

Sample-Preserving Search-to-Decision Reduction for Noisy Linear Equations over Large Moduli

Andrej Bogdanov* Kel Zin Tan† Prashant Nalini Vasudevan‡

October 9, 2026

Abstract

The Noisy Linear Equations problem involves finding the solution to a random linear system over a finite field given noisy evaluations. This is a generalisation of various learning problems widely used in cryptography, including Learning with Errors (LWE), Learning with Rounding (LWR), and Learning Parity with Noise (LPN).

We present a new sample-preserving search-to-decision reduction for Noisy Linear Equations that has complexity $\text{poly}(n, m, \log q, q/\sigma)$, where n is the dimension of the secret, m is the number of samples, q is the modulus, and σ is the magnitude of the noise (defined suitably for the kind of noise involved). In particular, this implies search-to-decision reductions for LWE and LWR that run in time $\text{poly}(n)$ even if the modulus q is exponentially large, as long as the noise-to-modulus ratio σ/q is non-negligible. Our reduction works for prime moduli q . If q is composite, it instead produces a list that contains a small multiple of the solution.

Prior to this work, known search-to-decision reductions for LWE and LWR either did not preserve the parameters of the problem (i.e., m , n , σ , and q), or had complexity that scaled polynomially with either σ or the largest prime factor of q .

*  abogdano@uottawa.ca. University of Ottawa.

†  kelzin@u.nus.edu. Department of Computer Science, National University of Singapore.

‡  prashvas@nus.edu.sg. Department of Computer Science, National University of Singapore.

Contents

1	Introduction	1
1.1	Our Results	3
1.2	Technical Overview	4
1.3	Paper Outline	6
2	Preliminaries	6
2.1	Coprime secret	7
2.2	Noise Tolerance	8
2.3	Fourier Analysis	9
3	Reduction	10
3.1	Threshold Distinguisher	10
3.2	Terminal Hybrid	12
3.3	Correlation With Threshold Function	12
3.4	Heavy Fourier Coefficients	14
3.5	Fourier analysis of f	16
3.6	Learning Significant Fourier Coefficients	17
3.7	Proof of Theorem 3.1	18
4	Corollaries	18
4.1	Exact Secret Recovery	19
5	Acknowledgments	20
A	Supplementary Proofs	22
A.1	Hybrid Argument	22

1 Introduction

The *Noisy Linear Equations* problem asks to recover the solution to a random system of linear equations over $\mathbb{Z}_q$ given noisy outputs. Let $\mathcal{R} : \mathbb{Z}_q^m \rightarrow \mathbb{Z}_p^m$ be a “noise” function. (This function could be randomised, and p may or may not be the same as q .) Given a random matrix $A \in \mathbb{Z}_q^{m \times n}$ and the noisy evaluation $\mathcal{R}(Ax)$ for a random $x \in \mathbb{Z}_q^n$, the goal for the *Search* problem is to recover the solution x . For the *Decision* problem, similarly given A , the goal is to distinguish the noisy evaluations $\mathcal{R}(Ax)$ from a random vector in $\mathbb{Z}_p^m$.

This problem is clearly easy for some choices of the noise function $\mathcal{R}$, such as the identity function. It is believed to be hard for certain choices of the noise function, such as the following:

- $\mathcal{R}(v) = v + e \bmod q$, where each entry in e is drawn from the discrete Gaussian distribution. This is known as the *Learning With Errors (LWE)* problem [Reg09]. The hardness of LWE can be shown based on the worst-case hardness of lattice problems [Reg09, Pei09, BLP⁺13]
- $\mathcal{R}(v) = \lfloor p \cdot v/q \rfloor \bmod p$, where $p < q$ and $\lfloor \cdot \rfloor$ represents rounding to the nearest integer. This is called the *Learning With Rounding (LWR)* problem [BPR12], whose hardness follows from reductions from LWE [BPR12, AKPW13, BGM⁺16, AA16].
- $\mathcal{R}(v)_i = v_i + a_i \cdot b_i$ for each $i \in [m]$, where a_i is a random element of $\mathbb{Z}_q$ and b_i is a biased random bit. This is the generalised *Learning Parity with Noise (LPN)* problem [BFKL93].

The above problems are of considerable importance to cryptography, and form the basis of numerous constructions of advanced cryptographic primitives [Reg09, BV11, BPR12, BLMR13, BP14].

Search-to-Decision Reductions. In the regimes of interest to public-key cryptography, the planted solution x is typically unique. Search is thus at least as hard as decision. A search-to-decision reduction establishes their equivalence: It efficiently converts a distinguisher (decision algorithm) into a solution finder (search algorithm). Cryptographic applications such as public-key encryption [Reg09], fully homomorphic encryption [BV11], and structured pseudorandom functions [BPR12, BLMR13, BP14] assume decision hardness. A search-to-decision reduction increases our confidence in their security. In order for this equivalence to be as tight as possible, it is desirable for such reductions to be computationally efficient, preserve the number of samples and variables in the problem instance (i.e., the parameters m and n), and preserve the noise function $\mathcal{R}$.

Recall that in the LWE problem, the noise function $\mathcal{R}(v)$ corresponds to adding to each entry in v a sample (called the “error”) from a discrete Gaussian distribution, say of standard deviation σ , modulo q . The first search-to-decision reduction for LWE is from Regev’s work that originally defined the problem [Reg09]. This reduction preserves the error distribution, but requires the modulus q to be prime. It is also not sample-preserving, with the search algorithm needing a $\tilde{O}(1/\varepsilon^2)$ -factor of extra samples, where ε is the advantage of the distinguisher. Further, the complexity¹ of the reduction scales polynomially with the modulus q .

Several subsequent works have since extended this reduction to composite moduli q , under various conditions. Peikert [Pei09] obtained a reduction for $q = p_1 \cdots p_k$ (for distinct primes p_i) that has runtime scaling polynomially with $(\sum_i p_i)$, under the requirement that the advantage of the distinguisher ε is larger than $\Omega(\sum_i \exp(-p_i^2 \sigma^2))$. Applebaum et al. [ACPS09] obtained a

¹By “complexity”, we typically mean the product of the running time and the inverse of the success probability.

Work	Problem	Moduli	Noise	Samples	Complexity
[Reg09]	LWE	Prime	Preserving	$\tilde{O}(m/\varepsilon^2)$	$\text{poly}(q)$
[MP12]	LWE	$p_1^{e_1} \cdots p_k^{e_k}$	Preserving	$\tilde{O}(m/\varepsilon^2)$	$\text{poly}(\sum_i p_i + e_i)$
[BLP ⁺ 13]+[MP12]	LWE	All	$O(\sqrt{n})$ loss	$\tilde{O}(m/\varepsilon^2)$	$\text{poly}(\sum_i p'_i + e'_i)$
[MM11]	LWE	Prime Power	Preserving	Preserving	$\text{poly}(\sigma)$
[BLP ⁺ 13]+[MM11]	LWE	All	$O(\sqrt{n})$ loss	$m + \tilde{O}(n)$	$\text{poly}(\log q, \sigma)$
[BGM ⁺ 16]	LWR	Prime	Preserving	Preserving	$\text{poly}(q)$
This Work	LWE/LWR	Prime	Preserving	Preserving	$\text{poly}(\log q, q/\sigma)$

Table 1: Comparison of search-to-decision reductions. n is the dimension of the secret, m is the number of samples, σ is the noise magnitude (ratio of moduli in case of LWR), q is the modulus, and ε is the advantage of the decision algorithm. Complexity omits $\text{poly}(m, n, 1/\varepsilon)$ factors. For reductions involving [BLP⁺13], the reduction is from Search LWE with a possibly different modulus $q' = p_1^{e'_1} \cdots p_k^{e'_k}$.

reduction for $q = p^e$ (for prime p) that has runtime scaling polynomially with p and e . Micciancio and Peikert [MP12] combined both results and obtained a reduction for $q = p_1^{e_1} \cdots p_k^{e_k}$ (for distinct primes p_i) that has runtime scaling polynomially with $\sum(p_i + e_i)$ and a similar requirement on the distinguisher's advantage: $\varepsilon > \Omega(\sum_i \exp(-p_i^{2e_i} \sigma^2 / q^2))$. All of these reductions also have a $\tilde{O}(1/\varepsilon^2)$ -factor increase in the number of samples needed by the search algorithm.

Later work by Brakerski et al. [BLP⁺13] showed how to reduce Decision LWE over a modulus q' to Decision LWE over any modulus $q \leq q'$, with runtime scaling only polylogarithmically in q' . Combined with the other reductions mentioned above, this enables reductions from Search LWE over a nice q' (that has efficient search-to-decision reductions) to Decision LWE over any $q \leq q'$.

$$\text{Search-LWE}_{m',q',\sigma'} \implies \text{Decision-LWE}_{m',q',\sigma'} \implies \text{Decision-LWE}_{m,q,\sigma}$$

The price of Brakerski et al.'s reduction is that the noise-to-modulus ratio of the resulting instance is larger: $\sigma/q = \Omega(\sqrt{n} \cdot \sigma'/q')$. The number of variables remains the same, while the number of samples decreases by $\Omega(n + \log \log q)$ (i.e., $m = m' - \Omega(n + \log \log q)$).

Most relevant to our work is the search-to-decision reduction of Micciancio and Mol [MM11], which is the only known sample-preserving classical reduction for LWE.² Their reduction avoids the dependence on q and has runtime scaling polynomially with the noise magnitude σ instead. In the special case when q is of the form p^e and the noise is restricted to be at most $(p/2)$ in each element, the complexity reduces to linear in p (and polylogarithmic in q). More recently, Sudo et al. [SHTY25] showed a quantum sample-preserving search-to-decision reduction for LWE that uses only two queries to the distinguisher, but succeeds only with probability inverse polynomial in q .

Using techniques related to those of Micciancio and Mol, Bogdanov et al. [BGM⁺16] effectively showed the only known search-to-decision reduction for LWR, which has complexity polynomial in the modulus q .³

Limitations of Existing Reductions. The easy reduction from decision to search for Noisy Linear Equations preserves all parameters: secret dimension, number of samples, error magnitude,

²The sample-preserving reduction of Applebaum et al. for LPN [AIK09] is based on similar ideas.

³The focus of their work is different, but this result is implicit in the proof of their Lemmas 3 and 4.

and modulus. To establish *equivalence* of search and decision, so should the reverse reduction. As described above, all currently known reductions either fail to preserve one or more of these, or else have running time that is polynomial in the modulus or error magnitude (and hence are not efficient when these are large). The features of these reductions are summarised in Table 1.

1.1 Our Results

We construct a new parameter-preserving search-to-decision reduction for the Noisy Linear Equations problem for noise functions that are robust to ℓ_∞ perturbations (see below for details). Where it applies, our reduction is sample-preserving and, roughly speaking, its complexity scales with the noise-to-modulus ratio rather than with the magnitude of the noise or the modulus. Among other corollaries (see Section 4), our results imply the following for any prime modulus q :

- Search-to-decision reduction for LWE that preserves the dimension of the secret n , number of samples m , modulus q , and error magnitude σ , with running time that is $\text{poly}(n, m, \log q, q/\sigma)$
- Search-to-decision reduction for LWR that preserves the dimension of the secret n , number of samples m , modulus q , and rounding modulus p , with running time that is $\text{poly}(n, m, \log q, p)$

In particular, our reduction for LWE runs in polynomial time even if the modulus q is exponentially large, as long as the error rate σ/q is inverse-polynomial.

More generally, our reduction applies to Noisy Linear Equations problems with a noise function $\mathcal{R}$ that can tolerate small perturbations. We say $\mathcal{R}$ is *B-smooth* if for every v of ℓ_∞ -norm at most B , with sufficiently high probability over the choice of A , the distribution of $\mathcal{R}(Ax + v)$ is very close to that of $\mathcal{R}(Ax)$. LWE is smooth when B is somewhat smaller than the error magnitude σ . LWR is smooth when the B is somewhat smaller than the ratio of the moduli q/p . (See Section 2 for the relevant formal definitions and statements.)

Theorem 1.1 (Informal, see Theorem 3.1 and Section 4). *Consider any $n, m, p, q \in \mathbb{Z}^+$, with q being prime, and a B -smooth noise function $\mathcal{R} : \mathbb{Z}_q^m \rightarrow \mathbb{Z}_p^m$. For the Noisy Linear Equations problem over $\mathbb{Z}_q$ with sample count m , secret dimension n , and noise function $\mathcal{R}$, there exists an algorithm that, given a decision algorithm that runs in time T and has advantage ε , solves the search problem in time $\text{poly}(n, m, \log q, q/B\varepsilon, T)$ with success probability $\tilde{\Omega}(B\varepsilon/q)$.*

Beyond prime moduli, our reductions also work as above for moduli q that do not have any small prime factors (see Section 4 for details). Our reductions also work for general moduli q in some special cases, such as the entries in the secret vector being very small.

Otherwise, for general q , our search algorithm instead produces a list of size $\tilde{O}((B/q\varepsilon)^2)$ of candidate vectors that contains a vector $c \cdot x$ for some $0 < |c| \leq \tilde{O}((B/q\varepsilon)^2)$, where x is the secret vector. If c turns out to be coprime with q , then by trial and error over all elements in the list and all c , we would be able to recover the secret if equipped with an algorithm to verify the solution.

When c is not coprime to q , the output of our reduction reveals only partial information about the secret. This is, in fact, a limitation of all known parameter-preserving reductions for the following reason. When q is not prime, an LWE distinguisher could censor substantial information about the secret. For example, if the modulus q is even, the distinguisher might base its decision only on the value of the instance reduced modulo $q/2$. The output of the distinguisher would then be identical for every pair of secrets congruent modulo $q/2$. All known parameter-preserving

reductions, including ours, either leave the secret intact or modify it by an affine transformation. Such reductions can therefore at best recover the secret up to congruencies modulo small factors of q . Bogdanov et al. [BGM⁺16, Lemma 4] point out a related limitation in a step of their reduction from search LWE to decision LWR.

Open Problems. Neither our algorithm nor that of Micciancio and Mol [MM11] runs in time $\text{poly}(n)$ in a parameter setting where the error magnitude is both $\omega(\text{poly}(n))$ and $o(q/\text{poly}(n))$. Ideally, we would like to have a parameter-preserving reduction whose complexity scales only with $\text{polylog}(q)$ for all noise parameter settings. We do not see a fundamental barrier to such a reduction, but it seems out of the reach of existing techniques. The other open problem is to perform exact secret recovery reduction when q is an arbitrary composite modulus, without any restrictions on the secret x .

1.2 Technical Overview

In this subsection, we first analyse why most existing reductions have a complexity that scales linearly with the size of the modulus. Then we discuss how [MM11] achieves a sample-preserving reduction that scales with the size of the error. Finally, we introduce our reduction technique.

Prior Reductions. Regev’s initial reduction for LWE [Reg09] uses a coordinate-guessing technique to recover the secret. Make a guess $g \in \mathbb{Z}_q$ for the first co-ordinate of secret x_1 . For each sample $(a, \langle a, x \rangle + e)$, it is modified to $(a + (b, 0, \dots, 0), \langle a, x \rangle + e + b \cdot g)$ where b is randomly sampled from $\mathbb{Z}_q$. If $g = x_1$, then the distribution stays the same. Otherwise, it becomes random. Because the reduction has to iterate through all possible guesses, the reduction complexity grows with the size of the domain. Later works by [Pei09, ACPS09, MP12] extend this technique to support larger modulus $q = p_1^{e_1} \cdots p_k^{e_k}$. The rough idea is to use the Chinese Remainder Theorem and guess the value of the secret modulo the small primes. Therefore, the complexity of the reduction has a polynomial dependence on $\sum_i (p_i + e_i)$.

Micciancio and Mol obtain a sample-preserving reduction [MM11] using an approach analogous to the sample-preserving reduction for LPN with $q = 2$ by Applebaum, Ishai and Kushilevitz [AIK09]. The main idea is to use the distinguisher to construct a weak inner product predictor $P(y)$ that “predicts” $\langle y, x \rangle \bmod q$, where $y \in \mathbb{Z}_q^n$ is an input and x is the secret. This can be done by transforming $(A, Ax + e)$ to $(A + ry^\top, Ax + e + rg)$ where r is a random vector from $\mathbb{Z}_q^m$ and $g \in \mathbb{F}_q$ is a guess for the value $\langle y, x \rangle$. If $g = y^\top x$, then the distribution stays the same. Otherwise, it becomes random. The predictor P performs this transformation, runs the LWE distinguisher on the resulting instance, and outputs whatever it outputs. They show that this weak predictor $P(y)$ has significant Fourier weight at x , and use the learning algorithm from Akavia, Goldwasser and Safra [AGS03] to recover its heavy Fourier coefficients, and thus the secret.

Making a guess for the value of $\langle y, x \rangle \bmod q$ degrades the advantage by $1/q$, which causes the overall learning algorithm to have a running time polynomial in q . To avoid this polynomial dependency on the modulus, they use the equivalence of LWE and Knapsack and instead construct a predictor for $\langle y, e \rangle$, where e is the error vector. The domain of the predictor $P(y)$ is changed to $y \in \mathbb{Z}_p^m$, where p is a prime and $p \geq \max e_i$, the output is then $P(y) = \langle y, e \rangle \bmod p$. This prime p scales with the error size and not with the original modulus q . When each term of the y_i is at most p , the predictor guessing range for $g = \langle y, e \rangle$ (without modulo p) is reduced to $p \cdot m \cdot \max_i e_i$.

They will make a guess $g \in [0, p \cdot m \cdot \max_i e_i]$ for the value of $\langle y, e \rangle$ without modulo p . The loss in advantage is then a factor of $1/\text{poly}(p, m)$. Running the learning algorithm on such a predictor has running time polynomial in $\text{poly}(p, \log q)$.

Ultimately, existing known reductions involve iterating through all possible guesses for some secret value, which incurs a reduction complexity that scales linearly with the size of the domain of this value (q in the case of [Reg09], $|e|_\infty$ in the case of [MM11], etc.). If iteration through the domain is used in the reduction, it will be difficult to obtain a reduction that runs in time polylogarithmic in the size of the modulus, secret and error.

Threshold Distinguisher. To obtain a better dependence on q , we would need to use the single-bit output of the distinguisher more effectively than to rule out just one possible guess each time. Ideally, we hope that a single query to the distinguisher would split the search space in half, similar to a binary search.

Consider a distinguishing algorithm D for the Noisy Linear Equation problem that takes as input $(A, \mathcal{R}(Ax))$ for some $A \in \mathbb{Z}_q^{m \times n}$ and $x \in \mathbb{Z}_q^n$, and outputs a bit from $\{-1, 1\}$. We use this to construct an algorithm P that takes as input $(A, \mathcal{R}(Ax))$ and also a vector $y \in \mathbb{Z}_q^n$, and somehow signals whether $|\langle y, x \rangle_q| \leq B$ or $|\langle y, x \rangle_q| > B$, for some parameter B . The closer B is to $q/2$, the better, because that reveals more information about inner product $\langle y, x \rangle$, and thus the secret x .

To see how the construction is done, consider taking the input $(A, \mathcal{R}(Ax))$ and adding $vy^\top \in \mathbb{Z}_q^{m \times n}$ to the matrix, where $v \leftarrow \{0, 1\}^m$. In other words, the input becomes

$$(A + vy^\top, \mathcal{R}(Ax))$$

Since A is random, this distribution is essentially

$$(A + vy^\top, \mathcal{R}(Ax)) \equiv (A, \mathcal{R}(Ax - vy^\top x))$$

If $y^\top x = |\langle y, x \rangle_q| \leq B$ is small, this is like adding a small vector to the already noisy $\mathcal{R}(Ax)$. If the noise function $\mathcal{R}$ absorbs it, then the distribution is close to the noisy linear equations distribution.

$$(A, \mathcal{R}(Ax - vy^\top x)) \approx (A, \mathcal{R}(Ax))$$

Otherwise, if $y^\top x = |\langle y, x \rangle_q|$ is large, it overwhelms the noise. And if we add many such $vy^\top$ to A , it essentially becomes randomised and independent of $\mathcal{R}(Ax)$.

Unfortunately, we cannot afford to set $B = q/2$ because it will overwhelm the noise in most natural noise distributions. The actual B that we use is the largest B such that for any vector z whose elements have magnitude at most B , the statistical distance between $\mathcal{R}(Ax + z)$ and $\mathcal{R}(Ax)$ is bounded by $O(\varepsilon)$.

In the proof, starting with the observation about adding $vy^\top$ mentioned above, we will use a hybrid argument to construct the algorithm P . Informally, what we will show is the following (Section 3.1):

$$\left| \Pr[P(y, A, \mathcal{R}(Ax)) = 1 \mid |\langle y, x \rangle_q| \leq B] - \Pr[P(y, A, \mathcal{R}(Ax)) = 1 \mid |\langle y, x \rangle_q| > B] \right| \geq \tilde{\Omega}(\varepsilon)$$

Where $y \leftarrow \mathbb{Z}_q^n$, $A \leftarrow \mathbb{Z}_q^{m \times n}$, $x \leftarrow \mathbb{Z}_q^n$

Heavy Fourier Coefficients. Just having P is still not enough to recover the secret x . As we would like to have a sample-preserving reduction, we use an idea similar to [MM11, BGM⁺16], where the information about x is obtained by looking into the Fourier coefficients of P . We show that for a good fixed input $(A, x, \mathcal{R}(Ax))$, the function $P(y) = P(y, A, \mathcal{R}(Ax))$ has a heavy Fourier coefficient at some character $c \cdot x$, where c is a small integer and x is the secret of the noisy linear equations. We can then recover a list containing $c \cdot x$ with the algorithm of Akavia, Goldwasser, and Safra [AGS03] (Section 3.6).

To analyse the Fourier spectrum of P , we use an intermediate threshold function S to isolate the important Fourier characters (Definition 3.6). The function S is defined as

$$S(y) = \begin{cases} -1 & \text{if } |\langle y, x \rangle_q| \leq B \\ 1 & \text{otherwise} \end{cases}$$

with y being the input vector, and x being the secret of the noisy linear equations. Because of the properties of P described earlier, we can show that P is correlated with S (Section 3.3).

$$\mathbf{E}[(P(y) - \mu_P)(S(y) - \mu_S)] \geq \tilde{\Omega}\left(\frac{B\varepsilon}{q}\right)$$

where $\mu_P = \mathbf{E}[P]$ and $\mu_S = \mathbf{E}[S]$. This correlation implies that P must have some non-negligible Fourier weight on the heavy Fourier characters of S .

By definition of S , it only has non-zero Fourier coefficients at multiples of x . Furthermore, we show that the Fourier weight of S concentrates on the characters $c \cdot x$ where c is small (Section 3.4). Due to their correlation, this implies that P also contains at least one heavy Fourier coefficient at a character $c \cdot x$ where c is small.

Recovering the Secret Finally, if q is a prime or c turns out to be coprime with q , we can go through every element in the output list and try every possible c . Since the list and $|c|$ are bounded by $\text{poly}(q/B\varepsilon)$, this process finishes in time that grows polynomially with $q/B\varepsilon$. If a verification algorithm exists, we can check each answer candidate and return the correct one. Alternatively, randomly pick an element from the list and choose a random c , incurring some loss in the success probability.

1.3 Paper Outline

In Section 2, we present the notation and technical tools used throughout the paper. In Section 3, we prove the main theorem. In Section 4, we derive various corollaries of our main theorem, including the reductions for LWE and LWR.

2 Preliminaries

Notation We use x_i to denote the i -th bit of the binary string (vector) x and A_i to denote the i -th row of the matrix A . Let $\mathbb{F}_p$ denote the order- p prime field. Let $\mathbb{Z}_q$ denote the integer ring modulo q . Write $[n] := \{1, 2, \dots, n\}$. Write $\lfloor x \rfloor$ as the number x rounded to the nearest integer. Write $\langle y, x \rangle_q \in \mathbb{Z}_q$ as the inner product of vectors x and y modulo q . Define $|x|$, where $x \in \mathbb{Z}_q$, as $|x| = \min(x', x' - q)$ where x' is x lifted to the integer ring.

Distributions Denote $D_1 \equiv D_2$ if the distributions D_1 and D_2 are the same. Let $U(D)$ be the uniform distribution on the set D . The notation $x \leftarrow D$ means that x is sampled from the distribution D ; when D is a set (we abuse notation), it means x is sampled from $U(D)$.

Definition 2.1 (Discrete Gaussian Distribution). Define D_σ as the discrete Gaussian distribution over integers with parameter $\sigma > 0$ and centre at 0.

$$\rho_\sigma(x) = \exp(-x^2/2\sigma^2)$$

$$\Pr[D_\sigma = x] = \rho_\sigma(x) / \sum_{y \in \mathbb{Z}} \rho_\sigma(y)$$

Definition 2.2 (Noisy Linear Equations). Let $n, m, p, q \in \mathbb{Z}^+$. For sample count m , secret dimension n , and a noise function $\mathcal{R} : \mathbb{Z}_q^m \rightarrow \mathbb{Z}_p^m$, algorithms S and D are said to have advantage $\varepsilon \in [0, 1]$ in solving the Search and Decision Noisy Linear Equations problem, respectively, if they satisfy the following conditions:

- Search: Suppose $A \leftarrow \mathbb{Z}_q^{m \times n}$, $x \leftarrow \mathbb{Z}_q^n$, S recovers the secret x with probability at least ε :

$$\Pr[S(A, \mathcal{R}(Ax)) = x] \geq \varepsilon.$$

- Decision: Suppose $A \leftarrow \mathbb{Z}_q^{m \times n}$, $x \leftarrow \mathbb{Z}_q^n$, $b \leftarrow \mathbb{Z}_q^m$, D distinguishes samples from uniform with advantage at least ε :

$$|\Pr[D(A, \mathcal{R}(Ax)) = 1] - \Pr[D(A, \mathcal{R}(b)) = 1]| \geq \varepsilon.$$

Definition 2.3 (Learning With Error (LWE)). The Learning With Error (LWE) problem with noise parameter σ is a Noisy Linear Equations problem with the noise function $\mathcal{R} : \mathbb{Z}_q^m \rightarrow \mathbb{Z}_q^m$ defined as $\mathcal{R}(y) = (y + e) \bmod q$, where $e \leftarrow D_\sigma^m$.

Definition 2.4 (Learning With Rounding (LWR)). The Learning With Rounding (LWR) problem with modulus $p < q$ is a Noisy Linear Equations problem with the noise function $\mathcal{R} : \mathbb{Z}_q^m \rightarrow \mathbb{Z}_p^m$ defined, for each $i \in [m]$, as $\mathcal{R}(y)_i = \left\lfloor \frac{p}{q} y'_i \right\rfloor \bmod p$, where y'_i is y_i lifted to the reals.

2.1 Coprime secret

Throughout the proof, we will only consider secrets of the noisy linear equations (that is, the vector x in Definition 2.2) that satisfy $\gcd(x, q) = \gcd(x_1, \dots, x_n, q) = 1$. This is a mild assumption when we sample x at random because the event of obtaining such a secret occurs with very high probability. We can precisely calculate the probability of this event using the Chinese Remainder Theorem.

$$\Pr[\gcd(x_1, \dots, x_n, q) = 1] = \prod_{\substack{p|q \\ p \text{ is prime}}} \left(1 - \frac{1}{p^n}\right) \geq \left(1 - \frac{1}{2^n}\right)^{\log q} \geq 1 - \frac{\log q}{2^n}$$

which is a value that is exponentially close to 1.

Definition 2.5. Define $\mathcal{P}(\mathbb{Z}_q^n) = \{x \in \mathbb{Z}_q^n \mid \gcd(x_1, \dots, x_n, q) = 1\}$

Proposition 2.6. For any $q \in \mathbb{Z}_{\geq 2}$, $TV(U(\mathcal{P}(\mathbb{Z}_q^n)), U(\mathbb{Z}_q^n)) \leq \frac{\log q}{2^n}$

2.2 Noise Tolerance

In this subsection, we show that both LWE noise and LWR noise can absorb small deviations in the noisy codeword (which is the term we shall use to refer to Ax). Specifically, we will be adding a vector $v \leftarrow \{0, B\}^m$ to the noisy codeword of LWE and LWR, where B is something much smaller than the noise magnitude. We show that the total variation distance to the original distribution will still be close despite adding v . This property is essential for our search-to-decision reduction.

Definition 2.7 (Noise Tolerance). *Let $n, m, p, q, B \in \mathbb{Z}^+$ and $\varepsilon \in [0, 1]$. Let $\mathcal{R} : \mathbb{Z}_q^m \rightarrow \mathbb{Z}_p^m$ be a noise function. We say that $\mathcal{R}$ is (B, ε) -noise tolerant if for every $|b| \leq B$ and every $v \in \{0, b\}^m$,*

$$TV\left((A, \mathcal{R}(Ax)), (A, \mathcal{R}(Ax + v))\right) \leq \varepsilon$$

where $A \leftarrow \mathbb{Z}_q^{m \times n}$ and $x \leftarrow \mathcal{P}(\mathbb{Z}_q^n)$.

Let $\mathcal{B}'(\mathcal{R}, \varepsilon) \in \mathbb{Z}$ be the largest B such that $\mathcal{R}$ is (B, ε) -noise tolerant. The tolerance bound function is defined as $\mathcal{B}(\mathcal{R}, \varepsilon) = \min(\mathcal{B}'(\mathcal{R}, \varepsilon), \lfloor q/4 - 1 \rfloor)$.

Proposition 2.8 (LWE Noise Tolerance). *Let $n, m, q, B \in \mathbb{Z}^+$, $\sigma \in \mathbb{R}$, and $\mathcal{R}$ be the LWE noise function with noise parameter σ . Suppose $A \in \mathbb{Z}_q^{m \times n}$, $x \in \mathbb{Z}_q^n$, $v \in \{0, b\}^m$ where $|b| \leq B$, then*

$$TV\left((A, \mathcal{R}(Ax)), (A, \mathcal{R}(Ax + v))\right) \leq \frac{B\sqrt{m}}{2\sigma}$$

Which implies $\mathcal{B}(\mathcal{R}, \varepsilon) \geq \min(2\varepsilon\sigma/\sqrt{m}, \lfloor q/4 - 1 \rfloor)$.

Proof. Expanding the noise function $\mathcal{R}$, then by data processing inequality

$$\begin{aligned} TV\left((A, \mathcal{R}(Ax)), (A, \mathcal{R}(Ax + v))\right) &= TV\left((A, Ax + e), (A, Ax + e + v)\right) \\ &\leq TV(e, e + v) \end{aligned}$$

Where $e \leftarrow D_\sigma^m$ and addition is done over $\mathbb{Z}$.

Without loss of generality, assume that $b \geq 0$. Looking at a single dimension $i \in [m]$ such that $v_i = b$, and write down the KL divergence

$$\begin{aligned} D_{KL}(b + e_i \parallel e_i) &= \sum_{z \in \mathbb{Z}} \Pr[b + e_i = z] \log \frac{\Pr[b + e_i = z]}{\Pr[e_i = z]} \\ &= \sum_{z \in \mathbb{Z}} \Pr[b + e_i = z] \log \exp\left(\frac{(z)^2}{2\sigma^2} - \frac{(z - b)^2}{2\sigma^2}\right) \\ &= \sum_{z \in \mathbb{Z}} \Pr[b + e_i = z] \frac{2bz - b^2}{2\sigma^2} \end{aligned}$$

Let $Y = b + e_i$, and simplify the statement above

$$\sum_z \Pr[Y = z] \frac{2bz - b^2}{2\sigma^2} = \mathbf{E}_{Y \sim b + D_\sigma} \left[\frac{2bY - b^2}{2\sigma^2} \right] = \frac{2b \mathbf{E}[Y] - b^2}{2\sigma^2} = \frac{b^2}{2\sigma^2} \leq \frac{B^2}{2\sigma^2}$$

For m dimensions, Pinsker's Inequality bounds the total variation by

$$TV(e + v, e) \leq \sqrt{\frac{1}{2} D_{KL}(e + v \parallel e)} = \sqrt{\frac{m}{2} D_{KL}(e_i + v_i \parallel e_i)} \leq \frac{B\sqrt{m}}{2\sigma}. \quad \square$$

Proposition 2.9 (LWR Noise Tolerance). *Let $n, m, p, q \in \mathbb{Z}^+$, $p < q$ and $\mathcal{R}$ be the LWR noise function with rounding modulus p . Suppose $A \leftarrow \mathbb{Z}_q^{m \times n}$, $x \in \mathcal{P}(\mathbb{Z}_q^n)$, $v \in \{0, b\}^m$ where $|b| \leq B < q/p$, then*

$$TV\left((A, \mathcal{R}(Ax)), (A, \mathcal{R}(Ax + v))\right) \leq \frac{pBm}{q}$$

Which implies $\mathcal{B}(\mathcal{R}, \varepsilon) \geq \min(q\varepsilon/pm, \lfloor q/4 - 1 \rfloor)$.

Proof. Without loss of generality, assume that $b \geq 0$. Let E be the event that $\mathcal{R}(Ax) = \mathcal{R}(Ax + v)$. This event happens when every value in Ax is far away from the boundary where rounding occurs. In other words, for each coordinate value $(Ax)_i$, lifted to the reals $(Ax)'_i$, we have that $\left\lfloor \frac{p}{q}(Ax)'_i \right\rfloor \equiv \left\lfloor \frac{p}{q}((Ax)'_i + b) \right\rfloor \pmod{p}$. The total variation distance can then be upper-bounded using this event.

$$TV\left((A, \mathcal{R}(Ax)), (A, \mathcal{R}(Ax + v))\right) \leq \Pr[\bar{E}]$$

As $\gcd(x, q) = 1$ and A is randomly sampled, Ax is distributed just as a random vector over $\mathbb{Z}_q^m$. For a list of elements in $\mathbb{Z}_q$ that round to the same element in $\mathbb{Z}_p$, only the last b elements in the list would round to a different element after adding the value b , while the rest stay the same.

This means that at most pB elements in $\mathbb{Z}_q$ will change value after addition. So for a random element in $\mathbb{Z}_q$ the probability that it falls into this boundary region is at most pB/q . By union bound, the probability of $\bar{E}$ is upper bounded.

$$TV\left((A, \mathcal{R}(Ax)), (A, \mathcal{R}(Ax + v))\right) \leq \Pr[\bar{E}] \leq \frac{pBm}{q}. \quad \square$$

2.3 Fourier Analysis

Here we describe the basics of Fourier Analysis and establish the choice of basis in the paper. Denote $\omega_q = e^{-2\pi i/q}$.

Definition 2.10 (Fourier Transform). *Let $n, q \in \mathbb{Z}^+$. The Fourier transform of a function $S : \mathbb{F}_q^n \rightarrow \mathbb{C}$ is another function $\hat{S} : \mathbb{F}_q^n \rightarrow \mathbb{C}$ defined as*

$$\hat{S}(\chi) = \mathbf{E}_{y \leftarrow \mathbb{Z}_q^n} \left[S(y) \omega_q^{\langle \chi, y \rangle} \right]$$

Proposition 2.11 (Inverse Fourier Transform). *Let $\hat{S}$ be the Fourier transform of $S : \mathbb{F}_q^n \rightarrow \mathbb{C}$. Then*

$$S(y) = \sum_{\chi \in \mathbb{Z}_q^n} \hat{S}(\chi) \omega_q^{-\langle \chi, y \rangle}$$

Proposition 2.12 (Parseval–Plancherel). *Let $n, q \in \mathbb{Z}^+$, functions $P, S : \mathbb{F}_q^m \rightarrow \mathbb{C}$*

$$\begin{aligned} \mathbf{E}_{y \leftarrow \mathbb{Z}_q^n} \left[P(y) \overline{S(y)} \right] &= \mathbf{E}_{y \leftarrow \mathbb{Z}_q^n} \left[\left(\sum_{\chi \in \mathbb{Z}_q^m} \hat{P}(\chi) \omega_q^{-\langle \chi, y \rangle} \right) \left(\sum_{\chi \in \mathbb{Z}_q^m} \overline{\hat{S}(\chi)} \omega_q^{\langle \chi, y \rangle} \right) \right] \\ &= \sum_{\chi \in \mathbb{Z}_q^m} \hat{P}(\chi) \overline{\hat{S}(\chi)} \end{aligned}$$

In particular, this implies the following for any such functions P

$$\mathbf{E}_{y \leftarrow \mathbb{Z}_q^n} [|P(y)|^2] = \sum_{\chi \in \mathbb{Z}_q^m} |\hat{P}(\chi)|^2$$

3 Reduction

This section describes the proof of the main theorem.

Theorem 3.1. *Consider any $n, m, p, q \in \mathbb{Z}^+$, a noise function $\mathcal{R} : \mathbb{Z}_q^m \rightarrow \mathbb{Z}_p^m$, and any $\varepsilon \in (0, 1]$. Let $t = \lceil 4 \log(qm/\varepsilon) \rceil$, denote by B the noise tolerance bound $\mathcal{B}(\mathcal{R}, \varepsilon/4t)$, and set $\varepsilon' = (B\varepsilon/2qt)$.*

Suppose there exists an algorithm for the Decision Noisy Linear Equations problem over $\mathbb{Z}_q$ with sample count m , secret dimension n , and noise function $\mathcal{R}$, and that the algorithm runs in time T and has advantage ε .

Then there exists an algorithm as follows. Given an input $(A, \mathcal{R}(Ax))$, where $A \leftarrow \mathbb{Z}_q^{m \times n}$ and $x \leftarrow \mathcal{P}(\mathbb{Z}_q^n)$, it outputs a list of size $\tilde{O}(1/(\varepsilon')^2)$ that contains $c \cdot x$ for some $c \in \mathbb{Z}_q \setminus \{0\}$ such that $|c| \leq \lceil 32/(\varepsilon')^2 \rceil$. This algorithm runs in time $\text{poly}(n, m, \log q, 1/\varepsilon', T)$ and successfully outputs such a list with probability $\Omega(\varepsilon')$.

Proof overview is described in Section 1.2

Section Outline In Section 3.1 and Section 3.2, we will talk about the threshold distinguisher. Section 3.3 is on the correlation of P and S . Section 3.4 and Section 3.5 are on the concentration of the Fourier weight of S . Finally, in Section 3.6 we state the algorithm from [AGS03] to learn heavy Fourier coefficients.

3.1 Threshold Distinguisher

In this subsection, we describe the construction of a threshold distinguisher P that can distinguish whether $|\langle y, x \rangle_q| \leq B$ or $|\langle y, x \rangle_q| > B$. Note that the inner product is performed over $\mathbb{Z}_q$.

Lemma 3.2. *Let $n, m, p, q \in \mathbb{Z}^+$, $\varepsilon \in (0, 1]$, a noise function $\mathcal{R} : \mathbb{Z}_q^m \rightarrow \mathbb{Z}_p^m$, $t = \lceil 4 \log(qm/\varepsilon) \rceil$, $B = \mathcal{B}(\mathcal{R}, \varepsilon/4t)$. Suppose there exists an algorithm D for the Decision Noisy Linear Equations problem over $\mathbb{Z}_q$ with sample count m , secret dimension n , noise function $\mathcal{R}$, that has advantage ε .*

Then there exists an algorithm P that outputs a value in $\{-1, 1\}$ such that

$$\left| \Pr[P(y, A, \mathcal{R}(Ax)) = 1 \mid |\langle y, x \rangle_q| \leq B] - \Pr[P(y, A, \mathcal{R}(Ax)) = 1 \mid |\langle y, x \rangle_q| > B] \right| \geq \varepsilon/4t$$

Where $y \leftarrow \mathbb{Z}_q^n$, $A \leftarrow \mathbb{Z}_q^{m \times n}$, $x \leftarrow \mathcal{P}(\mathbb{Z}_q^n)$.

Proof. The construction of P is as follows:

Algorithm P: On input $(y, A, b = \mathcal{R}(Ax))$ and Decision algorithm D ,

- 1 Sample $k \leftarrow [0, t-1]$ and $v \leftarrow \{0, 1\}^m$
- 2 Sample matrices $V_k \leftarrow \{0, 1\}^{m \times k}$ and $Y_k \leftarrow \mathbb{Z}_q^{k \times n}$
- 3 Output $D(A + V_k Y_k + v y^\top, b)$

We use the hybrid argument to prove the lemma. First, define the hybrid distribution $\mathcal{H}_k$ over $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_p^m$ as

$$\mathcal{H}_k = (A + V_k Y_k, \mathcal{R}(Ax))$$

Claim 3.3. *For any $k \in \mathbb{Z}_{\geq 0}$, let $H = (A + V_k Y_k + vy^\top, \mathcal{R}(Ax))$.*

- *If $|\langle y, x \rangle_q| \leq B$, then $TV(H, \mathcal{H}_k) \leq \varepsilon/4t$*
- *If no condition is imposed, then $H \equiv \mathcal{H}_{k+1}$*

Proof. Suppose that $|\langle y, x \rangle_q| \leq B$, rewrite

$$H \equiv (A + V_k Y_k + vy^\top, \mathcal{R}(Ax)) \equiv (A + V_k Y_k, \mathcal{R}(Ax - vy^\top x))$$

The vector $vy^\top x = v\langle y, x \rangle_q$ is uniformly random in $\{0, \langle y, x \rangle_q\}^m$. By the definition of noise tolerance (Definition 2.7), and the fact that $B = \mathcal{B}(\mathcal{R}, \varepsilon/4t)$,

$$TV(H, \mathcal{H}_k) \leq \varepsilon/4t$$

On the other hand, if no condition is imposed on y , it is uniformly random and

$$H \equiv (A + V_k Y_k + vy^\top, b) \equiv (A + V_{k+1} Y_{k+1}, b) \equiv \mathcal{H}_{k+1}. \quad \square$$

It is clear from the definition that $\mathcal{H}_0$ is distributed as a Noisy Linear Equation instance. We claim that once t is large enough, the terminal hybrid $\mathcal{H}_t$ is distributed as a random instance.

Claim 3.4. (Terminal hybrid). *Let $t = \lceil 4 \log(qm/\varepsilon) \rceil$, $A \leftarrow \mathbb{Z}_q^{m \times n}$, $b \leftarrow \mathbb{Z}_q^m$,*

$$TV(\mathcal{H}_t, (A, \mathcal{R}(b))) \leq \varepsilon^2/2qm \ll \varepsilon$$

We defer the proof of the claim to the next section (Section 3.2). With this claim, we can then use hybrid argument to conclude a $\varepsilon/2t$ advantage in distinguishing between $\mathcal{H}_k$ and $\mathcal{H}_{k+1}$ where $k \leftarrow [0, t-1]$.

Proposition 3.5.

$$\left| \Pr_{k \leftarrow [0, t-1]} [D(\mathcal{H}_k) = 1] - \Pr_{k \leftarrow [0, t-1]} [D(\mathcal{H}_{k+1}) = 1] \right| \geq \varepsilon/2t$$

Proof. Standard hybrid argument with Claim 3.4. Details deferred to Section A.1 $\square$

Let H be the distribution sent to the decision algorithm D in line 3 of the algorithm P . Using Claim 3.3 and Proposition 3.5, we obtain a distinguisher that behaves differently under the condition of $|\langle y, x \rangle_q| \leq B$ and no condition. i.e.

$$\left| \Pr_{y \leftarrow \mathbb{Z}_q^n} [D(H) = 1 \mid |\langle y, x \rangle_q| \leq B] - \Pr_{y \leftarrow \mathbb{Z}_q^n} [D(H) = 1] \right| \geq \varepsilon/2t - \varepsilon/4t = \varepsilon/4t$$

Where the loss of $\varepsilon/4t$ is from the total variation distance between H and $\mathcal{H}_k$ when $|\langle y, x \rangle_q| \leq B$.

The condition $|\langle y, x \rangle_q| > B$ can be imposed on the second term at no cost. Let E be the event that $|\langle y, x \rangle_q| \leq B$. By law of total probability,

$$\begin{aligned} & \left| \Pr[D(H) = 1|E] - \Pr[D(H) = 1] \right| \\ &= \Pr[\bar{E}] \cdot \left| \Pr[D(H) = 1|E] - \Pr[D(H) = 1|\bar{E}] \right| \end{aligned}$$

Which implies

$$\left| \Pr[D(H) = 1|E] - \Pr[D(H) = 1|\bar{E}] \right| \geq \frac{\varepsilon/4t}{\Pr[\bar{E}]} \geq \varepsilon/4t. \quad \square$$

3.2 Terminal Hybrid

Here, we show the proof of the claim that the terminal hybrid is close to the uniform distribution.

Claim 3.4. (Terminal hybrid). *Let $t = \lceil 4 \log(qm/\varepsilon) \rceil$, $A \leftarrow \mathbb{Z}_q^{m \times n}$, $b \leftarrow \mathbb{Z}_q^m$,*

$$TV(\mathcal{H}_t, (A, \mathcal{R}(b))) \leq \varepsilon^2/2qm \ll \varepsilon$$

Proof. Start by writing

$$\mathcal{H}_t = (A + VY, \mathcal{R}(Ax)) \equiv (A, \mathcal{R}(Ax - VYx))$$

Where $A \leftarrow \mathbb{Z}_q^{m \times n}$, $V \leftarrow \{0, 1\}^{m \times t}$, $Y \leftarrow \mathbb{Z}_q^{t \times n}$, $x \leftarrow \mathcal{P}(\mathbb{Z}_q)$.

Since Y is a random matrix in $\mathbb{Z}_q^{t \times n}$, and $\gcd(x, q) = 1$, $y = Yx$ is uniformly random in $\mathbb{Z}_q^m$, even given x . As every row v of V is a random $\{0, 1\}$ vector of min-entropy 2^{-t} , and dot product with y is a universal hash, by the leftover hash lemma [ILL89], $(y, \langle v, y \rangle)$ is $\frac{1}{2}\sqrt{q2^{-t}}$ -statistically close to (y, r) for a uniformly random r . By a hybrid argument, the statistical distance between

$$(y, \langle v_1, y \rangle, \dots, \langle v_m, y \rangle) \quad \text{and} \quad (y, r_1, \dots, r_m),$$

where v_i are the rows of V , is at most m times larger. In particular, Vy is $\frac{1}{2}m\sqrt{q2^{-t}}$ -close to uniform. By our choice of parameters, this is at most

$$\frac{1}{2}m\sqrt{q(\varepsilon/qm)^4} \leq \varepsilon^2/2qm \quad \square$$

3.3 Correlation With Threshold Function

Lemma 3.2 essentially shows that the algorithm P is correlated with the indicator for the event $|\langle y, x \rangle_q| \leq B$. For analytic purposes it will be convenient to express this correlation in terms of the covariance of P and a random variable defined by sampling the inputs to the following function at random.

Definition 3.6 (Threshold Function). *Let $n, q, B \in \mathbb{Z}^+$. Define $S : \mathbb{Z}_q^n \times \mathbb{Z}_q^n \rightarrow \{-1, 1\}$ as*

$$S(y, x) = \begin{cases} 1 & \text{if } |\langle y, x \rangle_q| \leq B \\ -1 & \text{otherwise} \end{cases}$$

The conversion from difference in conditional expectations to the covariance of ± 1 -valued random variables is given by the following formula.

Lemma 3.7. *For any $\{-1, 1\}$ -valued random variables X and Y ,*

$$\mathbf{E}[(X - \mu_X)(Y - \mu_Y)] = \frac{\mathbf{Var}(Y)}{2} \cdot (\mathbf{E}[X|Y = 1] - \mathbf{E}[X|Y = -1]),$$

where $\mu_X = \mathbf{E}[X]$, $\mu_Y = \mathbf{E}[Y]$.

Proof. We can rewrite the variance of Y in terms of probabilities

$$\begin{aligned} \mathbf{Var}(Y) &= \mathbf{E}[Y^2] - \mathbf{E}[Y]^2 \\ &= 1 - ((1 - \Pr[Y = -1]) - \Pr[Y = -1])^2 \\ &= -4\Pr[Y = -1]^2 + 4\Pr[Y = -1] \\ &= 4 \cdot \Pr[Y = -1](1 - \Pr[Y = -1]) \\ &= 4\Pr[Y = -1]\Pr[Y = 1] \end{aligned}$$

Expand $\mathbf{E}[(X - \mu_X)(Y - \mu_Y)]$ using law of total probability,

$$\begin{aligned} &\mathbf{E}[(X - \mu_X)(Y - \mu_Y)] \\ &= \mathbf{E}[XY] - \mu_X\mu_Y \\ &= \mathbf{E}[X|Y = 1]\Pr[Y = 1] - \mathbf{E}[X|Y = -1]\Pr[Y = -1] \\ &\quad - (\mathbf{E}[X|Y = 1]\Pr[Y = 1] + \mathbf{E}[X|Y = -1]\Pr[Y = -1])\mu_Y \\ &= \mathbf{E}[X|Y = 1]\Pr[Y = 1](1 - \mu_Y) + \mathbf{E}[X|Y = -1]\Pr[Y = -1](-1 - \mu_Y) \end{aligned}$$

Substitute $\mu_Y = \Pr[Y = 1] - \Pr[Y = -1]$, then

$$1 - \mu_Y = 2\Pr[Y = -1] \text{ and } (-1 - \mu_Y) = -2\Pr[Y = 1]$$

Finally with $\mathbf{Var}(Y) = 4\Pr[Y = -1]\Pr[Y = 1]$,

$$\begin{aligned} &\mathbf{E}[XY] - \mu_X\mu_Y \\ &= 2\mathbf{E}[X|Y = 1]\Pr[Y = 1]\Pr[Y = -1] - 2\mathbf{E}[X|Y = -1]\Pr[Y = -1]\Pr[Y = 1] \\ &= \frac{\mathbf{Var}(Y)}{2}(\mathbf{E}[X|Y = 1] - \mathbf{E}[X|Y = -1]). \end{aligned} \quad \square$$

Applying Lemma 3.7 to P and S gives the following corollary.

Corollary 3.8. *Using the same parameter definitions as in Lemma 3.2,*

$$\mathbf{E}[(P(y, A, \mathcal{R}(Ax)) - \mu_P)(S(y, x) - \mu_S)] \geq \left(\frac{\mathbf{Var}(S)}{2}\right)(\varepsilon/4t) \geq \frac{B\varepsilon}{2qt}$$

Where $A \leftarrow \mathbb{Z}_q^{m \times n}$, $x \leftarrow \mathcal{P}(\mathbb{Z}_q^n)$, $y \leftarrow \mathbb{Z}_q^n$.

Proof. From Lemma 3.2, without loss of generality,

$$\Pr[P(y, A, \mathcal{R}(Ax)) = 1 | S(y, x) = 1] - \Pr[P(y, A, \mathcal{R}(Ax)) = 1 | S(y, x) = -1] \geq \varepsilon/4t$$

Then the statement follows from Lemma 3.7. The calculation of the variance of S is shown as follows:

$$\mathbf{Var}(S) = 4\Pr[S = 1]\Pr[S = -1] = 4\left(\frac{2B+1}{q}\right)\left(1 - \frac{2B+1}{q}\right) \geq \frac{4B}{q} \quad \square$$

Where the last inequality is from $B \leq \lfloor q/4 - 1 \rfloor$

3.4 Heavy Fourier Coefficients

To analyse the Fourier coefficients of P , it is more convenient to fix the input problem instance $(A, x, \mathcal{R}(Ax))$ to be any good instance. We say an instance $(A, x, \mathcal{R}(Ax))$ is ε -good if

$$\mathbf{E}_{y \leftarrow \mathbb{Z}_q^n} [(P(y, A, \mathcal{R}(Ax)) - \mu_P)(S(y, x) - \mu_S)] \geq \varepsilon$$

By definition, $(P - \mu_P)(S - \mu_S)$ takes value in $[-4, 4]$. By Markov's inequality (Proposition 3.9),

$$\mathbf{E}_{y \leftarrow \mathbb{Z}_q^n} [(P(y, A, \mathcal{R}(Ax)) - \mu_P)(S(y, x) - \mu_S)] \geq \frac{\varepsilon}{2} \quad (1)$$

for a $\varepsilon/8$ fraction of A, x , and $\mathcal{R}$ (if randomized). We fix these choices from here on and view P and S as functions of y only.

$$P(y) = P(y, A, \mathcal{R}(Ax)) \text{ and } S(y) = S(y, x)$$

Proposition 3.9. *If $|X| \leq 4$ and $\mathbf{E}[X] \geq \varepsilon$, then for any random variable Y , we have $\mathbf{E}[X|Y] \geq \varepsilon/2$ with probability at least $\varepsilon/8$ over Y .*

Proof. Let E be the set of Y for which $\mathbf{E}[X|Y] \leq \varepsilon/2$. By the law of total expectation,

$$\varepsilon \leq \mathbf{E}[X] \leq \mathbf{E}[X|Y \notin E] \Pr(Y \notin E) + \mathbf{E}[X|Y \in E] \leq 4 \Pr(Y \notin E) + \varepsilon/2$$

so that $\Pr(Y \notin E) \geq \varepsilon/8$. □

By Plancherel's formula (see Proposition 2.12), (1) has an equivalent formulation in the Fourier domain:

$$\sum_{\chi \neq 0^m} \hat{P}(\chi) \cdot \hat{S}(\chi) \geq \frac{\varepsilon}{2}. \quad (2)$$

($\hat{S}$ will turn out to be real so we omit complex conjugation.)

The function $S(y)$ can be expressed as $2f(\langle y, x \rangle) - 1$, where $f: \mathbb{Z}_q \rightarrow \mathbb{C}$ is the univariate function $1(|t| \leq B)$. Making the substitution $t = \langle y, x \rangle$ in the Fourier transform of f

$$f(t) = \sum_{c \in \mathbb{Z}_q} \hat{f}(c) \omega^{-ct}$$

gives the Fourier expansion

$$S(y) = \sum_{c \in \mathbb{Z}_q} 2\hat{f}(c) \cdot \omega^{-c\langle y, x \rangle} - 1 = \sum_{\chi \text{ s.t. } \chi = c \cdot x} 2\hat{f}(c) \cdot \omega^{-\langle y, \chi \rangle} - 1$$

Comparing this to the inverse Fourier Transform (Proposition 2.11) of S , we have the following observations:

- For any χ that is not of the form $c \cdot x$ for some $c \in \mathbb{F}_q$, the Fourier coefficients $S(\chi)$ is 0.
- When $c \neq 0$, $\hat{S}(c \cdot x) = 2\hat{f}(c)$.
- $\hat{S}(0^n) = 2\hat{f}(0^n) - 1$

Inequality (2) therefore simplifies to

$$\sum_{c \neq 0} \hat{P}(c \cdot x) \cdot \hat{f}(c) \geq \frac{\varepsilon}{4}. \quad (3)$$

As f is a threshold function, most of its Fourier mass is concentrated on low frequencies (see Figure 1).

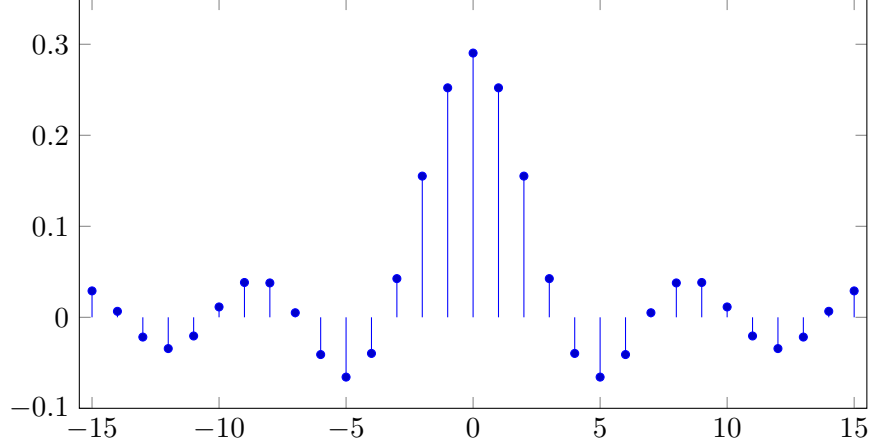


Figure 1: Fourier spectrum of f with $q = 31$ and $B = 4$.

Claim 3.10. Let $\tau \in \mathbb{Z}^+$ and define the light weight set $\mathcal{W}_l = \{c \in \mathbb{Z}_q \mid |c| > \tau\}$, then

$$\sum_{c \in \mathcal{W}_l} |\hat{f}(c)|^2 \leq 1/2\tau$$

Claim 3.11. Let $\tau \in \mathbb{Z}^+$ and define the heavy weight set $\mathcal{W}_h = \{c \in \mathbb{Z}_q \mid 0 < |c| \leq \tau\}$, then

$$\sum_{c \in \mathcal{W}_h} |\hat{f}(c)| \leq 1 + \ln \tau$$

Proofs of the two claims are deferred to the next section (Section 3.5). We now use these to show that (3) forces a heavy Fourier coefficient of the form $\hat{P}(c \cdot x)$.

Lemma 3.12. Assuming (1), there exists $c \in \mathbb{Z}_q$ such that $c \neq 0$, $|c| \leq \lceil 32/\varepsilon^2 \rceil$ and

$$|\hat{P}(c \cdot x)| \geq \frac{\varepsilon}{16 \ln(14/\varepsilon)}.$$

Proof. Set τ to $\lceil 32/\varepsilon^2 \rceil$. As explained in the text above, (1) is equivalent to (3). The sum in (3) splits as

$$\sum_{c \in \mathcal{W}_h} \hat{P}(c \cdot x) \hat{f}(c) + \sum_{c \in \mathcal{W}_l} \hat{P}(c \cdot x) \hat{f}(c) \geq \varepsilon/4.$$

By the Cauchy-Schwarz inequality, Parseval's formula, and Claim 3.10,

$$\begin{aligned} \sum_{c \in \mathcal{W}_l} \hat{P}(c \cdot x) \hat{f}(c) &\leq \sqrt{\sum_{c \in \mathcal{W}_l} |\hat{P}(c \cdot x)|^2} \sqrt{\sum_{c \in \mathcal{W}_l} |\hat{f}(c)|^2} \\ &\leq \sqrt{\mathbf{E}_{y \leftarrow \mathbb{Z}_q^n} [P(y)^2]} \sqrt{\sum_{c \in \mathcal{W}_l} |\hat{f}(c)|^2} \leq \sqrt{\varepsilon^2/64} = \varepsilon/8 \end{aligned}$$

Therefore, the sum over the heavy terms is lower bounded by

$$\sum_{c \in \mathcal{W}_h} \hat{P}(c \cdot x) \cdot \hat{f}(c) \geq \varepsilon/8$$

By Claim 3.11, the sum of all $|\hat{f}(c)|$ terms is at most $\ln \tau$. Therefore at least one of the terms $\hat{P}(c \cdot x)$ must have magnitude at least

$$\frac{\varepsilon/8}{1 + \ln \tau} = \frac{\varepsilon/8}{\ln \lceil 64e/\varepsilon^2 \rceil} \geq \frac{\varepsilon}{16 \ln(14/\varepsilon)}$$

by our choice of τ . □

3.5 Fourier analysis of f

Next, we prove Claims 3.10 and 3.11. We will make use of the following proposition.

Proposition 3.13. *For $c \in \mathbb{Z}_q$, $|\hat{f}(c)| \leq \frac{1}{q|\sin(c\pi/q)|}$.*

Proof. The Fourier coefficients of f are

$$\hat{f}(c) = \mathbf{E}_{t \leftarrow \mathbb{Z}_q} [\mathbf{1}(|t| \leq B) \cdot \omega_q^{c \cdot t}] = \frac{1}{q} \sum_{t=-B}^B \omega_q^{c \cdot t}$$

Using the geometric progression formula,

$$\frac{1}{q} \sum_{t=-B}^B \omega_q^{c \cdot t} = \frac{\omega_q^{-cB}(1 - \omega_q^{c(2B+1)})}{q(1 - \omega_q^c)} = \frac{\omega_q^{-cB} \omega_q^{c(2B+1)/2}}{q \omega_q^{c/2}} \cdot \frac{(\omega_q^{-c(2B+1)/2} - \omega_q^{c(2B+1)/2})}{(\omega_q^{-c/2} - \omega_q^{c/2})}$$

Since $\omega_q^{-a} - \omega_q^a = -2i \sin(2\pi a/q)$, we can simplify to

$$\frac{1}{q} \sum_{t=-B}^B \omega_q^{c \cdot t} = \frac{\sin(c(2B+1)\pi/q)}{q \sin(c\pi/q)}$$

Since the denominator is always a positive number when $c \in \mathbb{Z}_q$ and the numerator is within range $[-1, 1]$, $|\hat{f}(c)| \leq 1/q|\sin(c\pi/q)|$. □

Claim 3.10. *Let $\tau \in \mathbb{Z}^+$ and define the light weight set $\mathcal{W}_l = \{c \in \mathbb{Z}_q \mid |c| > \tau\}$, then*

$$\sum_{c \in \mathcal{W}_l} |\hat{f}(c)|^2 \leq 1/2\tau$$

Proof. From Proposition 3.13,

$$\sum_{c \in \mathcal{W}_l} |\hat{f}(c)|^2 \leq \frac{1}{q^2} \cdot \sum_{c \in \mathcal{W}_l} \frac{1}{\sin(c\pi/q)^2}$$

As $\sin^2$ is an even function,

$$\frac{1}{q^2} \sum_{c \in \mathcal{W}_l} \frac{1}{\sin(c\pi/q)^2} = \frac{2}{q^2} \sum_{c=\tau+1}^{\lfloor q/2 \rfloor} \frac{1}{\sin(c\pi/q)^2}$$

As the sin function is concave downward from 0 to $\pi/2$, the function is lower-bounded by the line that connects the origin and the point at $(\pi/2, 1)$, i.e., $\sin t \geq 2t/\pi$.

$$\frac{2}{q^2} \sum_{c=\tau+1}^{\lfloor q/2 \rfloor} \frac{1}{\sin(c\pi/q)^2} \leq \frac{2}{q^2} \sum_{c=\tau+1}^{\lfloor q/2 \rfloor} \frac{q^2}{4c^2} = \frac{1}{2} \sum_{c=\tau+1}^{\lfloor q/2 \rfloor} \frac{1}{c^2}$$

Because $1/t^2$ is a decreasing function for $t > 0$, we can apply the integral bound to obtain

$$\frac{1}{2} \sum_{c=\tau+1}^{\lfloor q/2 \rfloor} \frac{1}{c^2} \leq \frac{1}{2} \int_{\tau}^{q/2} \frac{1}{c^2} dc \leq \frac{1}{2\tau}. \quad \square$$

Claim 3.11. Let $\tau \in \mathbb{Z}^+$ and define the heavy weight set $\mathcal{W}_h = \{c \in \mathbb{Z}_q \mid 0 < |c| \leq \tau\}$, then

$$\sum_{c \in \mathcal{W}_h} |\hat{f}(c)| \leq 1 + \ln \tau$$

Proof. From Proposition 3.13, using $\sin t \geq 2t/\pi$ in the relevant range,

$$\begin{aligned} \sum_{c \in \mathcal{W}_h} |\hat{f}(c)| &\leq \sum_{c \in \mathcal{W}_h} \frac{1}{q |\sin(c\pi/q)|} = 2 \sum_{c=1}^{\tau} \frac{1}{q \sin(c\pi/q)} \\ &\leq \sum_{c=1}^{\tau} \frac{1}{c} \leq 1 + \int_1^{\tau} \frac{1}{c} dc \leq 1 + \ln \tau. \end{aligned} \quad \square$$

3.6 Learning Significant Fourier Coefficients

With Lemma 3.12, we know that there exists a small $|c|$ such that the Fourier coefficient $\hat{P}(c \cdot x)$ is large, say at least α . To find $c \cdot x$, we use the Learning Significant Fourier Coefficients algorithm from Akavia, Goldwasser and Safra [AGS03] to find all Fourier Coefficients with magnitude at least α .

First, we show that the number of such coefficients is at most $1/\alpha^2$.

Proposition 3.14. Let $\alpha \in (0, 1]$, $\mathcal{W} = \{\chi \in \mathbb{Z}_q^n \mid |\hat{P}(\chi)| \geq \alpha\}$, then $|\mathcal{W}| \leq 1/\alpha^2$

Proof. Since $P : \mathbb{Z}_q^n \rightarrow \{-1, 1\}$, $\mathbf{E}_{y \leftarrow \mathbb{Z}_q^n} [P(y)^2] = 1$. So

$$|\mathcal{W}| \alpha^2 = \sum_{\chi \in \mathcal{W}} \alpha^2 \leq \sum_{\chi \in \mathcal{W}} \hat{P}(\chi)^2 \leq \sum_{\chi \in \mathbb{Z}_q^n} \hat{P}(\chi)^2 = \mathbf{E}_{y \leftarrow \mathbb{Z}_q^n} [P(y)^2] \leq 1$$

$\square$

Theorem 3.15 ([AGS03]). *Let $\alpha \in (0, 1]$. There exists an algorithm that, given a function $P : \mathbb{Z}_q^n \rightarrow \{-1, 1\}$ and α , outputs $\mathcal{W} = \{\chi \in \mathbb{Z}_q^n \mid |\hat{P}(\chi)| \geq \alpha\}$, except with probability $o(\alpha)$, and runs in time $\text{poly}(n, \log q, 1/\alpha)$.*

3.7 Proof of Theorem 3.1

From Lemma 3.2 and Corollary 3.8, we get an algorithm P such that

$$\mathbf{E}[(P(y, A, \mathcal{R}(Ax)) - \mu_P)(S(y, x) - \mu_S)] \geq \frac{B\varepsilon}{2qt}$$

Let $\varepsilon' = (B\varepsilon/2qt)$. With probability at least $\varepsilon'/8$ we get a $\varepsilon'/2$ -good instance $(A, x, \mathcal{R}(Ax))$. In a good instance, Lemma 3.12 says that there exists $c \in \mathbb{Z}_q$ such that $c \neq 0$, $|c| \leq \lceil 32/(\varepsilon')^2 \rceil$ and

$$|\hat{P}(c \cdot x)| \geq \frac{\varepsilon'}{16 \ln(14/\varepsilon')} = \alpha.$$

Then, we run the [AGS03] algorithm from Theorem 3.15 on P with α to obtain a list that contains $c \cdot x$. By Proposition 3.14, this list has size at most $\tilde{O}(1/(\varepsilon')^2)$. The final runtime of the algorithm is $\text{poly}(n, m, \log q, 1/\varepsilon', T)$, where T is the running time of the algorithm for the Decision Noisy Linear Equations problem.

The success probability of the algorithm depends on whether we obtain an $\varepsilon'/2$ -good instance and whether the [AGS03] algorithm from Theorem 3.15 returns the list correctly. By the union bound, the algorithm will succeed with probability $\varepsilon'/8 - o(\varepsilon') = \Omega(\varepsilon')$

4 Corollaries

We can derive corollaries from Theorem 3.1 for the specific noise functions corresponding to LWE and LWR. From Proposition 2.6, the total variation distance of $U(\mathcal{P}(\mathbb{Z}_q^n))$ and $U(\mathbb{Z}_q^n)$ is small. Therefore, we write the following with the distribution of the secret being $U(\mathbb{Z}_q^n)$ instead.

Corollary 4.1 (Partial Secret Recovery for LWE/LWR). *Consider any $n, m, q \in \mathbb{Z}^+$, and any $\varepsilon \in (0, 1]$. Consider the LWE/LWR problem over $\mathbb{Z}_q$ with sample count m , secret dimension n .*

- *Under LWE noise with parameter $\sigma \in \mathbb{R}$, $\sigma < q$, set $\varepsilon' = \sigma\varepsilon^2/q\sqrt{m}$, or,*
- *Under LWR noise with rounding modulus $p \in \mathbb{Z}^+$, $p < q$, set $\varepsilon' = \varepsilon^2/pm$*

Suppose there exists an algorithm for the corresponding Decision LWE/LWR that runs in time T and has advantage ε .

Then there exists an algorithm that, given the problem instance, outputs a list of size $\tilde{O}(1/(\varepsilon')^2)$ that contains $c \cdot x$, where x is the solution, for some $c \in \mathbb{Z}_q \setminus \{0\}$ such that $|c| \leq \tilde{O}(1/(\varepsilon')^2)$. This algorithm runs in time $\text{poly}(n, m, \log q, 1/\varepsilon', T)$ and successfully outputs such a list with probability $\tilde{\Omega}(\varepsilon')$.

Proof. For LWE, by Proposition 2.8, set $B = \min(2\varepsilon\sigma/\sqrt{m}, \lfloor q/4 - 1 \rfloor)$. For LWR, by Proposition 2.9, set $B = \min(q\varepsilon/pm, \lfloor q/4 - 1 \rfloor)$. The rest follows from Theorem 3.1. $\square$

We note that, assuming that the search problem and decision problem are defined over the same secret distribution, our reduction also applies to secret distributions other than uniform over $\mathbb{Z}_q^m$, as long as the secret x satisfies

$$\gcd(x_1, \dots, x_n, q) = 1$$

with very high probability.

4.1 Exact Secret Recovery

So far, we have shown that our algorithm can compute a list of vectors that contains some small multiple of the secret. Under certain conditions, the exact secret can be recovered from this list. We describe a few sufficient conditions for the recovery.

The first requirement for recovery is the existence of an efficient verification algorithm V for the Noisy Linear Equations problem.

Definition 4.2 (Verification Algorithm). *A verification algorithm V for a Noisy Linear Equations problem is an algorithm in which the following holds with high probability over A, x , and $\mathcal{R}$:*

$$\forall x' \in \mathbb{Z}_q^n, V(x', A, \mathcal{R}(Ax)) = \begin{cases} 1 & \text{if } x' = x \\ -1 & \text{if } x' \neq x \end{cases}$$

For typical Noisy Linear Equations problems, such a verification algorithm exists when m is large enough, and the error is small enough for unique recovery. In particular, a verification algorithm for LWE exists when $n \log q \ll m \log(q/\sigma)$, while it exists for LWR when $n \log q \ll m \log p$.

The next requirement is to be able to recover $x \in \mathbb{Z}_q^n$ given $c \cdot x$ and c .

Definition 4.3 (Recovery Algorithm). *A recovery algorithm G for a Noisy Linear Equations problem is an algorithm such that, for all x , and $c \in \mathbb{Z}_q \setminus \{0\}$:*

$$G(c \cdot x, c) = x$$

Some settings where such a G exists are when q is a prime, or if we are guaranteed that $c \cdot \max_i x_i \leq q$. For example, when the secret x is uniform over $\{0, 1\}^n$ and $|c| < q$ (Binary secret LWE problem [BLP⁺13]).

Since the algorithm's output list size is $\tilde{O}(1/(\varepsilon')^2)$ and $|c| \leq \tilde{O}(1/(\varepsilon')^2)$, in time $\tilde{O}(1/(\varepsilon')^4)$ we can try all candidates c on every element in the output list. If G returns x on any one of them, the verification algorithm V would tell us that we got the right answer.

Suppose a verification algorithm does not exist, but G exists. We can still perform recovery at the cost of lowering the success probability by randomly choosing an element from the output list and randomly choosing a c . The success probability of the search algorithm would then become $\Omega((\varepsilon')^5)$.

Corollary 4.4 (Search-to-Decision Reduction for LWE/LWR). *Consider any $n, m, q, p \in \mathbb{Z}^+$, a noise function $\mathcal{R} : \mathbb{Z}_q^m \rightarrow \mathbb{Z}_p^m$, and any $\varepsilon \in (0, 1]$. For the LWE/LWR over $\mathbb{Z}_q$ with sample count m , secret dimension n ,*

- *Under LWE noise with parameter $\sigma \in \mathbb{R}$, $\sigma < q$, set $\varepsilon' = \sigma \varepsilon^2 / q \sqrt{m}$*

- Under LWR noise with rounding modulus $p \in \mathbb{Z}^+$, $p < q$, set $\varepsilon' = \varepsilon^2/pm$

Suppose there exists a Decision algorithm that runs in time T_D and has advantage ε , and a Recovery algorithm that runs in time T_G .

Then there exists a Search algorithm that runs in time $\text{poly}(n, m, \log q, 1/\varepsilon', T_D, T_G)$ with success probability at least $\tilde{\Omega}((\varepsilon')^5)$.

Furthermore, if a Verification algorithm that runs in time T_V exists, there exists a Search algorithm that runs in time $\text{poly}(n, m, \log q, 1/\varepsilon', T_D, T_V, T_G)$ with success probability at least $\tilde{\Omega}(\varepsilon')$. Specifically, this applies to LWE when $n \log q \ll m \log(q/\sigma)$, $\sigma \ll o(p)$ and LWR when $n \log q \ll m \log p$, $p \ll q$

5 Acknowledgments

This work was supported by the National Research Foundation, Singapore, under its NRF Fellowship programme, award no. NRF-NRFF14-2022-0010 and by an NSERC Discovery Grant. AI tools were used as typing assistants for grammar and basic editing.

References

- [AA16] Jacob Alperin-Sheriff and Daniel Apon. Dimension-preserving reductions from LWE to LWR. *IACR Cryptol. ePrint Arch.*, 2016:589, 2016.
- [ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009.
- [AGS03] Adi Akavia, Shafi Goldwasser, and Shmuel Safra. Proving hard-core predicates using list decoding. In *44th Symposium on Foundations of Computer Science, FOCS 2003, Cambridge, MA, USA, October 11-14, 2003, Proceedings*, pages 146–157. IEEE Computer Society, 2003.
- [AIK09] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography with constant input locality. *J. Cryptol.*, 22(4):429–469, 2009.
- [AKPW13] Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited - new reduction, properties and applications. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 57–74. Springer, 2013.
- [BFKL93] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 278–291. Springer, 1993.

- [BGM⁺16] Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen. On the hardness of learning with rounding over small modulus. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 209–224. Springer, 2016.
- [BLMR13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic prfs and their applications. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 410–428. Springer, 2013.
- [BLP⁺13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC’13, Palo Alto, CA, USA, June 1-4, 2013*, pages 575–584. ACM, 2013.
- [BP14] Abhishek Banerjee and Chris Peikert. New and improved key-homomorphic pseudorandom functions. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 353–370. Springer, 2014.
- [BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 719–737. Springer, 2012.
- [BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 97–106. IEEE Computer Society, 2011.
- [ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 12–24. ACM, 1989.
- [MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 465–484. Springer, 2011.
- [MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology*

- *EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, 2012.

- [Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 333–342. ACM, 2009.
- [Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.
- [SHTY25] Kyohei Sudo, Keisuke Hara, Masayuki Tezuka, and Yusuke Yoshida. Quantum search-to-decision reduction for the LWE problem. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 108(2):104–116, 2025.

A Supplementary Proofs

A.1 Hybrid Argument

Proposition 3.5.

$$\left| \Pr_{k \leftarrow [0, t-1]} [D(\mathcal{H}_k) = 1] - \Pr_{k \leftarrow [0, t-1]} [D(\mathcal{H}_{k+1}) = 1] \right| \geq \varepsilon/2t$$

Proof of Proposition 3.5. Using $TV(\mathcal{H}_t, (A, \mathcal{R}(b))) \leq \varepsilon^2/2qt$ from Claim 3.4,

$$\begin{aligned} & \left| \Pr_{i \leftarrow [0, t-1]} [D(\mathcal{H}_i) = 1] - \Pr_{i \leftarrow [0, t-1]} [D(\mathcal{H}_{i+1}) = 1] \right| \\ &= \left| \sum_{i=0}^{t-1} \frac{1}{t} \Pr[D(\mathcal{H}_i) = 1] - \frac{1}{t} \Pr[D(\mathcal{H}_{i+1}) = 1] \right| \\ &= \left| \frac{1}{t} (\Pr[D(\mathcal{H}_0) = 1] - \Pr[D(\mathcal{H}_t) = 1]) \right| \\ &\geq \varepsilon/t - \varepsilon^2/2qt \geq \varepsilon/2t \end{aligned}$$

□