Worst-Case Hardness for LPN and Cryptographic Hashing via Code Smoothing

Zvika Brakerski[*] Weizmann

Vadim Lyubashevsky[†] IBM Research – Zurich

Vinod Vaikuntanathan MIT

Daniel Wichs[‡] Northeastern

Abstract

We present a worst case decoding problem whose hardness reduces to that of solving the Learning Parity with Noise (LPN) problem, in some parameter regime. Prior to this work, no worst case hardness result was known for LPN (as opposed to syntactically similar problems such as Learning with Errors). The caveat is that this worst case problem is only mildly hard and in particular admits a quasi-polynomial time algorithm, whereas the LPN variant used in the reduction requires extremely high noise rate of $1/2 - 1/\mathrm{poly}(n)$. Thus we can only show that "very hard" LPN is harder than some "very mildly hard" worst case problem. We note that LPN with noise $1/2 - 1/\mathrm{poly}(n)$ already implies symmetric cryptography.

Specifically, we consider the (n, m, w)-nearest codeword problem ((n, m, w)-NCP) which takes as input a generating matrix for a binary linear code in m dimensions and rank n, and a target vector which is very close to the code (Hamming distance at most w), and asks to find the codeword nearest to the target vector. We show that for balanced (unbiased) codes and for relative error $w/m \approx \log^2 n/n$, (n, m, w)-NCP can be solved given oracle access to an LPN distinguisher with noise ratio $1/2 - 1/\mathrm{poly}(n)$.

Our proof relies on a smoothing lemma for codes which we show to have further implications: We show that (n, m, w)-NCP with the aforementioned parameters lies in the complexity class Search-$\mathcal{BPP}^{\mathcal{SZK}}$ (i.e. reducible to a problem that has a statistical zero knowledge protocol) implying that it is unlikely to be $\mathcal{NP}$-hard. We then show that the hardness of LPN with very low noise rate $\log^2(n)/n$ implies the existence of collision resistant hash functions (our aforementioned result implies that in this parameter regime LPN is also in $\mathcal{BPP}^{\mathcal{SZK}}$).

1 Introduction

The hardness of noisy learning problems such as learning parity with noise (LPN) [BFKL93, BKW03] and learning with errors (LWE) [Reg05] have proved to be a goldmine in modern cryptography. The hardness of LWE has been instrumental in solving long-standing problems such as fully homomorphic encryption [Gen09, BV11]. Both LPN and LWE have given us efficient and plausibly quantum-proof cryptographic constructions [KPC⁺11, BCD⁺16, ADPS16]. However, while we know several structural results about LWE, relatively little is known about the 25-year old LPN problem.

Before we proceed, let us define the LPN and LWE problems. In the (search version of the) LPN problem, the algorithm is given access to an oracle that produces samples $(\mathbf{a}_i, \mathbf{s}^T\mathbf{a}_i + e_i)$

[*]Supported by the Israel Science Foundation (Grant No. 468/14), Binational Science Foundation (Grants No. 2016726, 2014276), and by the European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482) and via Project PROMETHEUS (Grant 780701).

[†]Supported by the SNSF ERC Transfer Grant CRETP2-166734 – FELICITY.

[‡]Research supported by NSF grants CNS1314722, CNS-1413964.

where $\mathbf{s} \in \mathbb{Z}_2^n$ is the "secret" vector, $\mathbf{a}_i \in \mathbb{Z}_2^n$ are uniformly distributed and $e_i \in \mathbb{Z}_2$ come from the Bernoulli distribution (that is, it is 1 with probability ϵ and 0 otherwise). The goal is to recover $\mathbf{s}$. The (search version of the) LWE problem is the same but for two key changes: first, the vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$ are uniformly random with entries from some large enough finite field $\mathbb{Z}_q$ and second, each error term e_i is chosen from the discrete Gaussian distribution over the integers. The exact choice of the error distribution does not matter much: what is important is that in LWE, each sample has an error with bounded absolute value (at least with high probability). These seemingly minor differences seem to matter a great deal: we know worst-case to average-case reductions for LWE [Reg05, Pei09, BLP+13] while no such result is known for LPN;[1] we know that (a decisional version of) LWE is in the complexity class $\mathcal{SZK}$ [MV03] (statistical zero-knowledge) while no such result is known for LPN; and we can build a dizzying array of cryptographic primitives assuming the hardness of LWE (e.g. attribute based encryption and homomorphic encryption to name the more exotic examples) while the repertoire of LPN is essentially limited to one-way functions and public-key encryption (and primitives that can be constructed generically from it). In particular, we do not know how to construct even simple, seemingly "unstructured", primitives such as a collision-resistant hash function from the hardness of LPN, even with extreme parameter choices. Can we bridge this puzzling gap between LWE and LPN?

In a nutshell, the goal of this paper is to solve all three of these problems. Our main tool is a *smoothing lemma for binary linear codes.* We proceed to describe our results and techniques in more detail.

1.1 Overview of Our Results and Techniques

Worst-case to Average-case Reduction. We consider the promise nearest codeword problem (NCP), a worst-case analog of the learning parity with noise problem. Roughly speaking, in the search version of the (n, m, w)-promise nearest codeword problem, one is given the generator matrix $\mathbf{C} \in \mathbb{Z}_2^{n\times m}$ of a linear code, along with a vector $\mathbf{t} \in \mathbb{Z}_2^m$ such that $\mathbf{t} = \mathbf{s}^T\mathbf{C} + \mathbf{x}^T$ for some $\mathbf{s} \in \mathbb{Z}_2^n$ and $\mathbf{x} \in \mathbb{Z}_2^m$ *with the promise* that $\mathsf{wt}(\mathbf{x}) = w$. The problem is to find $\mathbf{s}$. The non-promise version of this problem (which is commonly called the nearest codeword problem) is known to be $\mathcal{NP}$-hard, even to approximately solve [ABSS93] and the promise problem is similarly $\mathcal{NP}$-hard in the large-error regime (that is, when the Hamming weight of $\mathbf{x}$ exceeds $(1/2+\epsilon)d$ where d is the minimum distance of the code and $\epsilon > 0$ is an arbitrarily small constant) [DMS99].

In terms of algorithms, Berman and Karpinski [BK02] show how to find an $O(n/\log n)$-approximate nearest codeword in polynomial time. In particular, this means that if the Hamming weight of $\mathbf{x}$ in the promise version is at most $O(d \cdot \log n/n)$, their algorithm finds the unique nearest codeword's $\mathbf{s}$ efficiently. To the best of our knowledge, this result is the current limit of polynomial-time solvability of the promise nearest codeword problem. Alon, Panigrahy and Yekhanin [APY09] show a deterministic nearly-polynomial time algorithm with the same parameters. In this work, we consider the promise NCP for *balanced codes*, where all nonzero codewords have Hamming weight between $(1/2 - \beta)m$ and $(1/2 + \beta)m$ for some balance parameter $\beta > 0$. We are not aware of improved NCP algorithms that apply to balanced codes.

Our first result (in Section 4) shows a reduction from the *worst-case* promise NCP for balanced codes where $w/m \approx \frac{\log^2 n}{n}$ to the *average-case* hardness of LPN_ϵ^n with very high error-rate $\epsilon =$

[1] Feldman et al. [FGKP09] showed a worst-case to average-case reduction with respect to the noise distribution, but not with respect to the samples themselves.

$1/2 - 1/O(n^4)$. We note that a random linear code is β-balanced with overwhelming probability when $\beta \geq 3\sqrt{n/m}$ so for a sufficiently large m the restriction on β is satisfied by most codes. Thus, qualitatively speaking, our result shows that solving LPN with very high error *on the average* implies solving NCP with very low error *for most codes*. While the parameters we achieve are extreme, we emphasize that no worst-case to average-case reduction for LPN was known prior to our work.

The worst-case to average-case reduction is a simple consequence of a smoothing lemma for codes that we define and prove in Section 3. In a nutshell, our smoothing lemma shows a simple randomized procedure that maps a worst-case linear code $\mathcal{C}$ and a vector $\mathbf{t}$ to a random linear code $\mathcal{C}'$ and a vector $\mathbf{t}'$ such that if $\mathbf{t}$ is super-close to $\mathcal{C}$, then $\mathbf{t}'$ is somewhat close to $\mathcal{C}'$. Our worst-case to average-case reduction then follows simply by applying the smoothing lemma to the worst-case code and vector. We show a simple Fourier-analytic proof of the smoothing lemma, in a way that is conceptually similar to analogous statements in the context of lattices [MR04] (see more details in the end of Section 3). Similar statements have been shown before in the list-decoding high-error regime [KS10], whereas our setting for NCP is in the unique decoding (low error) regime.

Statistical Zero-Knowledge. Another consequence of our smoothing lemma is a statistical zero-knowlege proof for the NCP problem for balanced codes with low noise, namely where $w/m \approx \frac{\log^2 n}{n}$. In particular, we show that the search problem is in $\mathcal{BPP}^{\mathcal{SZK}}$. Membership in $\mathcal{BPP}^{\mathcal{SZK}}$ should be viewed as an easiness result: a consequence of this result and a theorem of Mahmoody and Xiao [MX10] is that NCP with low noise cannot be $\mathcal{NP}$-hard unless the polynomial hierarchy collapses. Our result is the first non-$\mathcal{NP}$-hardness result we know for NCP, complementing the $\mathcal{NP}$-hardness result of Dumer, Micciancio and Sudan [DMS99] for noise slightly larger than half the minimum distance, namely where $w/m \approx 1/2$ (but leaves a large gap in between). This is the LPN/codes analog of a result for LWE/lattices that we have known for over a decade [MV03]. We refer the reader to Section 5 for this result.

Collision-Resistant Hashing. Finally, we show a new cryptographic consequence of the hardness of LPN with low noise, namely a construction of a collision-resistant hash (CRH) function. Again, collion-resistant hashing from LWE/lattices has been known for over two decades [Ajt96, GGH96] and we view this result as an LPN/codes analog. The construction is extremely simple: the family of hash functions is parameterized by a matrix $\mathbf{A} \in \mathbb{Z}_2^{n \times n^{1+c}}$ for some $c > 0$, its domain is the set of vectors $\mathbf{x} \in \mathbb{Z}_2^{n^{1+c}}$ with Hamming weight $2n/(c \log n)$ and the output is simply $\mathbf{Ax}$ (mod 2). This is similar to a CRH construction from the recent work of Applebaum et al. [AHI+17] modulo the setting of parameters; what is new in our work is a reduction from the LPN problem with error rate $O(\log^2 n/n)$ to breaking this CRH function.

Related Work. Our LPN-based collision-resistant hash function was used in [BLSV17] as a basis for constructing an identity based encryption scheme based on LPN with very low noise. Concurrently with, and independently from, our work, Yu et al. [YZW+17] constructed a family of collision-resistant hash functions based on the hardness of LPN using the same main idea as in Section 6 of the present work. While the core ideas of the construction in the two works is identical, [YZW+17] further discusses different parameter settings and some heuristics upon whose reliance one can obtain a tighter connection between the hardness of the CRH and the LPN problem.

2 Preliminaries

2.1 Notation

Throughout the paper, we will be working with elements in the additive group $\mathbb{Z}_2$ with the usual addition operation. We will denote by bold lower-case letters vectors over $\mathbb{Z}_2^n$ for $n > 1$, and by bold upper-case letters matrices over $\mathbb{Z}_2^{m \times n}$ for $m, n > 1$. We will make the assumption that all vectors are column vectors and write $\mathbf{a}^T$ to denote the row vector which is the transpose of $\mathbf{a}$. The Hamming weight of $\mathbf{a} \in \mathbb{Z}_2^n$, written as $\mathsf{wt}(\mathbf{a})$, denotes the number of 1's in $\mathbf{a}$. For a set S, we write $s \leftarrow S$ to denote that s is chosen uniformly at random from S. When D is some probability distribution, then $s \leftarrow D$ means that s is chosen according to D.

The Ber_ϵ distribution over $\mathbb{Z}_2$ is the Bernoulli distribution that outputs 1 with probability ϵ and 0 with probability $1 - \epsilon$. Let $\mathcal{S}_k^m$ be the set of all the elements $\mathbf{s} \in \mathbb{Z}_2^m$ such that $\mathsf{wt}(\mathbf{s}) = k$.

A negligible function $\text{negl}(n)$ is any function that grows slower than inverse polynomial in n. In particular, for every polynomial p there is an $n_0 \in \mathbb{N}$ such that for every $n > n_0$, $\text{negl}(n) < 1/p(n)$.

2.2 The Learning Parity with Noise (LPN) Problem

For an $\mathbf{s} \in \mathbb{Z}_2^n$, and an $\epsilon \in [0, .5]$ let $\mathcal{O}_{\mathbf{s},\epsilon}^n$ be an algorithm that, when invoked, chooses a random $\mathbf{a} \leftarrow \mathbb{Z}_2^n$ and $e \leftarrow \mathsf{Ber}_\epsilon$ and outputs $(\mathbf{a}, \mathbf{s}^T\mathbf{a} + e)$. An algorithm A is said to solve the search LPN_ϵ^n problem with probability δ if

$$\Pr[\mathsf{A}^{\mathcal{O}_{\mathbf{s},\epsilon}^n} \Rightarrow \mathbf{s}\ ;\ \mathbf{s} \leftarrow \mathbb{Z}_2^n] \geq \delta.$$

Let $\mathcal{U}^n$ be an algorithm that, when invoked, chooses random $\mathbf{a} \leftarrow \mathbb{Z}_2^n$ and $b \leftarrow \mathbb{Z}_2$ and outputs $(\mathbf{a}, b)$. We say that an algorithm A has advantage δ in solving the decisional LPN_ϵ^n problem if

$$\left|\Pr[\mathsf{A}^{\mathcal{O}_{\mathbf{s},\epsilon}^n} \Rightarrow 0;\ \mathbf{s} \leftarrow \mathbb{Z}_2^n] - \Pr[\mathsf{A}^{\mathcal{U}^n} \Rightarrow 0]\right| \geq \delta.$$

The LPN problem has a search to decision reduction (c.f. [KS06]). Namely, if there is an algorithm that runs in time t and has advantage δ in solving the decisional LPN_ϵ^n problem, then there is an algorithm that runs in time $O(nt/\delta)$ that solves the search LPN_ϵ^n problem with probability ≈ 1.

The following fact is known in some contexts as The Piling-Up Lemma [Mat93].

Lemma 2.1. *For all $\epsilon \in [0, \frac{1}{2}]$ it holds that* $\Pr[e_1 + \ldots + e_k = 0;\ e_i \leftarrow \mathsf{Ber}_\epsilon] = \frac{1}{2} + \frac{1}{2} \cdot (1 - 2\epsilon)^k$.

2.3 The Nearest Codeword Problem

An (binary) (n, m, d)-code $\mathcal{C}$ is a subset of $\{0,1\}^m$ such that $|\mathcal{C}| = 2^n$ and for any two codewords $\mathbf{x}, \mathbf{y} \in \mathcal{C}$, $\mathsf{wt}(\mathbf{x} \oplus \mathbf{y}) \leq d$. The code is linear (denoted $[n, m, d]$-code) if $\mathcal{C}$ is the row span of some matrix $\mathbf{C} \in \{0,1\}^{n \times m}$.

Definition 2.1 (Nearest Codeword Problem (NCP))**.** *The nearest codeword problem $\mathsf{NCP}_{n,m,w}$ is characterized by $n, m, w \in \mathbb{Z}$ and is defined as follows. The input consists of a matrix $\mathbf{C} \in \mathbb{Z}^{n \times m}$ which is the generator of a code, along with a vector $\mathbf{t} \in \mathbb{Z}^m$ such that $\mathbf{t} = \mathbf{s}^T\mathbf{C} + \mathbf{x}^T$ for some $\mathbf{s} \in \mathbb{Z}_2^n, \mathbf{x} \in \mathbb{Z}_2^m$ with $\mathsf{wt}(\mathbf{x}) = w$. The problem is to find $\mathbf{s}$.*

Note that our definition requires $\mathsf{wt}(\mathbf{x}) = w$, as opposed to the more relaxed requirement $\mathsf{wt}(\mathbf{x}) \leq w$. However since w comes from a polynomial domain $\{0, \ldots, m\}$ the difference is not very substantial (in particular, to solve the relaxed version one can go over all polynomially-many relevant values of w and try solving the exact version).

In this work, we consider a variant of the problem which is restricted to *balanced* codes, which are codes where all non-zero codewords have hamming weight close to 1/2. We start by defining balanced codes and then present balanced NCP.

Definition 2.2. *A code $\mathcal{C} \subseteq \{0,1\}^m$ is β-balanced if its minimum distance is at least $\frac{1}{2}(1-\beta)m$ and maximum distance is at most $\frac{1}{2}(1+\beta)m$.*

Definition 2.3 (balanced NCP (balNCP))**.** *The balanced nearest codeword problem $\mathsf{balNCP}_{n,m,w,\beta}$ is characterized by $n, m, w \in \mathbb{Z}$ and $\beta \in (0,1)$, and is defined as follows. The input consists of a matrix $\mathbf{C} \in \mathbb{Z}^{n \times m}$ which is the generator of a β-balanced code, along with a vector $\mathbf{t} \in \mathbb{Z}^m$ such that $\mathbf{t}^T = \mathbf{s}^T\mathbf{C} + \mathbf{x}^T$ for some $\mathbf{s} \in \mathbb{Z}_2^n, \mathbf{x} \in \mathbb{Z}_2^m$ with $\mathsf{wt}(\mathbf{x}) = w$. The problem is to find $\mathbf{s}$.*

The $\mathsf{balNCP}_{n,m,w,\beta}$ problem has a unique solution when $w \leq \frac{1}{4}(1-\beta)m$.

Standard decoding algorithms allow to solve NCP in polynomial time with success probability $(1-\frac{w}{m})^n$ [BK02] or even deterministically in time $(1-\frac{w}{m})^{-n} \cdot \mathrm{poly}(n,m)$ [APY09]. We are not aware of improved methods that apply to balanced codes.

To conclude this section we show via a straightforward probabilistic argument that most sparse linear codes are indeed balanced (this is essentially the Gilbert-Varshamov Bound). This is to serve as sanity check that the definition is not vacuous and will also be useful when we apply our $\mathcal{SZK}$ results to the LPN_ϵ^n problem which naturally induces random codes.

Lemma 2.2. *A random linear code $\mathcal{C} \subseteq \mathbb{Z}_2^m$ of dimension n is β-balanced with probability at least $1-2^{n-\beta^2 m/4+1}$. In particular, when $\beta \geq 3\sqrt{n/m}$ a random linear code is β-balanced with probability $1-\mathrm{negl}(n)$.*

Proof. Let $\mathbf{C} \leftarrow \mathbb{Z}_2^{n \times m}$ be a randomly chosen generator matrix. Then the associated code $\mathcal{C}$ fails to be β-balanced if and only if there exists some $\mathbf{s} \neq \mathbf{0} \in \mathbb{Z}_2^n$ such that $|\mathsf{wt}(\mathbf{s}^T\mathbf{C}) - \frac{m}{2}| > \frac{\beta}{2}m$. For any fixed $\mathbf{s} \neq \mathbf{0}$ the vector $\mathbf{s}^T\mathbf{C}$ is uniformly random in $\mathbb{Z}_2^m$ and therefore by the Chernoff bound:

$$\Pr\left[\left| \mathsf{wt}(\mathbf{s}^T\mathbf{C}) - \frac{m}{2}\right| > \tfrac{\beta m}{2}\right] \leq 2\exp\left(-\frac{\beta^2 m}{4}\right)$$

By the union bound, the probability that the code is not β-balanced is at most

$$2^{n+1}\exp\left(-\frac{\beta^2 m}{4}\right) \leq 2^{n-\frac{\beta^2 m}{4}+1}.$$

This is negligible in n when $\beta \geq 3\sqrt{n/m}$. □

2.4 Statistical Zero Knowledge

Statistical zero-knowledge ($\mathcal{SZK}$) is the class of all problems that admit a zero-knowledge proof [GMR89] with a statistically sound simulation. Sahai and Vadhan [SV03] showed that the following problem is complete for $\mathcal{SZK}$.

Definition 2.4. *The promise problem Statistical Distance (SD) is defined by the following YES and NO instances. For a circuit $C : \{0,1\}^n \to \{0,1\}^m$, we let $C(U_n)$ denote the probability distribution* *on m-bit strings obtained by running C on a uniformly random input. Let $\mathsf{SD}(D_0, D_1)$ denote the statistical (variation) distance between the distributions D_0 and D_1.*

$$\Pi_{YES} := \{(C_0, C_1) : \ C_0, C_1 : \{0,1\}^n \to \{0,1\}^m \ \textit{and} \ \mathsf{SD}(C_0(U_n), C_1(U_n)) \geq 2/3\}$$
$$\Pi_{NO} := \{(C_0, C_1) : \ C_0, C_1 : \{0,1\}^n \to \{0,1\}^m \ \textit{and} \ \mathsf{SD}(C_0(U_n), C_1(U_n)) \leq 1/3\}$$

By $\mathcal{BPP}^{\mathcal{SZK}}$, we mean decision problems that can be reduced to the statistical distance problem using randomized reductions. While in general such reduction could query the SD oracle on inputs that violate the promise (namely, a pair of circuits/distributions whose statistical distance lies strictly between 1/3 and 2/3), the reductions we present in this paper will respect the SD promise. Search-$\mathcal{BPP}^{\mathcal{SZK}}$ is defined analogously.

3 A Smoothing Lemma for Noisy Codewords

Let $\mathcal{C} \subseteq \mathbb{Z}_2^m$ be a binary linear code with generating matrix $\mathbf{C} \in \mathbb{Z}_2^{n \times m}$. We say that a distribution $\mathcal{R}$ over $\mathbb{Z}_2^m$ smooths $\mathcal{C}$ if the random variable $\mathbf{Cr}$ for $\mathbf{r} \leftarrow \mathcal{R}$ is statistically close to uniform over $\mathbb{Z}_2^n$. We say that $\mathcal{R}$ also smooths noisy codewords if for every vector $\mathbf{x}$ of sufficiently low Hamming weight, it holds that $(\mathbf{Cr}, \mathbf{x}^T\mathbf{r})$ is statistically close to the distribution $\mathcal{U}_{\mathbb{Z}_2^n} \times \mathsf{Ber}_\epsilon$ for some ϵ.

The notion of smoothing will play an important role in our reductions in this work. In particular, we would like to characterize families of codes that are smoothed by distributions supported over low Hamming weight vectors. To this end, we show that for balanced codes, there exist such smoothing distributions. (Similar statements have been shown before in the high-error regime, e.g., by Kopparty and Saraf [KS10].)

We note that while our proof uses harmonic analysis, it is also possible to prove it using the Vazirani XOR Lemma [Vaz86,Gol95]. However, we find that our method of using harmonic analysis demonstrates more straightforwardly the analogy of our lemma to smoothing in the lattice world (which is most often proved using harmonic analysis), see comparison in the end of this section. Furthermore, this suggests an approach if one wants to analyze the non-binary setting.

We start by defining our family of smoothing distributions $\mathcal{R}_{d,m}$.

Definition 3.1. *Let $d, m \in \mathbb{N}$. The distribution $\mathcal{R}_{d,m}$ over $\mathbb{Z}_2^m$ is defined as follows. Sample (with replacement) d elements $t_1, \ldots, t_d$ uniformly and independently from $[m]$. Output $\mathbf{x} = \oplus_{i=1}^d \mathbf{u}_{t_i}$, where $\mathbf{u}_j$ is the j-th standard basis vector. One can easily verify that $\mathcal{R}_{d,m}$ is supported only over vectors of Hamming weight at most d.*

We can now state and prove our smoothing lemma for noisy codewords.

Lemma 3.1. *Let $\beta \in (0,1)$ and let $\mathbf{C} \in \mathbb{Z}_2^{n \times m}$ be a generating matrix for a β-balanced binary linear code $\mathcal{C} \subseteq \mathbb{Z}_2^m$. Let $\mathbf{c} \in \mathbb{Z}_2^m$ be a word of distance w from $\mathcal{C}$. Let $\mathbf{s}, \mathbf{x}$ be s.t. $\mathbf{c}^T = \mathbf{s}^T\mathbf{C} + \mathbf{x}^T$ and* $\mathsf{wt}(\mathbf{x}) = w$.

Consider the distribution $(\mathbf{a}, b)$ generated as follows. Sample $\mathbf{r} \leftarrow \mathcal{R}_{d,m}$ and set $\mathbf{a} = \mathbf{Cr}$, $b = \mathbf{c}^T\mathbf{r}$. Then it holds that the joint distribution of $(\mathbf{a}, b - \mathbf{s}^T\mathbf{a})$ is within statistical distance δ from the product distribution $\mathcal{U}_{\mathbb{Z}_2^n} \times \mathsf{Ber}_\epsilon$, where

$$\begin{aligned} \delta &\leq 2^{(n+1)/2} \cdot (\beta + \tfrac{2w}{m})^d \ \textit{and} \\ \epsilon &= \tfrac{1}{2} - \tfrac{1}{2}(1 - \tfrac{2w}{m})^d. \end{aligned}$$

Proof. Let e denote the value $b - \mathbf{s}^T\mathbf{a}$. We bound the distance of $\left[\begin{smallmatrix}\mathbf{a}\\ e\end{smallmatrix}\right] = \left[\begin{smallmatrix}\mathbf{C}\\ \mathbf{x}^T\end{smallmatrix}\right]\mathbf{r}$ from $\mathcal{U}_{\{0,1\}^n} \times \mathsf{Ber}_\epsilon$ using simple harmonic analysis. Let f be the probability density function of $\left[\begin{smallmatrix}\mathbf{a}\\ e\end{smallmatrix}\right]$, and consider its (binary) Fourier Transform:

$$\hat{f}(\mathbf{y}, z) = \mathop{\mathbb{E}}_{\mathbf{a},e}[(-1)^{\mathbf{y}^T\mathbf{a}+ze}] = \mathop{\mathbb{E}}_{\mathbf{r}}[(-1)^{(\mathbf{y}^T\mathbf{C}+z\mathbf{x}^T)\mathbf{r}}] \;, \tag{1}$$

It immediately follows that $\hat{f}(\mathbf{0}, 0) = 1$. Moreover

$$\hat{f}(\mathbf{0}, 1) = \mathop{\mathbb{E}}_{\mathbf{r}}[(-1)^{\mathbf{x}^T\mathbf{r}}] \;. \tag{2}$$

Recalling that $\mathbf{r} = \oplus_{i=1}^d \mathbf{u}_{t_i}$ we have

$$\mathop{\mathbb{E}}_{\mathbf{r}}[(-1)^{\mathbf{x}^T\mathbf{r}}] = \prod_{i=1}^d \mathop{\mathbb{E}}_{t_i}[(-1)^{\mathbf{x}^T\mathbf{u}_{t_i}}] = (1 - \tfrac{2w}{m})^d \;,$$

since each t_i is sampled uniformly and independently in $[m]$ and thus has a $\frac{w}{m}$ probability to hit a coordinate where $\mathbf{x}$ is one. Recalling the definition of ϵ, we have $\hat{f}(\mathbf{0}, 1) = 1 - 2\epsilon$.

Now let us consider the setting where $\mathbf{y} \neq \mathbf{0}$. In that case, let us denote $\mathbf{v} = \mathbf{y}^T\mathbf{C}$, a nonzero codeword in $\mathcal{C}$. Since $\mathcal{C}$ is balanced it follows that $\mathsf{wt}(\mathbf{v}) \in [\frac{1}{2}(1-\beta)m, \frac{1}{2}(1+\beta)m]$. Let us further denote $(\mathbf{v}')^T = \mathbf{y}^T\mathbf{C} + z\mathbf{x}^T$, since $\mathsf{wt}(\mathbf{x}) \leq w$ it follows that $\mathsf{wt}(\mathbf{v}') \in \frac{1}{2}(1 \pm \beta')m$ for $\beta' = \beta + \frac{2w}{m}$. For $\mathbf{y} \neq \mathbf{0}$ we thus get

$$\hat{f}(\mathbf{y}, z) = \mathop{\mathbb{E}}_{\mathbf{r}}[(-1)^{(\mathbf{v}')^T\mathbf{r}}] = \prod_{i=1}^d \mathop{\mathbb{E}}_{t_i}[(-1)^{(\mathbf{v}')^T\mathbf{u}_{t_i}}] \;. \tag{3}$$

Since each t_i is sampled uniformly from $[m]$, it follows that $\mathbf{v}'\mathbf{u}_{t_i} \pmod 2 = 0$ with probability $\epsilon_i \in \frac{1}{2}(1 \pm \beta')$. Therefore for all $i \in [d]$ it holds that

$$\left|\mathop{\mathbb{E}}_{t_i}[(-1)^{\mathbf{v}'\mathbf{u}_{t_i}}]\right| = |1 - 2\epsilon_i| \leq \beta' \;. \tag{4}$$

We conclude that

$$\left|\hat{f}(\mathbf{y}, z)\right| \leq (\beta')^d \;. \tag{5}$$

Now we are ready to compare with $\mathcal{U}_{\mathbb{Z}_2^n} \times \mathsf{Ber}_\epsilon$. Let g be the probability density function of $\mathcal{U}_{\mathbb{Z}_2^n} \times \mathsf{Ber}_\epsilon$, and let $\hat{g}$ be its Fourier Transform. Then $\hat{g}(\mathbf{0}, 0) = 1$, $\hat{g}(\mathbf{0}, 1) = 1 - 2\epsilon$ and $\hat{g}(\mathbf{y}, z) = 0$ for all $\mathbf{y} \neq 0$. Therefore

$$\left\|\hat{f} - \hat{g}\right\|_2^2 = \sum_{\mathbf{y},z} \left|\hat{f}(\mathbf{y}) - \hat{g}(\mathbf{y})\right|^2 \leq \sum_{\substack{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{0\} \\ z\in\mathbb{Z}_2}} (\beta')^{2d} \leq 2^{n+1}(\beta')^{2d} \;. \tag{6}$$

By Parseval's theorem, we have that

$$\|f - g\|_2^2 = \frac{1}{2^{n+1}} \left\|\hat{f} - \hat{g}\right\|_2^2 \leq (\beta')^{2d} \;, \tag{7}$$

and going to ℓ_1 norm we have

$$\|f - g\|_1 \leq 2^{(n+1)/2} \cdot \left\|\hat{f} - \hat{g}\right\|_2 \leq 2^{(n+1)/2} \cdot (\beta')^d \;, \tag{8}$$

which completes the proof. $\square$

Relation to Lattice Smoothing. To conclude this section, let us briefly explain the analogy to smoothing lemmas for lattices [MR04]. Our explanation is intended mostly for readers who are familiar with lattice smoothing and the notion of q-ary lattices, and wish to better understand the connection to our notion of smoothing. Other readers may safely skip this paragraph. We recall that the goal of smoothing in the lattice world is to find a distribution $\mathcal{D}$ (a Gaussian in the lattice case), such that reducing it modulo a lattice $\mathcal{L}$ results in an almost uniform distribution over the cosets of the lattice. Let us restrict our attention to integer lattices, integer distributions and integer cosets. Formally, $\mathcal{D} \pmod{\mathcal{L}}$ is uniform over $\mathbb{Z}/\mathcal{L}$. Now let $\mathbf{C}$ be a generating matrix for a binary code, and consider the so called "perp lattice" $\mathcal{L} = \Lambda_2^{\perp}(\mathbf{C}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{C}\mathbf{x} = 0 \pmod 2\} = \mathcal{C}^{\perp} + 2\mathbb{Z}^m$. That is, the lattice corresponding to the dual code of $\mathcal{C}$ plus all even vectors. Each integer cosets of the lattice $\mathcal{L}$ corresponds to a vector $\mathbf{y}$ where the respective coset is $\mathcal{L}_{\mathbf{y}} = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{C}\mathbf{x} = \mathbf{y} \pmod 2\}$. Thus a smoothing distribution $\mathcal{D}$ is one where drawing $\mathbf{r}$ from $\mathcal{D}$ and computing $\mathbf{C}\mathbf{r} \pmod 2$ is close to uniform. Therefore, our smoothing lemma above shows that for 2-ary lattices one can devise non-trivial (and useful) smoothing distributions, and these distributions are not discrete Gaussians as usually considered in the lattice literature. Finally, the fact that we can smooth the code together with a noisy codeword is somewhat analogous to Gaussian leftover hash lemmas in the context of lattices.

4 A Worst Case Balanced NCP to Average Case LPN Reduction

Theorem 4.1. *Assume there is an algorithm that solves the search* $\mathsf{LPN}_{\epsilon}^{n}$ *problem with success probability* α *in the average case by running in time* T *and making* q *queries. Then, for every* $d \leq m \in \mathbb{Z}$ *there is an algorithm that solves search* $\mathsf{balNCP}_{n,m,w,\beta}$ *in the worst case in time* $T \cdot \mathrm{poly}(n,m)$ *with success probability at least* $\alpha - q \cdot \delta$ *where*

$$\begin{aligned} \delta &\leq 2^{(n+1)/2} \cdot (\beta + \tfrac{2w}{m})^d \\ \epsilon &= \tfrac{1}{2} - \tfrac{1}{2}(1 - \tfrac{2w}{m})^d . \end{aligned}$$

Proof. Assume $\mathcal{A}$ is an algorithm for the LPN problem as in the theorem. Define $\mathcal{B}$ as follows:

- Input: $\mathbf{C} \in \mathbb{Z}_2^{n \times m}$, $\mathbf{t} \in \{0,1\}^m$. By assumption $\mathbf{C}$ is the generator of a β-balanced code and $\mathbf{t}^T = \mathbf{s}^T\mathbf{C} + \mathbf{x}^T$ for some $\mathbf{s} \in \mathbb{Z}_2^n, \mathbf{x} \in \mathbb{Z}_2^m$ with $\mathsf{wt}(\mathbf{x}) \leq w$.

1. Sample $\mathbf{s}' \leftarrow \mathbb{Z}_2^n$ and set $\mathbf{c}^T = \mathbf{t}^T + (\mathbf{s}')^T\mathbf{C} = (\mathbf{s} + \mathbf{s}')^T\mathbf{C} + \mathbf{x}^T$.
2. Run the algorithm $\mathcal{A}$. Every time $\mathcal{A}$ request a new LPN sample, choose $\mathbf{r} \leftarrow \mathcal{R}_{d,m}$ and set $\mathbf{a} = \mathbf{C}\mathbf{r}$, $b = \mathbf{c}^T\mathbf{r}$ and give $\mathbf{a}, b$ to $\mathcal{A}$.
3. If at some point $\mathcal{A}$ outputs $\mathbf{s}^* \in \mathbb{Z}_2^n$ then output $\mathbf{s}^* - \mathbf{s}'$.

By Lemma 3.1 each of the values $(\mathbf{a}, b)$ given to $\mathcal{A}$ during step 2 is δ-close to a fresh sample from $\mathcal{O}_{\mathbf{s}^*,\epsilon}^{n}$ where $\mathbf{s}^* = \mathbf{s} + \mathbf{s}'$ is uniformly random over $\mathbb{Z}_2^n$. By assumption, if $\mathcal{A}$ were actually given samples from $\mathcal{O}_{\mathbf{s}^*,\epsilon}^{n}$ is step 2 it would output $\mathbf{s}^*$ in step 3 with probability α. Therefore if $\mathcal{A}$ makes q queries in step 2, the probability that it outputs $\mathbf{s}^*$ in step 3 is at least $\alpha - q\delta$. This proves the theorem. □

Corollary 4.2. *Let* $m = n^c$ *for some constant* $c > 1$, $\beta = \frac{1}{\sqrt{n}}, w = \lceil m\frac{\log^2 n}{n} \rceil$. *Assume that search* $\mathsf{balNCP}_{n,m,w,\beta}$ *is hard in the worst case, meaning that for every polynomial time algorithm* *its success probability on the worst case instance is at most* $\mathrm{negl}(n)$*. Then for some* $\epsilon < \frac{1}{2} - \frac{1}{O(n^4)}$ *search* LPN_ϵ^n *is hard in the average case, meaning that for every polynomial time algorithm its success probability on a random instance is at most* $\mathrm{negl}(n)$*.*

Proof. Follows directly from the theorem by setting $d = \lceil 2n/\log n \rceil$ and noting that:

$$\begin{aligned}\delta &\leq 2^{(n+1)/2} \cdot (\beta + \tfrac{2w}{m})^d \leq 2^{(n+1)/2 - (d/2)\log n + O(1)} \leq 2^{-n/2 + O(1)} = \mathrm{negl}(n) \\ \epsilon &= \tfrac{1}{2} - \tfrac{1}{2}(1 - \tfrac{2w}{m})^d \leq \frac{1}{2} - 2^{-(4\frac{w}{m}d + 1)} \leq \frac{1}{2} - 1/O(n^4)\end{aligned}$$

□

The above says that the wost-case hardness of balNCP with very low error-rate $w/m \approx \frac{\log^2 n}{n}$ implies the average-case hardness of LPN_ϵ^n with very high error-rate $\epsilon = 1/2 - 1/O(n^4)$. Note that a random linear code is β-balanced with overwhelming probability when $\beta \geq 3\sqrt{n/m}$ so for a sufficiently large m the restriction on β is satisfied by most codes.

Other choices of parameters may also be interesting. For example, we can set the error-rate to be $w/m \approx 1/\sqrt{n}$ and $d = 2n/\log n$ while keeping $m = n^c$ for some $c > 1$, $\beta = 1/\sqrt{n}$ the same as before. Then if we assume that $\mathsf{balNCP}_{n,m,w,\beta}$ is $(T(n), \delta(n))$ hard in the worst case (meaning that for every $T(n)$ time algorithm the success probability on the worst case instance is at most $\delta(n)$) this implies LPN_ϵ^n is $(T'(n), \delta'(n))$ hard in the average where $\epsilon(n) = 1/2 - 2^{-\sqrt{n}/\log n}$, $T'(n) = T(n)/\mathrm{poly}(n)$ and $\delta'(n) = \delta(n) + T'(n)2^{-(n-1)/2}$. Note that, as far as we know, the $\mathsf{balNCP}_{n,m,w,\beta}$ with noise rate $w/m = 1\sqrt{n}$ may be $(T(n), \delta(n))$ hard for some $T(n) = 2^{\Omega(\sqrt{n})}$, $\delta(n) = 2^{-\Omega(\sqrt{n})}$, which would imply the same asymptotic hardness for LPN_ϵ^n. Although the error-rate $\epsilon = 1/2 - 2^{-\sqrt{n}/\log n}$ is extremely high, it is not high enough for the conclusion to hold statistically and therefore this connection may also be of interest.

5 Statistical Zero Knowledge for Balanced NCP and LPN

In this section, we show that for certain parameter regimes, $\mathsf{balNCP} \in$ Search-$\mathcal{BPP}^{\mathcal{SZK}}$ and is thus unlikely to be $\mathcal{NP}$-hard [MX10]. Towards this end, we use a decision to search reduction analogous to the canonical one known for the LPN problem. We consider the following randomized samplers (with an additional implicit parameter d):

- Randomized sampler $\mathsf{Samp}_0(\mathbf{C}, \mathbf{t})$ takes as input a matrix $\mathbf{C} \in \{0,1\}^{n \times m}$ and a word $\mathbf{t} \in \{0,1\}^m$. It samples $\mathbf{r} \overset{\$}{\leftarrow} \mathcal{R}_{d,m}$ and outputs $(\mathbf{Cr}, \mathbf{t}^T\mathbf{r})$.
- Randomized sampler $\mathsf{Samp}_{i,\sigma}(\mathbf{C}, \mathbf{t})$ is parameterized by $i \in [n]$, $\sigma \in \{0,1\}$, takes as input a matrix $\mathbf{C} \in \{0,1\}^{n \times m}$ and a word $\mathbf{t} \in \{0,1\}^m$. It samples $\mathbf{r} \overset{\$}{\leftarrow} \mathcal{R}_{d,m}$ and $\rho \in \{0,1\}$ and outputs $(\mathbf{Cr} + \rho\mathbf{u}_i, \mathbf{t}^T\mathbf{r} + \rho\sigma)$.

Lemma 5.1. *Consider a generating matrix* $\mathbf{C} \in \{0,1\}^{n \times m}$ *for a* β*-balanced code, and let* $\mathbf{t} = \mathbf{s}^T\mathbf{C} + \mathbf{x}^T$ *for some* $\mathbf{s} \in \{0,1\}^n$ *and* $\mathbf{x}$ *with hamming weight* w*. Then the following hold:*

1. *The sampler* $\mathsf{Samp}_0(\mathbf{C}, \mathbf{t})$ *samples from a distribution that is* δ*-close to* $\mathcal{U}_{\{0,1\}^n} \times \mathsf{Ber}_\epsilon$*.*
2. *If* $\mathbf{s}_i = \sigma$ *then* $\mathsf{Samp}_{i,\sigma}(\mathbf{C}, \mathbf{t})$ *samples from a distribution that is* δ*-close to* $\mathcal{U}_{\{0,1\}^n} \times \mathsf{Ber}_\epsilon$*.*
3. *If* $\mathbf{s}_i \neq \sigma$ *then* $\mathsf{Samp}_{i,\sigma}(\mathbf{C}, \mathbf{t})$ *samples from a distribution that is* δ*-close to* $\mathcal{U}_{\{0,1\}^n} \times \mathcal{U}_{\{0,1\}}$*.*

Here, $\epsilon = \frac{1}{2} - \frac{1}{2}(1 - \frac{2w}{m})^d$, $\delta = 2^{(n+1)/2} \cdot (\beta + \frac{2w}{m})^d$.

Proof. Assertion 1 follows directly from Lemma 3.1.

For Assertion 2 we note that if $\mathbf{s}_i = \sigma$ then

$$(\mathbf{Cr} + \rho\mathbf{u}_i, \mathbf{t}^T\mathbf{r} + \rho\sigma) = (\mathbf{Cr}, \mathbf{t}^T\mathbf{r}) + \rho(\mathbf{u}_i, \sigma) = (\mathbf{Cr}, \mathbf{t}^T\mathbf{r}) + (\rho\mathbf{u}_i, \mathbf{s}^T(\rho\mathbf{u}_i)) \ .$$

By Lemma 3.1 this distribution is within δ statistical distance to

$$(\mathbf{a}, \mathbf{s}^T\mathbf{a} + e) + (\rho\mathbf{u}_i, \mathbf{s}^T(\rho\mathbf{u}_i)) \ ,$$

with $(\mathbf{a}, e)$ distributed $\mathcal{U}_{\{0,1\}^n} \times \mathsf{Ber}_\epsilon$. Finally, we can write

$$(\mathbf{a}, \mathbf{s}^T\mathbf{a} + e) + (\rho\mathbf{u}_i, \mathbf{s}^T(\rho\mathbf{u}_i)) = ((\mathbf{a} + \rho\mathbf{u}_i), \mathbf{s}^T(\mathbf{a} + \rho\mathbf{u}_i) + e) \ ,$$

and since $\mathbf{a}' = \mathbf{a} + \rho\mathbf{u}_i$ is also uniformly distributed, the assertion follows.

For Assertion 3 we note that when $\mathbf{s}_i \neq \sigma$, i.e. $\sigma = \mathbf{s}_i + 1$ then

$$(\mathbf{Cr} + \rho\mathbf{u}_i, \mathbf{t}^T\mathbf{r} + \rho\sigma) = (\mathbf{Cr}, \mathbf{t}^T\mathbf{r}) + \rho(\mathbf{u}_i, \sigma) = (\mathbf{Cr}, \mathbf{t}^T\mathbf{r}) + (\rho\mathbf{u}_i, \mathbf{s}^T(\rho\mathbf{u}_i)) + (\mathbf{0}, \rho) \ .$$

As above, by Lemma 3.1, this distribution is within δ statistical distance to

$$(\mathbf{a}, \mathbf{s}^T\mathbf{a} + e) + (\rho\mathbf{u}_i, \mathbf{s}^T(\rho\mathbf{u}_i)) = (\underbrace{(\mathbf{a} + \rho\mathbf{u}_i)}_{\mathbf{a}'}, \mathbf{s}^T(\mathbf{a} + \rho\mathbf{u}_i) + e) + (\mathbf{0}, \rho) = (\mathbf{a}', \mathbf{s}^T\mathbf{a}' + e + \rho) \ ,$$

with $(\mathbf{a}, e)$ distributed $\mathcal{U}_{\{0,1\}^n} \times \mathsf{Ber}_\epsilon$, and thus also $(\mathbf{a}', e)$ distributed $\mathcal{U}_{\{0,1\}^n} \times \mathsf{Ber}_\epsilon$ and independent of ρ. Since ρ is uniform and independent of $(\mathbf{a}', e)$ it follows that $(\mathbf{a}', \mathbf{s}^T\mathbf{a}' + e + \rho)$ is distributed $\mathcal{U}_{\{0,1\}^n} \times \mathcal{U}_{\{0,1\}}$. □

The following is an immediate corollary of Lemma 5.1.

Corollary 5.2. *If $\mathbf{s}_i = \sigma$ then the distributions generated by $\mathsf{Samp}_{i,\sigma}(\mathbf{C}, \mathbf{t})$ and $\mathsf{Samp}_0(\mathbf{C}, \mathbf{t})$ are within statistical distance* at most 2δ.

If $\mathbf{s}_i \neq \sigma$ then the distributions generated by $\mathsf{Samp}_{i,\sigma}(\mathbf{C}, \mathbf{t})$ and $\mathsf{Samp}_0(\mathbf{C}, \mathbf{t})$ are within statistical distance at least $(1 - 2\epsilon) - 2\delta$.

Proof. A direct calculation shows that the statistical distance between Ber_ϵ and $\mathcal{U}_{\{0,1\}}$ is $1 - 2\epsilon$. Plugging in Lemma 5.1, the result follows. □

We define the notion of a direct product sampler. This is just a sampler that outputs multiple samplers.

Definition 5.1. *Let $\mathcal{D}$ be a distribution and let $k \in \mathbb{N}$, then $\mathcal{D}^{(k)}$ is the distribution defined by k independent samples from $\mathcal{D}$.*

Lemma 5.3. *Consider distributions $\mathcal{D}_1, \mathcal{D}_2$ and values $0 \leq \delta_1 \leq \delta_2 \leq 1$ s.t. $\mathsf{dist}(\mathcal{D}_1, \mathcal{D}_2) \in (\delta_1, \delta_2)$. Let $k \in \mathbb{N}$ then $\mathsf{dist}(\mathcal{D}_1^{(k)}, \mathcal{D}_2^{(k)}) \in (1 - c_1 e^{-c_2\delta_1^2 k}, k\delta_2)$. For some positive constants c_1, c_2.*

Proof. The upper bound follows by union bound and the lower bound from the Chernoff bound. □

Theorem 5.4. *There exists a Search-$\mathcal{BPP}^{\mathcal{SZK}}$ algorithm for solving* balNCP *on instances of the following form. Letting* $\mathbf{C} \in \{0,1\}^{n \times m}$, $\mathbf{t} \in \{0,1\}^m$, $w \in [m]$ *denote the* balNCP *input, we require that the code generated by* $\mathbf{C}$ *is* β*-balanced and that* n, m, w, β *are such that there exist* $d \in [m]$ *and* $k \leq \text{poly}(n,m)$ *for which*

$$2\delta k < 1/3 \tag{9}$$

for $\delta = 2^{(n+1)/2} \cdot (\beta + \frac{2w}{m})^d$, *and*

$$c_1 e^{-c_2(1-2\epsilon-2\delta)^2 k} < 1/3 \tag{10}$$

for δ *as above,* $\epsilon = \frac{1}{2} - \frac{1}{2}(1 - \frac{2w}{m})^d$, *and* c_1, c_2 *are the constants from Lemma 5.3.*

Proof. We recall the problem Statistical Distance (SD) which is in $\mathcal{SZK}$. This problem takes as input two sampler circuits and outputs 0 if the inputs sample distributions that are within statistical distance $< 1/3$ and 1 if the distributions are within statistical distance $> 2/3$. We will show how to solve balNCP for the above parameters using an oracle to SD.

Specifically, for all $i = 1, \dots, n$ and $\sigma \in \{0,1\}$, the algorithm will call the SD oracle on input $(\mathsf{Samp}_0^{(k)}(\mathbf{C}, \mathbf{t}), \mathsf{Samp}_{i,\sigma}^{(k)}(\mathbf{C}, \mathbf{t}))$, where $\mathsf{Samp}_{(\cdot)}^{(k)}$ is the algorithm that runs the respective Samp k times and outputs all k generated samples.

Let $\alpha_{i,\sigma}$ denote the oracle response on the (i, σ) call. Then if for any i it holds that $\alpha_{i,0} = \alpha_{i,1}$, then return $\perp$. Otherwise set $\mathbf{s}_i$ to the value σ for which $\alpha_{i,\sigma} = 0$. Return $\mathbf{s}$.

By definition of our samplers, they run in polynomial time, so if k is polynomial then our inputs to SD are indeed valid. Combining Corollary 5.2 and Lemma 5.3, it holds that $\alpha_{i,\sigma} = 0$ if and only if $\mathbf{s}_i^* = \sigma$, where $\mathbf{s}^*$ is the vector for which $\mathbf{t}^T = (\mathbf{s}^*)^T \mathbf{C} + \mathbf{x}^T$ and $\mathsf{wt}(\mathbf{x}) = w$. The correctness of the algorithm follows. □

Corollary 5.5. *Let* $m = n^c$ *for some constant* $c > 1$, $\beta = \frac{1}{\sqrt{n}}, w = \lceil m \frac{\log^2 n}{n} \rceil$. *Then search* $\mathsf{balNCP}_{n,m,w,\beta} \in$ *Search-*$\mathcal{BPP}^{\mathcal{SZK}}$.

Proof. In Theorem 5.4 set $d = \lceil 2n / \log n \rceil$ and $k = n^9$. By the same calculation as in Corollary 4.2 we have $\delta = \text{negl}(n)$ and $\epsilon \leq \frac{1}{2} - 1/O(n^4)$. Therefore for large enough n we have $2\delta k < 1/3$ and $c_1 e^{-c_2(1-2\epsilon-2\delta)^2 k} = e^{-\Omega(n)} < 1/3$ as required by the theorem. □

On Statistical Zero Knowledge and LPN. We notice that since sparse random codes are balanced with overwhelming probability (Lemma 2.2), our results in this section also imply that the LPN problem is in Search-$\mathcal{BPP}^{\mathcal{SZK}}$ for error value $\frac{\log^2 n}{n}$. We note that even though in LPN the weight of the noise vector (the distance from the code) is not fixed as in our definition of balNCP, the domain of possible weights is polynomial and thus the exact weight can be guessed with polynomial success probability. Once a successful guess had been made, it can be verified once a solution had been found.

6 Collision-Resistant Hashing

In this section, we describe a collision-resistant hash function family whose security is based on the hardness of the (decisional) $\mathsf{LPN}^n_{O(\log^2 n/n)}$ problem. For any positive constant $c \in \mathbb{R}^+$ and a matrix $\mathbf{A} \in \mathbb{Z}_2^{n \times n^{1+c}}$, define the function

$$h_{\mathbf{A}} : \mathcal{S}^{n^{1+c}}_{2n/(c\log n)} \to \mathbb{Z}_2^n \quad \text{as} \quad h_{\mathbf{A}}(\mathbf{r}) := \mathbf{A}\mathbf{r}. \tag{11}$$

Notice that because

$$\left|\mathcal{S}^{n^{1+c}}_{2n/(c\log n)}\right| = \binom{n^{1+c}}{2n/(c\log n)} > \left(\frac{n^{1+c}}{2n/(c\log n)}\right)^{2n/(c\log n)} > 2^{2n}$$

and the size of $\mathbb{Z}_2^n$ is exactly 2^n, the function $h_{\mathbf{A}}$ is compressing.

We now relate the hardness of finding collisions in the function $h_{\mathbf{A}}$, for a random $\mathbf{A}$, to the hardness of the decisional LPN^n_ϵ problem.

Theorem 6.1. *For any constant $c > 0$, if there exists an algorithm A_1 running in time t such that*

$$\Pr\left[\mathsf{A}_1(h_{\mathbf{A}}) \Rightarrow (\mathbf{r}_1, \mathbf{r}_2) \in \mathcal{S}^{n^{1+c}}_{2n/(c\log n)} \text{ s.t. } \mathbf{r}_1 \neq \mathbf{r}_2 \text{ and } h_{\mathbf{A}}(\mathbf{r}_1) = h_{\mathbf{A}}(\mathbf{r}_2) \ ; \ \mathbf{A} \leftarrow \mathbb{Z}_2^{n\times n^{1+c}}\right] \geq \delta,$$

then there exists an algorithm A_2 that runs in time $\approx t$ and solves the decisional LPN^n_ϵ problem for any $\epsilon \leq \frac{1}{4}$ with advantage at least $\delta \cdot 2^{-16n\epsilon/(c\log n)-1}$.

In particular, for $\epsilon = O(\log^2 n/n)$ and any $\delta = 1/\mathrm{poly}(n)$, the advantage is $1/\mathrm{poly}(n)$.

Proof. The algorithm A_2 has access to an oracle that is either $\mathcal{O}^n_{\mathbf{s},\epsilon}$ or $\mathcal{U}^n$. He calls the oracle n^{1+c} times to obtain samples of the form $(\mathbf{a}_i, b_i)$. He arranges the $\mathbf{a}_i$ and b_i into a matrix $\mathbf{A}$ and vector $\mathbf{b}$ as

$$\mathbf{A} = \left[\ \mathbf{a}_1 \mid \cdots \mid \mathbf{a}_{n^{1+c}}\ \right] \in \mathbb{Z}_2^{n\times n^{1+c}}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \cdots \\ b_{n^{1+c}} \end{bmatrix} \in \mathbb{Z}_2^{n^{1+c}}$$

and sends $\mathbf{A}$ to A_1. If A_1 fails to return a valid answer, then A_2 outputs $\mathsf{ans} \leftarrow \{0,1\}$. If A_1 does return valid distinct $\mathbf{r}_1$ and $\mathbf{r}_2$ such that $h_{\mathbf{A}}(\mathbf{r}_1) = h_{\mathbf{A}}(\mathbf{r}_2)$, then A_2 returns $\mathsf{ans} = \mathbf{b}^T(\mathbf{r}_1 - \mathbf{r}_2)$.

We first look at the distribution of ans when the oracle that A_2 has access to is $\mathcal{U}^n$. In this case it's easy to see that regardless of whether A_1 returns a valid answer, we'll have $\Pr[\mathsf{ans} = 0] = \frac{1}{2}$ because $\mathbf{b}$ is completely uniform in $\mathbb{Z}_2^{n^{1+c}}$.

On the other hand, if the oracle is $\mathcal{O}^n_{\mathbf{s},\epsilon}$, then we know that for all i, $b_i = \mathbf{s}^T\mathbf{a}_i + e_i$, where $e_i \leftarrow \mathsf{Ber}_\epsilon$. This can be rewritten as $\mathbf{s}^T\mathbf{A} + \mathbf{e}^T = \mathbf{b}^T$ where $\mathbf{e} = \begin{bmatrix} e_1 \\ \cdots \\ e_{n^{1+c}} \end{bmatrix}$. Therefore

$$\mathbf{b}^T(\mathbf{r}_1 - \mathbf{r}_2) = \mathbf{A}(\mathbf{r}_1 - \mathbf{r}_2) + \mathbf{e}^T(\mathbf{r}_1 - \mathbf{r}_2) = \mathbf{e}^T(\mathbf{r}_1 - \mathbf{r}_2).$$

Since $\mathsf{wt}(\mathbf{r}_i) = 2n/(c\log n)$, we know that $\mathsf{wt}(\mathbf{r}_1 - \mathbf{r}_2) \leq 4n/(c\log n)$. Since the $\mathbf{A}$ that is sent to A_1 is independent of $\mathbf{e}$, we have that

$$\Pr[\mathbf{e}^T \cdot (\mathbf{r}_1 - \mathbf{r}_2) = 0 \ ; \ e_i \leftarrow \mathsf{Ber}_\epsilon] \geq \frac{1}{2} + \frac{1}{2}(1-2\epsilon)^{4n/(c\log n)} \geq \frac{1}{2} + 2^{-16n\epsilon/(c\log n)-1}, \tag{12}$$

where the first inequality follows from Lemma 2.1 and the second inequality is due to the assumption that $\epsilon \leq \frac{1}{4}$ and the fact that $1 - x \geq 2^{-2x}$ for $x \leq 1/2$.

Thus when the oracle is $\mathcal{O}^n_{\mathbf{s},\epsilon}$, we have

$$\Pr[\mathsf{ans} = 0] \geq \frac{1}{2}\cdot(1-\delta) + \left(\frac{1}{2} + 2^{-16n\epsilon/(c\log n)-1}\right)\cdot\delta = \frac{1}{2} + \delta\cdot 2^{-16n\epsilon/(c\log n)-1}.$$

□

6.1 Observations and Other Parameter Regimes.

As far as we know, the best attack against the hash function in (11) with $c = 1$ requires $2^{\Omega(n)}$ time, whereas the $\mathsf{LPN}^n_{\log^2 n/n}$ problem, from which we can show a polynomial-time reduction, can be solved in time $2^{O(\log^2 n)}$. Thus there is possibly a noticeable loss in the reduction for this parameter setting. It was observed in [YZW$^+$17, Theorem 2, Theorem 3] that there are other ways to set the parameters in Theorem 6.1 which achieve different connections between the hash function and the underlying LPN problem. For example, defining $n = \log^2 m$ and $c = \log m/\log\log m - 1$ implies that there exists a hash function defined by the matrix $\mathbf{A} \in \mathbb{Z}_2^{\log^2 m \times 2m}$ such that succeeding with probability δ in finding collisions in this hash function is at least as hard as solving $\mathsf{LPN}^{\log^2 m}_\epsilon$ problem with advantage $\delta \cdot m^{-O(\kappa\epsilon)}$ for a constant κ. This is exactly the parameter setting in [YZW$^+$17, Theorem 3].[2]

Based on the state of the art of today's algorithms, it's clear that using a hash function defined by an $n \times n^2$ matrix $\mathbf{A}$ is more secure than one defined by a $\log^2 n \times 2n$ matrix (since one can trivially find collisions in the latter in time $2^{O(\log^2 n)}$). There is, however, no connection that we're aware of between the LPN problems on which they are based via Theorem 6.1. In particular, we do not know of any polynomial-time (in n) reductions that relate the hardness of the $\mathsf{LPN}^n_{\log^2 n/n}$ problem to the $\mathsf{LPN}^{\log^2 n}_\epsilon$ problem for a constant ϵ.

Acknowledgments

The first author wishes to thank Ben Berger and Noga Ron-Zewi for discussions on the hardness of decoding problems.

References

[ABSS93] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optimia in lattices, codes, and systems of linear equations. In *34th Annual Symposium on Foundations of Computer Science, Palo Alto, California, USA, 3-5 November 1993*, pages 724–733. IEEE Computer Society, 1993.

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 327–343, 2016.

[AHI$^+$17] Benny Applebaum, Naama Haramaty, Yuval Ishai, Eyal Kushilevitz, and Vinod Vaikuntanathan. Low-complexity cryptographic hash functions. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 7:1–7:31, 2017.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 99–108, 1996.

[2]The error ϵ in [YZW$^+$17] can be any constant $< \frac{1}{2}$, whereas our Theorem 6.1 restricts it to $< \frac{1}{4}$. Our restriction is made simply for obtaining a "clean" inequality in Eq. (12) since we were only interested in LPN instances with much lower noise.

[APY09] Noga Alon, Rina Panigrahy, and Sergey Yekhanin. Deterministic approximation algorithms for the nearest codeword problem. In Irit Dinur, Klaus Jansen, Joseph Naor, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings*, volume 5687 of *Lecture Notes in Computer Science*, pages 339–351. Springer, 2009.

[BCD$^+$16] Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1006–1018, 2016.

[BFKL93] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 278–291, 1993.

[BK02] Piotr Berman and Marek Karpinski. Approximating minimum unsatisfiability of linear equations. In David Eppstein, editor, *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 6-8, 2002, San Francisco, CA, USA.*, pages 514–516. ACM/SIAM, 2002.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[BLP$^+$13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 575–584, 2013.

[BLSV17] Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. Anonymous ibe, leakage resilience and circular security from new assumptions. Cryptology ePrint Archive, Report 2017/967, 2017. https://eprint.iacr.org/2017/967, conference version in EUROCRYPT 2018 [BLSV18].

[BLSV18] Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. Anonymous ibe, leakage resilience and circular security from new assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 535–564. Springer, 2018.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 97–106, 2011.

[DMS99] Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 475–485. IEEE Computer Society, 1999.

[FGKP09] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM J. Comput.*, 39(2):606–645, 2009.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178, 2009.

[GGH96] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(42), 1996.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[Gol95] Oded Goldreich. Three xor-lemmas - an exposition. *Electronic Colloquium on Computational Complexity (ECCC)*, 2(56), 1995.

[KPC$^+$11] Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. Efficient authentication from hard learning problems. In *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, pages 7–26, 2011.

[KS06] Jonathan Katz and Ji Sun Shin. Parallel and concurrent security of the HB and hb$^+$ protocols. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, pages 73–87, 2006.

[KS10] Swastik Kopparty and Shubhangi Saraf. Local list-decoding and testing of random linear codes from high error. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 417–426. ACM, 2010.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, pages 386–397, 1993.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 372–381, 2004.

[MV03] Daniele Micciancio and Salil P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, pages 282–298, 2003.

[MX10] Mohammad Mahmoody and David Xiao. On the power of randomized reductions and the checkability of SAT. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, June 9-12, 2010*, pages 64–75, 2010.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 333–342, 2009.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93, 2005.

[SV03] Amit Sahai and Salil P. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2):196–249, 2003.

[Vaz86] Umesh Virkumar Vazirani. *Randomness, Adversaries and Computation (Random Polynomial Time)*. PhD thesis, 1986.

[YZW$^+$17] Yu Yu, Jiang Zhang, Jian Weng, Chun Guo, and Xiangxue Li. Learning parity with noise implies collision resistant hashing. Cryptology ePrint Archive, Report 2017/1260, 2017. https://eprint.iacr.org/2017/1260.