Smoothing Out Binary Linear Codes and Worst-case Sub-exponential Hardness for LPN

Yu Yu* Jiang Zhang†

Abstract

Learning parity with noise (LPN) is a notorious (average-case) hard problem that has been well studied in learning theory, coding theory and cryptography since the early 90's. It further inspires the Learning with Errors (LWE) problem [Regev, STOC 2005], which has become one of the central building blocks for post-quantum cryptography and advanced cryptographic primitives such as fully homomorphic encryption [Gentry, STOC 2009]. Unlike LWE whose hardness can be reducible from worst-case lattice problems, no corresponding worst-case hardness results were known for LPN until very recently. At Eurocrypt 2019, Brakerski et al. [BLVW19] established the first feasibility result that the worst-case hardness of nearest codeword problem (NCP) (on balanced linear code) at the extremely low noise rate $\frac{\log^2 n}{n}$ implies the quasi-polynomial hardness of LPN at the extremely high noise rate $1/2 - 1/\mathsf{poly}(n)$. It remained open whether a worst-case to average-case reduction can be established for standard (constant-noise) LPN, ideally with sub-exponential hardness.

In this paper, we carry on the worst-case to average-case reduction for LPN [BLVW19]. We first expand the underlying binary linear codes (of the worst-case NCP) to not only the balanced code considered in [BLVW19] but also to another code (in some sense dual to balanced code). At the core of our reduction is a new variant of smoothing lemma (for both binary codes) that circumvents the barriers (inherent in the underlying worst-case randomness extraction) and admits tradeoffs for a wider spectrum of parameter choices. In addition to the worst-case hardness result obtained in [BLVW19], we show that for any constant $0 < c < 1$ the constant-noise LPN problem is $(T = 2^{\Omega(n^{1-c})}, \epsilon = 2^{-\Omega(n^{\min(c,1-c)})}, q = 2^{\Omega(n^{\min(c,1-c)})})$-hard assuming that the NCP (on either code) at the low-noise rate $\tau = n^{-c}$ is $(T' = 2^{\Omega(\tau n)}, \epsilon' = 2^{-\Omega(\tau n)}, m = 2^{\Omega(\tau n)})$-hard in the worst case, where T, ϵ, q and m are time complexity, success rate, sample complexity, and codeword length respectively. Moreover, refuting the worst-case hardness assumption would imply arbitrary polynomial speedups over the current state-of-the-art algorithms for solving the NCP (and LPN), which is a win-win result. Unfortunately, public-key encryptions and collision resistant hash functions would need constant-noise LPN with $(T = 2^{\omega(\sqrt{n})}, \epsilon' = 2^{-\omega(\sqrt{n})}, q = 2^{\sqrt{n}})$-hardness (Yu et al., CRYPTO 2016 & ASIACRYPT 2019), which is almost (up to an arbitrary $\omega(1)$ factor in the exponent) what is reducible from the worst-case NCP when $c = 0.5$. We leave it as an open problem whether the gap can be closed or there is a separation in place.

Keywords: Foundations of Cryptography, Worst-case to average-case reduction, Learning Parity with Noise, Smoothing Lemma.

1 Introduction

1.1 Learning Parity with Noise

Learning parity with noise (LPN) [BFKL93] represents a noisy version of the "parity learning problem" in machine learning as well as the "decoding random linear codes" in coding theory. The conjectured hardness of the LPN problem and its variants implies various

*Shanghai Jiao Tong University, Shanghai 200240, China. Email: yuyuathk@gmail.com.

†State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China. Email:jiangzhang09@gmail.com.

cryptographic applications, such as symmetric encryption/authentication [HB01, JW05, KS06, ACPS09, KPC+11, DKPW12, LM13, CKT16], zero-knowledge proof for commitment schemes [JKPT12], oblivious transfer [DDN14], public-key cryptography [Ale03] and collision resistant hash functions [BLVW19, YZW+19]. Regev [Reg05] introduced the problem of learning with errors (LWE) by generalizing LPN to larger moduli and to a broader choice of noise distributions. Both LPN and LWE are believed to be hard problems not succumbing to quantum algorithms and thus constitute promising candidates for post-quantum cryptography. For the past fifteen years LWE has shown great success in founding upon worst-case hard lattice problems [Reg05, Pei09, BLP+13] and as a versatile building block for advanced cryptographic algorithms (such as fully homomorphic encryption [Gen09] and attribute-based encryption [GVW13, BGG+14]). In contrast to LWE, its twelve-year elder cousin LPN remains much less understood. For instance, it was not until recently did we get the first feasibility result about its root of worst-case hardness [BLVW19]. We carry on the research a step further.

The computational version of the Learning Parity with Noise (LPN) problem with secret size $n \in \mathbb{N}$ and noise rate $0 < \mu < 1/2$ asks to recover the random secret $\mathbf{x}$ given $(\mathbf{A},\ \mathbf{A}\cdot\mathbf{x}+\mathbf{e})$, where $\mathbf{x} \xleftarrow{\$} \mathbb{F}_2^n$, $\mathbf{A}$ is a random $q\times n$ Boolean matrix, $\mathbf{e}$ follows the q-fold Bernoulli distribution with parameter μ (i.e., taking the value 1 with probability μ and the value 0 with probability $1-\mu$), '$\cdot$' and '$+$' denote (matrix-vector) multiplication and addition modulo 2 respectively.[1] The decisional version of LPN challenges to distinguish $(\mathbf{A},\ \mathbf{A}\cdot\mathbf{x}+\mathbf{e})$ from uniform randomness. In terms of hardness, the two LPN versions are polynomially equivalent [KS06, AIK07].

LPN has been extensively studied in learning theory, and it was shown in [FGKP06] that an efficient algorithm for LPN would allow to learn several important function classes such as 2-DNF formulas, juntas, and any function with a sparse Fourier spectrum. Typically, the noise rate μ of LPN is constant (i.e., independent of secret size n). The BKW (Blum, Kalai and Wasserman) algorithm [BKW03] solves LPN in time/sample complexity $2^{O(n/\log n)}$. Lyubashevsky [Lyu05] introduced "sample amplification" trick to obtain a variant of the BKW attack with time complexity $2^{O(n/\log\log n)}$ and sample complexity $q = n^{1+\epsilon}$. If further restricted to linearly many samples (i.e., $q = O(n)$) then the best attacks run in exponential time [Ste88, MMT11, BJMM12]. Alekhnovich [Ale03] introduced an interesting noise regime (referred to as low-noise LPN) $\mu = 1/\sqrt{n}$ (or more generally $\mu = n^{-c}$ for $1/2 \le c < 1$) that implies public-key cryptography. More recently, Brakerski et al. [BLVW19] shows that LPN for noise rate $\mu = \frac{\log^2 n}{n}$ (called extremely low-noise LPN) implies collision resistant hash functions. Note that the best solvers for low-noise LPN runs in time $\mathsf{poly}(n)\cdot e^{\mu n}$ [CC98, BLP11, KF15], so the LPN at noise rate $\mu = \frac{\log^2 n}{n}$ is still polynomially hard despite the existence of quasi-polynomial attacks. Alternatively, public-key encryption [YZ16] and collision resistant hash functions [YZW+19] can be constructed under the assumption that the constant-noise LPN problem is $2^{\omega(\sqrt{n})}$-hard given $2^{\sqrt{n}}$ samples, known as the sub-exponential LPN assumption.

1.2 Nearest Codeword Problem and Worst-case Hardness

Quite naturally, the worst-case decoding problem considered in [BLVW19] and this work is the worst-case analogue of the LPN problem, known as the promise version of the Nearest Codeword Problem (NCP). Informally, the problem is about finding out $\mathbf{s}^\mathsf{T} \in \mathbb{F}_2^n$ given a generator matrix $\mathbf{C} \in \mathbb{F}_2^{n\times m}$ for some $[m,n]$ binary linear code ($m > n$) and a noisy codeword $\mathbf{t}^\mathsf{T} = (\mathbf{s}^\mathsf{T}\mathbf{C} + \mathbf{x}^\mathsf{T}) \in \mathbb{F}_2^m$ with the promise that the error vector $\mathbf{x} \in \mathbb{F}_2^m$ has exact Hamming weight $|\mathbf{x}| = w$, as opposed to the general requirement $|\mathbf{x}| \le w$. Note that the difference is not substantial since having smaller weight can only make the problem easier (seen by a simple reduction), and one can enumerate all possible values for w and invoke the corresponding solver (for the

[1] Another equivalent formulation is to find out $\mathbf{s}$ given as many (up to one's computing power and availability of the samples) random noisy inner product $\langle \mathbf{a}_i, \mathbf{s}\rangle + \mathbf{e}_i$ as possible. In this paper we stick to the $\mathbf{Ax}+\mathbf{e}$ representation which is more consistent with that of the worst-case decoding problems.

exact weight). The non-promise version of the NCP problem is known to be NP-hard even to approximate [ABSS97] and the promise version is also NP-hard in the high-noise regime where the Hamming weight of error vector $|\mathbf{x}| \geq (1/2+\epsilon)d$ for minimal distance d of the code and any arbitrarily small constant ϵ [DMS03]. As for the algorithms, Berman and Karpinski [BK02] showed how to search for the $O(n/\log n)$-approximate nearest codeword in polynomial time and Alon, Panigrahy and Yekhanin [APY09] gives a deterministic algorithm with the same parameters, which is the current state-of-the-art for solving NCP.

1.3 Worst-case to Average-case Reductions for LPN

We start with the "sample amplification" technique [Lyu05] that bears some resemblance to the smoothing lemma in [BLVW19]. The idea is to use polynomially many LPN samples, say $(\mathbf{C}, \mathbf{t}^\mathsf{T} = (\mathbf{s}^\mathsf{T}\mathbf{C} + \mathbf{x}^\mathsf{T}))$, as a basis to generate much more samples (with a higher noise), which enables meaningful tradeoff between sample and time complexities for the BKW algorithm. In more details, a "sample amplification" oracle take as input $(\mathbf{C}, \mathbf{t}^\mathsf{T})$ and responds with $(\mathbf{C}\mathbf{r}_i, \mathbf{t}^\mathsf{T}\mathbf{r}_i = \mathbf{s}^\mathsf{T}\mathbf{C}\mathbf{r}_i + \mathbf{x}^\mathsf{T}\mathbf{r}_i)$ as the i-th re-randomized LPN sample, where $\mathbf{r}_i \leftarrow R$ and $(\mathbf{C}, \mathbf{C}\mathbf{r}_i, \mathbf{x}^\mathsf{T}\mathbf{r}_i)$ is statically close to $(\mathbf{C}, \mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r}_i)$ by the leftover hash lemma. Preferably distribution R should be maximized with min-entropy (of more than n bits) while keeping as small Hamming weight as possible (to make $\mathbf{x}^\mathsf{T}\mathbf{r}_i$ biased) at the same time, so a natural candidate can be a random length-m-weight-d distribution or similar (e.g., m-fold Bernoulli distribution for parameter $\frac{d}{m}$), where $d \ll m$ is a tunable parameter. Döttling [Döt15] used a computational version of this technique which yields better parameters by relying on the dual-LPN assumption (in place of the leftover hash lemma) for pseudorandomness generation.

In the context of reducing worst-case hard promise-NCP to average-case hard LPN [BLVW19], let $(\mathbf{C}, \mathbf{t}^\mathsf{T} = (\mathbf{s}^\mathsf{T}\mathbf{C} + \mathbf{x}^\mathsf{T}))$ be an NCP instance, where $\mathbf{C} \in \mathbb{F}_2^{n\times m}$, $\mathbf{s} \in \mathbb{F}_2^n$, $\mathbf{x} \in \mathbb{F}_2^m$ with $|\mathbf{x}| = w$ are all fixed values, and the goal to generate randomized LPN sample $(\mathbf{C}\mathbf{r}_i, \mathbf{t}^\mathsf{T}\mathbf{r}_i + \mathbf{u}^\mathsf{T}\mathbf{C}\mathbf{r}_i = (\mathbf{s}^\mathsf{T} + \mathbf{u}^\mathsf{T})\mathbf{C}\mathbf{r}_i + \mathbf{x}^\mathsf{T}\mathbf{r}_i)$ with random $\mathbf{u} \xleftarrow{\$} \mathbb{F}_2^n$ and each $\mathbf{r}_i$ drawn from a random weight-d distribution[2]. The difference is that $\mathbf{C}$ is a generator matrix for a specific code (instead of being sampled from uniform), and $\mathbf{s}$ is masked by random $\mathbf{u}$. Brakerski et al. [BLVW19] showed that if $\mathbf{C}$ belongs to a β-balanced code for $\beta = O(\sqrt{n/m})$, i.e., the Hamming distance lies in between $(1/2-\beta)m$ and $(1/2+\beta)m$, then $(\mathbf{C}\mathbf{r}_i, \mathbf{x}^\mathsf{T}\mathbf{r}_i)$ is $2^{\frac{n}{2}} \cdot (\frac{2w}{m} + \beta)^d$ close to $(\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r}_i)$, where $\Pr[\mathbf{x}^\mathsf{T}\mathbf{r}_i = 1] = 1/2 - e^{-\Theta(\frac{w}{m}d)}$ is noise rate of the LPN. As a main result, the worst-case hardness of the NCP on balanced code of noise rate $\frac{w}{m} = \frac{\log^2 n}{n}$ implies the average-case hardness of LPN of noise rate[3] $\mu = 1/2 - 1/\mathsf{poly}(n)$. This was the only known result for basing LPN on worst-case hardness assumptions. It mainly establishes the feasibility result, i.e., assuming polynomial hardness for extremely low-noise NCP (for which quasi-polynomial attacks are known) only to reach the conservative conclusion that extremely high-noise LPN is quasi-polynomially hard. Therefore, it remained open whether worst-hardness guarantee can be secured for LPN of a lower noise, such as constant-noise LPN with sub-exponential hardness shown in this paper.

Curiously, one can investigate existentially (using probabilistic method) the possibility of extending the reduction to constant-noise LPN. Think of a uniformly random $\mathbf{C} \xleftarrow{\$} \mathbb{F}_2^{n\times m}$, and by the leftover hash lemma $(\mathbf{C}, \mathbf{C}\mathbf{r}, \mathbf{x}^\mathsf{T}\mathbf{r})$ is 2^{-n}-close to $(\mathbf{C}, \mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})$ provided that the random m-choose-d distribution $\mathbf{r}$ has sufficient min-entropy $d\log(m/d) = \Omega(n)$. It follows by Markov inequality that there exists at least a $(1-2^{-n/2})$-fraction of "good" $\mathbf{C}$ satisfying $(\mathbf{C}\mathbf{r}, \mathbf{x}^\mathsf{T}\mathbf{r})$ is $2^{-n/2}$-close to $(\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})$. Take into account that $\mathbf{x}$ has $\binom{m}{w}$ possible values, the fraction of "bad" $\mathbf{C}$ amounts up to $\binom{m}{w}2^{-\frac{n}{2}}$. In terms of parameters, we set $\frac{w}{m}d = \Theta(1)$ for constant-noise LPN, noise rate $\frac{w}{m} = \omega(\frac{\log n}{n})$ is necessary for the hardness assumption to hold and recall the

[2] Strictly speaking, $\mathbf{r}_i$ is sampled from $R_{d,m}$ whose definition is deferred to Section 2.1.

[3] The result of [BLVW19] only reduces to LPN of noise rate $1/2 - 2^{-\omega(1)}$, see Remark 3.1 for more discussions.

entropy condition $d\log(m/d) = \Omega(n)$, which implies $d = o(\frac{n}{\log n})$ and thus

$$\log \binom{m}{w} \approx w\log(m/w) = \Omega(\frac{m}{d}\log d) = 2^{\Omega(n/d)}\log d = n^{\omega(1)} \ .$$

This means the upper bound $\binom{m}{w}2^{-O(n)}$ on the fraction of "bad" $\mathbf{C}$ is useless (i.e., greater than 1). In other words, we don't have a straightforward non-constructive proof that the worst-case hardness of NCP problem (on any binary linear codes) implies the hardness of constant-noise LPN, and solving this problem needs new ideas to beat the union bound.

1.4 Our Contributions

We consider the promise version of NCP on two classes of binary linear codes, i.e., balanced code considered in [BLVW19] and (a relaxed form of) independent code. Informally, a β-balanced $[m,n]$ code is a strengthened form of $[m,n,m(1/2-\beta)]$ code whose maximal distance is $m(1/2+\beta)$, and a k-independent $[m,n]$ code is dual to a $[m,m-n,k+1]$ code. Instead of sampling $\mathbf{r}$ from a random weight-d distribution, we let $\mathbf{r}$ follow Bernoulli distribution $\mathcal{B}^m_{\frac{d}{m}}$ (i.e., with expected Hamming weight d). While this looks like a weakening of the distribution ($\mathbf{r}$ is now only $2^{-\Omega(d)}$-close to a random weight-roughly-d distribution), the condition that all bits of $\mathbf{r}$ are independent is crucial for proving a tighter version of smooth lemma that avoids the accumulative loss due to union bound. For proper parameter choice that guarantees: (1) $\mathbf{r}$ is $2^{-\Omega(d)}$-close to having min-entropy $d\log(m/d) = \Omega(n)$ and (2) the code exists in overwhelming abundance, we prove for each code a corresponding smooth lemma that $(\mathbf{Cr}, \mathbf{x}^\mathsf{T}\mathbf{r})$ is $\frac{2^{-\Omega(d)}}{1-2\mu}$-close to $(\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})$, where $\mu \stackrel{\mathsf{def}}{=} \Pr[\mathbf{x}^\mathsf{T}\mathbf{r}=1] = 1/2 - e^{-\Theta(\frac{w}{m}d)}$ is the noise rate of the LPN. Compared to the unconditional case where (it can be shown that) $\mathbf{Cr}$ is $2^{-\Omega(d)}$-close to $\mathbf{U}_n$, the result is worsened by only a factor of $\frac{1}{1-2\mu}$, rather than suffering from the multiplicative factor $\binom{m}{w}$ in the aforementioned non-constructive analysis. The result of [BLVW19] falls into a corollary by setting $\frac{w}{m} = \frac{\log^2 n}{n}$, $m = \mathsf{poly}(n)$, $d = 2n/\log n$ such that $\mu = 1/2 - 1/\mathsf{poly}(n)$. Furthermore, our smoothing lemma allows to transform sub-exponential worst-case hardness of NCP into the sub-exponential average-case hardness for constant-LPN, where the underlying NCP lies in the low-noise regime $\frac{w}{m} = n^{-c}$ $(0 < c < 1)$. In particular, we assume there exists some constant $0 < \varepsilon < 1$ such that NCP problem is $2^{\varepsilon\frac{w}{m}n}$-hard on either code of codeword length, say[4] $m = 2^{\frac{\varepsilon}{8}\frac{w}{m}n}$. To our best knowledge, the state-of-the-art algorithms [BK02, APY09] solve the worst-case NCP with complexity $\mathsf{poly}(n,m)e^{\frac{w}{m}n}$, and we are not aware of any algorithms with additional accelerations for the balanced/independent codes. In fact, we don't even know a much better algorithm for its average-case analogue, i.e., the LPN problem of noise rate $\mu = n^{-c}$ $(0 < c < 1)$ needs time $\mathsf{poly}(n)e^{\mu n}$ to solve with overwhelming success [KF15, Appendix C]. Falsifying our assumption would imply arbitrary polynomial speedups over the current state-of-the-art, i.e., for every constant $\varepsilon > 0$ there exists an algorithm that runs in time $2^{\varepsilon\frac{w}{m}n}$ and solves the problem in worst case (for at least infinitely many values of n), which is a win-win situation.

Theorem 1.1 (main result, informal) *Assume that the NCP problem at noise rate $\frac{w}{m} = n^{-c}$, on either balanced code or independent code, is $(T = 2^{\Omega(n^{1-c})}, m = 2^{\Omega(n^{1-c})})$-hard. Then,*
(1) for $0 < c < 1/2$, the constant-noise LPN is $(T = 2^{\Omega(n^{1-c})}, \epsilon = 2^{-\Omega(n^c)}, q = 2^{\Omega(n^c)})$-hard;
(2) for $1/2 \le c < 1$, the constant-noise LPN is $(T = 2^{\Omega(n^{1-c})}, \epsilon = 2^{-\Omega(n^{1-c})}, q = 2^{\Omega(n^{1-c})})$-hard.

Here the (T,ϵ,q)-hardness of LPN refers to that no algorithm of time T can solve LPN of q samples with probability better than ϵ. The constant-noise LPN with sub-exponential hardness already implies efficient symmetric-key cryptographic applications, and we further discuss

[4] We just need $T \ge \mathsf{poly}(m,n)$. One may replace $m = 2^{\frac{\varepsilon}{8}\frac{w}{m}n}$ with $m = 2^{\delta\varepsilon\frac{w}{m}n}$ for any small constant δ. In general, the hardness of NCP (resp., LPN) is insensitive to codeword length m (resp., sample complexity q).

possibilities of going beyond minicrypt[5]. Unfortunately, for whatever reason that could be interesting, public-key cryptography and collision resistant hash functions require constant-noise LPN with $(T = 2^{\omega(n^{0.5})}, \epsilon = 2^{-\omega(n^{0.5})}, q = 2^{n^{0.5}})$-hardness [YZ16, YZW+19], in contrast to the $(T = 2^{\Omega(n^{0.5})}, \epsilon = 2^{-\Omega(n^{0.5})}, q = 2^{\Omega(n^{0.5})})$-hardness we established for LPN when $c = 0.5$, where $\omega(\cdot)$ omits a (arbitrarily small) super-constant (see more discussions in Section 3.6). One might try to set $c = 0.5 - \delta$ to obtain $(T = 2^{\Omega(n^{0.5+\delta})}, \epsilon = 2^{-\Omega(n^{0.5-\delta})}, q = 2^{\Omega(n^{0.5-\delta})})$-hard LPN, and then rebalance T and $1/\epsilon$ to be of the same order (as in a typical hardness assumption). However, we don't know if such a time/success-rate tradeoff for LPN can be obtained in general (without sacrificing q). We leave it as an open problem whether such a gap can be closed with tighter proofs or there's in a strict hierarchy in place. On the other hand, the attempt to use our reduction for cryptanalysis, i.e., to turn the BKW algorithm (for LPN) into a worst-case solver for constant-noise NCP (i.e., $\frac{w}{m} = O(1)$), is not successful again due to some small gap. We refer to Section 3.6 for further details.

2 Preliminaries

2.1 Notations, Definitions and Inequalities

Column vectors are represented by bold lower-case letters (e.g., $\mathbf{s}$), row vectors are denoted as their transpose (e.g., $\mathbf{s}^\mathsf{T}$), and matrices are denoted by bold capital letters (e.g., $\mathbf{A}$). $|\mathbf{s}|$ refers to the Hamming weight of bit string $\mathbf{s}$. We use notations for sets and distributions as follows.

- R_d^m: the uniform distribution over set $\mathcal{R}_d^m \stackrel{\text{def}}{=} \{\mathbf{r} \in \mathbb{F}_2^m : |\mathbf{r}| = d\}$.
- $R_{d,m}$: the distribution that first samples $\mathbf{t}_1, \cdots, \mathbf{t}_m$ uniformly and independent from $\mathcal{R}_1^m$ and then produces as output their XOR sum $\bigoplus_{i=1}^m \mathbf{t}_i$.
- $\mathcal{B}_\mu^q \stackrel{\text{def}}{=} \underbrace{\mathcal{B}_\mu \times \cdots \times \mathcal{B}_\mu}_{q}$, where $\mathcal{B}_\mu$ is Bernoulli distribution with parameter μ.

We use e for the natural constant and $\log(\cdot)$ for binary logarithm. $\mathbf{x} \stackrel{\$}{\leftarrow} \mathcal{X}$ refers to drawing $\mathbf{x}$ from set $\mathcal{X}$ uniformly at random, and $\mathbf{x} \leftarrow X$ means drawing $\mathbf{x}$ according to distribution X. The collision probability of Y is defined as $\mathsf{Col}(Y) \stackrel{\text{def}}{=} \sum_y \Pr[Y = y]^2$. We denote by $\mathbf{H}_\infty(Y)$ the min-entropy of random variable Y. $\mathsf{poly}(\cdot)$ refers to a certain polynomial. The *statistical distance* between X and Y, denoted by $\mathsf{SD}(X, Y) \stackrel{\text{def}}{=} \frac{1}{2}\sum_x |\Pr[X = x] - \Pr[Y = x]|$. We say that X and Y are ε-close if $\mathsf{SD}(X, Y) \le \varepsilon$. We refer to Appendix A for proofs omitted in the main body and Appendix B for the inequalities, lemmas and theorems used in this paper.

2.2 Binary Linear Codes

Coding theory terminology typically refers to a linear code as $[n, k]$-code or $[n, k, d]$-code, but we choose to use $[m,n]$-code $(m > n)$ in order to be more compatible with the LPN problem and [BLVW19], where n is the size of message (secret to be decoded) and m is codeword length.

Definition 2.1 (binary linear code) *A binary (m,n)-code is a set of codewords $\mathcal{C} \subset \mathbb{F}_2^m$ with $|\mathcal{C}| = 2^n$ $(n < m)$, and a binary linear $[m,n]$-code $\mathcal{C}$ is a binary (m,n)-code that is the row span of some generator matrix $\mathbf{C} \in \mathbb{F}_2^{n\times m}$, i.e., $\mathcal{C} \stackrel{def}{=} \{\mathbf{s}^\mathsf{T}\mathbf{C} \in \mathbb{F}_2^m | \mathbf{s}^\mathsf{T} \in \mathbb{F}_2^n\}$.*

Definition 2.2 (dual code) *The dual code of a binary linear $[m,n]$-code $\mathcal{C}$, denoted by $\mathcal{C}^\perp$, is a binary $[m,m-n]$-code $\mathcal{C}^\perp \stackrel{def}{=} \{\mathbf{d} \in \mathbb{F}_2^m | \forall \mathbf{c} \in \mathcal{C} : \mathbf{d}^\mathsf{T}\mathbf{c} = \mathbf{0}\}$.*

[5] minicrypt is Impagliazzo's [Imp95] hypothetical world where one-way functions exist but public-key cryptography does not.

Definition 2.3 (minimum/maximum distance) *The minimum (resp., maximum) distance of a binary linear $[m,n]$-code $\mathcal{C}$ is d if for any distinct codewords $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ it holds that $|\mathbf{x}-\mathbf{y}| \geq d$ (resp., $|\mathbf{x}-\mathbf{y}| \leq d$). A linear $[m,n]$-code with minimum distance d is called a $[m,n,d]$-code.*

A β-balanced code is a $[m,n,\frac{1}{2}(1-\beta)m]$ code with maximal distance bounded by $\frac{1}{2}(1+\beta)m$. A binary linear code is k-independent if its generator matrix has k-wise independent columns. In other words, k-independent $[m,n]$-code and $[m,m-n,k+1]$-code are dual to each other. In the extreme case $k = n$, k-independent $[m,n]$ code becomes a maximum distance separable (MDS) code, but since binary MDS codes are trivial we use $k < n$ with further relaxed conditions.

Definition 2.4 (balanced code) *A binary linear $[m,n]$ code $\mathcal{C} \subseteq \mathbb{F}_2^m$ 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 2.5 (independent code) *For a binary linear $[m,n]$ code $\mathcal{C} \subseteq \mathbb{F}_2^m$,*

- *$\mathcal{C}$ is k-independent if every k columns of its generator matrix $\mathbf{C}$ are linearly independent, i.e., $\forall i \in [1,\ldots,k] : \Pr[\mathbf{Cr} = \mathbf{0} : \mathbf{r} \leftarrow R_i^m] = 0$.*
- *$\mathcal{C}$ is (k,ζ)-independent if $\forall i \in \{\frac{k}{2}, \frac{k}{2}+1, \ldots, k\} : \Pr[\mathbf{Cr} = \mathbf{0} : \mathbf{r} \leftarrow R_i^m] \leq 2^{-n}(1+\zeta)$.*

The latter relaxes the independence condition by only enforcing it for $i \in [n/2, n]$ (instead of for all $i \in (0, n]$) and even for $i \in [n/2, n]$ a slackness of ζ is allowed, in the spirit of almost universal hash functions [Sti02]. Note that there is nothing special with the cut-off point $n/2$, which can be replaced with δn for any constant $0 < \delta < 1$ without affecting our results asymptotically.

The following lemmas state that balanced code and independent code exist in abundance and they both account for an overwhelming portion of linear code (for the parameter choices of this paper). Otherwise said, one just samples a random matrix to get the generator matrix for both codes at the same time except with exponentially small error. The first lemma for balanced code (essentially the Gilbert-Varshamov bound) follows by a straightforward probabilistic argument and in contrast Lemma 2.2 has a less trivial proof, which applies the Chebyshev's inequality to pairwise independent variables (somewhat reminiscent of the Goldreich-Levin theorem [GL89]). We refer interested readers to Appendix A and Remark A.1 for its proof and discussions.

Lemma 2.1 (Existence of balanced code [BLVW19]) *A random binary linear $[n,m]$-code is β-balanced with probability at least $1-2^{n+1}e^{-\frac{\beta^2 m}{4}}$. In particular, for $\beta \geq 2\sqrt{n/m}$ the random binary linear code is β-balanced with probability $1-2^{-\Omega(n)}$.*

Lemma 2.2 (Existence of independent code) *A random binary linear $[m,n]$ code $\mathcal{C}$ is (k, ζ)-independent with probability at least $(1-\frac{k2^{n+\frac{\log m}{2}-\frac{k}{2}\log\frac{m}{k}}}{\zeta^2})$. In particular, for $k\log(m/k) \geq 16n$ and $\log m = o(n)$ the random binary linear code is $(k, 2^{-n})$-independent with probability at least $1-2^{-4n}$.*

2.3 The NCP and LPN problem

Throughout, n is the main security parameter, and other parameters, e.g., $\mu = \mu(n)$, $q = q(n)$, $m = m(n)$ and $T = T(n)$, can be seen as functions of n.

Definition 2.6 (Nearest Codeword Problem (NCP)) *The nearest codeword problem $\mathsf{NCP}_{n,m,w}$ for $n, m, w \in \mathbb{N}$ refers to that given the input of a matrix $\mathbf{C} \in \mathbb{F}_2^{n\times m}$ of a binary linear code $\mathcal{C}$ and a noisy codeword $\mathbf{t}^\mathsf{T} = \mathbf{s}^\mathsf{T}\mathbf{C} + \mathbf{x}^\mathsf{T}$ for some $\mathbf{s} \in \mathbb{F}_2^n$ and $\mathbf{x} \in \mathcal{R}_w^m$, and the challenge is to find out a solution $\mathbf{s}'$ such that $\mathbf{s}^\mathsf{T}\mathbf{C} + \mathbf{x}^\mathsf{T} = \mathbf{s}'^\mathsf{T}\mathbf{C} + \mathbf{x}'^\mathsf{T}$ for some $\mathbf{x}' \in \mathcal{R}_w^m$. In particular, we consider the NCP on the following codes:*

- ***(Balanced NCP)***. *The balanced nearest codeword problem, referred to as* $\mathsf{balNCP}_{n,m,w,\beta}$, *is the* $\mathsf{NCP}_{n,m,w}$ *on* β*-balanced linear* $[m,n]$*-code.*
- ***(Independent NCP)***. *The independent nearest codeword problem, denoted by* $\mathsf{indNCP}_{n,m,w,k,\zeta}$, *refers to the* $\mathsf{NCP}_{n,m,w}$ *on* (k,ζ)*-independent linear* $[m,n]$*-code.*

Similar to one-way function, an instance of the NCP is considered solved as long as a decoding algorithm comes up with any legitimate solution $\mathbf{x}'$, which does not necessarily equal the original $\mathbf{x}$. In general, linear codes have unique solutions except for a $2^{-m+n+2w\log m}$ fraction (see Lemma 2.3), which is super-exponentially small for our parameter setting $w\log m = O(n)$ and $m = \Omega(n^{1+\varepsilon})$. Moreover, $\mathsf{balNCP}_{n,m,w,\beta}$ has unique solution for $w < \frac{1}{4}(1-\beta)m$. Decisional and computational LPN are polynomially equivalent even for the same sample complexity [AIK07].

Definition 2.7 (Learning Parity with Noise (LPN)) *The (computational) LPN problem with secret length* n*, noise rate* $\mu \in (0,1/2)$ *and sample complexity* q*, denoted by* $\mathsf{LPN}_{n,\mu,q}$*, asks to find out* $\mathbf{x}$ *given* $(\mathbf{A},\ \mathbf{A}\cdot\mathbf{x}+\mathbf{e})$*; and the decisional LPN problem* $\mathsf{DLPN}_{n,\mu,q}$ *challenges to distinguish* $(\mathbf{A},\ \mathbf{A}\cdot\mathbf{x}+\mathbf{e})$ *and* $(\mathbf{A},\ \mathbf{U}_q)$*, where matrix* $\mathbf{A} \xleftarrow{\$} \mathbb{F}_2^{q\times n}$, $\mathbf{x} \xleftarrow{\$} \mathbb{F}_2^n$, $\mathbf{y} \xleftarrow{\$} \mathbb{F}_2^q$, *and* $\mathbf{e} \leftarrow \mathcal{B}_\mu^q$.

Computational hardness. We say that a computational/decisional problem is (T,ϵ)-hard, if every probabilistic algorithm running in time T solves it with probability/advantage at most ϵ. We say that NCP (resp., LPN) is (T,ϵ,q)-hard if the problem is (T,ϵ)-hard when the codeword length (resp., sample complexity) does not exceed q. When the success-rate term $\epsilon = 1/T$ we often omit ϵ. Recall that standard polynomial hardness requires that $T > \mathsf{poly}(n)$ and $\epsilon < 1/\mathsf{poly}(n)$ for every poly and all sufficiently large n's.

Lemma 2.3 (Unique decoding of binary NCP/LPN) *For* $w/m < 1/4$,

$$\Pr_{\mathbf{C}\xleftarrow{\$}\mathbb{F}_2^{m\times n}}\left[\exists \mathbf{s}_1 \neq \mathbf{s}_2 \in \mathbb{F}_2^n, \exists \mathbf{x}_1,\mathbf{x}_2 \in \mathbb{F}_2^m : |\mathbf{x}_1|,|\mathbf{x}_2| \le w \wedge \left(\mathbf{s}_1^\mathsf{T}\mathbf{C}+\mathbf{x}_1^\mathsf{T} = \mathbf{s}_2^\mathsf{T}\mathbf{C}+\mathbf{x}_2^\mathsf{T}\right)\right]$$

is upper bounded by $2^{-m+n+2w\log m}$.

3 Worst-case to Average-case Reductions for NCP/LPN

3.1 The worst-case to average-case reduction from [BLVW19]

Brakerski et al. [BLVW19] showed that the worst-case hardness of the extremely low-noise NCP problem on balanced code implies the (average-case) hardness of extremely high-noise LPN.

Theorem 3.1 ([BLVW19]) *Assume that* $\mathsf{balNCP}_{n,m,w,\beta}$ *is hard in the worst case for noise rate* $\frac{w}{m} = \frac{\log^2 n}{n}$, $m = 4n^2$, $\beta = 1/\sqrt{n}$ *then* $\mathsf{LPN}_{n,\mu,q}$ *is hard (in the average case) for* $\mu = 1/2 - \frac{1}{n^{O(1)}}$ *and any* $q = \mathsf{poly}(n)$.

As detailed in Algorithm 1, the idea is to convert an NCP instance $(\mathbf{C},\ \mathbf{t}^\mathsf{T})$ into LPN samples. By Theorem 3.2, the conversion produces q LPN samples of noise rate μ up to error $q\delta$, where

$$\mu = \frac{1}{2} - \frac{1}{2}(1-\frac{2w}{m})^d,\ \ q\delta = O(q)2^{\frac{n}{2}}\cdot(\frac{2w}{m}+\beta)^d\ .$$

Thus, the conclusion follows by setting $\frac{w}{m} = \frac{\log^2 n}{n}$, $\beta = 2\sqrt{n/m} = 1/\sqrt{n}$, $d = 2n/\log n$ such that $\mu = 1/2 - 1/n^{O(1)}$ and $q\delta = \mathsf{negl}(n)$.

Remark 3.1 (possibilities and limitations) *Other possible parameter choices are also discussed in [BLVW19], e.g., assume that* $\mathsf{balNCP}_{n,m,w,\beta}$ *is* $2^{\Omega(\sqrt{n})}$*-hard for* $\frac{w}{m} = \frac{1}{\sqrt{n}}$ *(while keeping* $\beta = \frac{1}{\sqrt{n}}$ *and* $d = 2n/\log n$*) then* $\mathsf{LPN}_{n,\mu,q}$ *is* $2^{\Omega(\sqrt{n})}$*-hard for noise* $\mu = 1/2 - 2^{-\sqrt{n}/\log n}$ *and* $q = 2^{\Omega(\sqrt{n})}$*. This result is non-trivial since the noise rate* μ *(although quite close to uniform already) isn't high enough for the conclusion to hold statistically. However, it does not seem to yield efficient (a.k.a. polynomial-time) cryptographic applications. In fact, the barriers are inherent in its smoothing lemma Theorem 3.2: it is necessary to have NCP's noise rate* $\frac{w}{m} = \frac{\omega(\log n)}{n}$ *for the hardness assumption to hold, and then regardless of the value of* β *it requires* $d = \Omega(n/\log n)$ *to make* $(\frac{2w}{m} + \beta)^d < 2^{-n/2}$*, which lower bounds the noise rate of LPN, i.e.,* $\mu = \frac{1}{2} - \frac{1}{2}(1 - \frac{2w}{m})^d = 1/2 - 2^{-\omega(1)}$*.*

Algorithm 1 Converting an NCP instance to LPN samples.

Input: $(\mathbf{C}, \mathbf{t}^\mathsf{T} = \mathbf{s}^\mathsf{T}\mathbf{C} + \mathbf{x}^\mathsf{T})$, where $\mathbf{C} \in \mathbb{F}_2^{n\times m}$, $\mathbf{s} \in \mathbb{F}_2^n$, $\mathbf{x} \in \mathcal{R}_w^m$

$\mathbf{u} \xleftarrow{\$} \mathbb{F}_2^n$

Sample $\mathbf{R} \stackrel{\text{def}}{=} [\mathbf{r}_1, \ldots, \mathbf{r}_q] \in \mathbb{F}_2^{m\times q}$, where every column $\mathbf{r}_i \leftarrow R_{d,m}$ $(1 \le i \le q)$

Output: $(\mathbf{CR}, \mathbf{t}^\mathsf{T}\mathbf{R} + \mathbf{u}^\mathsf{T}\mathbf{CR}) = (\mathbf{CR}, (\mathbf{s}^\mathsf{T} + \mathbf{u}^\mathsf{T})\mathbf{CR} + \mathbf{x}^\mathsf{T}\mathbf{R})$

Theorem 3.2 (W/A-case reduction via code smoothing [BLVW19]) *Assume that* $\mathsf{balNCP}_{n,m,w,\beta}$ *is* T*-hard in the worst case, then* $\mathsf{LPN}_{n,\mu,q}$ *is* $(T - O(nmq), \frac{1}{T} + q\delta)$*-hard (in the average case) for any* $w, d \le m$*, any* q *and*

$$\delta = \max_{\mathbf{x}\in\mathcal{R}_{m,w}} \mathsf{SD}\Big((\mathbf{Cr}, \mathbf{x}^\mathsf{T}\mathbf{r}) \ , \ (\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})\Big) \le 2^{\frac{n+1}{2}} \cdot (\frac{2w}{m} + \beta)^d \ , \tag{1}$$

$$\mu = \max_{\mathbf{x}\in\mathcal{R}_{m,w}} \Pr[\mathbf{x}^\mathsf{T}\mathbf{r} = 1] = \frac{1}{2} - \frac{1}{2}(1 - \frac{2w}{m})^d \ . \tag{2}$$

where $\mathbf{r} \leftarrow R_{d,m}$*,* $\mathbf{C} \in \mathbb{F}_2^{n\times m}$ *is a generator matrix of any* β*-balanced* $[m, n]$ *code and* $O(mnq)$ *accounts for the complexity of Algorithm 1.*

The authors of [BLVW19] proved the above smoothing lemma using harmonic analysis. We give an alternative proof via Vazirani's XOR lemma [Vaz86, Gol11]. We stress that the approach serves to simplify the presentation to readers by establishing the proof under a well-known theorem. In other words, the proof below is not essentially different from that in [BLVW19] after unrolling out the proof of the XOR lemma.

Lemma 3.1 (Vazirani's XOR lemma [Vaz86, Gol11]) *For any r.v.* $\mathbf{v} \in \mathbb{F}_2^n$*, we have*

$$\mathsf{SD}(\mathbf{v}, \mathbf{U}_n) \ \le \ \sqrt{\sum_{\mathbf{0}\neq\mathbf{a}\in\mathbb{F}_2^n} \mathsf{SD}(\mathbf{a}^\mathsf{T}\mathbf{v}, \mathbf{U}_1)^2} \ .$$

A SIMPLIFIED PROOF FOR THEOREM 3.2. We denote with $\mathbf{C}_\mathbf{x} \in \mathbb{F}_2^{n\times(m-w)}$ and $\mathbf{r}_\mathbf{x} \in \mathbb{F}_2^{m-w}$ be the submatrix and substring of $\mathbf{C}$ and $\mathbf{r}$ respectively by keeping columns and bits that correspond to the positions of 0's in $\mathbf{x}^\mathsf{T}$. Recall that $\mathbf{r} \leftarrow R_{d,m}$ refers to $\mathbf{r} := \oplus_{i=1}^d \mathbf{t}_i$ for random weight-1 strings $\mathbf{t}_1, \cdots, \mathbf{t}_d \in \mathbb{F}_2^m$. Similarly, let $\mathbf{t}_i^{\bar{\mathbf{x}}}$ denote $\mathbf{t}_i$'s w-bit substring corresponding to the positions of 1's in $\mathbf{x}^\mathsf{T}$. Further, let $\mathcal{E}_j$ denote the event that the Hamming weight sum $\sum_{i=1}^d |\mathbf{t}_i^{\bar{\mathbf{x}}}| = j$, and thus $\mathbf{r}_\mathbf{x}$ conditioned on $\mathcal{E}_j$, denoted by $\mathbf{r}_{\mathbf{x},j}$, follows distribution $R_{d-j,m-w}$.

$$
\begin{aligned}
&\mathsf{SD}\Big((\mathbf{C}\mathbf{r}, \mathbf{x}^\mathsf{T}\mathbf{r})\ ,\ (\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})\ \Big) \\
\le\ &\mathsf{SD}\Big((\mathbf{C}_\mathbf{x}\mathbf{r}_\mathbf{x}, \mathbf{t}_1^{\bar{\mathbf{x}}}, \ldots, \mathbf{t}_d^{\bar{\mathbf{x}}})\ ,\ (\mathbf{U}_n, \mathbf{t}_1^{\bar{\mathbf{x}}}, \ldots, \mathbf{t}_d^{\bar{\mathbf{x}}})\Big) \\
\le\ &\sum_{j=0}^{d} \Pr[\mathcal{E}_j] \cdot \sqrt{\sum_{\mathbf{0}\neq\mathbf{a}\in\mathbb{F}_2^n} \mathsf{SD}\Big(\ \mathbf{a}^\mathsf{T}\mathbf{C}_\mathbf{x}\mathbf{r}_{\mathbf{x},j},\ \mathbf{U}_1\ \Big)^2} \\
\le\ &\sum_{j=0}^{d} \Pr[\mathcal{E}_j] \cdot \sqrt{2^n \cdot \left((\frac{w+\beta m}{m-w})^{d-j}\right)^2} \\
=\ &2^{\frac{n}{2}} \sum_{j=0}^{d} \underbrace{\binom{d}{j}(\frac{w}{m})^j(1-\frac{w}{m})^{d-j}}_{\Pr[\mathcal{E}_j]} \cdot (\frac{w+\beta m}{m-w})^{d-j} \ =\ 2^{\frac{n}{2}}(\beta+\frac{2w}{m})^d\ ,
\end{aligned}
$$

where the first inequality is due to that $\mathbf{x}^\mathsf{T}\mathbf{r}$ is implied by $\mathbf{t}_1^{\bar{\mathbf{x}}}, \ldots, \mathbf{t}_d^{\bar{\mathbf{x}}}$ (i.e., $\mathbf{x}^\mathsf{T}\mathbf{r}$ is the parity bit of $\oplus_{i=1}^d \mathbf{t}_i^{\bar{\mathbf{x}}}$), the second inequality follows from Vazirani's XOR lemma and the third inequality is due to Piling-up lemma, in particular, $\mathbf{a}^\mathsf{T}\mathbf{C} \in \mathbb{F}_2^m$ is a balanced string with $\frac{(1\pm\beta)m}{2}$ 1's and thus its substring $\mathbf{a}^\mathsf{T}\mathbf{C}_\mathbf{x} \in \mathbb{F}_2^{m-w}$ has $(\frac{m-w}{2}) \pm (\frac{w+\beta m}{2})$ 1's and each bit 1 of $\mathbf{r}_{\mathbf{x},j}$ hits the 1's in $\mathbf{a}^\mathsf{T}\mathbf{C}_\mathbf{x}$ with probability $\frac{1}{2} \pm \frac{w+\beta m}{2(m-w)}$. Finally, we compute noise rate μ by the following:

$$
1 - 2\mu = \Pr[\mathbf{x}^\mathsf{T}\mathbf{r} = 0] - \Pr[\mathbf{x}^\mathsf{T}\mathbf{r} = 1] = \sum_{i=0}^{d} \binom{d}{i} (\frac{-w}{m})^i (1-\frac{w}{m})^{d-i} = (1-\frac{2w}{m})^d\ .
$$

3.2 On the Non-triviality of Code Smoothing

As discussed in Remark 3.1, the worst-case to average-case reduction in [BLVW19] may only give rise to LPN on noise rate $\mu = 1/2 - 1/n^{\Omega(1)}$. The code smoothing lemma, see (1), can be seen as deterministic randomness extraction from Bernoulli-like distributions. We now investigate the possibilities of smoothing linear binary codes existentially using a probabilistic argument. Consider $\mathbf{C}$ to be uniform over $\in \mathbb{F}_2^{n\times m}$ instead of a fixed one, then $\mathbf{r}$ has average min-entropy roughly $d\log(m/d)$ even given the single bit leakage $\mathbf{x}^\mathsf{T}\mathbf{r}$, and thus by the leftover hash lemma $\mathsf{SD}\Big((\mathbf{C}, \mathbf{C}\mathbf{r}, \mathbf{x}^\mathsf{T}\mathbf{r}), (\mathbf{C}, \mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})\Big) \le 2^{-n}$ for $d\log(m/d) = 3n$. It follows by Markov inequality that there exists at least a $(1-2^{-n/2})$-fraction of "good" $\mathbf{C}$ satisfying $\mathsf{SD}\Big((\mathbf{C}\mathbf{r}, \mathbf{x}^\mathsf{T}\mathbf{r}), (\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})\Big) \le 2^{-n/2}$. This seemingly opens new possibilities especially in the sub-exponential hardness regime. For example, assume the NCP problem on a "good" code is $2^{\Omega(\sqrt{n})}$-hard (in the worst case) for noise rate $\frac{w}{m} = \frac{1}{\sqrt{n}}$, $d = O(\sqrt{n})$, and $m = 2^{O(\sqrt{n})}$ then $\mathsf{LPN}_{n,\mu,q}$ is $2^{\Omega(\sqrt{n})}$-hard against constant noise (see (2)). However, so far we only consider a specific value of $\mathbf{x}$ for which there is a $2^{-n/2}$ fraction of $\mathbf{C}$ that fails the randomness extraction, and by summing over all the possible $\mathbf{x} \in \mathcal{R}_w^m$ the fraction of "bad" $\mathbf{C}$ amounts up to $\binom{m}{w}2^{-n/2}$, which is useless since $\binom{m}{w}$ is super-exponential for $w = O(2^{\sqrt{n}}/\sqrt{n})$. To summarize, the existence of more meaningful smoothing lemma for binary linear code crucially relies on tighter proof techniques and better exploitation of the actual linear code in consideration to beat the union bound (so that "bad" $\mathbf{C}$ for different values of $\mathbf{x}$ mostly coincide and they jointly constitute only a negligible fraction).

3.3 Worst-case Sub-exponential Hardness for LPN

We obtain the following worst-case to average-case reductions for LPN, where $d\log(m/d) = \Omega(n)$ is a necessary entropy condition (which is implicit in (1) of Theorem 3.2) and the values of β,

k, and ζ are chosen to ensure the existence of respective codes (Lemma 2.1 and Lemma 2.2).

Theorem 3.3 (W/A-reduction for balanced codes) *Assume that* $\mathsf{balNCP}_{n,m,w,\beta}$ *is* (T,ϵ)*-hard in the worst case for* $\beta = 2\sqrt{n/m}$*, then* $\mathsf{LPN}_{n,\mu,q}$ *is* $(T - O(nmq),\ \epsilon + \frac{q\cdot 2^{-\Omega(d)}}{1-2\mu})$*-hard for* $\mu = \frac{1}{2} - \frac{1}{2}(1 - \frac{2d}{m})^w$*, any* q *and* d *satisfying* $d\log(m/d) \geq 4n$*.*

Theorem 3.4 (W/A-reduction for independent codes) *Assume that* $\mathsf{indNCP}_{n,m,w,k,\zeta}$ *is* (T,ϵ)*-hard in the worst case for* $k = \frac{16d}{7}$ *and* $\zeta = 2^{-n}$*, then* $\mathsf{LPN}_{n,\mu,q}$ *is* $(T - O(nmq),\ \epsilon + \frac{q\cdot 2^{-\Omega(d)}}{1-2\mu})$*-hard for* $\mu = \frac{1}{2} - \frac{1}{2}(1 - \frac{2d}{m})^w$*, any* q *and* d *satisfying* $d\log(m/d) \geq 7n$*.*

Proof sketch. The proofs of Theorem 3.3 and Theorem 3.4 use the NCP instance to LPN sample conversion as described in Algorithm 1 except for sampling every $\mathbf{r}_i \leftarrow \mathcal{B}_{\frac{d}{m}}$ instead of $\mathbf{r}_i \leftarrow R_{d,m}$. The conclusions follow from the respective smoothing lemmas (Lemma 3.4 and Lemma 3.6). While replacing $R_{d,m}$ with $\mathcal{B}_{\frac{d}{m}}$ seems equivalent in terms of the resulting noise rate μ (almost same as (2) except that d and w are swapped), the fact that bits of $\mathbf{r}_i$ are all independent is crucial in obtaining more generic security bounds for δ that allow for a wider range of parameter choices. □

We can see that Theorem 3.1 falls into a corollary of Theorem 3.3 by setting $\frac{w}{m} = \frac{\log^2 n}{n}$, $d = O(n/\log n)$ and $m = \mathsf{poly}(n)$ to get $\mu = 1/2 - 1/n^{O(1)}$. Moreover, it admits new possibilities as stated in the theorem below. In particular, our assumption is that there exists constant ε such that the NCP problem is is $2^{\varepsilon\frac{w}{m}n}$-hard at noise rate $\frac{w}{m}$ and codeword length $m = 2^{\frac{\varepsilon}{8}\frac{w}{m}n}$. Note that refuting this assumption means that we can do arbitrary polynomial speedup over the current best known algorithms in solving the respective NCPs, which is a win-win situation.

Theorem 3.5 (Sub-exponential hardness for LPN) *Assume that either (1)* $\mathsf{balNCP}_{n,m,w,\beta}$ *with* $\beta = 2\sqrt{n/m}$*, or (2)* $\mathsf{indNCP}_{n,m,w,k,\zeta}$ *with* $k = \frac{16d}{7}$ *and* $\zeta = 2^{-n}$*, is* $2^{\Omega(n^{1-c})}$*-hard at noise rate* $\frac{w}{m} = n^{-c}$ *and codeword size* $m = 2^{\Omega(n^{1-c})}$*, then depending on the value of* c *we have*

Case $0 < c < 1/2$**:** $\mathsf{LPN}_{n,\mu,q}$ *is* $(2^{\Omega(n^{1-c})}, 2^{-\Omega(n^c)})$*-hard for* $0 < \mu = O(1) < 1/2$ *and* $q = 2^{\Omega(n^c)}$*;*

Case $1/2 \leq c < 1$**:** $\mathsf{LPN}_{n,\mu,q}$ *is* $2^{\Omega(n^{1-c})}$*-hard for* $0 < \mu = O(1) < 1/2$ *and* $q = 2^{\Omega(n^{1-c})}$*.*

Proof sketch. This is a corollary of Theorem 3.3 and Theorem 3.4 (from the respective assumptions) for $\frac{w}{m} = n^{-c}$, $\mu = O(1)$ (s.t. $\frac{w}{m}d = O(1)$), $d\log(m/d) = O(n)$ and $T = \Omega(n^{1-c})$. Note that $1/T + q2^{-\Omega(d)} = 2^{-\Omega(n^{1-c})} + 2^{-\Omega(n^c)}$, which is why the value of c is considered. □

3.4 Smoothing balanced codes

Our smoothing lemma benefits from Lemma 3.2 which tightly relates the bound on the conditional case $\mathsf{SD}\Big((\mathbf{Cr}, \mathbf{x}^\mathsf{T}\mathbf{r}),\ (\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})\Big)$ to that of the unconditional case $\mathsf{SD}(\mathbf{Cr}, \mathbf{U}_n)$, regardless of which $\mathbf{x}$ is used. Note that this would not have been possible if $\mathbf{r}$ were not sampled from the Bernoulli distribution. The proof of Lemma 3.2 further refers to Lemma 3.3.

Lemma 3.2 *For any matrix* $\mathbf{C} \in \mathbb{F}_2^{n\times m}$*, any* $\mathbf{x} \in \mathcal{R}_w^m$ *and any* $0 \leq a \leq 1/2$ *we have*

$$\mathsf{SD}(\mathbf{C}_\mathbf{x}\mathbf{r}_\mathbf{x}, \mathbf{U}_n) \ \leq\ \frac{\mathsf{SD}(\mathbf{Cr}, \mathbf{U}_n)}{(1-2a)^w}$$

where $\mathbf{r} \leftarrow \mathcal{B}_a^m$*,* $\mathbf{C}_\mathbf{x} \in \mathbb{F}_2^{n\times(m-w)}$ *(resp.,* $\mathbf{r}_\mathbf{x} \in \mathbb{F}_2^{m-w}$*) denotes the submatrix of* $\mathbf{C}$ *(resp., subvector of* $\mathbf{r}$*) by keeping only columns (resp., bits) corresponding to the positions of bit-0 in* $\mathbf{x}$ *respectively.*

Proof. We have $\mathbf{Cr} = \mathbf{C_x r_x} + \bigoplus_{i=1}^{w} e_i \mathbf{c}_i$ where $e_i \leftarrow \mathcal{B}_a$ and $\mathbf{c}_i$ is the i-th column vector of $\mathbf{C} \setminus \mathbf{C_x}$ (i.e., the columns of $\mathbf{C}$ that are excluded from $\mathbf{C_x}$). By applying Lemma 3.3 w times we get

$$\mathsf{SD}(\mathbf{Cr}, \mathbf{U}_n) \;\geq\; (1-2a)^w \cdot \mathsf{SD}(\mathbf{C_x r_x}, \mathbf{U}_n) \;\; .$$

□

Lemma 3.3 *Let* $\mathbf{p}$ *be a random variable over* $\mathbb{F}_2^n$*, and let* $\mathbf{c}$ *be any constant vector over* $\mathbb{F}_2^n$*. Then, we have*

$$\mathsf{SD}(\mathbf{p} \oplus (e_1\mathbf{c}), \mathbf{U}_n) \geq (1-2a) \cdot \mathsf{SD}(\mathbf{p}, \mathbf{U}_n) \;\; ,$$

where $e_1 \xleftarrow{\$} \mathcal{B}_a$ *(*$0 \leq a \leq 1/2$*) and* $e_1\mathbf{c}$ *denotes scalar vector multiplication between* e_1 *and* $\mathbf{c}$*.*

Proof. We use the shorthand $p_x \stackrel{\mathsf{def}}{=} \Pr[\mathbf{p} = x] - 2^{-n}$ for any $x \in \mathbb{F}_2^n$. Observe that any non-zero $\mathbf{c}$ divides $\mathbb{F}_2^n$ into two disjoint equal-size subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{F}_2^n$ such that every $\mathbf{p} \in \mathcal{S}_1$ implies $(\mathbf{p} + \mathbf{c}) \in \mathcal{S}_2$ and vice versa. Therefore,

$$\begin{aligned}
\mathsf{SD}\Big(\mathbf{p} \oplus (e_1\mathbf{c}), \mathbf{U}_n\Big) &= \frac{1}{2} \sum_{x \in \mathbb{F}_2^n} \Big| p_x(1-a) + p_{x \oplus \mathbf{c}} a \Big| \\
&\geq \frac{1}{2} \sum_{x \in \mathbb{F}_2^n} \Big(|p_x|(1-a) - |p_{x\oplus\mathbf{c}}| a \Big) \\
&= \frac{1}{2} \sum_{x \in \mathbb{F}_2^n} \Big(|p_x|(1-2a) \Big) = (1-2a) \cdot \mathsf{SD}(\mathbf{x}, \mathbf{U}_n) \;\; .
\end{aligned}$$

□

Lemma 3.4 (Smoothing lemma for balanced codes) *Let* $\beta \leq 2\sqrt{n/m}$*,* $d\log(m/d) \geq 4n$*, and let* $\mathbf{C} \in \mathbb{F}_2^{n \times m}$ *be any generator matrix for a* β*-balanced* $[m, n]$*-linear code, then for every* $\mathbf{x} \in \mathcal{R}_w^m$ *and* $\mathbf{r} \leftarrow \mathcal{B}_{\frac{d}{m}}^m$ *it holds that* $\mu = \Pr[\mathbf{x}^\mathsf{T}\mathbf{r} = 1] = \frac{1}{2} - \frac{1}{2}(1 - \frac{2d}{m})^w$ *and*

$$\delta_{\mathbf{C},\mathbf{x}} = \mathsf{SD}\Big((\mathbf{Cr}, \mathbf{x}^\mathsf{T}\mathbf{r}) \;,\; (\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r}) \Big) \leq \frac{2^{-\Omega(d)}}{1-2\mu} \;\; .$$

Proof. The noise rate μ directly follows from the Piling-up lemma.

$$\begin{aligned}
& \mathsf{SD}(\mathbf{Cr}, \mathbf{U}_n \;) \\
\leq\;& \mathsf{SD}(\mathbf{Cr}', \mathbf{U}_n) + 2^{-\Omega(d)} \\
\leq\;& \sqrt{\sum_{\mathbf{0} \neq \mathbf{a} \in \mathbb{F}_2^n} \mathsf{SD}\Big(\underbrace{\mathbf{a}^\mathsf{T}\mathbf{C}}_{\mathbf{b}^\mathsf{T}}\mathbf{r}',\; \mathbf{U}_1 \Big)^2} + 2^{-\Omega(d)} \\
\leq\;& \sqrt{\sum_{\mathbf{0} \neq \mathbf{a} \in \mathbb{F}_2^n} \mathsf{SD}\Big((\mathbf{b}^\mathsf{T}, \mathbf{b}^\mathsf{T}\mathbf{r}'),\; (\mathbf{b}^\mathsf{T}, \mathbf{U}_1) \Big)^2} + 2^{-\Omega(d)} \\
\leq\;& 2^{\frac{n}{2}} \cdot 2^{\frac{\frac{(\log e)\beta^2}{2} m + \frac{\log m}{2} - d(1-\delta)\log(\frac{m}{d(1-\delta)})}{2}} + 2^{-\Omega(d)} \\
=\;& 2^{-\Omega(d)}
\end{aligned}$$

where the first inequality follows from a Chernoff bound that $\mathbf{r}$ is $2^{-\Omega(d)}$-close to some $\mathbf{r}'$ that is a convex combination of $R^m_{d(1-\delta)}$, $R^m_{d(1-\delta)+1}$, $\cdots$, $R^m_{d(1+\delta)}$ for any small constant $\delta > 0$, the

second is due to Vazirani's XOR lemma. By the definition of balanced code $\mathbf{b}^\mathsf{T} \stackrel{\mathsf{def}}{=} \mathbf{a}^\mathsf{T}\mathbf{C} \in \mathbb{F}_2^m$ satisfies $\frac{(1-\beta)m}{2} \le |\mathbf{b}^\mathsf{T}| \le \frac{(1+\beta)m}{2}$ and we assume WLOG $|\mathbf{b}^\mathsf{T}| = \frac{(1-\beta)m}{2}$ so that $\mathbf{b}^\mathsf{T}\mathbf{r}'$ is maximally biased. As for the third inequality, we observe that for every $\mathbf{b}^\mathsf{T} \in \mathcal{R}^m_{\frac{(1-\beta)m}{2}}$ distribution $\mathbf{b}^\mathsf{T}\mathbf{r}'$ will always be the same regardless of which $\mathbf{b}^\mathsf{T}$ is picked due to the symmetry of $\mathbf{r}'$, and therefore it is convenient to assume that $\mathbf{b}^\mathsf{T}$ is uniform over $\mathcal{R}^m_{\frac{(1-\beta)m}{2}}$ instead of a fixed string, which enables to apply strong two-source extraction and Fact 2 to obtain the fourth inequality. Finally, we set $\beta = 2\sqrt{n/m}$, $d\log(m/d) = 4n$ and sufficiently small δ to complete the proof. Following the proof of Theorem 3.2, let $\mathbf{C}_\mathbf{x} \in \mathbb{F}_2^{n\times(m-w)}$ and $\mathbf{C}_{\bar{\mathbf{x}}} \in \mathbb{F}_2^{n\times w}$ denote the submatrices of $\mathbf{C}$ by keeping columns corresponding to the 0's and 1's in $\mathbf{x}^\mathsf{T}$ respectively, and let $\mathbf{r}_\mathbf{x} \in \mathbb{F}_2^{m-w}$ and $\mathbf{r}_{\bar{\mathbf{x}}} \in \mathbb{F}_2^w$ denote the subvectors of $\mathbf{r}$ that correspond to the positions of 0's and 1's in $\mathbf{x}^\mathsf{T}$ respectively. This allows to complete the proof by

$$\begin{aligned}
&\mathsf{SD}\Big((\mathbf{Cr}, \mathbf{x}^\mathsf{T}\mathbf{r}) \ , \ (\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})\Big) \le \mathsf{SD}\Big((\mathbf{C}_\mathbf{x}\mathbf{r}_\mathbf{x}, \mathbf{r}_{\bar{\mathbf{x}}}) \ , \ (\mathbf{U}_n, \mathbf{r}_{\bar{\mathbf{x}}})\Big) \\
= \ &\mathsf{SD}(\mathbf{C}_\mathbf{x}\mathbf{r}_\mathbf{x}, \mathbf{U}_n) \ \le \ \frac{\mathsf{SD}(\mathbf{Cr}, \mathbf{U}_n)}{(1-\frac{2d}{m})^w} = \frac{2^{-\Omega(d)}}{(1-\frac{2d}{m})^w} \ .
\end{aligned}$$

where the first inequality is due to that $(\mathbf{Cr}, \mathbf{x}^\mathsf{T}\mathbf{r})$ is implied by $(\mathbf{C}_\mathbf{x}\mathbf{r}_\mathbf{x}, \mathbf{r}_{\bar{\mathbf{x}}})$, i.e., $\mathbf{Cr} = \mathbf{C}_\mathbf{x}\mathbf{r}_\mathbf{x} + \mathbf{C}_{\bar{\mathbf{x}}}\mathbf{r}_{\bar{\mathbf{x}}}$ and $\mathbf{x}^\mathsf{T}\mathbf{r} = \langle 1^w, \mathbf{r}_{\bar{\mathbf{x}}}\rangle$, and so is $(\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})$ by $(\mathbf{U}_n, \mathbf{r}_{\bar{\mathbf{x}}})$, the equality is due to the independence of $\mathbf{r}_\mathbf{x}$ and $\mathbf{r}_{\bar{\mathbf{x}}}$, and the last inequality follows from Lemma 3.2. □

Lemma 3.5 (Two-source extraction via inner product) *For independent random variables $\mathbf{b}^\mathsf{T}$, $\mathbf{r} \in \mathbb{F}_2^m$ with $\mathbf{H}_\infty(\mathbf{b}^\mathsf{T}) = k_b$ and $\mathbf{H}_\infty(\mathbf{r}) = k_r$ we have*

$$\mathsf{SD}\Big(\ (\mathbf{b}^\mathsf{T}, \mathbf{b}^\mathsf{T}\mathbf{r}), \ (\mathbf{b}^\mathsf{T}, U_1) \ \Big) \le 2^{-(\frac{k_b+k_r-m}{2}+1)} \ .$$

3.5 Smoothing Independent Codes

Lemma 3.6 (Smoothing lemma for independent codes) *Let $d\log(m/d) \ge 7n$ and $\log m = o(n)$, and let $\mathbf{C} \in \mathbb{F}_2^{n\times m}$ be any generator matrix for a $(k = \frac{16d}{7}, 2^{-n})$-independent $[m, n]$-linear code $\mathcal{C} \in \mathbb{F}_2^m$, then for every $\mathbf{x} \in \mathcal{R}_{m,w}$ and $\mathbf{r} \leftarrow \mathcal{B}^m_{\frac{d}{m}}$ it holds that*

$$\delta_{\mathbf{C},\mathbf{x}} = \mathsf{SD}\Big((\mathbf{Cr}, \mathbf{x}^\mathsf{T}\mathbf{r}) \ , \ (\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})\Big) \le \frac{2^{-\Omega(d)}}{(1-\frac{2d}{m})^w} \ ,$$

$$\mu = \Pr[\mathbf{x}^\mathsf{T}\mathbf{r} = 1] = \frac{1}{2} - \frac{1}{2}(1-\frac{2d}{m})^w \ .$$

Proof. For any constant $0 < \delta < 1$, $\mathbf{r}$ is $2^{-\Omega(d)}$-close to some convex combination of $R^m_{d(1-\delta)}$, $R^m_{d(1-\delta)+1}$, $\cdots$, $R^m_{d(1+\delta)}$, which is denoted by $\mathbf{r}'$. By Lemma 3.7,

$$\mathsf{SD}(\mathbf{Cr} \ , \ \mathbf{U}_n) \le 2^{-\Omega(d)} + \sqrt{2^n \cdot \mathsf{Col}(\mathbf{Cr}') - 1} \ .$$

We assume WLOG $\mathbf{r}' \leftarrow R^m_{d(1-\delta)}$ and consider i.i.d. $\mathbf{r}_1, \mathbf{r}_2 \leftarrow R^m_{d(1-\delta)}$ such that

$$\mathsf{Col}(\mathbf{Cr}') = \Pr[\mathbf{Cr}_1 = \mathbf{Cr}_2] = \Pr[\mathbf{C}\ddot{\mathbf{r}}]$$

where for constant $0 < \Delta < 1$ variable $\ddot{\mathbf{r}} \stackrel{\mathsf{def}}{=} \mathbf{r}_1 - \mathbf{r}_2$ follows a convex combination of $R^m_{2d(1-\delta)(1-\Delta)}$, $R^m_{2d(1-\delta)(1-\Delta)+1}$, …, $R^m_{2d(1-\delta)}$ whose weights lie in between[6]

$$k/2 = 2d(1-\delta)(1-\Delta) \le \mathsf{weight} \le 2d(1+\delta) = k$$

[6]For up limit on $|\ddot{\mathbf{r}}|$ we need to consider the other extreme case $\mathbf{r}' \leftarrow R^m_{d(1+\delta)}$, where the corresponding $\ddot{\mathbf{r}}$ is a convex combination of $R^m_{2d(1+\delta)(1-\Delta)}$, $R^m_{2d(1+\delta)(1-\Delta)+1}$, …, $R^m_{2d(1+\delta)}$ up to small error.

for $\delta = 1/7$ and $\Delta = 1/3$ except with error

$$\sum_{i=d(1-\delta)\Delta}^{d(1-\delta)} \frac{\binom{d(1-\delta)}{i}\binom{m-d(1-\delta)}{d(1-\delta)-i}}{\binom{m}{d(1-\delta)}} \leq \frac{2^{d(1-\delta)(1-\Delta)\log\frac{m}{d(1-\delta)(1-\Delta)}}}{2^{d(1-\delta)\log\frac{m}{d(1-\delta)}}} \leq 2^{-d(1-\delta)\Delta\log\frac{m}{d(1-\delta)}} \quad .$$

The error is upper bounded by 2^{-2n} (for $\delta = 1/7$ and $\Delta = 1/3$). Thus,

$$\sqrt{2^n \cdot \mathsf{Col}(\mathbf{Cr}') - 1} \leq \sqrt{2^n \cdot (2^{-n}(1+2^{-n}) + 2^{-2n}) - 1} = 2^{-\Omega(n)} \quad .$$

and $\mathsf{SD}(\mathbf{Cr}\ ,\ \mathbf{U}_n) \leq 2^{-\Omega(d)}$. The rest follow the same steps as in the proof of Lemma 3.4. □

Lemma 3.7 is abstracted out from the leftover hash lemma (see Appendix A for its proof).

Lemma 3.7 (Generalized Hash Lemma) *For any function $h : \mathbb{F}_2^m \to \mathbb{F}_2^n$ and any random variable $\mathbf{r}$ over $\mathbb{F}_2^m$ we have*

$$\mathsf{SD}\Big(h(\mathbf{r}),\ \mathbf{U}_n\Big) \leq \frac{1}{2}\sqrt{2^n \cdot \mathsf{Col}(h(\mathbf{r})) - 1} \quad .$$

3.6 Discussions

We conclude that the constant-noise LPN problem is $(T = 2^{\Omega(n^{1-c})}, \epsilon = 2^{-\Omega(n^{\min(c,1-c)})}, q = 2^{\Omega(n^{\min(c,1-c)})})$-hard assuming that the NCP (on the balanced/independent code) at the low-noise rate $\tau = n^{-c}$ is $(T' = 2^{\Omega(\tau n)}, \epsilon' = 2^{-\Omega(\tau n)}, m = 2^{\Omega(\tau n)})$-hard in the worst case. Unfortunately, we need $(T = 2^{\omega(n^{0.5})}, \epsilon = 2^{-\omega(n^{0.5})}, q = 2^{\Omega(n^{0.5})})$-hardness for constructing collision resistant hash functions and public-key encryptions [YZ16, YZW+19], where the super-constant omitted by $\omega(\cdot)$ (representing the gap between what we prove for $c = 0.5$ and what is needed for PKE/CRH) can be arbitrarily small.[7] We explain in details below.

Theorem 3.6 ([YZW+19]) *Let n be the security parameter, and let $\mu = \mu(n)$, $k = k(n)$, $q = q(n)$, $t = t(n)$ and $T = T(n)$ such that $t^2 \leq q \leq T = 2^{\frac{8\mu t}{\ln 2(1-2\mu)}}$. For each $\mathbf{A} \in \mathbb{F}_2^{n\times q}$, define compressing function $h_{\mathbf{A}} : \mathbb{F}_2^{\log(\frac{q}{t})t} \to \mathbb{F}_2^n$ with $\log(\frac{q}{t})t > n$ by $h_{\mathbf{A}}(\mathbf{x}) = \mathbf{A} \cdot \mathsf{Expand}(\mathbf{x})$, where Expand expands any string of length $\log(\frac{q}{t})t$ into one of length q with Hamming weight no greater than t, and $h_{\mathbf{A}}$ is computable in time $O(q \log q)$ (see [YZW+19, Construction 3.1] for concrete instantiation of $h_{\mathbf{A}}$). Assume that the $\mathsf{DLPN}_{n,\mu,q}$ is T-hard, then for every probabilistic adversary $\mathcal{A}$ of running time $T' = 2^{\frac{4\mu t}{\ln 2(1-2\mu)}-1}$*

$$\Pr_{\mathbf{A}\xleftarrow{\$}\mathbb{F}_2^{n\times q}}\left[\ (\mathbf{y},\mathbf{y}') \leftarrow \mathcal{A}(\mathbf{A}) : \mathbf{y} \neq \mathbf{y}' \wedge h_{\mathbf{A}}(\mathbf{y}) = h_{\mathbf{A}}(\mathbf{y}')\ \right] \leq \frac{1}{T'} \quad .$$

Note that the above theorem does not state "$h_{\mathbf{A}}$ is a T'-hard collision resistant hash (CRH)" as it is computable in time $O(q \log q)$ while $q = 2^{\Omega(\sqrt{n})}$ is not polynomial in the security parameter n. In particular, length requirement $q \leq T$ (any adversary making q queries runs in time at least q) implies, by taking a logarithm, $\log(q) = O(t)$ (recall that μ is constant). Since the compressing condition requires $\log(\frac{q}{t})t > n$ we need to set q and t to be at least $2^{\Omega(\sqrt{n})}$ and $\Omega(\sqrt{n})$ respectively. The authors of [YZW+19] offers a remedy to solve this problem. Switch to a new security parameter $\lambda = q$, and let $t = \log\lambda \cdot \omega(1)$ for any arbitrarily small $\omega(1)$. This ensures that $h_{\mathbf{A}}$ is computable in time $\mathsf{poly}(\lambda)$ while remaining $\lambda^{\omega(1)}$-collision resistant. Therefore, we need $(T = 2^{\omega(n^{0.5})}, \epsilon = 2^{-\omega(n^{0.5})}, q = 2^{\Omega(n^{0.5})})$-hardness for constant-noise LPN to construct collision resistant hash functions, where $\omega(\cdot)$ omits an arbitrary super constant.

[7] The difference between decisional and computational LPN is omitted since 2^p-hard $\mathsf{LPN}_{n,\mu,q}$ implies $2^{\Omega(p)}$-hard $\mathsf{DLPN}_{n,\mu,q}$ for any $p = \omega(\log n)$, $\mu = O(1)$ and $q \geq \mathsf{poly}(n)$ due to the sample-preserving reduction [AIK07].

Neither can we construct public-key encryptions from ($T = 2^{\Omega(n^{0.5})}$, $\epsilon = 2^{-\Omega(n^{0.5})}$, $q = 2^{\Omega(n^{0.5})}$)-hard LPN due to the same $\omega(1)$ gap factor (see Theorem 3.7). The reason is essentially similar to the case of CRH. In fact, in some extent CRH and PKE are dual to each when being constructed from LPN. The authors of [YZ16] already minimized the hardness needed for LPN to construct PKE, and also used the parameter switching technique. We restate the main results of [YZ16] below.

Theorem 3.7 ([YZ16]) *Assume that* $\mathsf{DLPN}_{n,\mu,q}$ *is* ($T = 2^{\omega(n^{0.5})}$, $\epsilon = 2^{-\omega(n^{0.5})}$, $q = 2^{n^{0.5}}$)*-hard for any constant* $0 < \mu \leq 1/10$*, there exist IND-CCA secure public-key encryption schemes.*

We also mention that our result fails to transform the BKW algorithm (for LPN) into a worst-case solver for constant-noise NCP (i.e., $\frac{w}{m} = O(1)$) again due to some small gap. In particular, we recall the variant of BKW algorithm in Theorem 3.8 below, and we informally state our reduction results (Theorem 3.3 and Theorem 3.4) in Lemma 3.8. In order for Lemma 3.8 to compose with Theorem 3.8, we need $q = n^{1+\varepsilon}$ and $d = O(\log n)$ to make $\frac{q \cdot 2^{-\Omega(d)}}{1-2\mu} < 1$ and thus $\mu = \frac{1}{2} - e^{-O(\frac{w}{m}d)} = \frac{1}{2} - 2^{-O(\log n)}$, which does not meet the noise rate of Theorem 3.8, i.e., $\mu = 1/2 - 2^{-(\log n)^{\delta}}$ for any constant $0 < \delta < 1$.

Theorem 3.8 ([Lyu05]) *Let* $q = n^{1+\varepsilon}$ *and* $\mu = 1/2 - 2^{-(\log n)^{\delta}}$ *for any constants* $\varepsilon > 0$ *and* $0 < \delta < 1$. $\mathsf{LPN}_{n,\mu,q}$ *can be solved in time* $2^{O(n/\log\log n)}$ *with overwhelming probability,*

Lemma 3.8 (Our reduction, informal) *Any algorithm that solves* $\mathsf{LPN}_{n,\mu,q}$ *in time* T *with success rate* p*, implies another worst-case algorithm (for the NCP considered in Theorem 3.3 and Theorem 3.4) of running time* $T + O(nmq)$ *with success rate* $p - \frac{q \cdot 2^{-\Omega(d)}}{1-2\mu}$*, where* $\mu = \frac{1}{2} - e^{-O(\frac{w}{m}d)}$.

4 Concluding Remarks

We show that constant-noise LPN is ($T = 2^{\Omega(n^{1-c})}$, $\epsilon = 2^{-\Omega(n^{\min(c,1-c)})}$, $q = 2^{\Omega(n^{\min(c,1-c)})}$)-hard assuming that the NCP (on the balanced/independent code) at the low-noise rate $\tau = n^{-c}$ is ($T' = 2^{\Omega(\tau n)}$, $\epsilon' = 2^{-\Omega(\tau n)}$,$m = 2^{\Omega(\tau n)}$)-hard in the worst case, which significantly improves upon [BLVW19] in terms of the end results obtained. However, the result is not strong enough to imply collision resistant hash functions or public-key encryptions due to the $\omega(1)$ gap term. We leave it as an open problem whether the gap can be closed or there is a separation in place.

References

[ABSS97] Sanjeev Arora, László Babai, Jacques Stern, and Z Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Advances in Cryptology - CRYPTO 2009*, pages 595–618, 2009.

[AIK07] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography with constant input locality. In *Advances in Cryptology - CRYPTO 2007*, pages 92–110, 2007. Full version available at `http://www.eng.tau.ac.il/~bennyap/pubs/input-locality-full-revised-1.pdf`.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th Annual Symposium on Foundations of Computer Science*, pages 298–307, Cambridge, Massachusetts, October 2003. IEEE.

[APY09] Noga Alon, Rina Panigrahy, and Sergey Yekhanin. Deterministic approximation algorithms for the nearest codeword problem. In *Algebraic Methods in Computational Complexity*, 2009.

[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*, volume 773 of *LNCS*, pages 278–291. Springer-Verlag, 22–26 August 1993.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In *Advances in Cryptology - EUROCRYPT 2012*, pages 520–536, 2012.

[BK02] Piotr Berman and Marek Karpinski. Approximating minimum unsatisfiability of linear equations. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 514–516. Society for Industrial and Applied Mathematics, 2002.

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

[BLP11] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: Ball-collision decoding. In *Advances in Cryptology - CRYPTO 2011*, pages 743–760, 2011.

[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, *45th ACM STOC*, pages 575–584, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.

[BLVW19] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. Worst-case hardness for LPN and cryptographic hashing via code smoothing. In *Advances in Cryptology - EUROCRYPT 2019*, volume 11478 of *Lecture Notes in Computer Science*, pages 619–635. Springer, 2019.

[CC98] Anne Canteaut and Florent Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to mceliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998.

[CKT16] David Cash, Eike Kiltz, and Stefano Tessaro. Two-round man-in-the-middle security from LPN. In *Proceedings of the 13th Theory of Cryptography (TCC 2016-A)*, pages 225–248, 2016.

[DDN14] Bernardo David, Rafael Dowsley, and Anderson C. A. Nascimento. Universally composable oblivious transfer based on a variant of LPN. In *Proceedings of the 13th International Conference on Cryptology and Network Security (CANS 2014)*, pages 143–158, 2014.

[DKPW12] Yevgeniy Dodis, Eike Kiltz, Krzysztof Pietrzak, and Daniel Wichs. Message authentication, revisited. In *Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2012)*, pages 355–374, 2012.

[DMS03] Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory*, 49(1):22–37, 2003.

[Döt15] Nico Döttling. Low noise LPN: KDM secure public key encryption and sample amplification. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 604–626, Gaithersburg, MD, USA, March 30 – April 1, 2015. Springer, Heidelberg, Germany.

[FGKP06] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *47th Symposium on Foundations of Computer Science*, pages 563–574, Berkeley, CA, USA, October 21–24 2006. IEEE.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178, Bethesda, MD, USA, May 31 – June 2, 2009. ACM Press.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In D. S. Johnson, editor, *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 25–32, Seattle, Washington, 15–17 May 1989.

[Gol11] Oded Goldreich. Three XOR-lemmas - an exposition. In *Studies in Complexity and Cryptography*, pages 248–272. 2011.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.

[HB01] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2001)*, pages 52–66, 2001.

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Structure in Complexity Theory Conference*, pages 134–147, 1995.

[JKPT12] Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In *Advances in Cryptology – ASIACRYPT 2012*, pages 663–680, 2012.

[JW05] Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In Victor Shoup, editor, *Advances in Cryptology—CRYPTO 2005*, volume 3621 of *LNCS*, pages 293–308. Springer-Verlag, 14–18 August 2005.

[KF15] Paul Kirchner and Pierre-Alain Fouque. An improved BKW algorithm for LWE with applications to cryptography and lattices. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 43–62, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[KPC+11] Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. Efficient authentication from hard learning problems. In *Proceedings of the 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2011)*, pages 7–26, 2011.

[KS06] Jonathan Katz and Ji Sun Shin. Parallel and concurrent security of the hb and hb$^+$ protocols. In Serge Vaudenay, editor, *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 73–87. Springer-Verlag, 2006.

[LM13] Vadim Lyubashevsky and Daniel Masny. Man-in-the-middle secure authentication schemes from lpn and weak prfs. In *Advances in Cryptology - CRYPTO 2013*, pages 308–325, 2013.

[Lyu05] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *Proceedings of the 9th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM 2005)*, pages 378–389, 2005.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In *Proceedings of the 17th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2011)*, pages 107–124, 2011.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 333–342, Bethesda, MD, USA, May 31 – June 2, 2009. ACM Press.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press.

[Ste88] Jacques Stern. A method for finding codewords of small weight. In *Coding Theory and Applications, 3rd International Colloquium*, pages 106–113, 1988.

[Sti02] D. R. Stinson. Universal hash families and the leftover hash lemma, and applications to cryptography and computing. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 42:3–31, 2002. Available at http://www.cacr.math.uwaterloo.ca/~dstinson/publist.html.

[Vaz86] Umesh Virkumar Vazirani. *Randomness, Adversaries and Computation (Random Polynomial Time)*. PhD thesis, 1986. AAI8718194.

[YZ16] Yu Yu and Jiang Zhang. Cryptography with auxiliary input and trapdoor from constant-noise LPN. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 214–243, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.

[YZW+19] Yu Yu, Jiang Zhang, Jian Weng, Chun Guo, and Xiangxue Li. Collision resistant hashing from sub-exponential learning parity with noise. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part II*, volume 11922 of *LNCS*, pages 3–24, Kobe, Japan, December 8–12, 2019. Springer, Heidelberg, Germany.

A Proofs Omitted

Proof of Lemma 2.2. For $\mathbf{C} \xleftarrow{\$} \mathbb{F}_2^{n\times m}$ and every $\mathbf{r} \in \mathbb{F}_2^m$ define

$$z_{\mathbf{C},\mathbf{r}} \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } \mathbf{C}\cdot\mathbf{r} = \mathbf{0} \\ 0, & \text{otherwise } \mathbf{C}\cdot\mathbf{r} \neq \mathbf{0} \end{cases}$$

For every $\mathbf{r} \neq \mathbf{0}$, the expectation $\mathbb{E}_{\mathbf{C}\xleftarrow{\$}\mathbb{F}_2^{n\times m}}[z_{\mathbf{C},\mathbf{r}}] = 2^{-n}$, and for every two distinct $\mathbf{r}_1 \neq \mathbf{r}_2$ variables $z_{\mathbf{C},\mathbf{r}_1}$ and $z_{\mathbf{C},\mathbf{r}_2}$ are pair-wise independent. For any $k/2 \le i \le k$,

$$\begin{aligned} & \Pr_{\mathbf{C}\xleftarrow{\$}\mathbb{F}_2^{n\times m}}\Big[\ \sum_{\mathbf{r}\in\mathcal{R}_i^m} z_{\mathbf{C},\mathbf{r}} \ge N\cdot 2^{-n}(1+\zeta)\ \Big] \\ \le\ & \Pr_{\mathbf{C}\xleftarrow{\$}\mathbb{F}_2^{n\times m}}\Big[\ \Big|\sum_{\mathbf{r}\in\mathcal{R}_i^m} z_{\mathbf{C},\mathbf{r}} - N\cdot 2^{-n}\Big| \ge N2^{-n}\zeta\ \Big] \\ \le\ & \frac{Var[\ \sum_{\mathbf{r}\in\mathcal{R}_i^m} z_{\mathbf{C},\mathbf{r}}\ ]}{(N2^{-n}\zeta)^2} = \frac{\sum_{\mathbf{r}\in\mathcal{R}_i^m} Var[\ z_{\mathbf{C},\mathbf{r}}\ ]}{(N2^{-n}\zeta)^2} \\ \le\ & \frac{Var[\ z_{\mathbf{C},\mathbf{r}}\ ]}{N2^{-2n}\zeta^2} \le \frac{2^{-n}}{N2^{-2n}\zeta^2} \le \frac{2^{n+\frac{\log m}{2}-\frac{k}{2}\log(m/k)}}{\zeta^2}\ , \end{aligned}$$

where $N \stackrel{\text{def}}{=} |\mathcal{R}_i^m| \ge \binom{m}{k/2}$, the second inequality is by Chebyshev, and the equality is due to $Var[X_i + X_j] = Var[X_i] + Var[X_j] - 2cov(X_i, X_j)$ whose covariance $cov(X_i, X_j) = \mathbb{E}[X_i \cdot X_j] - \mathbb{E}[X_i]\cdot\mathbb{E}[X_j] = 0$ for pairwise independent (X_i, X_j), and the third inequality is due to that for 0-1 valued random variable X_i

$$Var[X_i] = \mathbb{E}[(X_i)^2] - \mathbb{E}[X_i]^2 = \mathbb{E}[X_i] - \mathbb{E}[X_i]^2\ .$$

We complete the proof by a union bound on all possible values of i. □

Remark A.1 (Why not $i \in (0, k/2)$) *Note that the above considers only $i \ge k/2$. As we can see from the above proof, this is because $\log N = \log|\mathcal{R}_i^m| = \log\binom{m}{i}$ needs to be $\Omega(n)$ to make the bound meaningful. For small values of i, it is not possible since m is only sub-exponential.*

Proof of Lemma 2.3. Let $\mathbf{s} \stackrel{\text{def}}{=} \mathbf{s}_1 - \mathbf{s}_2$ and $\mathbf{x} \stackrel{\text{def}}{=} \mathbf{x}_1 - \mathbf{x}_2$. For any $\mathbf{s} \neq \mathbf{0}$ the random variable $\mathbf{s}^\mathsf{T}\mathbf{C}$ is uniform over $\mathbb{F}_2^m$ and thus it hits $\{\mathbf{x} \in \mathbb{F}_2^m\colon |\mathbf{x}| \le 2w\}$ with probability at most $\sum_{i=0}^{2w}\binom{m}{i}/2^m$. The conclusion follows by a union bound on all possible $\mathbf{s} \in \mathbb{F}_2^n$. □

Proof of Lemma 3.7. We denote $\mathcal{S} \stackrel{\text{def}}{=} \mathbb{F}_2^n$ and $p_s = \Pr[h(\mathbf{r}) = s]$.

$$\begin{aligned} & \mathsf{SD}\Big(h(\mathbf{r}),\ \mathbf{U}_n\Big) \\ =\ & \frac{1}{2}\sum_{s\in\mathcal{S}} |p_s - \frac{1}{|\mathcal{S}|}| \\ =\ & \frac{1}{2}\sum_{s\in\mathcal{S}} \sqrt{\frac{1}{|\mathcal{S}|}}\cdot\Big(\sqrt{|\mathcal{S}|}\cdot\Big|p_s - \frac{1}{|\mathcal{S}|}\Big|\Big) \\ \le\ & \frac{1}{2}\sqrt{\sum_{s\in\mathcal{S}}(\frac{1}{|\mathcal{S}|})\cdot\sum_{s\in\mathcal{S}}|\mathcal{S}|(p_s - \frac{1}{|\mathcal{S}|})^2} \\ \le\ & \frac{1}{2}\sqrt{2^n(\sum_{s\in\mathcal{S}} p_s^2) - 1} \\ =\ & \frac{1}{2}\sqrt{2^n\cdot\mathsf{Col}(h(\mathbf{r})) - 1}\ , \end{aligned}$$

where the first inequality is Cauchy-Schwartz, i.e., $|\sum_i a_i b_i| \le \sqrt{(\sum_i a_i^2)\cdot(\sum_i b_i^2)}$.

□

B Inequalities, Theorems and Lemmas

Lemma B.1 (Piling-up lemma) *For $0 < \mu < 1/2$ and $\ell \in \mathbb{N}^+$ we have*

$$\Pr\Big[\bigoplus_{i=1}^{\ell} E_i = 0 : E_1, \ldots, E_\ell \leftarrow \mathcal{B}_\mu\Big] = \frac{1}{2}(1 + (1-2\mu)^\ell) \ .$$

Lemma B.2 (Chebyshev's inequality) *Let Y be any random variable (taking real values) with expectation μ and standard deviation σ (i.e., $Var[Y] = \sigma^2 = \mathbb{E}[(Y-\mu)^2]$). Then, for any $\delta > 0$ we have $\Pr[\ |Y-\mu| \ \geq \delta\sigma] \ \leq \ 1/\delta^2$.*

Lemma B.3 (Chernoff bound) *Let $X_1, \ldots, X_n$ be independent random variables and let $\bar{X} = \sum_{i=1}^n X_i$, where $\Pr[0 \leq X_i \leq 1] = 1$ holds for every $1 \leq i \leq n$. Then, for any $\Delta_1 > 0$ and $0 < \Delta_2 < 1$,*

$$\Pr[\ \bar{X} \ > \ (1+\Delta_1)\cdot\mathbb{E}[\bar{X}]\ ] < e^{-\frac{\min(\Delta_1,\Delta_1^2)}{3}\mathbb{E}[\bar{X}]} \ ,$$
$$\Pr[\ \bar{X} \ < \ (1-\Delta_2)\cdot\mathbb{E}[\bar{X}]\ ] < e^{-\frac{\Delta_2^2}{2}\mathbb{E}[\bar{X}]} \ .$$

Fact 1 *For any $0 \leq x \leq 1$, $\log(1+x) \geq x$; and for any $x > -1$ we have $\log(1+x) \leq x/\ln 2$.*

Fact 2 *For $k = o(m)$ we have $\log\binom{m}{k} = (1+o(1))k\log\frac{m}{k}$; and for $\beta = o(1)$, $\log\binom{m}{\frac{m}{2}(1-\beta)} = m(1-\frac{\beta^2}{2}(\log e + o(1))) - \frac{\log m}{2} + O(1)$.*

Proof of Fact 2. The first inequality follows from the approximation $\log(n!) = \log\big(O(\sqrt{n}(\frac{n}{e})^n)\big) = \frac{1}{2}\log n + n\log n - n\log e + O(1)$ and for the second one we have

$$\begin{aligned}
\log\binom{m}{\frac{m}{2}(1-\beta)} &= \log\frac{m!}{\big(\frac{m}{2}(1-\beta)\big)!\big(\frac{m}{2}(1+\beta)\big)!} \\
&= m\log m - \frac{m}{2}(1-\beta)\log\Big(\frac{m}{2}(1-\beta)\Big) \\
&\quad -\frac{m}{2}(1+\beta)\log\Big(\frac{m}{2}(1+\beta)\Big) - \frac{1}{2}\log m + O(1) \\
&= m\Big(1 - \frac{\log e}{2}(1-\beta)\big(-\beta - \frac{1}{2}\beta^2 + o(\beta^2)\big) - \frac{\log e}{2}(1+\beta)\big(\beta - \frac{1}{2}\beta^2 + o(\beta^2)\big)\Big) \\
&\quad -\frac{1}{2}\log m + O(1) \\
&= m(1 - \frac{\log e}{2}\beta^2 + o(\beta^2)) - \frac{1}{2}\log m + O(1) \ ,
\end{aligned}$$

where we use the approximation of $\log(n!)$ and for $x = o(1)$, $\log(1+x) = \log e(x - \frac{1}{2}x^2 + o(x^2))$. □

Lemma B.4 (Sample-preserving reduction [AIK07]) *Any distinguisher $\mathcal{D}$ of running time T with*

$$\Pr_{\mathbf{A}\xleftarrow{\$}\mathbb{F}_2^{q\times n},\mathbf{s}\leftarrow S,\mathbf{e}\leftarrow E}[\mathcal{D}(\mathbf{A}, \mathbf{A}\mathbf{s}+\mathbf{e}) = 1] - \Pr[\mathcal{D}(\mathbf{A}, \mathbf{U}_n) = 1] \geq \varepsilon$$

implies another algorithm $\mathcal{D}'$ of running time $T + O(nq)$ such that

$$\Pr_{\mathbf{A}\xleftarrow{\$}\mathbb{F}_2^{q\times n},\mathbf{s}\leftarrow S,\mathbf{e}\leftarrow E}[\mathcal{D}'(\mathbf{A}, \mathbf{A}\mathbf{s}+\mathbf{e}, \mathbf{r}^\mathsf{T}) = \mathbf{r}^\mathsf{T}\mathbf{s}] \geq \frac{1}{2} + \frac{\varepsilon}{2} \ ,$$

where S and E are any distributions over $\mathbb{F}_2^n$ and $\mathbb{F}_2^q$ respectively.

Lemma B.5 (Goldreich-Levin Theorem [GL89]) *Any algorithm $\mathcal{D}$ of running time T with*

$$\Pr[\mathcal{D}'(f(\mathbf{s}), \mathbf{r}^\mathsf{T}) = \mathbf{r}^\mathsf{T}\mathbf{s}] \geq \frac{1}{2} + \varepsilon$$

implies algorithm $\mathcal{A}$ of running time $O(\frac{n^2}{\varepsilon^2}T)$ such that $\Pr_{\mathbf{s}\leftarrow S}[\mathcal{A}(f(\mathbf{s})) = f^{-1}(f(\mathbf{s}))] = \frac{\Omega(\varepsilon^3)}{n}$, where f is any function on input $s \leftarrow S \in \mathbb{F}_2^n$ and $\mathbf{r} \xleftarrow{\$} \mathbb{F}_2^n$.