT RLWE samples, but we still need to convince ourselves that the resulting problem remains hard, and as a consequence, that the scheme we build is secure. To do so, we show (in Thm. 7.1) that this problem can be reduced to standard RLWE (with preprocessing).

Theorem 1.3 (Hardness of $\{0, 1\}$-CRT RLWE (informal)). *Let q be a large LWE modulus, subject to some technical constraints. Assume that RLWE on $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ with error of size $q/\mathrm{poly}(n)$ is hard,* *even when the adversary is allowed to compute (and later use) a polynomially sized hint during a (possibly exponential-time) preprocessing phase that only receives R and q as input. Then $\{0,1\}$-CRT RLWE on R_q with error increased by a constant factor is also hard.*

To prove Thm. 7.1, we introduce various new techniques. At the core of our argument is the notion of an arbitrary (finite) subset of $\mathbb{R}^n/q\mathbb{Z}^n$ to be "evenly distributed". A suitable formalization is given by the smoothing parameter [MR04], which was previously only defined for lattices, and which we thus have to generalize to arbitrary sets.

Using this formalization, we can split the proof into two parts. Firstly, we show the generic hardness of RLWE restricted to b's from a set S with small smoothing parameter (Thm. 6.3). Intuitively, this can be achieved just by "rounding" an input b to the closest element of S, but we need to find a way to move the computation of the closest element of S to the preprocessing phase (depending on the structure of S, this might take exponential time). This is similar to the problem of CVP with preprocessing, which is well-studied, both from an algorithmic perspective [LLM06; DRS14] and in terms of NP-hardness [Mic01]. However, the sets in our case are not restricted to be lattices.

The second part of the proof is then to show that the $\{0,1\}$-CRT set indeed has small smoothing parameter (Lemma 7.6 and Corollary 7.8). Using Fourier-analytic techniques similar to the proof of Banaszczyk's transference theorem [Ban93], we can reduce this to a number-theoretic condition on the primes dividing q. We then proceed to show that this condition is fulfilled with very high probability for a suitable, random choice of prime divisors of q. While our bounds are quite loose and can be proven without using much of the available number-theoretic structure, they are enough to complete the reduction. However, getting a better understanding of this number-theoretic setting and its connection to the geometric structure of S remains an interesting open problem.

1.2 Related work

In the following, we discuss related work on RLWE variants other than our $\{0,1\}$-CRT RLWE, and prior/-concurrent work on constructing/optimizing/implementing DEPIR.

Other RLWE Variants Ring-LWE has become a fundamental tool in modern cryptography, and has given rise to a panoply of variants and hardness reductions between them. The standard definition of an LWE or RLWE instance consists of samples $(a, b = as + e)$ that are built from uniformly randomly chosen values a and s with an error e chosen from a Gaussian distribution. It is a natural question what impact a change of the underlying distributions of a, s, or e has.

Much work has been done on analysing LWE with non-uniform distributions of the secret key s, for example, [ACPS09; GKPV10; BLP+13; Mic18; BD20a; BJRW23; BBPS19]. In the RLWE setting, similar results also exist [LPR10; BD20b; LWW20]. Non-standard error distributions were considered for example in [MP13; AG11; BGPW16; May21]. Related problems such as Learning with Rounding [BPR12] can also be seen as LWE with a non-standard error distribution. To the best of our knowledge, only one other work [JLS24] analyses (R)LWE with a non-standard distribution of a (or, equivalently, b).

Work on DEPIR Before the breakthrough of [LMW23], there were already some attempts [CHR17; BIPW17] for constructing DEPIR. Instead of taking an approach based on homomorphic encryption, they approached the problem from the direction of locally-decodable codes. However, the security of these attempts remains heuristic.

Since then, the only papers that focus on building single-server DEPIR are the already mentioned ones [LMW23; OPPW24]. Additionally, single-server DEPIR is used by [DHMW24] to construct sublinear-time laconic function evaluation. There is also some work on multi-server DEPIR schemes [LLFP24; GLM+24]. Such PIR schemes rely on the client interacting with multiple servers who must not communicate with each other (i.e., require a non-collusion assumption). In such a setting, one can use secret-sharing methods and achieve information-theoretic security.

For this, both schemes [LLFP24; GLM+24] add a polynomial evaluation datastructure on top of the multi-server PIR protocol of [WY05], which is again based on polynomials. Although these schemes do not use ASHE or SHFs, they share the very general idea of evaluating a polynomial at a point by using evaluations of it and its derivatives at different points. We also want to remark that, while these schemes significantly improve on previous work, PIR protocols with client-independent preprocessing and sublinear online time have been known for a long time in the multi-server setting [BIM00].

Finally, the recent work of [FMS24] proposes a black-box compiler to provide malicious security for any PIR scheme, including DEPIR schemes.

2 Preliminaries

Notation. We extend functions $f : \mathbb{R}^n \to \mathbb{R}$ to sets in the usual way as

$$f(A) := \sum_{a \in A} f(a)$$

We will only do so when this is well-defined. We write $B_{\mathbb{R}^n}(r)$ for the n-dimensional ball of radius r centred around the origin. Moreover, for a set A (finite or measurable), we write $\$A$ for the uniform probability distribution on A. In some places we sum or scale probability distributions, which should be understood as the distribution we get when performing the same operations on independent random variables distributed according to the corresponding distributions. When n, m are integers, we write $n \perp m$ for "n and m are coprime".

All rings appearing in this work are assumed to be commutative and unital, except when it is explicitly mentioned that they are not. The most important ring is the ring of integers $R = \mathbb{Z}[X]/(X^n + 1)$ in a power-of-two cyclotomic number field, i.e., n is a power of two. By using the coefficient embedding

$$\mathbb{Z}^n \xrightarrow{\sim} R, \qquad (a_i)_i \mapsto \sum_i a_i X^i$$

we can identify R with $\mathbb{Z}^n$, and thus also get the ℓ_2-norm on R. Note that it is more natural to consider the canonical norm of a number ring, but in this case (power-of-two cyclotomics), both norms turn out to be equivalent up to scaling.

We use calligraphic letters $\mathcal{A}, \mathcal{B}, \mathcal{C}$ to denote probability distributions (or algorithms), upper case letters A, B, C, S for matrices and sets, and lower case letters to denote both scalars and vectors. The letter p is used only for prime numbers, and any occurrence should be understood to refer to a prime number, even when it is not explicitly mentioned.

2.1 Distributions and statistical distance

For $s > 0$, we consider the Gaussian function

$$\rho_s : \mathbb{R}^n \to \mathbb{R}_{>0}, \quad x \mapsto \exp(-\pi \|x\|^2 / s^2)$$

It defines the continuous Gaussian distribution $\mathcal{D}_{\mathbb{R}^n,s,u}$ of width s and center $u \in \mathbb{R}^n$ over $\mathbb{R}^n$, which has density function $\rho_s(x - u)/s^n$. We often consider also the continuous Gaussian distribution over $\mathbb{R}^n/L$ for a full-rank lattice $L \subseteq \mathbb{R}^n$ (usually $L = \mathbb{Z}^n$). We use $\mathrm{lift}(\cdot)$ as the shortest-lift map, which assigns for every element of the torus $x \in \mathbb{T} := \mathbb{R}/\mathbb{Z}$ (or quotients of other metric spaces) the shortest element $x' \in \mathbb{R}$ with $x' \equiv x \bmod \mathbb{Z}$. The distribution $\mathcal{D}_{\mathbb{R}^n/L,s,u}$ is then the distribution with density function

$$x \mapsto s^{-n} \sum_{t \in L} \rho_s(\mathrm{lift}(x) + t - u)$$

Note that this is exactly the distribution we get from sampling $x \leftarrow \mathcal{D}_{\mathbb{R}^n,s,u}$ and reducing it modulo L.

For a discrete, non-empty set $A \subseteq \mathbb{R}^n$, we also need the discrete Gaussian distribution $\mathcal{D}_{A,s,u}$, which is the discrete probability distribution on A with probability weight function

$$A \to \mathbb{R}_{\geq 0}, \quad x \mapsto \frac{\rho_s(x-u)}{\rho_s(A-u)}$$

As for the continuous version, we also define it on a quotient space. Let $L \subseteq \mathbb{R}^n$ be a lattice such that for all $t \in L$ we have $t + A \subseteq A$. Then $\mathcal{D}_{A/L,s,u}$ is the probability distribution over A/L with weight function

$$A/L \to \mathbb{R}_{\geq 0}, \quad x \mapsto \sum_{t \in L} \frac{\rho_s(\operatorname{lift}(x) + t - \operatorname{lift}(u))}{\rho_s(A - \operatorname{lift}(u))}$$

An important property of ρ_s is that it is concentrated around the origin, and it approaches 0 very fast as its argument moves away from the origin. The following folklore theorems formalize this property.

Lemma 2.1 (Gaussian tail bound, discrete version). *Let $L \subseteq \mathbb{R}^n$ be any lattice, $u \in \mathbb{R}^n$, and $\gamma \geq 1$. Then*

$$\rho_s((L+u) \setminus B_{\mathbb{R}^n}(\gamma s \sqrt{n})) \leq 2^{-\gamma n} \rho_s(L)$$

Proof. See [Ban93, Lemma 1.5 (ii)]. □

Lemma 2.2 (Gaussian tail bound, continuous version). *Let $\gamma \geq 1$. Then*

$$\int_{\mathbb{R}^n \setminus B_{\mathbb{R}^n}(\gamma s \sqrt{n})} \rho_s(x) dx \leq 2^{-\gamma n} \int_{\mathbb{R}^n} \rho_s(x) dx = 2^{-\gamma n} s^n$$

Proof. Take the limit over $L = \frac{1}{n}\mathbb{Z}^n$, $n \to \infty$ in the previous statement. □

At one point in this work, we require the Chernoff bound for a sum of independent random variables. It is similar to the more well-known Hoeffding inequality, but relies on the expectation of e^{X_i} instead of bounds on X_i.

Lemma 2.3 (Chernoff bound). *Let $X_1, \dots, X_n$ be independent random variables. Then, for any a, we have*

$$\Pr\Big[\sum_i X_i \geq a\Big] \leq \inf_{t > 0} e^{-ta} \prod_i \mathrm{E}\left[e^{tX_i}\right]$$

A very important tool for reduction in lattice-based cryptography is the statistical distance.

Definition 2.4. *For two discrete probability distributions $\mathcal{A}, \mathcal{A}'$ we define the ℓ_1-statistical distance (or just statistical distance) as*

$$\varDelta(\mathcal{A}, \mathcal{A}') := \sum_{x \in \operatorname{supp}(\mathcal{A}) \cup \operatorname{supp}(\mathcal{A}')} \Big| \Pr_{a \leftarrow \mathcal{A}}[a = x] - \Pr_{a \leftarrow \mathcal{A}'}[a = x] \Big|$$

For two continuous distributions $\mathcal{C}, \mathcal{C}'$ on $\mathbb{R}^n$ with density functions f, f', we define the ℓ_1-statistical distance as

$$\varDelta(\mathcal{C}, \mathcal{C}') := \int_{\mathbb{R}^n} |f(x) - f'(x)| dx$$

The statistical distance has the property that applying a (possibly randomized) function will never decrease the statistical distance, i.e., when writing $\mathcal{B}$ for the distribution of $f(X)$ where $X \leftarrow \mathcal{A}$, we have $\varDelta(\mathcal{B}, \mathcal{B}') \leq \varDelta(\mathcal{A}, \mathcal{A}')$. Therefore, if an algorithm outputs the answer with probability p for a certain input distribution $\mathcal{A}$, then replacing the input distribution by $\mathcal{A}'$ will result in the algorithm giving the answer with probability at least $p - \varDelta(\mathcal{A}, \mathcal{A}')$.

Later in this work, we will need a bound on the statistical distance between two Gaussians whose centres are not too far away from each other.

Lemma 2.5. *Let $s > 0$, $\delta > 0$ and $u \in \mathbb{R}^n$ with $\|u\| \leq \epsilon$. If $s \geq n^{\delta}\epsilon$, then the statistical distance between $\mathcal{D}_{\mathbb{T}^n,s}$ and $\mathcal{D}_{\mathbb{T}^n,s,u}$ is at most $7n^{1/2-\delta}$ for sufficiently large n.*

Proof. First, we note that it suffices to show the bound for $\Delta(\mathcal{D}_{\mathbb{R}^n,s}, \mathcal{D}_{\mathbb{R}^n,s,u})$, since applying the reduction modulo $\mathbb{Z}^n$ will not increase the statistical distance.

Now $\Delta(\mathcal{D}_{\mathbb{R}^n,s}, \mathcal{D}_{\mathbb{R}^n,s,u})$ is given by

$$\begin{aligned} & s^{-n} \int_{\mathbb{R}^n} |\rho_s(x) - \rho_s(x-u)| dx \\ \leq & s^{-n} \int_{\mathbb{R}^n} \rho_s(x) \Big| \exp\left(-\pi s^{-2} \left(\|x+u\|^2 - \|x\|^2\right)\right) - 1 \Big| dx \end{aligned}$$

We have

$$\big| \|x+u\|^2 - \|x\|^2 \big| = \big| 2\langle x, u\rangle + \|u\|^2 \big| \leq \epsilon \ (\epsilon + 2\|x\|)$$

By assumption $\epsilon/s \leq n^{-\delta}$, so for $\|x\| \leq \sqrt{n}s$, we have

$$s^{-2} \big| \|x+u\|^2 - \|x\|^2 \big| \leq n^{-2\delta} + 2n^{1/2-\delta} \leq 3n^{1/2-\delta}$$

and thus

$$\Big| \exp\left(-\pi s^{-2} \left(\|x+u\|^2 - \|x\|^2\right)\right) - 1 \Big| \leq 6n^{1/2-\delta}$$

for sufficiently large n (except when $\delta \leq 1/2$, but then claim is trivial anyway). It follows that

$$\begin{aligned} & \int_{\mathbb{R}^n} \rho_s(x) \Big| \exp\left(-\pi s^{-2} \left(\|x+u\|^2 - \|x\|^2\right)\right) - 1 \Big| dx \\ \leq & \ 2 \int_{\mathbb{R}^n \setminus B_{\mathbb{R}^n}(\sqrt{n}s)} \rho_s(x) dx + 6n^{1/2-\delta} \int_{\mathcal{B}_{\mathbb{R}^n}(\sqrt{n}s)} \rho_s(x) dx \end{aligned}$$

Now we can bound the first integral by using Lemma 2.2, while for the second integral, the trivial bound $\leq \int_{\mathbb{R}^n} \rho_s(x)dx$ is sufficient. In total, we find

$$\begin{aligned} & \Delta(\mathcal{D}_{\mathbb{R}^n,s}, \mathcal{D}_{\mathbb{R}^n,s,u}) \leq s^{-n} \Big(2 \cdot 2^{-n} + 6n^{1/2-\delta} \Big) \int_{\mathbb{R}^n} \rho_s(x) dx \\ & = 2^{1-n} + 6n^{1/2-\delta} \leq 7n^{1/2-\delta} \end{aligned}$$

as claimed. □

2.2 (Ring-) Learning with Errors

The security of many major homomorphic encryption schemes, including those being considered for standardization, relies on the Ring Learning with Errors (RLWE) problem [LPR10; SSTX09], which introduces additional structure to the standard Learning with Errors (LWE) problem proposed by [Reg05]. In this work, we will consider a slight generalization of the standard version of RLWE, where the distributions of all involved elements might be non-uniform. Unfortunately, this introduces some necessary technical details. However, on an intuitive level, the core of the problem remains unchanged.

Definition 2.6 ((Primal) RLWE). *Let R be a number ring, q, m be positive integers, $\mathcal{B}, \mathcal{E}$ distributions over R_q, and $\mathcal{S}$ a distribution over the unit group R_q^*. We say an algorithm solves the Ring Learning with Errors problem* $\mathrm{RLWE}(R, q, \mathcal{B}, \mathcal{E}, \mathcal{S}, m)$ *if it has non-negligible advantage in distinguishing the following two scenarios:*

- *In Scenario 1, the algorithm is given access to an oracle that can be called at most m times and, on input M, returns a sample (a, b) where $b \leftarrow \mathcal{B}$, $e \leftarrow \mathcal{E}$ and $a = s^{-1}(b - M - e)$.*

- *In Scenario 2, the oracle again can be called at most m times and answers each time with a new sample from the distribution $\$R_q \times \mathcal{B}$.*

Here, s is sampled once from $\mathcal{S}$ and remains the same in each oracle call.

The more common definition of RLWE is to choose $a_i \leftarrow \$R_q$ uniformly and to define $b_i = a_i s + e_i$. The problem is then simply to distinguish these samples (a_i, b_i) from uniform samples. Note that in this situation (we continue to assume $s \in R_q^*$), b_i is uniformly distributed on R_q and independent of e_i or s. This makes it easy to mask a message M by adding an RLWE sample, as $(a, b + M)$.

Unfortunately, this simple appraoch does not work anymore in our case. Since we want to choose b non-uniformly, we cannot mask M via $M + b$, as depending on the distribution $\mathcal{B}$, this might leak information about M. For example, for the constant distribution $\mathcal{B} = 0$, the RLWE problem becomes a variant of NTRU, but we clearly cannot mask m via $(a, m + 0)$. Therefore, it is necessary to use the oracle with input M when defining the RLWE problem. However, we argue that in the case $\mathcal{B} = \$R_q$, this new way of stating RLWE is equivalent to the standard version.

In particular, when given (standard-version) RLWE samples (a_i, b_i) with $b_i = a_i s + e_i$ and oracle inputs M_i, we can just output $(a_i, b_i + M)$. Now $b_i + M_i$ is distributed according to $\$R_q + M_i = \$R_q$, and obviously $a_i = s^{-1}((b_i + M_i) - M_i - e_i)$. Vice versa, when we are given access to an oracle as in our version of RLWE for $\mathcal{B} = \$R_q$, we can trivially generate standard RLWE samples by calling the oracle with $M = 0$. This argument now fails when we have a non-translation invariant distribution $\mathcal{B}$, so in general we need the oracle with parameter M to prove the security of standard encryption schemes.

As a last interesting point, we want to mention that our version of RLWE highlights an interesting connection to the NTRU problem.

Remark 2.7 (Relationship to NTRU). The NTRU problem asks to distinguish quotients f_i/s from uniform, where both s and f_i are small elements of R_q (say sampled from $\mathcal{D}_{R_q,\sigma}$). The standard way of encrypting a message M using NTRU is to mask it via $M + f_i/s$, but it has also been proposed [XZD+23] to encrypt the message "in the numerator", i.e., as $(f_i + M)/s$. The security of this encryption scheme does not follow from NTRU anymore, but instead corresponds exactly to $\mathrm{RLWE}(R, q, 0, \mathcal{E}, \mathcal{S}, m)$ with $\mathcal{B} = 0$ the constant 0 distribution. Unfortunately, our security reduction does not apply in this case, so the hardness of $\mathrm{RLWE}(R, q, 0, \mathcal{E}, \mathcal{S}, m)$ and the security of [XZD+23] remains unproven.

In this sense, we believe that $\mathrm{RLWE}(R, q, \mathcal{B}, \mathcal{E}, \mathcal{S}, m)$ for a $\mathcal{B}$ with less entropy than $\$R_q$ can be thought of as an intermediate problem between RLWE and NTRU. As such, it might be of independent interest for studying the hardness of NTRU, which seems somewhat more complicated than the hardness of RLWE [ABD16; DW21; KF17; BN24].

The scheme used in our work (as well as the previous work [LMW23; OPPW24]) very closely follows the [BV11] scheme. We now give a short proof of its security in our new RLWE setting, highlighting the changes necessary for a non-translation invariant $\mathcal{B}$.

Lemma 2.8. *Consider a positive integer $t \perp q$ and assume that the problem* $\mathrm{RLWE}(R, q, t^{-1}\mathcal{B}, \mathcal{E}, t^{-1}\mathcal{S}, m(n))$ *is hard (for all polynomially bounded $m(n)$). Then the symmetric cipher for messages $M \in R_t$ defined as*

$$\begin{aligned} \mathrm{Gen}(1^n) &= s \quad \textit{where } s \leftarrow \mathcal{S} \\ \mathrm{Enc}(M, s) &= \ (-s^{-1}(b - \mathrm{lift}(M) - te), b) \quad \textit{where } b \leftarrow \mathcal{B},\ e \leftarrow \mathcal{E} \\ \mathrm{Dec}((a, b), s) &= \mathrm{lift}(as + b) \bmod t \end{aligned}$$

is IND-CPA secure.

Note that its correctness depends on the choice of the error distribution $\mathcal{E}$.

Proof. The proof proceeds exactly as for the BGV scheme [BGV12]. Assume we are given an efficient adversary $\mathcal{A}$ that wins the IND-CPA game with non-negligible advantage. First, a hybrid argument shows that $\mathcal{A}$ also distinguishes valid encryptions and uniform encryption oracle outputs with non-negligible probability.

To complete the proof, consider now the algorithm $\mathcal{A}'$ that tries to solve RLWE by running $\mathcal{A}$. On each encryption oracle call with message M, it calls the RLWE oracle with $t^{-1}\mathrm{lift}(M)$ to get (a, b) and returns $(-a, t^{-1}b)$. If the RLWE oracle is uniform, the resulting answer remains uniform. Otherwise, the answer is a valid encryption of M w.r.t. secret ts. Thus, $\mathcal{A}'$ solves RLWE. □

Definition 2.9 (RLWE with preprocessing). *Define R, q, m, $\mathcal{B}$, $\mathcal{E}$, $\mathcal{S}$ as in Definition 2.6.* RLWE-P$(R, q, \mathcal{B}, \mathcal{S}, \mathcal{E}, m)$ *is a problem to solve* RLWE$(R, q, \mathcal{B}, \mathcal{S}, \mathcal{E}, m)$ *in polynomial time, where the adversary is allowed to execute a (possibly superpolynomial time) algorithm before querying* RLWE *oracle.*

2.3 Homomorphic Encryption

In this work, we only consider the private-key version of HE.

Definition 2.10 ((Somewhat) Homomorphic Encryption). *A (Somewhat) Homomorphic Encryption (SHE) scheme for a set of "permissible" arithmetic circuits $\mathfrak{C}$ is a private key encryption scheme* (KeyGen, Enc, Dec) *whose plaintext space P is a ring, together with an additional algorithm* Eval*, such that for every arithmetic circuit $\mathit{\Gamma} \in \mathfrak{C}$ with k inputs, we have*

$$\mathrm{Dec}(\mathrm{Eval}(\mathit{\Gamma}, \mathrm{Enc}(m_1, \mathrm{sk}), \ldots, \mathrm{Enc}(m_k, \mathrm{sk})), \mathrm{sk}) = \mathit{\Gamma}(m_1, \ldots, m_k)$$

where $\mathrm{sk} \leftarrow \mathrm{KeyGen}(1^n)$*.*

Usually, $\mathfrak{C}$ will contain all circuits of bounded additive and multiplicative depth. As mentioned in the introduction, one main goal of this paper is to investigate HE schemes with the additional, very strong property that homomorphic addition and multiplication are just addition and multiplication of ciphertexts (requiring that these are defined).

Definition 2.11 (Algebraic Homomorphic Encryption [LMW23]). *An SHE scheme* (KeyGen, Enc, Dec, Eval) *is called Algebraic (Somewhat) Homomorphic Encryption (ASHE), if its ciphertext space C is a ring, and for all permissible circuits $\mathit{\Gamma} \in \mathfrak{C}$, we have*

$$\mathrm{Eval}(\mathit{\Gamma}, \mathrm{ct}_1, \ldots, \mathrm{ct}_k) = \mathit{\Gamma}(\mathrm{ct}_1, \ldots, \mathrm{ct}_k)$$

We remark that unless explicitly mentioned, all rings are assumed to be commutative, so this framework of ASHE does not apply to GSW-style encryptions [GSW13] that are based on matrix rings.

The ASHE scheme by Brakerski and Vaikuntanathan. The existing ASHE schemes in the literature [Gen09; vGHV10; BV11] are fairly old as, prior to this work, there has been no attempt to build an HE scheme with the goal of making it ASHE. We also did not succeed in proposing a new, more efficient ASHE construction. Instead, we show how the existing BV scheme [BV11], which was used as the ASHE scheme in prior DEPIR constructions [LMW23; OPPW24], can be modified to support a more efficient DEPIR construction.

Let $R = \mathbb{Z}[X]/(X^n + 1)$ for n a power of two be a cyclotomic number ring, and set $R_q = R/qR$ to be its quotient by the "ciphertext modulus" q. Then, in the BV scheme, a message $M \in R_t = R/tR$ for $t \perp q$ is encrypted by hiding it in the lower bits of a standard RLWE sample (a, b). More precisely, the public-key encryption scheme underlying BV comprises the following algorithms:

$$\begin{aligned} \mathrm{Gen}(1^n) &= s \quad \text{where } s \leftarrow \$R_q \\ \mathrm{Enc}(M, s) &= \ (-as + te + \mathrm{lift}(M), a) \quad \text{where } a \leftarrow \$R_q,\ e \leftarrow \mathcal{D}_{\mathbb{Z}^n/q\mathbb{Z}^n, \sigma} \\ \mathrm{Dec}((b, a), s) &= \mathrm{lift}(as + b) \bmod t \end{aligned}$$

Note in particular the similarity to Lemma 2.8, from which we directly get the security of this scheme. BV becomes an ASHE scheme over the ciphertext ring $R_q[Y]$ by interpreting tuples (b, a) as polynomials $b + aY$. Therefore, the degree of ciphertexts will increase during multiplications, and we have to extend the decryption as

$$\mathrm{Dec}\Big(\sum_i Y^i a_i, s\Big) = \mathrm{lift}\Big(\sum_i a_i s^i\Big) \bmod t$$

This results in a correct homomorphic encryption as long as the error term te remains $\ll q$.

Theorem 2.12 ([BV11, Thm. 2]). *Let n, q, t, σ be the parameters of the BV scheme. Let $f(X_1, \dots, X_m) \in \mathbb{Z}[X_1, \dots, X_m]$ be a multivariate polynomial of total degree D and coefficient-bounded by $\|f\|_\infty$ (in absolute value). If $\|f\|_\infty (t\sigma n)^D \leq q/2$, then the scheme can (with probability exponentially close to 1) correctly and homomorphically evaluate f on fresh ciphertexts.*

Note that because the scheme is ASHE, it does not matter which circuit we use to evaluate a polynomial $f(X_1, \dots, X_m)$, the resulting ciphertext is always the same, namely $f(\mathrm{ct}_1, \dots, \mathrm{ct}_m)$.

2.4 The datastructure of [KU11] and DEPIR

The basic idea of the construction of Doubly-Efficient PIR by [LMW23] is to use the datastructure proposed by [KU11] that, once built, allows evaluating a multivariate polynomial in time logarithmic in the number of monomials.

Theorem 2.13 ([LMW23, Thm. 2.1], [KU11, Thm. 5.1]). *Let F, G be monic polynomials and $R = \mathbb{Z}_q[X, Y]/(F(X), G(Y))$. Let $f \in R[T_1, \dots, T_m]$ be a polynomial of individual degree $< d$ in every variable. Then there is an algorithm that, given f, computes a datastructure of size*

$$\mathrm{poly}(m, d, \ln \#R)(dm \ln \ln \#R)^m$$

in the same time. Using that datastructure, we can then compute $f(x_1, \dots, x_m)$ for any $x_1, \dots, x_m \in \mathbb{R}$ in time $\mathrm{poly}(d, m, \ln \#R)$*.*

The proof relies on the following idea. Assume for now that $R = \mathbb{F}_p$, for a large prime p. Consider the map

$$\phi_{\text{lift}} : \mathbb{Z}_Q \to \mathbb{F}_p, \quad x \mapsto \mathrm{lift}(x) \bmod p \tag{2}$$

where $Q = p_1 \cdots p_r$ is product of many small primes. If Q is very large, this map is "somewhat homomorphic", i.e., it is homomorphic on small values. Indeed, we will see that it satisfies our definition of a somewhat homomorphic function. The main point is now that (when p is large) we can find a large enough $Q \gg p$ such that every prime $p_1, \dots, p_r \ll p$. This means, for polynomial $f \in \mathbb{Z}[X_1, \dots, X_m]$ of bounded degree, we have

$$f(x_1, \dots, x_m) = \phi_{\text{lift}}(f(\hat{\phi}_{\text{lift}}(x_1), \dots, \hat{\phi}_{\text{lift}}(x_m)))$$

where

$$\hat{\phi}_{\text{lift}} : \mathbb{F}_p \to \mathbb{Z}_Q, \quad x \mapsto \mathrm{lift}(x) \bmod Q$$

By using the CRT isomorphism

$$\mathbb{Z}_Q \cong \mathbb{F}_{p_1} \times \dots \times \mathbb{F}_{p_r}$$

it is thus enough to evaluate f at points in $\mathbb{F}_{p_i}$. If the p_i are small enough, we can do the latter by reading an entry from a precomputed table of $f(x_1, \dots, x_m)$ for all $(x_1, \dots, x_m) \in \mathbb{F}_{p_i}^m$. Otherwise, we can repeat this idea, setting $p = p_i$. We call each such repetition a "reduction step", and usually consider the datastructure with two reduction steps. The resulting algorithm is displayed in Alg. 1.

This idea can be extended to all finite, reduced rings. However, as shown by [OPPW24], this is not necessary for DEPIR, as the ciphertext ring of the BV scheme can be decomposed into a product of finite fields. More concretely, if for every $p \mid q$, we have $p \equiv 1 \bmod 2Dn$, then we have the "double-CRT isomorphism"

$$R_q[Y]/(Y^D - 1) \cong \bigoplus_{j=1}^{Dn} \bigoplus_{p \mid q} \mathbb{F}_p \tag{3}$$

and it suffices to apply Thm. 2.13 only for $\mathbb{F}_p$, where $p \mid q$.

This isomorphism is an extension of the classical double-CRT isomorphism (also sometimes called double-RNS isomorphism)

$$R_q \cong \bigoplus_{j=1}^{n} \bigoplus_{p \mid q} \mathbb{F}_p$$

and is very commonly used in the literature, for example to speed up computations in $\mathbb{R}_q$. It was introduced by [GHS12] and since then has been used by all major implementations of FHE, like [GHS12; HS20; BBB+22; Zam22].

Algorithm 1: The evaluation datastructure [KU11] with two reduction steps.

```
Preprocessing
Input: multivariate polynomial $f \in \mathbb{Z}[X_1, \dots, X_m]$ of total degree $D$ and coefficients of size $\|f\|_\infty = \tilde{O}(1)$; the
       ambient finite ring $\mathbb{Z}_q$
For all primes $p$ of size $(1 + o(1)) \ (D \ln D + D \ln \ln q)$ compute
    $$T_p := (x, f(x))_x \quad \text{for } x \in \mathbb{Z}_p^m$$

Evaluation
Input: point $x \in \mathbb{Z}_q^m$
for all primes $p_1$ of size $(1 + o(1))D \ln q$ do
    Compute $x_{p_1} := \mathrm{lift}(x) \bmod p_1$
    for all primes $p_2$ of size $(1 + o(1))D \ln p_1$ do
        Compute $x_{p_1,p_2} := \mathrm{lift}(x_{p_1}) \bmod p_2$
        Using $T_{p_2}$, lookup
            $$y_{p_1,p_2} = f(x_{p_1,p_2}) \in \mathbb{Z}_{p_2}$$
    end
    Set $y_{p_1} = \mathrm{lift}(y'_{p_1}) \bmod p_1$ where $y'_{p_1}$ is the preimage of $(y_{p_1,p_2})_{p_2}$ under the CRT isomorphism
        $\mathbb{Z}_{\prod p_2} \xrightarrow{\sim} \bigoplus \mathbb{Z}_{p_2}$
end
Set $y = \mathrm{lift}(y') \bmod q$ where $y'$ is the preimage of $(y_{p_1})_{p_1}$ under the CRT isomorphism $\mathbb{Z}_{\prod p_1} \xrightarrow{\sim} \bigoplus \mathbb{Z}_{p_1}$
return $y$
```

Performance. In [OPPW24], the authors have argued that counting the number of entries that have to be read from the precomputed datastructure is a very good metric for measuring performance. They also gave a more precise asymptotic expression for this number of random storage accesses.

Proposition 2.14 ([OPPW24, Prop. 5.1]). *Let $f \in \mathbb{Z}[X_1, \dots, X_m]$ be a polynomial of total degree D, with all coefficients bounded by $\|f\|_\infty = \tilde{O}(1)$ in absolute value. Then, using the datastructure Thm. 2.13 with two reduction steps (i.e., applications of the shortest-lift map Eq. (2)) to evaluate $f(x_1, \dots, x_m)$ on $x_1, \dots, x_m \in \mathbb{Z}_q$ results in a datastructure size of $\tilde{O}(1)^m D^{m+1}$. The evaluation itself requires $\tilde{O}(D^2 \ln q)$ random storage accesses. Here, $\tilde{O}$ hides factors $\ln D$ and $\ln \ln q$.*

Building DEPIR. We now want to briefly sketch how this datastructure can be used to construct DEPIR. The basic idea is to "interpolate" the database into a polynomial. If we say the database DB is indexed by some index set I consisting of m-tuples, we can compute a multivariate polynomial $f \in \mathbb{Z}_t[X_1, \dots, X_m]$ such that

$$f(i_1, \dots, i_m) = \mathrm{DB}[(i_1, \dots, i_m)]$$

for all indices $(i_1, \dots, i_m) \in I$. In this case, the number of monomials of f will be equal to the size of the database N, but (depending on the choice of I), one can achieve that $\deg_{\mathrm{total}}(f) = O(N^{1/m})$.

To retrieve the $(i_1, \dots, i_m)$-th entry of the database, the client can now encrypt each i_j under an ASHE scheme, and the server uses the datastructure replies with an encryption of $f(i_1, \dots, i_m)$. If m is chosen large enough, the server can perform this homomorphic computation in time $o(N)$ by using the datastructure. Afterwards, the client just decrypts the reply and ends up with $\mathrm{DB}[(i_1, \dots, i_m)]$ as desired.

Corollary 2.15. *Consider the DEPIR scheme for a database of size N, and assume $\sigma = O(1)$ and $t = O(1)$. Then, using the datastructure Alg. 1 with two reduction steps, we need a datastructure of size $\tilde{O}(1)^m N^{1+1/m}$. Evaluating a PIR query requires $\tilde{O}(nN^{4/m})$ random storage accesses, each having size $\tilde{O}(1)$. Here, $\tilde{O}$ hides polynomial factors in $\ln n$ and $\ln N$.*

Proof. By making use of the decomposition Eq. (3), we have to run the evaluation algorithm Alg. 1 Dn times for each prime $p \mid n$. Since a multivariate polynomial of total degree D in m variables has $\binom{m+D}{m} = \varOmega(D^m)$ monomials, we can choose $D = O(N^{1/m})$. Primes that split in R are not too rare by the prime number theorem for primes in arithmetic progressions, thus we can choose all prime divisors of q to be of size $\mathrm{poly}(n)$. In particular, $\ln(p) \in \tilde{O}(1)$ for $p \mid q$, and using Prop. 2.14, we find that the required datastructure is of size

$$\tilde{O}(1)^m D^{m+1} = \tilde{O}(1)^m \; (N^{1/m})^{m+1} = \tilde{O}(1)^m N^{1+1/m}.$$

Now we apply Thm. 2.12 and see that we require $q = \varOmega(n^D)$, i.e., for the number of prime factors r of q, we can choose

$$r = O(\ln(n^D)/\ln(\mathrm{poly}(n))) = \tilde{O}(D).$$

Therefore, we find (using again Prop. 2.14) the total number of random read accesses to be

$$Dnr \cdot \tilde{O}(D^2) = \tilde{O}(n(N^{1/m})^4) = \tilde{O}(nN^{4/m}). \qquad \square$$

3 Somewhat Homomorphic Functions and ASHE

Given an ASHE scheme with plaintext space P and ciphertext space C, observe that for a fixed secret key sk and fixed encryption randomness r, the diagram

$$\begin{tikzcd}
P^k \arrow[r, "\mathrm{Enc}_r(\cdot, \mathrm{sk})"] \arrow[d, "\varGamma(\cdot, \ldots, \cdot)"'] & C^k \arrow[d, "\varGamma(\cdot, \ldots, \cdot)"] \\
P^k & C^k \arrow[l, "\mathrm{Dec}(\cdot, \mathrm{sk})"']
\end{tikzcd}$$

commutes for all permissible circuits $\varGamma(X_1, \ldots, X_k)$. Various formalizations of maps that make this (or similar) diagrams commute have already been studied before, for example, reverse-multiplication friendly embeddings (RMFE) [CCXY18; EHL+23] or packing methods [CL22]. The basic idea is always to "embed" a smaller space (e.g., P) into a larger space (e.g., C) in a way that is compatible with a limited number of arithmetic operations. This motivates our definition of a "somewhat homomorphic function" (SHF). The main difference to previous notions is that packing methods and RMFEs are required to be linear, while SHFs only have to be compatible with a limited number of additions.

Definition 3.1 (Somewhat Homomorphic Function). *Let R, S be rings. Then for $d_+, d_* > 0$, we say a function $\phi : R \to S$ is (d_+, d_*)-somewhat homomorphic (or a (d_+, d_*)-Somewhat Homomorphic Function, SHF), if it comes with a function $\hat{\phi} : S \to R$ such that for all arithmetic circuits $\varGamma(X_1, \ldots, X_k)$ of additive depth at most d_+ and multiplicative depth at most d_*, we have*

$$\forall x_1, \ldots, x_k \in S : \;\; \phi(\varGamma(\hat{\phi}(x_1), \ldots, \hat{\phi}(x_k))) = \varGamma(x_1, \ldots, x_m)$$

With this definition, a $(\infty, 1)$-somewhat homomorphic function $\mathbb{F}_{p^k} \to \mathbb{F}_{p^{kd}}$ is a RMFE. The definition does not exactly match packing methods as in [CL22], since for a packing method, the "unpacking" function (which corresponds to ϕ) may depend on the amount of multiplications performed by $\varGamma$. Finally, the shortest-lift map $\phi_{\mathrm{lift}} : \mathbb{F}_q \to \mathbb{Z}_N$ as used for the polynomial evaluation datastructure [KU11] (cf. Section 2.4) together with

$$\hat{\phi}_{\mathrm{lift}} : \mathbb{Z}_N \to \mathbb{F}_q, \quad x \mapsto \mathrm{lift}(x) \bmod q$$

is also an SHF.

Most importantly, following the diagram at the beginning of this section, we observe that ASHE schemes are closely related to SHFs, too. Given an ASHE scheme with plaintext ring P and ciphertext ring C, we immediately get a family of SHFs. In particular, for any secret key sk and encryption randomness r, the maps

$$\phi : C \to P, \quad \mathrm{ct} \mapsto \mathrm{Dec}(\mathrm{ct}, \mathrm{sk})$$
$$\hat{\phi} : P \to C, \quad m \mapsto \mathrm{Enc}_r(m, \mathrm{sk})$$

form an SHF with the same additive/multiplicative depth as supported by the ASHE scheme.

Note that any surjective ring homomorphism $\phi : R \to S$, with $\hat{\phi}$ being any left-inverse of ϕ, is always a (∞, ∞)-somewhat homomorphic map. However, such SHFs that are actually homomorphisms cannot be the result of post-quantum ASHE schemes. This has been shown by [AGKP14, Thm. 9]. While their notation differs from ours, the idea is that whenever the decryption map (for a fixed secret key) is $\mathbb{Z}$-linear (they call such schemes exact additively homomorphic encryption), a quantum adversary can use a Shor's algorithm-inspired method to decrypt arbitrary ciphertexts. Back in our setting, if the decryption map is a ring homomorphism, this clearly means that the ASHE scheme is such an exact additively homomorphic encryption scheme, thus vulnerable to quantum adversaries. We also note that while no impossibility theorem exists in the literature, no classically secure, arbitrary-depth ASHE scheme is known[6].

Moreover, if the SHF is a ring homomorphism when restricted to the subset of the ring that can actually be reached by homomorphic operations, then this restricted ring homomorphism "collapses back" to the original impossibility result. Hence, we also need to exclude such SHFs, which we call "trivial". Formally, we say an SHF $\phi : R \to S$ is trivial if it restricts to a non-unital ring homomorphism $R' \to S$ with $R' \subseteq R$ a subring and $\hat{\phi}(S) \subseteq R'$. In particular, this includes cases where S is either a quotient or a subring of R.

Since we are interested in ASHE schemes with small ciphertexts, this leads us to the main question of this section.

What dependency of $\#R$ on d_* can we achieve for a nontrivial SHF?

We also want to mention that [CL22] answered this question for packing methods by giving lower bounds on $\#R$ in terms of both d_* and $\#S$. Our bounds look somewhat similar, but the lack of linearity in SHFs makes the proofs much harder. Because of this, and since it is most relevant for our work, we lower-bound $\#R$ only in terms of d_* but ignore the size of S.

The following theorem is the main result of this section and directly addresses our motivating question. It will be proved in the remainder of this section. We restrict our attention to reduced rings, i.e., rings without nilpotents.

Theorem 3.2. *Let $\phi : R \to S$ be a (d_+, d_*)-somewhat homomorphic function between reduced rings R, S. Let $A := \hat{\phi}(S) \subseteq R$ and R_A be the non-unital subring generated by A.*

If $d_* \geq \log\log\#R + 5$ and $d_+ \geq 2\log\log\#R + 6$, then $\phi\big|_{R_A} : R_A \to S$ is a non-unital ring homomorphism, i.e., a trivial SHF.

We end up with a lower bound on the ciphertext ring size of ASHE schemes.

Corollary 3.3. *Let* $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *be a post-quantum ASHE scheme that can evaluate circuits of additive depth $2d$ and multiplicative depth d. Assume that its ciphertext ring C is reduced. Then $\#C \in 2^{\Omega(2^d)}$.*

To compare, the BV scheme requires a ciphertext ring of size $2^{\Omega(4^d)}$, which leads to the natural question whether there exists a scheme that matches our lower bound (Question 1.2). On the other hand, the ciphertext space (not a ring) of modern, non-ASHE schemes like BGV [BGV12] or BFV [FV12] (without bootstrapping) is of size $2^{O(d)}$, and thus provably better than what we can achieve with ASHE. This is also shown in Table 1.

3.1 Proof of Thm. 3.2

Instead of linear algebra as in [CL22], our proof relies on elementary additive combinatorics. At its core is the following lemma, from which it follows that $\log\log B$ subtractions and multiplications are enough to "produce" any integer of size up to B when starting from $\{0, 1\}$. Note that when we apply $+$, $-$, or $\cdot$ to sets, we refer to the set of all elements that are sums/differences/products of elements from the left and right set. This notation is standard in additive combinatorics.

[6] However, exact additively homomorphic encryption schemes under factorization-style assumptions do exist, e.g., Paillier [Pai99].

Table 1. Comparison of asymptotic ciphertext space resp. ciphertext ring sizes that are possible / achieved by known schemes. Parameters are chosen such that the schemes can evaluate circuits of multiplicative depth d and additive depth poly(d).

Scheme	Type	Ciphertext space size
Lower bound for ASHE	algebraic	$2^{\Omega(2^d)}$
BF [BV11]	algebraic	$2^{O(2^{2d})}$
BGV/BFV [BGV12; FV12]	non-algebraic	$2^{O(d)}$

Lemma 3.4. *Let n be a positive integer. Then*

$$\{-n^2, \dots, n^2\} \subseteq \{-n, \dots, n\}^2 - \{-n, \dots, n\}^2$$

Proof. By symmetry, it suffices to show

$$\{0, \dots, n^2\} \subseteq \{0, \dots, n\}^2 - \{0, \dots, n\}^2$$

So let $0 \leq x \leq n^2$, and let $y = \lceil \sqrt{x} \rceil \leq n$. Now we have either

$$(y-1)y < x \leq y^2$$

or

$$(y-1)^2 < x \leq (y-1)y$$

In the first case, it follows that $x = y^2 - z$ with $z < y^2 - y(y-1) = y \leq n$, and, in the second case, we see $x = y(y-1) - z$ with $z < y(y-1) - (y-1)^2 = y - 1 \leq n$. Both y^2 and $y(y-1)$ are in $\{0, \dots, n\}^2$ and any $z \leq n$ is $z = 1 \cdot z \in \{0, \dots, n\}^2$. The claim follows. □

Corollary 3.5. *Let R be a finite ring. Let $A \subseteq R$ and $\delta \geq \lceil \log \log \#R \rceil$. Then for every $x = ca$ with $a \in A^{2^\delta}$ and $c \in \mathbb{Z}$, there exists an arithmetic circuit $\Gamma(X_1, \dots, X_m)$ of additive and multiplicative depth at most $\delta + 2$ such that*

$$x = \Gamma(a_1, \dots, a_m) \quad \textit{with } a_1, \dots, a_m \in A$$

Proof. Write $\mathrm{reach}_d(A)$ for the set of elements that are reachable with a circuit of additive and multiplicative depth at most d, i.e.

$$\mathrm{reach}_d(A) := \{\Gamma(a_1, \dots, a_m) \mid \Gamma \text{ circuit of add./mult. depth} \leq d,\, a_i \in A\}$$

We show inductively that

$$\{-2^{2^d}, \dots, 2^{2^d}\}A^{2^d} \subseteq \mathrm{reach}_{d+2}(A)$$

The case $d = 0$ is clear, so assume $d > 0$. We have

$$\{-2^{2^{d-1}}, \dots, 2^{2^{d-1}}\}A^{2^{d-1}} \subseteq \mathrm{reach}_{d+1}(A)$$

Since the difference resp. product of two circuits gives a new circuit of additive resp. multiplicative depth increased by one, we have

$$\mathrm{reach}_{d+1}(A)^2 - \mathrm{reach}_{d+1}(A)^2 \subseteq \mathrm{reach}_{d+2}(A)$$

Thus by Lemma 3.4

$$\begin{aligned}\{-2^{2^d} \dots, 2^{2^d}\}A^{2^d} &\subseteq \left(\{-2^{2^{d-1}}, \dots, 2^{2^{d-1}}\}^2 - \{-2^{2^{d-1}}, \dots, 2^{2^{d-1}}\}^2\right)\left(A^{2^{d-1}}\right)^2 \\ &\subseteq \mathrm{reach}_{d+1}(A)^2 - \mathrm{reach}_{d+1}(A)^2 \subseteq \mathrm{reach}_{d+2}(A)\end{aligned}$$

This shows the claim, since $R \cap \mathbb{Z} \subseteq \{-\#R, \dots, \#R\}$. □

Lemma 3.6. *Let G be a finite group. Then every chain of subgroups of G has at most $\log \#G$ elements.*

Proof. Assume we have a proper chain of subgroups

$$A_1 \subsetneq A_2 \subsetneq \cdots \subsetneq A_k = R$$

then, for each $i < k$, it clearly holds $\#A_i \neq \#A_{i+1}$ and $\#A_i \mid \#A_{i+1}$. Thus $\#A_i \leq \frac{1}{2}\#A_{i+1}$ and the claim follows. □

Lemma 3.7. *Let G be a finite abelian group and $A \subseteq G$ a generating set. Then each element of G can be written as a $\mathbb{Z}$-linear combination of at most $\log \#G$ elements from A.*

Proof. As a finite abelian group, G is of the form

$$G = \bigoplus_p G^{(p)}$$

where $\#G^{(p)} = p^{k_p}$ for some positive integer k_p (in other words, $G^{(p)}$ is the localization at the prime ideal $(p) \leq \mathbb{Z}$ of the $\mathbb{Z}$-module G). Moreover, we have

$$\log \#G = \sum_p \log \#G^{(p)}$$

and the set

$$\underbrace{(\#Gp^{-k_p})}_{\in \mathbb{Z}} A \subseteq G$$

generates $G^{(p)}$, while being $\{0\}$ under any projection $G \twoheadrightarrow G^{(p')}$ for $p' \neq p$. Thus, it suffices to show the claim separately for each $G^{(p)}$.

So assume that $\#G = p^k$. This means that

$$G \cong \bigoplus_j \mathbb{Z}/p^{l_l}\mathbb{Z} \quad \text{for } l_j \text{ with } \sum l_j = k$$

Clearly, the case holds when $k = 1$, and thus by induction, we can assume that it holds for G/pG and pG. This means for any $x \in G$, there are $a_i \in A$ and $c_i \in \mathbb{Z}$ with

$$x \equiv \sum_{i=1}^{\lfloor \log \#G/pG \rfloor} c_i a_i \bmod pG$$

By using the assumption for pG, there must also exist $pa_i' \in pA$ and $c_i' \in \mathbb{Z}$ with

$$x - \sum_{i=1}^{\lfloor \log \#G/pG \rfloor} c_i a_i = \sum_{i=1}^{\lfloor \log \#pG \rfloor} c_i' \, (pa_i')$$

Thus

$$x = \sum_{i=1}^{\lfloor \log \#G/pG \rfloor} c_i a_i + \sum_{i=1}^{\lfloor \log \#pG \rfloor} (pc_i')a_i'$$

and the claim follows, since $\lfloor \log \#G/pG \rfloor + \lfloor \log \#pG \rfloor \leq \log \#G$. □

Lemma 3.8. *Let R be a finite ring and $A \subseteq R$. Let $k = \lceil \log \#R \rceil$ and R_A be the non-unital subring generated by A. Then for each $x \in \mathfrak{a}$ where*

$$\mathfrak{a} := \Big\{ \sum_i c_i a_i \;\Big|\; c_i \in R_A, a_i \in A^{2k} \Big\}$$

there is a circuit $\Gamma(X_1, \ldots, X_m)$ of additive depth at most $2\lceil \log k \rceil + 3$ and multiplicative depth at most $\lceil \log k \rceil + 3$ such that

$$x = \Gamma(a_1, \ldots, a_m) \quad \text{for } a_1, \ldots, a_m \in A$$

Proof. For any $B \subseteq R$, write $\langle B \rangle_{\mathbb{Z}}$ for the additive subgroup generated by B. Consider the chain

$$\langle A^{2k} \rangle_{\mathbb{Z}} \subseteq \sum_{i=2k}^{2k+1} \langle A^i \rangle_{\mathbb{Z}} \subseteq \sum_{i=2k}^{2k+2} \langle A^i \rangle_{\mathbb{Z}} \subseteq \dots \tag{4}$$

By Lemma 3.6, it has at most k elements. Furthermore, as soon as

$$\sum_{i=2k}^{2k+j} \langle A^i \rangle_{\mathbb{Z}} = \sum_{i=2k}^{2k+j+1} \langle A^i \rangle_{\mathbb{Z}}$$

we also have

$$\sum_{i=2k}^{2k+j+1} \langle A^i \rangle_{\mathbb{Z}} = \left\langle A \cdot \sum_{i=2k}^{2k+j} \langle A^i \rangle_{\mathbb{Z}} \right\rangle_{\mathbb{Z}} = \left\langle A \cdot \sum_{i=2k}^{2k+j+1} \langle A^i \rangle_{\mathbb{Z}} \right\rangle_{\mathbb{Z}} = \sum_{i=2k}^{2k+j+2} \langle A^i \rangle_{\mathbb{Z}}$$

Thus, the chain Eq. (4) becomes stationary after at most k elements. Now observe that the k-th element is closed under multiplication by R_A, and so

$$\mathfrak{a} = \sum_{i=2k}^{3k-1} \langle A^i \rangle_{\mathbb{Z}} = \langle A^{2k} \cup \dots \cup A^{3k-1} \rangle_{\mathbb{Z}}$$

Now consider some $x \in \mathfrak{a}$. By Lemma 3.7, we know that x is of the form

$$x = \sum_{j=1}^{k} c_j \prod_{l=1}^{m_j} a_{jl} \quad \text{for } c_j \in \mathbb{Z}, m_j \in \{2k, \dots, 3k-1\}, a_{jl} \in A$$

Now Corollary 3.5 with $\delta = \lceil \log k \rceil$ shows that we can write each

$$c_j \prod_{l=1}^{m_j} a_{jl} = \mathit{\Gamma}_j(a_{j1}, \dots, a_{jm_j}) \prod_{l=2^{\lceil \log k \rceil}+1}^{m_j} a_{jl}$$

for $\mathit{\Gamma}_j(X_1, \dots, X_{m_j})$ a circuit of add./mult. depth at most $\lceil \log k \rceil + 2$. Clearly, the product over $m_j - 2^{\lceil \log k \rceil} \leq 2k$ elements can also be computed within multiplicative depth $\lceil \log k \rceil + 2$. Thus, after summing the $\mathit{\Gamma}_j$, we are left with a circuit of multiplicative depth $\lceil \log k \rceil + 3$ and additive depth $\lceil \log k \rceil + 3 + \lceil \log k \rceil = 2\lceil \log k \rceil + 3$. □

Lemma 3.9. *Let $\mathfrak{a}$ be an ideal in a finite, non-unital, reduced ring R. Then for every $k > 0$, we have $\mathfrak{a} = \mathfrak{a}^k$.*

Proof. By adding a unit to R (i.e., considering the ring $R' = R + \mathbb{Z}$), it suffices to show the claim for unital rings. We show the claim locally at every prime ideal $\mathfrak{p}$ of R. However, the localization $R_{\mathfrak{p}}$ is a field for every prime $\mathfrak{p}$, thus $\mathfrak{a}_{\mathfrak{p}} \in \{0, R_{\mathfrak{p}}\}$ and the claim is trivial in this case. □

Now we can prove the main theorem.

Proof of Thm. 3.2. Let also $k = \lceil \log \#R \rceil$. By Lemma 3.9, we see that $R_A = R_A^{2k}$ is equal to the R_A-ideal generated by A^{2k}. Thus, Lemma 3.8 shows that we can reach every element of R_A with a bounded-depth circuit and inputs in A. Using this, we can easily show that $\phi\big|_{R_A}$ is a ring homomorphism. Let $x, y \in R_A$. By our argument, there must be circuits $\mathit{\Gamma}_x(X_1, \dots, X_m)$ and $\mathit{\Gamma}_y(X_1, \dots, X_m)$ of additive depth $\leq 2\lceil \log k \rceil + 3 \leq 2 \log\log \#R + 5$ and multiplicative depth $\leq \lceil \log k \rceil + 3 \leq \log\log \#R + 4$ such that

$$\mathit{\Gamma}_x(a_1, \dots, a_m) = x \quad \text{and} \quad \mathit{\Gamma}_y(a_1, \dots, a_m) = y$$

for some $a_1 = \hat{\phi}(s_1), \dots, a_m = \hat{\phi}(s_m) \in A$. However, then

$$\begin{aligned} \phi(x + y) = \phi((\mathit{\Gamma}_x + \mathit{\Gamma}_y)(a_1, \dots, a_m)) = \ & (\mathit{\Gamma}_x + \mathit{\Gamma}_y)(s_1, \dots, s_m) \\ = \ & \mathit{\Gamma}_x(s_1, \dots, s_m) + \mathit{\Gamma}_y(s_1, \dots, s_m) \\ = \ & \phi(\mathit{\Gamma}_x(a_1, \dots, a_m)) + \phi(\mathit{\Gamma}_y(a_1, \dots, a_m)) = \phi(x) + \phi(y) \end{aligned}$$

and similar for xy. In other words, $\phi\big|_{R_A}$ is a non-unital ring homomorphism as claimed. □

4 Generalizing ASHE for Applications in DEPIR

In the previous section, we have seen that there is an asymptotic performance gap between our lower bound on ASHE and known ASHE schemes, which leads us to the question whether one can construct an ASHE scheme matching the lower bound (Question 1.2). Unfortunately, we were unable to construct a scheme that would answer this question affirmatively. However, instead, we construct a scheme that almost matches the lower bound in Corollary 3.3, but is only algebraic in a generalized sense. While not being an ASHE scheme anymore, its Eval algorithm is still built from polynomial evaluations, and thus can be executed by using the polynomial evaluation datastructure introduced in Thm. 2.13. Therefore, we can use it to build DEPIR.

Consider the cyclotomic ring $R_q[X]/(X^n+1)$ and a multivariate polynomial $f \in \mathbb{Z}[X_1, ..., X_m]$ that represents the database. As sketched in the introduction, we now want to "decompose the ring" $R_q[Y]$ and "transform" the polynomial f in a compatible way. More concretely, we use the fact that input elements of $R_q[Y]$ only have degree one (since they are derived from freshly encrypted ciphertexts), and look at the evaluation of f on generic degree-1 polynomials:

$$f(X_1 + YZ_1, \dots, X_m + YZ_m) = \sum_i Y^i f_i(X_1, \dots, X_m, Z_1, \dots, Z_m). \tag{5}$$

Note that, as a polynomial in $Z_1, \dots, Z_m$, each f_i is homogeneous of degree i, which will later allow us to eliminate an additional variable.

Building on this, we present a new variant of BV. We first base its security on "$\{0,1\}$-CRT RLWE", a novel hardness assumption whose security we will later relate to standard assumptions in Section 6.

Construction 4.1 ("Decomposed" $\{0,1\}$-CRT BV)**.** *Let $q \perp t$ be positive integers, n a power of two, and $R = \mathbb{Z}[X]/(X^n+1)$. Define the SHE scheme given by the following algorithms.*

KeyGen(1^n) *Output $s \leftarrow \$R_q$ (alternatively, any other secret distribution that leads to a hard variant of RLWE is also fine).*

Enc(M, s) *Output (b, a) where*

$$a = -s^{-1}(b - M - te), \quad b \leftarrow \mathcal{B}, \quad e \leftarrow \mathcal{D}_{\mathbb{R}^n/q\mathbb{Z}^n}$$

and $\mathcal{B}$ is the uniform distribution over the set of elements (of R_q) with $\{0,1\}$-CRT components.

Dec$((c_0, \dots, c_D), s)$ *Output* $\operatorname{lift}(c_0 + c_1 s + \dots + c_D s^D) \bmod t$.

Eval$(f,\ (b_1, a_1), \dots (b_m, a_m))$ *Output*

$$(c_i)_{0 \le i \le D} := \left(a_m^{-1} f'_{i,b_1,\dots,b_m}(a_1, \dots, a_{m-1})\right)_{0 \le i \le D}$$

where, using the f_i from Eq. (5), we set

$$f'_{i,b_1,\dots,b_m} = f_i(b_1, \dots, b_m, X_1, \dots, X_{m-1}, 1) \in \mathbb{Z}[X_1, \dots, X_{m-1}].$$

Proposition 4.2. *Assume $n, q,$ and σ are chosen to satisfy the conditions of Thm. 7.1. Then Construction 4.1 is secure under the standard RLWE assumption, i.e.,* $\mathrm{RLWE}(R, q, \$R_q, \mathcal{D}_{\mathbb{R}^n/q\mathbb{Z}^n, \sqrt{\sigma^2+\beta^{-2}}}, \$R_q, \mathrm{poly(n)})$ *for any polynomial β.*

Proof. Using Lemma 2.8, we see that the scheme is secure under the $\{0,1\}$-CRT RLWE assumption. Thus, by Thm. 7.1, it is also secure under the standard RLWE assumption. □

The evaluation algorithm in Construction 4.1 outputs the coefficients of $f(b_1 + a_1 Y, \dots, b_m + a_m Y)$, exactly like the standard BV scheme. In other words, on a technical level, the only difference is that we now use $\{0,1\}$-CRT RLWE instead of standard RLWE. However, this new approach for implementing Eval directly leads to an improved DEPIR. The main point is that the polynomials $f'_{i,b_1,\dots,b_m}$ can be precomputed for each combination $(b_1,\dots,b_m) \in \{0,1\}^m$. Moreover, the polynomials

$$f'_{1,b_1,\dots,b_m}, \dots, f'_{D,b_1,\dots,b_m}$$

are all evaluated at the same point $(a_1,\dots,a_m)$. At least on an implementation level, this is a significant advantage, as we can store the precomputed point-value tables in an interleaved manner, and thus lookup an evaluation using a single (larger) read instead of D (smaller) ones. The resulting DEPIR algorithm is presented in Alg. 2.

Algorithm 2: Adjusted DEPIR algorithm for our "Decomposed $\{0,1\}$-CRT BV" homomorphic encryption scheme (cf. Construction 4.1).

```
Preprocessing
Compute $f(X_1,\dots,X_m) \in \mathbb{Z}[X_1,\dots,X_m]$ interpolating the database
for $(b_1,\dots,b_m) \in \{0,1\}^m$, $i \in \{0,\dots,D\}$ do
    Create a polynomial evaluation datastructure $T_{i,b_1,\dots,b_m}$ for $f'_{i,b_1,\dots,b_m} = f_i(b_1,\dots,b_m,X_1,\dots,X_{m-1},1)$
end

Client
Receive input index $(i_1,\dots,i_m)$
Encrypt $i_j$ as $b_j + a_jY = \mathrm{Enc}(i_j)$
Send $(b_1 + a_1Y,\dots,b_m + a_mY)$ to Server
Receive $r'$ from Server
return $\mathrm{Dec}(r')$

Server
Receive $(b_1 + a_1Y,\dots,b_m + a_mY)$ from Client
Compute the image $a_{kjp}$ resp. $b_{kjp}$ of $a_k, b_k$ under the isomorphism
    $$R_q \xrightarrow{\sim} \bigoplus_{j=1}^{n} \bigoplus_{p \mid q} \mathbb{F}_p \tag{6}$$
Call Alg. 1 with datastructure $T_{l,b_{1jp},\dots,b_{mjp}}$ to compute $r_{1jp},\dots,r_{Djp}$ as the evaluations
  of $f'_{1,b_1,\dots,b_m},\dots,f'_{D,b_1,\dots,b_m}$ at
    $$(a_{1jp}a_{mjp}^{-1},\dots,a_{(m-1)jp}a_{mjp}^{-1})$$
Compute the preimage $r_l$ of the $r_{ljp}$ under the isomorphism Eq. (6)
Send $r' = \sum_{l=0}^{D} a_m^l r_l$ to Client
```

Next, we argue that the performance in practice of the DEPIR scheme given in Alg. 2 is an improvement by a factor of $N^{1/m}$, despite it reading the same amount of data from the datastructure (namely $\tilde{O}(nN^{4/m})$). This is the consequence of replacing D smaller reads by one larger read, as just explained. Since reading contiguous memory location is much faster than randomly distributed ones, this results in a speedup of almost $\tilde{O}(N^{1/m})$. We will further back this claim up in Section 9.

Proposition 4.3. *Consider the adjusted DEPIR scheme of Alg. 2 for a database of size N, and assume $\sigma = O(1)$ as well as $t = O(1)$. Then, using the datastructure from Thm. 2.13 with two repetitions of the main shortest-lift step (cf. Eq. (2)), we need a datastructure of size $\tilde{O}(1)^m N^{1+1/m}$. Evaluating a PIR query requires $\tilde{O}(nN^{3/m})$ random storage accesses, each having size $\tilde{O}(N^{1/m})$. Here, $\tilde{O}$ hides factors $\ln n$ and $\ln N$.*

Proof. We begin by analysing the size of the datastructure(s). In total, we prepare $2^m D = \tilde{O}(1)^m N^{1/m}$ datastructures, each for a polynomial in $m-1$ variables. Using Prop. 2.14, we see that thus the total size is

$$\tilde{O}(1)^m N^{1/m} \tilde{O}(1)^m D^{m-1+1} = \tilde{O}(1)^m N^{1+1/m}.$$

Now we come to the number of storage accesses. As explained before, by storing the datastructures for $f'_{1,b_1,\ldots,b_m}, \ldots, f'_{D,b_1,\ldots,b_m}$ in an interleaved manner, we can evaluate them all with the same number of storage accesses as we would need for one. Thus (again by Prop. 2.14), we find the total number to be

$$rn \cdot \tilde{O}(D^2) = \tilde{O}(nD^3) = \tilde{O}(nN^{3/m}). \qquad \square$$

We remark that, since we need a new datastructure for each $\{0,1\}^m$, the required storage space increased by a factor of 2^m (although this is hidden by the asymptotic factor $\tilde{O}(1)^m$). In practice, we would choose very small m, so the impact is not too big. Nevertheless, in Section 5, we introduce another optimization that can get rid of this factor.

5 Improving the Polynomial Evaluation Datastructure

Algorithm 3: The two-step version of the evaluation datastructure that additionally uses the p-adic decomposition in the last step.

```
Preprocessing
Input: multivariate polynomial $f \in \mathbb{Z}[X_1, \ldots, X_m]$ of total degree $D$ and coefficients of size $\|f\|_\infty = \tilde{O}(1)$; the
       ambient finite ring $\mathbb{Z}_q$
Decompose $f$ into the $f_j$ as in Eq. (7)
For all primes $p$ of size $(1 + o(1))\ (D \ln D + D \ln\ln q)$ compute
    $T_p := (x, f(x))_x$   and   $T_{p,j} := (x, f_j(x))_x$   for $x \in \{0, \ldots, p-1\}^m \subseteq \mathbb{Z}_{p^2}^m$

Evaluation
Input: point $x \in \mathbb{Z}_q^m$
for all primes $p_1$ of size $(1 + o(1)) D \ln q$ do
    Compute $x_{p_1} := \mathrm{lift}(x) \bmod p_1$
    for all primes $p_2$ of size $(\frac{1}{2} + o(1)) D \ln p_1$ do
        Compute $x_{p_1,p_2} := \mathrm{lift}(x_{p_1}) \bmod p_2^2$
        Decompose $(x_{p_1,p_2})_j = (x^{(0)}_{p_1,p_2})_j + p\ (x^{(1)}_{p_1,p_2})_j$
        Using $T_{p_2}$ and the $T_{p_2,j}$, lookup
            $y_{p_1,p_2} = f(x^{(0)}_{p_1,p_2}) + p \sum_{j=1}^{m} (x^{(1)}_{p_1,p_2})_j f_j(x^{(0)}_{p_1,p_2}) \in \mathbb{Z}_{p_2^2}$
    end
    Set $y_{p_1} = \mathrm{lift}(y'_{p_1}) \bmod p_1$ where $y'_{p_1}$ is the preimage of $(y_{p_1,p_2})_{p_2}$ under
        $\mathbb{Z}_{\prod p_2^2} \xrightarrow{\sim} \bigoplus \mathbb{Z}_{p_2^2}$
end
Set $y = \mathrm{lift}(y') \bmod q$ where $y'$ is the preimage of $(y_{p_1})_{p_1}$ under the CRT isomorphism $\mathbb{Z}_{\prod p_1} \xrightarrow{\sim} \bigoplus \mathbb{Z}_{p_1}$
return $y$
```

In this section we present a second improvement that follows the same idea of "decomposing the ring" and "transforming the polynomial". Recall that the proof of Thm. 2.13 relies on the fact that the shortest-lift map is a Somewhat Homomorphic Function. We can thus "move" the evaluation of f from $\mathbb{F}_p$ to $\mathbb{F}_{p_i}$ for multiple p_i along the maps

$$\mathbb{F}_p \xleftarrow{\mathrm{lift}(\cdot)} \mathbb{Z}_q \xleftarrow{\sim} \mathbb{F}_{p_1} \times \cdots \times \mathbb{F}_{p_r}.$$

However, in the same way that our modified BV scheme is not ASHE anymore, here we also do not necessarily require an SHF. Instead, we use the p-adic decomposition map

$$\phi_p : \mathbb{Z}_{p^2} \to \underbrace{\{0, \dots, p-1\}}_{\subseteq \mathbb{Z}_{p^2}} \times \mathbb{Z}_p, \quad a + bp \mapsto (a, b).$$

To formulate the evaluation of f at a point x in terms of $\phi_p(x)$, we consider the evaluation of f at generic, decomposed points $X_j + pY_j$:

$$f(X_1 + pY_1, \dots, X_m + pY_m) = f(X_1, \dots, X_m) + p \sum_{j=1}^{m} Y_j f_j(X_1, \dots, X_m). \tag{7}$$

The important point here is that all the terms of total degree in the Y_j larger than 1 vanish, due to working in $\mathbb{Z}_{p^2}$. Thus, we can evaluate f at points in $\mathbb{Z}_p$ through $m+1$ evaluations of f resp. f_j at points in $\{0, \dots, p-1\}$. While technically we remain in the ring $\mathbb{Z}_{p^2}$, the reduced number of points directly leads to a smaller lookup table.

Hence, we now have two ways to reduce the evaluation of f from a larger to a smaller ring:

- The shortest lift map (together with a CRT-decomposition) will reduce an evaluation in $\mathbb{F}_q$ to $(1 + o(1))D \ln q / (\ln D + \ln \ln q)$ evaluations at points in $\mathbb{F}_{p_i}$, with $p_i \leq (1 + o(1))D \ln q$.
- The p-adic decomposition technique will reduce an evaluation in $\mathbb{Z}_q$ to $m + 1$ evaluations at points in $\{0, \dots, \sqrt{q} - 1\}$ (assuming $q = p^2$).

This means that for large p, the shortest-lift map is much better, as it conceptually reduces the space from q to $D \ln q$. On the other hand, the p-adic decomposition technique reduces the space from q to $\sqrt{q}$, and thus is better for "small" q, i.e., $q \approx D \ln D$. This leads us to an improved evaluation algorithm given in Alg. 3.

Since the tables used by Alg. 3 use primes half as large as the ones considered in [OPPW24], we expect each of them to use about a factor of 2^m less space. On the other hand, we require $m + 1$ of them, so the total decrease in datastructure size should be about a factor of $\frac{1}{m} 2^m$. This is hidden by the term $\tilde{O}(1)^m$, so the asymptotic datastructure size remains $\tilde{O}(1)^m D^{m+1}$. However, it does make a difference in practice, by cancelling out the factor 2^m introduced by our main optimization (Alg. 2).

6 Security of RLWE with Reduced Entropy

Section 4 showed that we are able to improve DEPIR, and come a little closer to answering Question 1.2, by using a special, reduced-entropy variant of RLWE. For this to work, we have to reduce the entropy in a way compatible with the algebraic structure, and so we settle for $\{0, 1\}$-CRT RLWE. To gain trust in the security of this variant, we want to find a reduction from standard RLWE.

In this section, we lay the groundwork for this reduction, by introducing a wide class of reduced-entropy variants of RLWE and giving a reduction argument for all of them. To then completely demonstrate the security of $\{0, 1\}$-CRT RLWE, we then only have to show that it lies within this class (we defer this to Section 7). Note that there might also be interest in other reduced-entropy variants of RLWE, for which the first part of our reduction then applies as well.

More concretely, we want to consider RLWE samples where not both elements are uniformly random. This leaves us two choices: we might either reduce the entropy in a or in b. For our construction, it turns out that the second choice, i.e. choosing a nonstandard distribution $\mathcal{B}$, is better. This also has the additional advantage that we can give a security reduction that does not yield a randomized error distribution, which is a common technical detail that appears whenever an error is multiplied by the secret.

In this section, we consider $\mathcal{B}$ to be the distribution $\sum_j \$A_j$, where the A_j are "evenly distributed" sets. The notion of "evenly distributed" is given by the smoothing parameter, as introduced by [MR04]. This is very well-studied quantity in the case of lattices, but there is no reason why we cannot use it for general sets.

Definition 6.1 (Smoothing parameter). *Let $A \subseteq \mathbb{T}^n$ be a nonempty and finite set. For $\epsilon > 0$, we define the smoothing parameter $\eta_\epsilon(A)$ as the infimum of all s such that*

$$\forall u \in \mathbb{T}^n : \ |\rho_s(u + A) - \rho_s(A)| \leq \epsilon \rho_s(A)$$

Before we come to the main theorem, we show that the correct formalization of "rounding to an evenly distributed set A" gives us a uniform element of A and a Gaussian error.

Lemma 6.2. *Let $A \subseteq \mathbb{T}^n$ be a nonempty and finite set, $\frac{1}{2} > \epsilon > 0$ and let $s \geq \eta_\epsilon(A)$.*

Now consider $a \leftarrow \$\mathbb{T}^n$ and $b \leftarrow \mathcal{D}_{A,s,a}$. In this case, the tuple $(b, a - b)$ is within statistical distance 8ϵ to $\$A \times \mathcal{D}_{\mathbb{T}^n,s}$.

Proof. Consider some $x \in A$. The probability of $b = x$ is given by

$$\begin{aligned} \Pr[b = x] &= \int_{\mathbb{T}^n} \frac{\rho_s(x-a)}{\rho_s(A-a)} da \\ &= \int_{\mathbb{T}^n} \frac{1}{(1+\epsilon(a))\rho_s(A)} \rho_s(x-a) da \\ &= \frac{1}{\rho_s(A)} \int_{\mathbb{T}^n} \frac{1}{1+\epsilon(a)} \rho_s(a) da \end{aligned}$$

with $|\epsilon(a)| \leq \epsilon$ for all a. Thus

$$(1-2\epsilon)\frac{s}{\rho_s(A)} \leq \frac{s}{(1+\epsilon)\rho_s(A)} \leq \Pr[b = x] \leq \frac{s}{(1-\epsilon)\rho_s(A)} \leq \ (1+2\epsilon)\frac{s}{\rho_s(A)}$$

Note that the probabilities must sum to 1, so we find

$$1 - \epsilon \leq \#A \frac{s}{\rho_s(A)} = \sum_{x \in A} \frac{s}{\rho_s(A)} \leq 1 + \epsilon$$

It follows that the statistical distance of b from $\$A$ is bounded as

$$\begin{aligned} \sum_{a \in A} \Big| \Pr[b = x] - \frac{1}{\#A} \Big| \leq \sum_{a \in A} \frac{2\epsilon s}{\rho_s(A)} + \frac{\epsilon}{\#A} &\leq \#A \Big(\frac{2\epsilon s}{\rho_s(A)} + \frac{\epsilon}{\#A} \Big) \\ &\leq \#A \frac{2\epsilon(1+\epsilon) + \epsilon}{\#A} \leq 4\epsilon \end{aligned}$$

For the second part, consider now the probability density function of $b - a$. Up to normalization, it is given by

$$g(y) \propto \sum_{b \in A} \frac{\rho_s(b - \ (b-y))}{\rho_s(A - \ (b-y))} = \sum_{b \in A} \frac{\rho_s(y)}{\rho_s(A - \ (b-y))}$$

Now we condition on $b = x$ for a fixed $x \in A$. The conditioned density function is now given by

$$g_{b=x}(y) = \gamma \frac{\rho_s(y)}{\rho_s(A - \ (b-y))}$$

for some normalization constant $\gamma > 0$. Using the smoothing parameter, we now estimate as before that

$$(1-2\epsilon)\frac{\gamma\rho_s(y)}{\rho_s(A)} \leq \frac{\gamma\rho_s(y)}{(1+\epsilon)\rho_s(A)} \leq g_{b=x}(y) \leq \frac{\gamma\rho_s(y)}{(1-\epsilon)\rho_s(A)} \leq \ (1+2\epsilon)\frac{\gamma\rho_s(y)}{\rho_s(A)}$$

Since γ is the normalization constant, we find

$$1-\epsilon \leq \frac{\gamma s^n}{\rho_s(A)} = \int_{\mathbb{T}^n} \frac{\gamma \rho_s(y)}{\rho_s(A)} dy \leq 1+\epsilon$$

Thus

$$\begin{aligned}
\int_{\mathbb{T}^n} \Big| g_{b=x}(y) - \frac{\rho_s(y)}{s^n} \Big| dy \leq & \int_{\mathbb{T}^n} \Big(\frac{2\epsilon\gamma}{\rho_s(A)} + \frac{\epsilon}{s^n} \Big) \rho_s(y) dy \\
\leq & \Big(\frac{2\epsilon(1+\epsilon)+\epsilon}{s^n} \Big) \int_{\mathbb{T}^n} \rho_s(y) dy \\
\leq & \frac{4\epsilon}{s^n} \int_{\mathbb{T}^n} \rho_s(y) dy \\
= & \frac{4\epsilon}{s^n} s^n = 4\epsilon
\end{aligned}$$

□

We are now able to give the main result of this section.

Theorem 6.3. *Let $c > 0$ be a constant, and let $m = o(n^c)$, q, and γ be positive integers. Assume that $q \geq \gamma$ and $\gamma = \operatorname{poly}(n)$. Let finally $\mathcal{S}$ be a distribution over R_q and $\sigma > 0$ with $\gamma\sigma \geq n^{c+1}$.*

Assume we have discrete, nonempty and finite sets $qA_1, \ldots, qA_n \subseteq R_q$ with $\eta_{2^{-n}}(A_i) \leq 1/\gamma$. If there exists an efficient algorithm (with preprocessing) for

$$\mathit{RLWE}\text{-}\mathit{P}(R, q, \sum_i q\$A_i, q\mathcal{D}_{\mathbb{T}^n, \sqrt{\sigma^2+1/\gamma^2}}, \mathcal{S}, m)$$

that has advantage $\Omega(1)$, then this implies also an efficient solver for standard RLWE with preprocessing, i.e., RLWE-P$(R, q, \$R_q, q\mathcal{D}_{\mathbb{T}^n,\sigma}, \mathcal{S}, m)$.

Intuition Given Lemma 6.2, the simplest approach would be to take an RLWE sample (a, b) and "round" it to A by outputting (a, b') for $b' \leftarrow q\mathcal{D}_{A,\sigma,b}$. In general this is not possible since we do not know anything about the structure of A, and thus cannot efficiently sample from $\mathcal{D}_{A,\sigma,b}$. Instead, we prepare a large number of samples around certain, well-chosen points during the preprocessing phase, and then use some of them to get a point close to b. This is why we then get a sum distribution $\sum_i q\$A_i$ in the result. We want to choose the initial points as scalar multiples of unit vectors of the cubic grid $\frac{1}{\gamma}\mathbb{Z}^n/\mathbb{Z}^n$, but we additionally have to mask them to make each one essentially uniform. This could be done either by multiplication with a random, invertible matrix, or by translation with a random vector. We choose the second method, as it is simpler.

Proof. We assume there is an algorithm $\mathcal{A}$ for RLWE-P with $\mathcal{B} = \sum_i q\$A_i$. We now construct the solver $\mathcal{A}'$ for RLWE-P$(R, q, \$R_q, q\mathcal{D}_{\mathbb{T}^n, \sqrt{\sigma^2+1/\gamma^2}}, \mathcal{S}, m)$ by using $\mathcal{A}$. In the preprocessing phase, we compute for each $i \in \{1, \ldots, m\}$ the following:

1. For each $j \in \{1, \ldots, n\}$, sample $r_j^{(i)} \leftarrow \$\mathbb{T}^n$ (the "mask").
2. For each $j \in \{1, \ldots, n\}$ and $k \in \{0, \ldots, \gamma - 1\}$ sample

$$s_{j,k}^{(i)} \leftarrow \mathcal{D}_{A_j, 1/\gamma, r_j^{(i)} + ke_j/\gamma}$$

   This step might require exponential time.
3. The outputs of the preprocessing phase are $r_j^{(i)}$ and $s_{j,k}^{(i)}$.

In the main phase, we now run $\mathcal{A}$, and when it calls the oracle the i-th time for some $M \in R_q$ to get a new RLWE sample, we do the following:

1. Call the standard RLWE oracle with M to get (a, b).

2. Define $r := \sum_j r_j^{(i)} \in \mathbb{T}^n$.
3. Sample $r' \leftarrow \$[-\frac{1}{2q}, \frac{1}{2q}]^n$.
4. Compute $x = (x_1, ..., x_n) = \lfloor \gamma\ (b/q + r' - r) \rceil \in \mathbb{Z}^n/\gamma\mathbb{Z}^n$.
5. Compute
$$b' := \sum_{j=1}^{n} q s_{j,x_j}^{(i)} \in \sum_j qA_j$$
6. Return (a, b').

Now we have to show first that b' indeed follows the distribution $\mathcal{B} = \sum_j q\$A_j$, and second that (a, b) with $a = s^{-1}(b - e - M)$ are mapped to (a, b') with $a = s^{-1}(b' - e' - M)$. Note that, clearly, tuples (a, b) with a uniform a are mapped to (a, b') with a uniform a (a and b resp. b' are also independent). From now on, we drop the superscript $\cdot^{(i)}$ when referring to $r_j^{(i)}$ or $s_{j,k}^{(i)}$, as we analyse the distributions of a single sample.

1. The distribution of b'. First, note that since $b/q + r'$ and r are both uniformly random on $\mathbb{T}^n$ and independent, it follows that $b/q + r' - r$ and r are independent. Thus, the random variables $x, r_1, \ldots, r_n$ are jointly independent. Furthermore, since again each r_j is uniformly random on $\mathbb{T}^n$, it follows that also $r_j + x_j e_j/\gamma$ is uniformly random on $\mathbb{T}^n$. Now Lemma 6.2 gives that s_{j,x_j} is distributed within negligible statistical distance of $\$A_j$. Since $s_{1,x_1}, \ldots, s_{n,x_n}$ are jointly independent, it follows that the distribution of b' is within negligible statistical distance to $\mathcal{B} = \sum_j q\$A_j$ as claimed.

2. The error distribution. Assume now that $b = as + qe + M$ with e distributed according to $\mathcal{D}_{\mathbb{T}^n,\sigma}$. We now consider the error
$$b' - M - as = b' - b + qe = \Big(b' - \frac{q}{\gamma}x\Big) + \Big(\frac{q}{\gamma}x - b\Big) + qe$$
For the first term, we find
$$b' - \frac{q}{\gamma}x = \sum_j q s_{j,x_j} - \frac{q}{\gamma}x = q \underbrace{\sum_j \left(s_{j,x_j} - x_j e_j/\gamma - r_j\right)}_{=:e_{\text{main}}} + q \underbrace{\sum_j r_j}_{=r}$$
Now we again use Lemma 6.2 and see that, up to negligible statistical distance, e_{main} is independent of b' and distributed according to $\mathcal{D}_{\mathbb{T}^n,1/\gamma}$.

For the second term, we find
$$\frac{q}{\gamma}x - b = qe_{\text{rnd}} + \ (b + qr' - qr - b) = qe_{\text{rnd}} + qr' - qr$$
where $e_{\text{rnd}} \in [-\frac{1}{2\gamma}, \frac{1}{2\gamma}]^n$ and $qr' \in [-\frac{1}{2}, \frac{1}{2}]^n$.

Now the term qr cancels out, and we see
$$b' - M - as = q(e + e_{\text{main}} + e_{\text{rnd}} + r')$$
Since e, e_{main} are independent, we see that their sum is distributed according to $\mathcal{D}_{\mathbb{T}^n,s}$ where $s = \sqrt{\sigma^2 + 1/\gamma^2}$. On the other hand, since $\gamma \leq q$, we have $e_{\text{rnd}} + r' \in [-1/\gamma, 1/\gamma]^n$. This implies $\|e_{\text{rnd}} + r'\| \leq \sqrt{n}/\gamma$.

By assumption, we now have $s \geq \sigma \geq n^{1+c}/\gamma \geq n^{1/2+c}\|e_{\text{rnd}} + r'\|$. Thus we can apply Lemma 2.5 for $\delta = 1/2 + c$, which shows that for any $e_{\text{rnd}} + r'$, the distribution $\mathcal{D}_{\mathbb{T}^n,s,e_{\text{rnd}}+r'}$ is within statistical distance $7n^{1/2-\delta} = 7n^{-c}$ from $\mathcal{D}_{\mathbb{T}^n,s}$. Thus, the statistical distance of the distribution of all errors, i.e.,
$$\bigtimes_{i=1}^{m} q\mathcal{D}_{\mathbb{T}^n,s,e_{\text{rnd}}^{(i)} + \ (r')^{(i)}}$$
to $q\mathcal{D}_{\mathbb{T}^n,s}^m$ is at most $7mn^{-c} = o(1)$.

To complete the proof, note now that the advantage of $\mathcal{A}$ w.r.t. the exact distributions is $\Omega(1)$, so the advantage on the distributions produced by our reduction is at least $\Omega(1) - o(1)$. This is clearly non-negligible, thus $\mathcal{A}'$ successfully solves RLWE-P$(R, q, \$R_q, q\mathcal{D}_{\mathbb{T}^n,\sigma}, \mathcal{S}, m)$. □

7 Security of $\{0,1\}$-CRT RLWE

We now instantiate Thm. 6.3 with a concrete class of evenly distributed sets that additionally have nice algebraic properties.

The $\{0,1\}$*-CRT set.* We already mentioned the importance of the double-CRT isomorphism

$$\begin{aligned} \iota:\ R_q = \mathbb{Z}_q[X]/(X^n+1) \ &\cong \bigoplus_{i \in\ (\mathbb{Z}/2n\mathbb{Z})^*}\ \bigoplus_{p\ \mid\ q} \mathbb{F}_p \\ a &\mapsto\ (a \bmod\ (p, X-\zeta^i))_{i,p} \end{aligned} \tag{8}$$

which exists whenever $p \equiv 1 \bmod 2n$ for all $p \mid q$.

We consider the set

$$S = \{a \in \mathbb{Z}_q \mid a \bmod p \in \{0,1\}\} \subseteq \mathbb{Z}_q \tag{9}$$

and the matrix

$$B = (\zeta^{ij})_{i \in \mathbb{Z}/n\mathbb{Z}, j \in (\mathbb{Z}/2n\mathbb{Z})^*} \subseteq \mathbb{Z}_q^{n \times n} \tag{10}$$

where $\zeta \in \mathbb{Z}_q$ is a primitive $2n$-th root of unity. With these definitions, we find that the set of elements that have 0 or 1 in their CRT components is

$$\{a \in R_q \mid \underbrace{a \bmod\ (p, X-\zeta^i)}_{=\iota(a)_{i,p}} \in \{0,1\}\} = BS^n$$

where (as before) we identify R with $\mathbb{Z}^n$ via the (isometric) coefficient embedding

$$R \xrightarrow{\sim} \mathbb{Z}^n, \quad \sum_{i=0}^{n-1} a_i X^i \mapsto\ (a_i)$$

Furthermore, we interpret vectors in S^n to be indexed by $(\mathbb{Z}/2n\mathbb{Z})^*$ so that the product BS^n is well-defined. The goal of this section is now to investigate the hardness of RLWE with distribution $\mathcal{B} = \$(BS^n)$, the uniform distribution on all elements with $\{0,1\}$-CRT components. More precisely, we want to prove the following theorem.

Theorem 7.1. *Let $r = \Omega(n^2 \log(n)^2)$ be divisible by $2n$, let $\beta = \mathrm{poly}(n)$, and let $p_1, \ldots, p_{r/2}$ be fixed primes of size at least $2\sqrt{n}\beta\log(\beta)$ that split in R. Let further $P_1, \ldots, P_{r/2}$ be finite sets of primes of size at least $2\sqrt{n}\beta\log(\beta)$ that split in R, such that for $p \leftarrow \$P_l$ the random variable $p \bmod p_l$ is distributed as $\$\mathbb{F}_{p_l}$. Define a set S and a matrix B as in Eq. (9) and Eq. (10), respectively. Finally, let $m = O(n^c)$ be an integer with $c > 0$, $\sigma \geq 4n^{c+1}/\beta$ and $\mathcal{S}$ any distribution over R_q. With probability exponentially close to 1 over the choice of $p_{r/2+1}, \ldots, p_r$ from $p_{r/2+l} \leftarrow \$P_l$, then for $q = p_1 \cdots p_r$, if there exists an efficient algorithm (with preprocessing) for*

$$\textit{RLWE-P}(R, q, BS^n, q\mathcal{D}_{\mathbb{T}^n,\sqrt{\sigma^2+16/\beta^2}}, \mathcal{S}, m)$$

that has advantage $\Omega(1)$, then this implies an efficient solver for standard RLWE with preprocessing, i.e.,

$$\textit{RLWE-P}(R, q, \$R_q, q\mathcal{D}_{\mathbb{T}^n,\sigma}, \mathcal{S}, m)$$

Proof. Up to permutation of the p_l, this directly follows from combining Thm. 6.3 with Corollary 7.8 and Lemma 7.6, which will be proven in this section. □

7.1 Small-CRT RLWE

We now present the main ideas of the proof of Thm. 7.1. In order to simplify the technical details we first present a simpler argument after which we will end up with a distribution supported on elements with small (but not binary) CRT components. We then give the corresponding results for $\{0,1\}$-CRT-RLWE, which are proved in exactly the same way.

In order to apply the general reduction Thm. 6.3, we have to show that BS^n is evenly distributed. Under a number-theoretic assumption (concretely Eq. (11)) that we show holds later, this is done by the next lemma. This lemma captures the core intuition of the proof technique.

Lemma 7.2. *Let $q = p_1 \cdots p_r$. Assume that for some $\beta > 0$ and all $y \in \mathbb{Z}^n \setminus \{0\}$ with $\|y\| \leq 4\sqrt{n}\beta \log(\beta)$ we have*

$$\prod_{j=1}^{n} \prod_{l=1}^{r} \Big| 1 + \exp\left(2\pi i \langle B^T y, e_j \rangle p_l^* / p_l\right) \Big| \leq 2^{nr} / \beta^n \tag{11}$$

where $p_l^* = \prod_{j \neq l} p_j^{-1} \in \mathbb{F}_{p_l}$. Then

$$\eta_{2^{3-n}}(BS^n/q) \leq 4/\beta$$

Intuition The proof proceeds similarly to the proof of Banaszczyk's transference theorem [Ban93] that relates the smoothing parameter $\eta_\epsilon(L)$ to the length of the shortest nonzero dual vector $\lambda_1(L^*)$[7]. Its main ingredient is the Poisson summation formula, which relates a function over a lattice with its Fourier transform over the dual. Using gaussians (which are eigenvectors of the Fourier transform) and concentration inequalities (like Lemma 2.1), one can then show the claim.

Since the set BS^n is not a lattice, we cannot show the required bounds on Fourier coefficients in analogue to [Ban93] by bounding "$\lambda_1((BS^n)^*)$", as this is not defined. Instead, for now, we directly assume Eq. (11), which looks somewhat like a Fourier coefficient. Proving that bound is a task we defer to later.

Proof. Consider an arbitrary $u \in \mathbb{R}^n$. We set $s = 4/\beta$ and use the Poisson summation formula over $\mathbb{Z}^n$ to find that

$$\begin{aligned} \rho_s(\mathbb{Z}^n + BS^n/q + u) = &\sum_{\delta \in BS^n/q} \sum_{x \in \mathbb{Z}^n} \rho_s(x + \delta + u) \\ = &s^n \sum_{y \in \mathbb{Z}^n} \rho_{1/s}(y) \exp(2\pi i \langle y, u \rangle) \sum_{\delta \in S^n/q} \exp(2\pi i \langle B^T y, \delta \rangle) \end{aligned}$$

since $\hat{\rho}_s = s^n \rho_{1/s}$. The summand for $y = 0$ simplifies to $s^n 2^{rn}$, since $\#S = 2^{rn}$ and $\langle 0, u \rangle = \langle 0, \delta \rangle = 0$. Thus, it suffices to bound the other terms. Observe now that

$$\sum_{\delta \in S^n/q} \exp(2\pi i \langle B^T y, \delta \rangle) = \prod_{j=1}^{n} \sum_{\delta \in S/q} \exp(2\pi i \langle B^T y, e_j \rangle \delta)$$

Furthermore, we can write S as

$$S = \Big\{ \sum a_l p_l^* \prod_{k \neq l} p_k \in \mathbb{Z}_q \;\Big|\; a_l \in \{0,1\} \Big\}$$

This means that

$$\sum_{\delta \in S/q} \exp(2\pi i \langle B^T y, e_j \rangle \delta) = \prod_{l=1}^{r} \Big(\exp(0) + \exp\left(2\pi i \langle B^T y, e_j \rangle p_l^* / p_l\right) \Big)$$

[7] Actually, it is usually stated as a bound on the covering radius $\mu(L)$ or on $\lambda_n(L)$ instead of $\eta_\epsilon(L)$ but, as we will see later, that is an easy corollary.

In other words, the assumption Eq. (11) gives that for $y \in \mathbb{Z}^n \setminus \{0\}$ with $\|y\| \leq 4\sqrt{n}\beta\log(\beta)$ we have

$$\Big|\sum_{\delta\in S^n/q} \exp(2\pi i\langle B^T y, \delta\rangle)\Big| \leq 2^{nr}/\beta^n$$

Thus, we can estimate

$$\Big|\sum_{\substack{y\in\mathbb{Z}^n\setminus\{0\}\\ \|y\|\leq 4\sqrt{n}\beta\log(\beta)}} \rho_{1/s}(y)\exp(2\pi i\langle y,u\rangle)\sum_{\delta\in S^n/q}\exp(2\pi i\langle B^T y,\delta\rangle)\ \Big| \leq \rho_{1/s}(\mathbb{Z}^n)2^{nr}/\beta^n$$

Finally, the y outside of the ball of radius $4\sqrt{n}\beta\log(\beta)$ do not contribute much anyway, as by Lemma 2.1 we have

$$\begin{aligned}
&\Big|\sum_{\substack{y\in\mathbb{Z}^n\\ \|y\|>4\sqrt{n}\beta\log(\beta)}} \rho_{1/s}(y)\exp(2\pi i\langle y,u\rangle)\sum_{\delta\in S^n/q}\exp(2\pi i\langle B^T y,\delta\rangle)\ \Big| \\
&\leq \sum_{\delta\in S^n/q}\Big|\sum_{\substack{y\in\mathbb{Z}^n\\ \|y\|>4\sqrt{n}\beta\log(\beta)}} \rho_{1/s}(y)\Big| \\
&\leq 2^{nr}\rho_{1/s}(\mathbb{Z}^n\setminus\mathcal{B}_{\mathbb{R}^n}(\frac{1}{s}\sqrt{n}\log(\beta))) \\
&\leq 2^{nr}2^{-\log(\beta)n}\rho_{1/s}(\mathbb{Z}^n) = 2^{nr}\beta^{-n}\rho_{1/s}(\mathbb{Z}^n)
\end{aligned}$$

Together, we bound the “error term” in our expression for $\rho_s(BS^n/q+u)$ as

$$\Big|s^n\sum_{y\in\mathbb{Z}^n\setminus\{0\}}\rho_{1/s}(y)\exp(2\pi i\langle y,u\rangle)\sum_{\delta\in S^n/q}\exp(2\pi i\langle B^T y,\delta\rangle)\Big| \leq 2\cdot 2^{nr}s^n\rho_{1/s}(\mathbb{Z}^n)/\beta^n$$

This directly implies

$$\rho_s(\mathbb{Z}^n + BS^n + u) \geq s^n 2^{nr}\ (1 - 2\rho_{1/s}(\mathbb{Z}^n)/\beta^{-n}) \geq s^n 2^{nr}\ (1-2^{1-n}) \tag{12}$$

and similarly

$$\rho_s(\mathbb{Z}^n + BS^n + u) \leq s^n 2^{nr}\ (1+2^{1-n})$$

since $\rho_{1/s}(\mathbb{Z}^n) = \rho_{1/s}(\mathbb{Z})^n \leq\ (2/s)^n = 2^{-n}\beta^n$. Now we use Eq. (12) with both u and $u' = 0$ to find

$$|\rho_s(BS^n+u) - \rho_s(BS^n)| \leq s^n 2^{nr}2^{2-n} \leq 2^{3-n}\rho_s(BS^n)$$

The claim follows. □

To continue, we now have to show Eq. (11) for a suitable choice of primes. Unfortunately, providing tight bounds here seems like a very difficult problem. We were not able to fully exploit the number-theoretic structure of the CRT isomorphism Eq. (8) and the matrix B. We only make use of the fact that $B \bmod p$ is invertible for all $p \mid q$. However, we believe that much stronger bounds could be achieved, if one were able to prove the following statements.

- If $y \neq 0$ is not too large, then at least a constant fraction of coefficients of $B^T y$ are in $\mathbb{Z}_q^*$.
- If y is random and small, then $B^T y$ behaves (in a suitable sense) somewhat like uniformly random on $\mathbb{Z}_q^n$.

We are, however, able to give a bound that is sufficient to apply Thm. 6.3 and holds with high probability for a random choice of primes.

We start with some technical lemmas.

Lemma 7.3. *Let $p_1, \dots, p_r$ be distinct primes. Then, if all p_l are sufficiently large, we have*

$$\Pr_{x_l \leftarrow \$ F_{p_l}} \left[\left| \prod_{l \leq r} 1 + e^{2\pi i x/p_l} \right| \geq (16/3)^{r/3} \right] \leq (2/3)^{2r/3} \tag{13}$$

Proof. Let $Y_l := \ln \left|1 + e^{2\pi i x_l/p_l}\right|$. Then we have to show that

$$\Pr_{x_l \leftarrow \$ F_{p_l}} \left[\sum_{l \leq r} Y_l \geq \frac{r}{3} \ln(16/3) \right] \leq (2/3)^{2r/3}.$$

To do so, we apply the Chernoff bound (Lemma 2.3) for the Y_l and get

$$\begin{aligned} \Pr_{x_l \leftarrow \$ F_{p_l}} \left[\sum_{l \leq r} Y_l \geq \frac{r}{3} \ln(16/3) \right] &\leq \inf_{t>0} \, (16/3)^{-rt/3} \prod_{l \leq r} \mathrm{E}\left[|1 + e^{2\pi i x_l/p_l}|^t \right] \\ &\leq (16/3)^{-r/3} \prod_{l \leq r} \mathrm{E}\left[|1 + e^{2\pi i x_l/p_l}| \right]. \end{aligned}$$

Observe that

$$\mathrm{E}\left[|1 + e^{2\pi i x_l/p_l}| \right] = \frac{1}{p_l} \sum_{x \in \frac{1}{p_l}\mathbb{Z}/\mathbb{Z}} |1 + e^{2\pi i x}|$$

As p_l increases, this gets arbitrarily close to

$$\int_0^1 |1 + e^{2\pi i x}| dx = \int_0^1 \sqrt{2 + 2\cos(2\pi x)} dx = \frac{4}{\pi}$$

Thus, if all the p_l are sufficiently large, we find

$$\mathrm{E}\left[|1 + e^{2\pi i x_l/p_l}| \right] \leq \frac{4}{3}$$

Now the claim follows, since

$$\Pr_{x_l \leftarrow \$ F_{p_l}} \left[\sum_{l \leq r} Y_l \geq \frac{r}{3} \ln(16/3) \right] \leq \; (16/3)^{-r/3} \; (4/3)^r \leq \left(\frac{3 \cdot 4^3}{16 \cdot 3^3} \right)^{r/3} = (2/3)^{2r/3} \qquad \square$$

The statement Eq. (13) required the x_l to be uniformly random in $\mathbb{F}_{p_l}$. In the bound Eq. (11) that we actually want, there are no additional constants, so the idea is to only bound the product for half the primes – say $p_1, \dots, p_{r/2}$ – and choose the other primes $p_{r/2+1}, \dots, p_r$ at random. Since the "CRT unit vectors" $\tilde{e}_l$ of $\mathbb{Z}_q$ are of the form $\tilde{e}_l = p_l^* \prod_{k \neq l} p_k \in \mathbb{Z}_q$, where p_l^* are as defined in Lemma 7.2, we see that $\tilde{e}_l/q = p_l^*/p_l$ includes the factor p_l^*, which depends on the other primes. Hence, by choosing these other primes randomly, we can use Lemma 7.3.

Lemma 7.4. *Let $r = \Omega(n \log(n)^2)$ be an even integer, and $p_1, \dots, p_{r/2}$ be sufficiently large primes that split in R. Let further $\beta = \mathrm{poly}(n)$ and $y \in \mathbb{Z}^n$ with at least one entry in $\mathbb{Z}_q^*$. Finally, let $P_1, \dots, P_r$ be finite sets of sufficiently large primes that split in R, and assume that the distribution of $p \bmod p_i$ for $p \leftarrow \$ P_i$ is uniform on $\mathbb{F}_{p_i}$. Then, for sufficiently large n, we have*

$$\Pr_{p_{r/2+l} \leftarrow \$ P_l} \left[\prod_{j=1}^{n} \prod_{l=1}^{r} \left| 1 + \exp\left(2\pi i \langle B^T y, e_j \rangle p_l^*/p_l \right) \right| \geq 2^{nr}/\beta^n \right] \leq \; (2/3)^{r/3}.$$

Proof. Since at least one entry of y is in $\mathbb{Z}_q^*$, we see that $y \bmod p_l$ is nonzero for all l. Note that every $B \bmod p_l$ is invertible, so for each l, there is some $j(l)$ with $\langle B^T y, e_{j(l)} \rangle \neq 0 \bmod p_l$. For this $j(l)$, we have then by Lemma 7.3

$$\Pr_{p_{r/2+l} \leftarrow \$ P_l} \left[\prod_{l=1}^{r/2} \left| 1 + \exp(2\pi i \langle B^T y, e_{j(l)} \rangle p_l^*/p_l) \right| \geq \; (16/3)^{r/6} \right] \leq \; (2/3)^{r/3}$$

since $\langle B^T y, e_{j(l)}\rangle p_l^*$ is distributed uniformly on $\mathbb{F}_{p_l}$. On the other hand, we can additionally use the trivial bound

$$\left|1+\exp(2\pi i \langle B^T y, e_j\rangle p_l^*/p_l)\right| \leq 2$$

for the remaining (j,l). In total, we find

$$\Pr_{p_{r/2+l}\leftarrow \$P_l}\left[\prod_{j=1}^{n}\prod_{l=1}^{r}\left|1+\exp\left(2\pi i \langle B^T y, e_j\rangle p_l^*/p_l\right)\right| \geq \; (16/3)^{r/6}\cdot 2^{nr-r/2}\right] \leq \; (2/3)^{r/3}$$

Finally, observe that $(16/3)^{1/6} < 2^{1/2}$, so

$$\begin{aligned}(16/3)^{r/6}\cdot 2^{nr-r/2} &= 2^{nr}\frac{(16/3)^{r/6}}{2^{r/2}}\\ &= 2^{nr}\exp(r\ln(16/3)/6 - r\ln(2)/2) \in 2^{nr}\exp(-\Omega(r))\end{aligned}$$

and the claim follows, since $-\Omega(r) = -\Omega(n\log(n)^2)$ is eventually smaller than $-\Omega(n\log(n)) = -\Omega(n\log(\beta))$, thus $\exp(-\Omega(r)) \leq \beta^{-n}$. □

Corollary 7.5. *Let $r, \beta, p_1, \ldots, p_{r/2}, P_1, \ldots, P_{r/2}$ be defined as in Lemma 7.4. Assume further that p_l and all elements of P_l are larger than $4\sqrt{n}\beta\log(\beta)$. Then, for sufficiently large n, we have*

$$\begin{aligned}\Pr_{p_{r/2+l}\leftarrow \$P_l}\Big[&\exists y \in \mathbb{Z}^n\setminus\{0\},\ \|y\| \leq 4\sqrt{n}\beta\log(\beta):\\ &\prod_{j=1}^{n}\prod_{l=1}^{r}\left|1+\exp\left(2\pi i\langle B^T y, e_j\rangle p_l^*/p_l\right)\right| \geq 2^{nr}/\beta^n\Big] \leq 2^{-n}\end{aligned} \tag{14}$$

Proof. First, note that the lower bound on the p_j means that for all $y \neq 0$ with $|y| \leq 4\sqrt{n}\beta\log(\beta)$, we immediately have $y \in \mathbb{Z}_q^*$. In particular, for $y \in \mathbb{Z}^n \setminus \{0\}$ with $\|y\| \leq 4\sqrt{n}\beta\log(\beta)$, at least one entry of y is in $\mathbb{Z}_q^*$. Hence, we can apply Lemma 7.4 for such a y. Taking now the union bound over all $O((4\sqrt{n}\beta\log(\beta))^n)$ possible y, we find that the probability of Eq. (14) is

$$\begin{aligned}&O\left((2/3)^{r/3}\cdot\ (4\sqrt{n}\beta\log(\beta))^n\right)\\ \leq&\exp(-\Omega(r - n\log(\beta) - n\log(n) - n\log\log(n)))\\ =&\exp(-\Omega(n\log(n)^2 - n\log(n))) = \exp(-\Omega(n\log(n)^2))\end{aligned}$$

The claim follows. □

Corollary 7.5 shows that the assumption Eq. (11) of Lemma 7.2 is satisfied for any $\beta = \mathrm{poly}(n)$ and a random choice of large enough primes with probability exponentially close to 1. Combining this with Lemma 7.2, we see that our set of $\{0,1\}$-CRT elements BS^n indeed has a small smoothing parameter (for a random choice of $r = \Omega(n\log(n)^2)$ primes). However, the reduction of Thm. 6.3 only gives a distribution $\mathcal{B} = \sum_j q\$A_j$ where the A_j have small smoothing parameter. In other words, we get the hardness of

$$\text{RLWE-P}\Big(R, q, \sum_{j=1}^{n}\$(BS^n), q\mathcal{D}_{\mathbb{T}^n,\sigma}, \mathcal{S}, \mathrm{poly}(n)\Big)$$

but not, as we really want, of

$$\text{RLWE-P}\Big(R, q, \$(BS^n), q\mathcal{D}_{\mathbb{T}^n,\sigma}, \mathcal{S}, \mathrm{poly}(n)\Big).$$

That is, we have established the hardness of RLWE where the b have CRT components chosen from the binomial distribution $B(\frac{1}{2}, n)$. In the folllowing subsection we will refine the argument to establish the hardness of RLWE where the b have CRT components chosen from $\{0,1\}$.

7.2 $\{0,1\}$-CRT RLWE

The main idea to complete the proof is to choose $r = \Omega(n^2 \log(n)^2)$ divisible by $2n$, so a factor of n larger than previously. This means we also have a factor of n more CRT components, which we group into n "groups" of equal size. Now we consider the sets

$$A_i = \left\{ a \in R_q \;\middle|\; \begin{array}{c} \text{The CRT components from the } i\text{-th group of } a \text{ are in } \{0,1\} \\ \text{The CRT components from all other groups of } a \text{ are } 0 \end{array} \right\}.$$

In particular, this means that $\sum A_i = BS^n$ and, moreover, $\sum \$A_i = \$(BS^n)$. Hence, it remains to show that each individual A_i has small smoothing parameter. The proofs above hold in exactly the same way for this situation, but with more complicated formalism, since we need to keep track of an additional index. For this reason, we will only sketch the proofs in this new setting.

For the rest of this section, consider fixed primes

$$p_1, \ldots, p_{\frac{1}{2}r/n},\ p_{r/n+1}, \ldots, p_{\frac{3}{2}r/n},\ \ldots,\ p_{r-r/n+1}, \ldots, p_{r-\frac{1}{2}r/n}$$

of sufficient size that split in R. Consider furthermore primes

$$p_{\frac{1}{2}r/n+1}, \ldots, p_{r/n},\ p_{\frac{3}{2}r/n+1}, \ldots, p_{2r/n},\ \ldots,\ p_{r-\frac{1}{2}r/n+1}, \ldots, p_r$$

chosen from suitable sets P_l. Now the CRT components from the "i-th group" are the ones corresponding to the prime ideals $(p_{ir/n+l}, X - \zeta^i)$ with $1 \leq l < r/n$ and $i \in\ (\mathbb{Z}/2n\mathbb{Z})^*$. Now define the sets

$$\begin{aligned} S_k := \{a \in \mathbb{Z}^q \mid \forall l \in \{1, \ldots, r/n\} : \\ a \bmod p_{ir/n+l} \in \{0,1\},\ \forall k \neq i :\ a \bmod p_{kr/n+l} = 0\} \end{aligned}$$

and

$$\begin{aligned} A_i &:= BS_i^n \\ &= \{a \in BS^n \mid \text{CRT components from all groups except } i \text{ of } a \text{ are } 0\}. \end{aligned}$$

The analogue of Lemma 7.2 is then given by Lemma 7.6, and the analogue of Corollary 7.5 is given by Corollary 7.8, which is a corollary of Lemma 7.7.

Lemma 7.6. *Let $q = p_1 \cdots p_r$ and $0 \leq i < n$. Assume that for some $\beta > 0$ and all $y \in \mathbb{Z}^n \setminus \{0\}$ with $\|y\| \leq 4\sqrt{n}\beta \log(\beta)$ we have*

$$\prod_{j=1}^{n} \prod_{l=1}^{r/n} \left| 1 + \exp\left(2\pi i \langle B^T y, e_j \rangle p^*_{ir/n+l} / p_{ir/n+l} \right) \right| \leq 2^{nr}/\beta^n$$

where $p_l^* = \prod_{j \neq l} p_j^{-1} \in \mathbb{F}_{p_l}$. Then

$$\eta_{2^{3-n}}(BS_i^n/q) \leq 4/\beta$$

Proof. Exactly as for Lemma 7.2. □

Lemma 7.7. *Let $r = \Omega(n^2 \log(n)^2)$ be a positive integer divisible by $2n$, and define $p_1, \ldots, p_r$, $P_1, \ldots, P_{\frac{1}{2}r/n}$, $\ldots, P_{r-r/n+1}, \ldots, P_{r-\frac{1}{2}r/n}$ as described at the beginning of this section. Let furthermore $0 \leq i < n$, $\beta = \mathrm{poly}(n)$, and $y \in \mathbb{Z}^n$ with at least one entry in $\mathbb{Z}_q^*$. Then, for sufficiently large n and $p_{kr/n+r/2+l} \leftarrow \$P_{kr/n+l}$, we have*

$$\Pr\left[\prod_{j=1}^{n} \prod_{l=1}^{r/n} \left| 1 + \exp\left(2\pi i \langle B^T y, e_j \rangle p^*_{ir/n+l} / p_{ir/n+l} \right) \right| \geq 2^{nr}/\beta^n \right] \leq\ (2/3)^{r/(3n)}.$$

Proof. Like in Lemma 7.4, observe that for every l, we have some $j(l)$ with $\langle B^T y, e_{j(l)}\rangle \neq 0 \bmod p_{ir/n+l}$. Again, this means that $\langle B^T y, e_j\rangle p^*_{ir/n+l}$ is uniformly random on $\mathbb{F}_{p_{ir/n+l}}$, and so Lemma 7.3 yields

$$\Pr\Big[\prod_{l=1}^{r/n}\Big|1+\exp\Big(2\pi i\langle B^T y, e_{j(l)}\rangle p^*_{ir/n+l}/p_{ir/n+l}\Big)\Big| \geq \; (16/3)^{r/(6n)}\Big] \leq \; (2/3)^{r/(3n)}.$$

Using the trivial bound

$$\Big|1+\exp\Big(2\pi i\langle B^T y, e_j\rangle p^*_{ir/n+l}/p_{ir/n+l}\Big)\Big| \leq 2$$

for the remaining j, l, we find

$$\begin{aligned}\Pr\Big[\prod_{j=1}^{n}\prod_{l=1}^{r/n}&\Big|1+\exp\Big(2\pi i\langle B^T y, e_j\rangle p^*_{ir/n+l}/p_{ir/n+l}\Big)\Big| \\ &\geq \; (16/3)^{r/(6n)}\cdot 2^{nr-r/(2n)}\Big] \leq \; (2/3)^{r/(3n)}.\end{aligned}$$

Now observe that $(16/3)^{1/6} < 2^{1/2}$, so

$$\begin{aligned}(16/3)^{r/(6n)}\cdot 2^{nr-r/(2n)} =& 2^{nr}\exp\Big(\frac{r}{6n}\ln(16/3)-\frac{r}{2n}\ln(2)\Big)\\ \in& 2^{nr}\exp(O(-r/n)).\end{aligned}$$

The claim now follows, since $O(-r/n) = O(-n\log(n)^2) \leq O(-n\log(n)) = O(\log(\beta^{-n}))$. □

Corollary 7.8. *Let $r = \Omega(n^2\log(n)^2)$ be a positive integer divisible by $2n$, and define $p_1, \dots, p_r$, $P_1, \dots, P_{\frac{1}{2}r/n}$, $\dots, P_{r-r/n+1}, \dots, P_{r-\frac{1}{2}r/n}$ as described at the beginning of this section. Let further $0 \leq i < n$ and $\beta = \operatorname{poly}(n)$. Assume furthermore that p_l and all elements of P_l are larger than $2\sqrt{n}\beta\log(\beta)$. Then, for sufficiently large n, we have*

$$\begin{aligned}\Pr\Big[&\exists y \in \mathbb{Z}^n\setminus\{0\},\; \|y\| \leq 2\sqrt{n}\beta\log(\beta) :\\ &\prod_{j=1}^{n}\prod_{l=1}^{r/n}\Big|1+\exp\Big(2\pi i\langle B^T y, e_j\rangle p^*_{ir/n+l}/p_{ir/n+l}\Big)\Big| \geq 2^{nr}/\beta^r\Big] \leq 2^{-n}.\end{aligned}$$

Proof. Again, we proceed exactly as in Corollary 7.5. There are $O((2\sqrt{n}\beta\log(\beta))^n)$ different $y \in \mathbb{Z}^n$ with $\|y\| \leq 2\sqrt{n}\beta\log(\beta)$. Thus, we can bound the probability by

$$\begin{aligned}(2/3)^{r/(3n)}\;(2\sqrt{n}\beta\log(\beta))^n \in& \exp\Big(-\frac{r}{3n}\ln(3/2)+O(n\log(\beta))+O(n\log(n))\Big)\\ \in& \exp\Big(O(n\log(\beta))-O(-n\log(n)^2)\Big)\end{aligned}$$

and the claim follows. □

8 Concrete Security of {0, 1}-CRT RLWE

The results of the previous section (cf. Thm. 7.1) show that asymptotically, $\{0,1\}$-CRT RLWE is indeed a hard problem, at least for a certain regime of parameters. In this section, we discuss the security of $\{0,1\}$-CRT RLWE for concrete parameters that are relevant for practical implementations of our protocol. Our concrete security estimates for $\{0,1\}$-CRT RLWE are summarised in Table 2.

For this, in the remainder of this section, we mainly examine the applicability of algorithms for solving RLWE to solving $\{0,1\}$-CRT RLWE. We present several arguments to support the conjecture that, concretely, $\{0,1\}$-CRT RLWE is not significantly easier than RLWE. However, we recognise that this is a new assumption and strongly encourage further cryptanalytic research to support or disprove this conjecture.

Geometric attacks Among the most successful attacks against RLWE are geometric attacks based on lattice reduction. They either solve BDD [LN13] or (u)SVP [ADPS16, Section 6.3] in the "primal" lattice $(a_1, ..., a_m)^T R$, or they solve SVP in the dual lattice of a projected sublattice thereof (the "dual" approach [MR09]). In the end, the core of the problem is always to find a short basis of a given lattice, which is usually done using the BKZ algorithm [SE94]. Note that these (as well as most other attacks) ignore the additional ring/ideal structure of RLWE, and just treat an RLWE instance as an LWE instance.

We were not able to find any efficient approach to exploit the special structure of the set of $\{0,1\}$-CRT elements BS^n that would enable an improvement in lattice reduction approaches. Additionally, note that any "norm-preserving" features of the (single)-CRT-isomorphism

$$\mathbb{Z}_q[X]/(X^n+1) \xrightarrow{\sim} \bigoplus_{j=1}^{n} \mathbb{Z}_q$$

would directly lead to a serious vulnerability of standard RLWE, since multiplication on the right-hand side is component-wise, thus a fixed component of $a_i s$ depends only on one component of the secret s. In other words, if there was a non-negligible chance that a short element of $\mathbb{Z}_q[X]/(X^n+1)$ has short CRT components, we could mount a distinguishing attack against RLWE by only considering these "1-dimensional" CRT component. By a cardinality argument, this means that also the reverse direction does not occur often, i.e. an element of $\mathbb{Z}_q[X]/(X^n+1)$ with small CRT components (like in the case of $\{0,1\}$-CRT RLWE) is unlikely to be small itself. This is further supported by our bound Lemma 7.6, which demonstrates that BS^n is evenly distributed. In particular, BS^n neither contains unusually many short vectors, or clusters of vectors that might be useful for performing lattice reduction. Thus, in Table 2, to concretely estimate the security of $\{0,1\}$-CRT RLWE against geometric attacks, we treat it as a corresponding RLWE instance.

Combinatorial attacks The situation is different for combinatorial attacks, as we can potentially exploit the special structure of the $\{0,1\}$-CRT set to improve known combinatorial attacks. In particular, in this subsection we will detail a modified version of the BKW algorithm [BKW03], which is the most famous combinatorial attack against (R)LWE. However, we find that this BKW variant does not significantly outperform lattice reduction approaches.

In essence, the BKW algorithm subtracts in stages elements $a = \sum a_i X^i \in R_q$ that agree on a subset of coefficients, and thus creates new elements with these coefficients set to zero. This results in lower-dimensional LWE instances with slightly increased noises, such that at the final stage, the remaining LWE instance is easy to solve. The observation in our setting is that instead of zeroing coefficients, we can equivalently zero CRT components. When the CRT components are in $\{0,1\}$ or otherwise small, we expect it to be much easier to find collisions, i.e., pairs of elements with the same values on a subset of CRT components.

We cannot translate the idea directly to our case as it is the b part of LWE samples has the $\{0,1\}$-CRT structure, rather than the a part. In other words, even if we find (e.g., using BKW) multiple short vectors $y^{(1)}, ..., y^{(r)}$ such that each of them annihilates the a_j as $\sum_j y_j^{(i)} b_j = 0$, then we have not yet solved our LWE problem. However, we have reduced it to the NTRU problem, since we find "samples"

$$s \sum_j y_j^{(i)} a_j + e^{(i)} = 0 \quad \Leftrightarrow \quad \sum_j y_j^{(i)} a_j = -\frac{e^{(i)}}{s} \quad \text{where } e^{(i)} = \sum_j y_j^{(i)} e_j$$

Taking the quotient of the "samples" associated with two different $y^{(i)}, y^{(i')}$, we can then cancel out s and find

$$\frac{\sum_j y_j^{(i)} a_j}{\sum_j y_j^{(1)} a_j} = \frac{e^{(i)}}{e^{(1)}}$$

which is the quotient of two short elements, hence an NTRU instance.

For the parameters in question, namely a small σ and $q \gg n^{c_{\text{fatigue}}}$ for some small constant $c_{\text{fatigue}} > 0$, the NTRU problem is said to be "overstretched" and significantly easier than RLWE [ABD16; DW21; KF17].

The exact value of c_{fatigue} depends on σ and is the topic of ongoing research, but [DW21] estimates it to be about $c_{\text{fatigue}} \approx 2.484$. Therefore, in our concrete estimates for the cost of this BKW variant, we avoid discussing the hardness of this version of NTRU and treat it as completely broken, i.e., we assume that it can be solved with no cost. Obviously, this is an underestimate, hence the concrete cost of the BKW approach will always be more than this.

At this point, we remark that more modern variants of BKW like coded-BKW [GJS15; GJMS17; GJMW19; GMW21] must be classified as "geometric approaches", since they significantly improve the performance of BKW for large q by using geometric structure. Thus, they are unlikely to be able to use the structure of $\{0, 1\}$-CRT RLWE, as mentioned above. However, it is possible to combine geometric approaches (e.g., coded-BKW or lattice reduction techniques) with the standard-BKW approach outlined above, and get a hybrid algorithm that outperforms both BKW and standard coded-BKW on $\{0, 1\}$-CRT RLWE samples. In particular, note that only the elements b of the first few tables T_i for $i \ll \log \max_{p \mid q} p$ during an execution of BKW actually have very narrowly distributed CRT components. Concretely, the elements in the i^{th} table T_i are the result of adding/subtracting 2^i initial elements, thus their CRT components are supported on $\{-2^i, ..., 2^i\}$. As a result, when i increases, the advantage of using BKW diminishes.

Instead of continuing normal BKW in this case, it thus makes sense to switch to a geometric algorithm. In detail, we choose prime ideals $\mathfrak{p}_0, ..., \mathfrak{p}_{rs-1}$ of R_q corresponding to the first $s = \lfloor \frac{1}{r} \sum_{l=0}^{t-1} \beta_l \rfloor$ CRT-components where β_l is the block size of the l^{th} stage of BKW. Then, after running BKW for t stages, we end up with RLWE samples (a, b) such that $b \equiv 0 \bmod \mathfrak{p}_j$, i.e., the b are part of an $(n - s)$-dimensional subspace. This leads to Alg. 4. We estimate its cost in Estimate 8.1. Note that our estimate is very conservative, since we completely ignore the cost of the NTRU oracle.

Estimate 8.1. *Using BKZ as SIS-oracle, we expect Alg. 4 running t stages of BKW with block sizes $\beta_l = \gamma/(l + 1)$ for $\gamma > 0$ and $0 \leq l \leq t - 1$ to have cost roughly*

$$\inf_{\gamma \geq 0} \; C_{\text{BKZ}}\Big(\exp\Big(\frac{(\ln q - \frac{1}{2} t \ln 2 - \ln \sigma)^2}{4(n - \gamma \ln t/r) \ln q}\Big)\Big) + t2^\gamma$$

when solving RLWE$(R, q, \$BS^n, q\mathcal{D}_{\mathbb{T}^n,\sigma}, \mathcal{S})$ for $q = p_1 \cdots p_r$. Here, $\mathcal{C}_{\text{BKZ}}(\delta_0)$ refers to the cost of BKZ with a sufficient block size to achieve a root-Hermite factor δ_0, and $t = \log \max_i p_i$.

Justification. During Alg. 4, we call the SIS oracle to find a short vector $y \in \mathbb{Z}^m$ in the lattice

$$L = \Big\{ y \in \mathbb{Z}^m \;\Big|\; \sum_j y_j b_j^{(t)} \equiv 0 \bmod q \Big\}$$

The value m can be chosen freely since we assume unlimited input samples (a_j, b_j) and hence can construct unlimited $b_j^{(t)}$. By construction, the $b_j^{(t)}$ have the first s CRT components set to 0 and we can assume the rest have a uniformly random value in $\mathbb{Z}_q^n$. In other words, they are all part of an $(n - s)$-dimensional subspace of $\mathbb{Z}_q^n$. Hence, the volume of L is with high probability $\text{vol}(L) = q^{n-s}$. Thus, we want to choose m to maximize the root-Hermite factor

$$\delta_0 = \frac{\|y\|^{1/m}}{\text{vol}(L)^{1/m^2}}$$

This results in $m = 2 \ln \text{vol}(L)/\ln \|y\|$, or

$$\delta_0 = \exp\Big(\frac{(\ln \|y\|)^2}{4 \ln \text{vol}(L)}\Big) = \exp\Big(\frac{(\ln \|y\|)^2}{4(n - s) \ln q}\Big)$$

Next, we estimate what size of y we need to achieve. Note that the samples $(a_j^{(t)}, b_j^{(t)})$ result from adding/subtracting 2^t input samples, thus their error is heuristically $\sqrt{2^t}\sigma$. Making a conservative estimate, it is necessary that at least $\|y\| \leq q2^{-t/2}/\sigma$, since otherwise the error would overflow q and the result would

Algorithm 4: Hybrid BKW for $\{0,1\}$-CRT-RLWE

```
   Input: Input samples $(a_j, b_j)$ in $\mathbb{Z}_q[X]/(X^n+1)$ where $b_j$ have $\{0,1\}$-CRT components; block sizes $\beta_l$ for each
          stage $0 \leq l \leq t-1$; SIS solver $\mathcal{A}$ that is used once we reach uniform $b$; NTRU oracle $\mathcal{B}$ for the last step
   Output: Whether $(a_j, b_j)$ are uniform or RLWE samples
 1 Initialize all tables $T_l := \emptyset$
 2 Let $j := 0$
 3 Take a new input sample $(a_j, b_j)$ and set $b_j^{(0)} := b_j \in \mathbb{Z}_q[X]/(X^n+1)$
 4 Set $\beta := 0$
 5 for $l = 0, 1, ..., t-1$ where $t = \lceil \log(\max_{p \mid q} p) \rceil$ do
 6     if $T_l$ contains $b$ with $b \equiv x_j^{(l)} \bmod \mathfrak{p}_i \ \forall \beta \leq i < \beta + \beta_l$ then
 7         Set $b_j^{(l+1)} := b_j^{(l)} - b$
 8         Update $\beta \leftarrow \beta + \beta_l$
 9     else
10         Add $b_j^{(l)}$ to $T_l$
11         Update $j \leftarrow j+1$ and go to step 3
12     end
13 end
14 if $T_t$ has enough elements to run $\mathcal{A}$ then
15     Find the vectors $y^{(j)}$ of $\ell_1$-norm at most $2^t$ such that $b_j^{(t)} = \sum_l y_l^{(j)} b_l$
16     Call $\mathcal{A}$ on the $b_j^{(t)}$ to find multiple short vectors $z^{(1)}, ..., z^{(k)}$ with $\sum_j z_j^{(i)} b_j^{(t)} = 0$
17     return Output of $\mathcal{B}$ on samples
           $\Big(\sum_{j,l} y_l^{(j)} z_j^{(i)} a_l\Big)\Big(\sum_{j,l} y_l^{(j)} z_j^{(1)} a_l\Big)^{-1}$  for $i \in \{2, ..., k\}$
18 else
19     Update $j \leftarrow j+1$ and go to step 3
20 end
```

be trivially unsolvable with high probability. This can be made precise by considering the exact distribution, but this only changes some small constants, as shown, e.g., by [Pla18]. Plugging this in, we get

$$\delta_0 = \exp\Big(\frac{(\ln q - \frac{1}{2} t \ln 2 - \ln \sigma)^2}{4(n-s)\ln q}\Big)$$

Finally, it is left to choose the block sizes β_l, which then define s via

$$s = \frac{1}{r} \sum_{l=0}^{t-1} \beta_l$$

It makes sense to choose them such that every stage of Alg. 4 has the same complexity. Note that the values $b_j^{(l)}$ all have coefficients in $\{-2^{l-1}, ..., 2^{l-1}\}$ for $l \geq 1$, thus the complexity of the l-th stage is bounded by $2^{(l+1)\beta_l}$. This motivates us to choose $\beta_l = \gamma/(l+1)$, in which case

$$s = \frac{1}{r} \sum_{l=1}^{t} \frac{\gamma}{l} \approx \frac{\gamma \ln(t)}{r}$$

and the total complexity of Alg. 4 is at least $t2^\gamma$ executions of the loop body at line 5.

To complete the estimate, we now note that γ can be chosen freely, thus the cost is

$$\inf_{\gamma \geq 0} C_{\mathrm{BKZ}}\Big(\exp\Big(\frac{(\ln q - \frac{1}{2} t \ln 2 - \ln \sigma)^2}{4(n - \gamma \ln t / r)\ln q}\Big)\Big) + t2^\gamma \qquad \square$$

In Table 2 we estimate the cost of Alg. 4 by including this formula in a custom script, built on top of the lattice estimator [APS15][8]. To estimate $C_{\mathrm{BKZ}}(\delta_0)$, we use the default cost model of the lattice estimator, which is derived from [MAT22]. Table 2 shows that Alg. 4 does not outperform the standard uSVP attack, even in our very conservative setting where the NTRU cost is estimated as zero. Overall, we can conclude that $\{0,1\}$-CRT RLWE is not concretely easier than an equivalent RLWE instance.

Table 2. Estimated concrete security (in bits) of $\{0,1\}$-CRT RLWE for various parameters. The row "Hybrid BKW" is estimated using a custom formula on top of the lattice estimator [APS15] based on Estimate 8.1. The cost of the other attacks is directly estimated using the lattice estimator, i.e., ignores the $\{0,1\}$-CRT structure.

n	2^{15}	2^{15}	2^{16}	2^{16}
$\log(q)$	721	841	1894	1924
r	24	30	63	64
Log estimated cost (rops)				
Hybrid BKW Alg. 4	157	133	118	116
Classical Dual Attack	158	134	119	117
uSVP Attack	156	133	118	116

Algebraic attacks The idea underlying algebraic attacks against LWE is to use LWE samples to create a polynomial system

$$I = \langle f_1(X_1, ..., X_n), ..., f_k(X_1, ..., X_n) \rangle \subseteq \mathbb{Z}_q[X_1, ..., X_n]$$

that, which high probability, has the LWE secret as only solution. This system can then be solved, either using linearization [AG11] or using Groebner basis [ACFP14]. Note that for known methods to construct I, the degree of the polynomials f_i depends strongly on the support of the error distribution. Thus, algebraic attacks have been most successful against LWE with unusually small errors.

The values of a or b only appear as constants in the above system I. Thus, restricting the values that b can take (as in our $\{0,1\}$-CRT RLWE case) seems to have basically no influence on the difficulty of finding a solution to I. Even significant progress in algebraic attacks against $\{0,1\}$-CRT RLWE would not invalidate our concrete security estimates, since the known attacks [AG11; ACFP14] are estimated to have huge complexity ($\gg 1000$ bits) when attacking RWLE with parameters in the regime we consider. Thus, we believe that we can ignore algebraic attacks when estimating the concrete security of $\{0,1\}$-CRT RLWE, and hence their cost is omitted from Table 2.

9 Implementation and Evaluation

We implement our DEPIR scheme utilizing our "decomposed $\{0,1\}$-CRT BV" homomorphic encryption scheme (cf. Alg. 2 and Alg. 3). Our implementation is written in Rust, includes multithreading support, and extends the open-source code provided by [OPPW24][9].

We then benchmark our DEPIR implementation and compare it to prior work [OPPW24], the only other implementation of DEPIR. All experiments use multithreading (utilizing all 24 cores) and are run on a system with an Intel Core i7-13700K CPU, 32 GB DDR5 RAM (5600 MHz), and a 4 TB Western Digital Black SN850X NVMe SSD. We limit the amount of RAM used to store parts of the datastructure to 26 GB, since additional RAM is needed for computation itself. However, the implemented splitting of the evaluation datastructure into RAM and disk is quite coarse-grained, thus the used amount of RAM is often significantly

[8] We have commit `e9f6a48b5995a89d17745da21f64fa9da821f5e5` from June 5th 2024.

[9] Our code is available at https://github.com/FeanorTheElf/ashe-depir

less than 26 GB. We provide the detailed parameter choices and results in Table 3, and plot the required number of queries as well as the size of the preprocessing datastructure in Figure 2 and Figure 3, respectively.

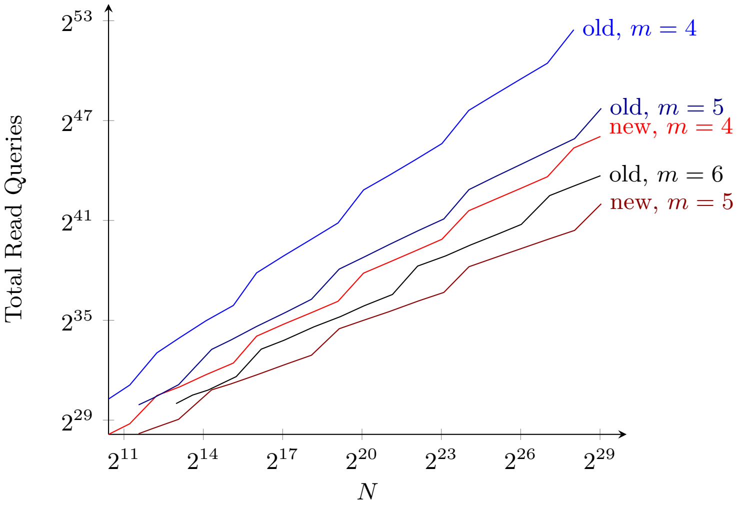

Fig. 2. Total number of queries (i.e., read accesses to the datastructure) performed by our implementation ("new") and [OPPW24] ("old", always performing two complete reduction steps) for different database sizes N and parameter choices for m.

Choice of parameters. We estimate the error (more precisely critical quantity) caused by the homomorphic computation to be approximately

$$e_{\text{exp}} = \sqrt{N}\left(\operatorname{erf}^{-1}(2^{-1/(nm)})t\sqrt{n}\sigma\right)^d \tag{15}$$

in canonical ℓ_∞-norm, i.e., $\|e\|_{\text{can},\infty} = \max|\sigma(e)|$ where $\sigma : R \to \mathbb{C}$ runs through all complex embeddings $R \hookrightarrow \mathbb{C}$. This then determines the size of q via $q \geq e_{\text{exp}}$. Together with an estimate of concrete security as described in Section 8, this gives the parameters as displayed in Table 3 and used also for Figures 2 and 3.

Our concrete estimate of the error Eq. (15) seems quite tight, and is equal to the experimentally chosen parameters of [OPPW24], up to a few bits. Thus, the parameter settings for $N \in \{46376, 15020334, 185250786\}$ can be considered to be almost the same as in [OPPW24]. To get reliable timings, we nevertheless run the code of [OPPW24] again on our system.

Results. Overall, as is evident from Table 3 (and as expected), we achieve a reduction in the number of queries of ×10 or more compared to [OPPW24] for large values of N resp. d. For the parameters for which we can actually run the protocol, we still achieve a speedup of more than ×4. For very small parameters however, our implementation runs slower than the one of [OPPW24], since we start storing data on disk from a smaller N on. On the storage side, we do slightly better than [OPPW24], but the difference is not very pronounced.

Note that our algorithm performs read accesses to larger chunks of data, i.e., about 50-100 bytes instead of only 2 bytes as in [OPPW24]. However, reading adjacent bytes is much faster than randomly distributed memory locations, so this does not offset our advantage. This is particularly in the case that the target data is stored on an SSD, since SSDs only support reading blocks of 4 KiB. Thus our larger reads come at (essentially) the same cost as the previous, single-byte reads.

Interestingly, due to the increased speed of RAM on our system, the storage-speed-tradeoff proposed by [OPPW24] is no longer optimal in our case. Thus, we also run the implementation from [OPPW24]

Table 3. Comparison of our implementation with previous work [OPPW24]. We ran the previous implementation on our system, once with the parameters proposed by [OPPW24] and once with two complete reduction steps, which is more similar to our implementation. Security levels are estimated as described in Section 8.

N	46376	142506	15020334	185250786
(m, D)	$(4, 30)$	$(5, 25)$	$(5, 68)$	$(6, 68)$
n	2^{15}	2^{15}	2^{16}	2^{16}
[OPPW24] (proposed parameters)				
$(\lfloor \log(q) \rceil, t, r)$	$(833, 29, 30)$	not given in [OPPW24]	$(1881, 32, 62)$	$(1913, 32, 63)$
Security (bits)	134		119	117
Total storage	1087 GB		873979 GB	411745167 GB
in RAM	26 GB		20 GB	15 GB
on Disk	1061 GB		873959 GB	411745152 GB
Queries total	$31.9 \cdot 2^{30}$		$3154.3 \cdot 2^{30}$	$3174.7 \cdot 2^{30}$
to RAM	$30.9 \cdot 2^{30}$		$815.3 \cdot 2^{30}$	$467.0 \cdot 2^{30}$
to Disk	$1.0 \cdot 2^{30}$		$2339.0 \cdot 2^{30}$	$2707.7 \cdot 2^{30}$
Running time	$1479.0s$		not possible to run	
[OPPW24] (when doing two complete reduction steps)				
$(\lfloor \log(q) \rceil, t, r)$	$(833, 29, 30)$	$(707, 28, 27)$	as above	
Security (bits)	134	160		
Total storage	18 GB	2040 GB		
in RAM	18 GB	25 GB		
on Disk	0 GB	2015 GB		
Queries total	$73.2 \cdot 2^{30}$	$44.6 \cdot 2^{30}$		
to RAM	$73.2 \cdot 2^{30}$	$26.8 \cdot 2^{30}$		
to Disk	0	$17.8 \cdot 2^{30}$		
Running time	$453.0s$	$27532.6s$		
Our work				
$(\lfloor \log(q) \rceil, t, r)$	$(841, 30, 28)$	$(721, 30, 24)$	$(1894, 30, 63)$	$(1924, 30, 64)$
Security (bits)	132	156	118	116
Total storage	44 GB	2927 GB	750922 GB	394474940 GB
in RAM	25 GB	17 GB	11 GB	9 GB
on Disk	19 GB	2910 GB	750911 GB	394474931 GB
Queries total	$7.1 \cdot 2^{30}$	$5.1 \cdot 2^{30}$	$133.9 \cdot 2^{30}$	$335.1 \cdot 2^{30}$
to RAM	$6.6 \cdot 2^{30}$	$2.1 \cdot 2^{30}$	$19.1 \cdot 2^{30}$	$20.4 \cdot 2^{30}$
to Disk	$0.5 \cdot 2^{30}$	$3.0 \cdot 2^{30}$	$114.8 \cdot 2^{30}$	$314.7 \cdot 2^{30}$
Running time	$1102.0s$	$6267.5s$	not possible to run	

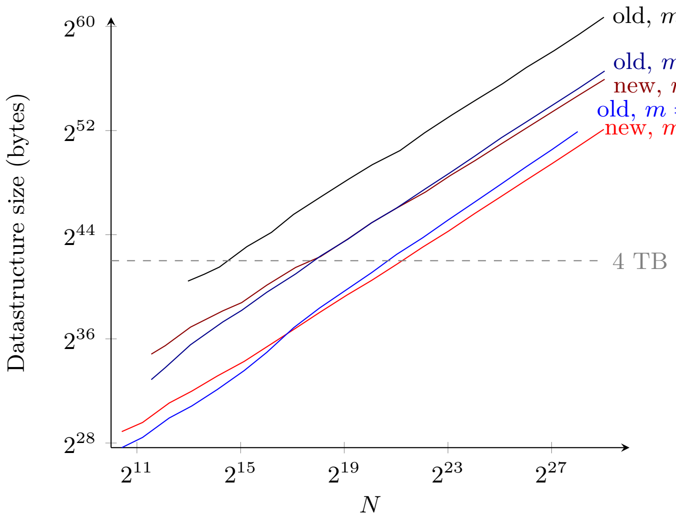

Fig. 3. Total size of the datastructure used by our implementation ("new" and [OPPW24] ("old", always performing two complete reduction steps) for different database sizes N and parameter choices for m. We also marked the 4 TB boundary, which is the amount of storage available on our system.

without limiting the number of "level 1 primes", i.e., performing two complete reduction steps as in our implementation.

Finally, we also count the exact number of read accesses and the datastructure size for N between 2^{10} and 2^{29}, and $m \in \{4, 5, 6\}$ (cf. Figure 2 and Figure 3, respectively). As we expect, the storage size for each $m \in \{4, 5, 6\}$ of our improved scheme is about the same as previously, while the read query count for $m \in \{4, 5\}$ very roughly corresponds to the read query count of [OPPW24] for $m' = m + 1$. In other words, we get one variable "for free". Asymptotically, as indicated by Prop. 4.3 and Alg. 3, our advantage scales approximately with $\frac{1}{m}N^{1/m}$.

We remark that [OPPW24] also used a simple model to estimate the runtime given the number of read access, based on a RAM access speed of about $7 \cdot 10^8$ IOPS and an SSD access speed of about 10^7 IOPS. However, we do not use this model, as due to different system specifications, it does not give accurate results in our setting.

Acknowledgements. We thank Martin R. Albrecht for discussing potential cryptanalytic approaches and the security of concrete parameter choices for our RLWE variant. We also thank Benjamin Curtis for his help with using the lattice estimator. Furthermore, we thank Damien Stehlé for providing valuable feedback on the first draft of our security reduction to RLWE with preprocessing. Simon Pohmann was supported by the EPSRC and the UK Government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/S021817/1).

References

[ABD16] M. Albrecht, S. Bai, and L. Ducas. "A subfield lattice attack on overstretched NTRU assumptions: Cryptanalysis of some FHE and graded encoding schemes". *Annual International Cryptology Conference*. 2016, pp. 153–178.

[APS15] M. R. Albrecht, R. Player, and S. Scott. "On the concrete hardness of learning with errors". In: *Journal of Mathematical Cryptology* 9.3 (2015), pp. 169–203.

[ACFP14] M. R. Albrecht, C. Cid, J.-C. Faugère, and L. Perret. *Algebraic Algorithms for LWE*. https://eprint.iacr.org/2014/1018. 2014. URL: https://eprint.iacr.org/2014/1018.

[ADPS16] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. "Post-quantum key {Exchange—A} new hope". *25th USENIX Security Symposium (USENIX Security 16)*. 2016, pp. 327–343.

[ACPS09] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems". *Advances in Cryptology – CRYPTO 2009*. Vol. 5677. Aug. 2009, pp. 595–618.

[AGKP14] F. Armknecht, T. Gagliardoni, S. Katzenbeisser, and A. Peter. "General Impossibility of Group Homomorphic Encryption in the Quantum World". *PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography*. Vol. 8383. Mar. 2014, pp. 556–573.

[AG11] S. Arora and R. Ge. "New Algorithms for Learning in Presence of Errors". *ICALP 2011: 38th International Colloquium on Automata, Languages and Programming, Part I*. Vol. 6755. July 2011, pp. 403–415.

[BBB+22] A. A. Badawi, J. Bates, F. Bergamaschi, D. B. Cousins, S. Erabelli, N. Genise, S. Halevi, H. Hunt, A. Kim, Y. Lee, Z. Liu, D. Micciancio, I. Quah, Y. Polyakov, S. R.V., K. Rohloff, J. Saylor, D. Suponitsky, M. Triplett, V. Vaikuntanathan, and V. Zucca. *OpenFHE: Open-Source Fully Homomorphic Encryption Library*. 2022. URL: https://eprint.iacr.org/2022/915.

[BN24] H. Bambury and P. Q. Nguyen. "Improved Provable Reduction of NTRU and Hypercubic Lattices". *International Conference on Post-Quantum Cryptography*. 2024, pp. 343–370.

[Ban93] W. Banaszczyk. "New bounds in some transference theorems in the geometry of numbers". In: *Mathematische Annalen* 296 (1993), pp. 625–635.

[BPR12] A. Banerjee, C. Peikert, and A. Rosen. "Pseudorandom Functions and Lattices". *Advances in Cryptology – EUROCRYPT 2012*. Vol. 7237. Apr. 2012, pp. 719–737.

[BIM00] A. Beimel, Y. Ishai, and T. Malkin. "Reducing the Servers Computation in Private Information Retrieval: PIR with Preprocessing". *Advances in Cryptology – CRYPTO 2000*. Vol. 1880. Aug. 2000, pp. 55–73.

[BKW03] A. Blum, A. Kalai, and H. Wasserman. "Noise-tolerant learning, the parity problem, and the statistical query model". In: *Journal of the ACM (JACM)* 50.4 (2003), pp. 506–519.

[BBPS19] M. Bolboceanu, Z. Brakerski, R. Perlman, and D. Sharma. "Order-LWE and the Hardness of Ring-LWE with Entropic Secrets". *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part II*. Vol. 11922. 2019, pp. 91–120. URL: https://doi.org/10.1007/978-3-030-34621-8%5C_4.

[BJRW23] K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen. "On the hardness of module learning with errors with short distributions". In: *Journal of Cryptology* 36.1 (2023), p. 1.

[BIPW17] E. Boyle, Y. Ishai, R. Pass, and M. Wootters. "Can We Access a Database Both Locally and Privately?" *TCC 2017: 15th Theory of Cryptography Conference, Part II*. Vol. 10678. Nov. 2017, pp. 662–693.

[Bra12] Z. Brakerski. "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP". *Advances in Cryptology – CRYPTO 2012*. Vol. 7417. Aug. 2012, pp. 868–886.

[BD20a] Z. Brakerski and N. Döttling. "Hardness of LWE on general entropic distributions". *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part II 30*. 2020, pp. 551–575.

[BD20b] Z. Brakerski and N. Döttling. "Lossiness and Entropic Hardness for Ring-LWE". *TCC 2020: 18th Theory of Cryptography Conference, Part I*. Vol. 12550. Nov. 2020, pp. 1–27.

[BGV12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping". *ITCS 2012: 3rd Innovations in Theoretical Computer Science*. Jan. 2012, pp. 309–325.

[BLP+13] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. "Classical hardness of learning with errors". *45th Annual ACM Symposium on Theory of Computing*. June 2013, pp. 575–584.

[BV11] Z. Brakerski and V. Vaikuntanathan. "Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages". *Advances in Cryptology – CRYPTO 2011*. Vol. 6841. Aug. 2011, pp. 505–524.

[BGPW16] J. A. Buchmann, F. Göpfert, R. Player, and T. Wunderer. "On the Hardness of LWE with Binary Error: Revisiting the Hybrid Lattice-Reduction and Meet-in-the-Middle Attack". *AFRICACRYPT 16: 8th International Conference on Cryptology in Africa*. Vol. 9646. Apr. 2016, pp. 24–43.

[CHR17] R. Canetti, J. Holmgren, and S. Richelson. "Towards Doubly Efficient Private Information Retrieval". *TCC 2017: 15th Theory of Cryptography Conference, Part II*. Vol. 10678. Nov. 2017, pp. 694–726.

[CCXY18] I. Cascudo, R. Cramer, C. Xing, and C. Yuan. "Amortized Complexity of Information-Theoretically Secure MPC Revisited". *Advances in Cryptology – CRYPTO 2018, Part III*. Vol. 10993. Aug. 2018, pp. 395–426.

[CL22] J. H. Cheon and K. Lee. "Limits of Polynomial Packings for $\mathbb{Z}_{p^k}$ and $\mathbb{F}_{p^k}$". *Advances in Cryptology – EUROCRYPT 2022, Part I*. Vol. 13275. May 2022, pp. 521–550.

[CGGI20] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. "TFHE: Fast Fully Homomorphic Encryption Over the Torus". In: *Journal of Cryptology* 33.1 (Jan. 2020), pp. 34–91.

[CHK22] H. Corrigan-Gibbs, A. Henzinger, and D. Kogan. "Single-Server Private Information Retrieval with Sublinear Amortized Time". *Advances in Cryptology – EUROCRYPT 2022, Part II*. Vol. 13276. May 2022, pp. 3–33.

[CK20] H. Corrigan-Gibbs and D. Kogan. "Private Information Retrieval with Sublinear Online Time". *Advances in Cryptology – EUROCRYPT 2020, Part I*. Vol. 12105. May 2020, pp. 44–75.

[DRS14] D. Dadush, O. Regev, and N. Stephens-Davidowitz. "On the closest vector problem with a distance guarantee". *2014 IEEE 29th Conference on Computational Complexity (CCC)*. 2014, pp. 98–109.

[DHMW24] F. Dong, Z. Hao, E. Mook, and D. Wichs. "Laconic function evaluation, functional encryption and obfuscation for rams with sublinear computation". *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. 2024, pp. 190–218.

[DM15] L. Ducas and D. Micciancio. "FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second". *Advances in Cryptology – EUROCRYPT 2015, Part I*. Vol. 9056. Apr. 2015, pp. 617–640.

[DW21] L. Ducas and W. van Woerden. "NTRU fatigue: how stretched is overstretched?" *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 IV 27*. 2021, pp. 3–32.

[EHL+23] D. Escudero, C. Hong, H. Liu, C. Xing, and C. Yuan. "Degree-D Reverse Multiplication-Friendly Embeddings: Constructions and Applications". *Advances in Cryptology – ASIACRYPT 2023, Part I*. Vol. 14438. Dec. 2023, pp. 106–138.

[FMS24] B. Falk, P. Mishra, and M. Shtepel. *Malicious Security for PIR (almost) for Free*. 2024. URL: https://eprint.iacr.org/2024/964.

[FV12] J. Fan and F. Vercauteren. *Somewhat Practical Fully Homomorphic Encryption*. 2012. URL: https://eprint.iacr.org/2012/144.

[Gen09] C. Gentry. "Fully homomorphic encryption using ideal lattices". *41st Annual ACM Symposium on Theory of Computing*. May 2009, pp. 169–178.

[GHS12] C. Gentry, S. Halevi, and N. P. Smart. *Homomorphic Evaluation of the AES Circuit*. 2012. URL: https://eprint.iacr.org/2012/099.

[GSW13] C. Gentry, A. Sahai, and B. Waters. "Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based". *Advances in Cryptology – CRYPTO 2013, Part I*. Vol. 8042. Aug. 2013, pp. 75–92.

[GLM+24] A. Ghoshal, B. Li, Y. Ma, C. Dai, and E. Shi. *Information-Theoretic Multi-Server PIR with Global Preprocessing*. 2024. URL: https://eprint.iacr.org/2024/765.

[GKPV10] S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. "Robustness of the Learning with Errors Assumption". *ICS 2010: 1st Innovations in Computer Science*. Jan. 2010, pp. 230–240.

[GJMS17] Q. Guo, T. Johansson, E. Mårtensson, and P. Stankovski. "Coded-BKW with sieving". *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 23*. 2017, pp. 323–346.

[GJMW19] Q. Guo, T. Johansson, E. Mårtensson, and P. S. Wagner. "On the asymptotics of solving the LWE problem using coded-BKW with sieving". In: *IEEE Transactions on Information Theory* 65.8 (2019), pp. 5243–5259.

[GJS15] Q. Guo, T. Johansson, and P. Stankovski. "Coded-BKW: Solving LWE using lattice codes". *Annual Cryptology Conference*. 2015, pp. 23–42.

[GMW21] Q. Guo, E. Mårtensson, and P. S. Wagner. "On the sample complexity of solving LWE using BKW-style algorithms". *2021 IEEE International Symposium on Information Theory (ISIT)*. 2021, pp. 2405–2410.

[HS20] S. Halevi and V. Shoup. *Design and implementation of HElib: a homomorphic encryption library*. 2020. URL: https://eprint.iacr.org/2020/1481.

[JLS24] A. Jain, H. Lin, and S. Saha. *A Systematic Study of Sparse LWE*. 2024.

[KU11] K. S. Kedlaya and C. Umans. "Fast polynomial factorization and modular composition". In: *SIAM Journal on Computing* 40.6 (2011), pp. 1767–1802.

[KF17] P. Kirchner and P.-A. Fouque. "Revisiting lattice attacks on overstretched NTRU parameters". *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. 2017, pp. 3–26.

[LLFP24] A. Lazzaretti, Z. Liu, B. Fisch, and C. Papamanthou. *Multi-Server Doubly Efficient PIR*. 2024. URL: https://eprint.iacr.org/2024/829.

[LWW20] H. Lin, Y. Wang, and M. Wang. "Hardness of Module-LWE and Ring-LWE on General Entropic Distributions." In: *IACR Cryptol. ePrint Arch.* 2020 (2020), p. 1238.

[LMW23] W.-K. Lin, E. Mook, and D. Wichs. "Doubly Efficient Private Information Retrieval and Fully Homomorphic RAM Computation from Ring LWE". *55th Annual ACM Symposium on Theory of Computing*. June 2023, pp. 595–608.

[LLM06] Y.-K. Liu, V. Lyubashevsky, and D. Micciancio. "On bounded distance decoding for general lattices". *International Workshop on Approximation Algorithms for Combinatorial Optimization*. 2006, pp. 450–461.

[LN13] M. Liu and P. Q. Nguyen. "Solving BDD by enumeration: An update". *Cryptographers' Track at the RSA Conference*. 2013, pp. 293–309.

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. "On Ideal Lattices and Learning with Errors over Rings". *Advances in Cryptology – EUROCRYPT 2010*. Vol. 6110. May 2010, pp. 1–23.

[MAT22] MATZOV. *Report on the Security of LWE: Improved Dual Lattice Attack*. 2022. URL: https://doi.org/10.5281/zenodo.6412487.

[May21] A. May. "How to Meet Ternary LWE Keys". *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part II*. Vol. 12826. 2021, pp. 701–731. URL: https://doi.org/10.1007/978-3-030-84245-1%5C_24.

[MW22] S. J. Menon and D. J. Wu. "SPIRAL: Fast, High-Rate Single-Server PIR via FHE Composition". *2022 IEEE Symposium on Security and Privacy*. May 2022, pp. 930–947.

[MR04] D. Micciancio and O. Regev. "Worst-case to average-case reductions based on Gaussian measures". *45th Annual IEEE Symposium on Foundations of Computer Science*. 2004, pp. 372–381.

[Mic01] D. Micciancio. "The hardness of the closest vector problem with preprocessing". In: *IEEE Transactions on Information Theory* 47.3 (2001), pp. 1212–1215.

[Mic18] D. Micciancio. *On the Hardness of Learning With Errors with Binary Secrets.* https://eprint.iacr.org/2018/988. 2018.

[MP13] D. Micciancio and C. Peikert. “Hardness of SIS and LWE with Small Parameters”. *Advances in Cryptology – CRYPTO 2013, Part I.* Vol. 8042. Aug. 2013, pp. 21–39.

[MR09] D. Micciancio and O. Regev. “Lattice-based cryptography”. *Post-quantum cryptography.* 2009, pp. 147–191.

[OPPW24] H. Okada, R. Player, S. Pohmann, and C. Weinert. “Towards Practical Doubly-Efficient Private Information Retrieval”. *Financial Cryptography.* 2024. URL: https://eprint.iacr.org/2023/1510.

[Pai99] P. Paillier. “Public-key cryptosystems based on composite degree residuosity classes”. *International conference on the theory and applications of cryptographic techniques.* 1999, pp. 223–238.

[Pla18] R. Player. “Parameter selection in lattice-based cryptography”. PhD thesis. Royal Holloway, University of London, 2018.

[Reg05] O. Regev. “On lattices, learning with errors, random linear codes, and cryptography”. *37th Annual ACM Symposium on Theory of Computing.* May 2005, pp. 84–93.

[SE94] C.-P. Schnorr and M. Euchner. “Lattice basis reduction: Improved practical algorithms and solving subset sum problems”. In: *Mathematical programming* 66 (1994), pp. 181–199.

[SSTX09] D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. “Efficient Public Key Encryption Based on Ideal Lattices”. *Advances in Cryptology – ASIACRYPT 2009.* Vol. 5912. Dec. 2009, pp. 617–635.

[vGHV10] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. “Fully Homomorphic Encryption over the Integers”. *Advances in Cryptology – EUROCRYPT 2010.* Vol. 6110. May 2010, pp. 24–43.

[WY05] D. Woodruff and S. Yekhanin. “A Geometric Approach to Information-Theoretic Private Information Retrieval”. In: *Electronic Colloquium on Computational Complexity (ECCC)* 37 (Jan. 2005).

[XZD+23] B. Xiang, J. Zhang, Y. Deng, Y. Dai, and D. Feng. “Fast blind rotation for bootstrapping FHEs”. *Annual International Cryptology Conference.* 2023, pp. 3–36.

[Zam22] Zama. *TFHE-rs: A Pure Rust Implementation of the TFHE Scheme for Boolean and Integer Arithmetics Over Encrypted Data.* 2022. URL: https://github.com/zama-ai/tfhe-rs.

[ZLTS23] M. Zhou, W.-K. Lin, Y. Tselekounis, and E. Shi. “Optimal Single-Server Private Information Retrieval”. *Advances in Cryptology – EUROCRYPT 2023, Part I.* Vol. 14004. Apr. 2023, pp. 395–425.

[ZPSZ23] M. Zhou, A. Park, E. Shi, and W. Zheng. “Piano: Extremely simple, single-server PIR with sublinear server computation”. In: *Cryptology ePrint Archive* (2023).