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

Yu Yu[1,2,3] and Jiang Zhang[4]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China
[2] Shanghai Qi Zhi Institute, 701 Yunjin Road, Shanghai 200232, China
[3] Shanghai Key Laboratory of Privacy-Preserving Computation, 701 Yunjin Road, Shanghai 200232, China
[4] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
E-mail: {yuyuathk, jiangzhang09}@gmail.com

**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. 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 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.

We start with a simple observation that the hardness of high-noise LPN over large fields is implied by that of the LWE of the same modulus, and is thus reducible from worst-case hardness of lattice problems. We then revisit [BLVW19], which is the main focus of this work. We first expand the underlying binary linear codes (of the NCP) to not only the balanced code considered in [BLVW19] but also to another code (with a minimum dual distance). 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 trade-offs for a wider spectrum of parameter choices. In addition to similar 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 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 $T$, $\epsilon$, $q$ 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 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) [13] 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 implies various cryptographic applications, such as symmetric encryption and authentication [4,20,23,34,39–41,44], zero-knowledge proof for commitment schemes [38], oblivious transfer [22], public-key cryptography [1] and collision resistant hash functions [19, 53]. Regev [47] 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 [18, 46, 47] and as a versatile building block for advanced cryptographic algorithms (such as fully homomorphic encryption [29] and attribute-based encryption [15,33]). In contrast, 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 [19].

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 $\mu$ (i.e., taking the value 1 with probability $\mu$ 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 [6, 28, 40].

[1] Another equivalent formulation is to find out $\mathbf{s}$ given as many (up to one's resource capacity) random noisy inner product $\langle \mathbf{a}_i, \mathbf{s} \rangle + \mathbf{e}_i$ as possible. In this paper we use the $\mathbf{Ax} + \mathbf{e}$ representation that is consistent with that of the decoding problems.

LPN has been extensively studied in learning theory, and it was shown in [27] 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 $\mu$ of LPN is constant (i.e., independent of secret size $n$). The BKW (Blum, Kalai and Wasserman) algorithm [14] solves LPN in time/sample complexity $2^{O(n/\log n)}$. Lyubashevsky [43] 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. Alekhnovich's work [1] implies an interesting variant of LPN (referred to as low-noise LPN) in the noise regime of $\mu = 1/\sqrt{n}$ (or more generally $\mu = n^{-c}$ for $1/2 \le c < 1$) that can be used to construct public-key crypto-systems. More recently, Brakerski et al. [19] 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 $\mathsf{poly}(n) \cdot e^{\mu n}$ [7, 11, 42], 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 [52] and collision resistant hash functions [53] 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 [19] 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 exact weight). The non-promise version of the NCP problem is known to be NP-hard even to approximate [8] and the promise version is also NP-hard in the high-noise regime where the Hamming weight of error vector $|\mathbf{x}| \ge (1/2+\epsilon)d$ for minimal distance $d$ of the code and any arbitrarily small constant $\epsilon$ [26]. As for the algorithms, Berman and Karpinski [10] showed how to search for the $O(n/\log n)$-approximate nearest codeword in polynomial time and Alon, Panigrahy and Yekhanin [2] 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 [43] that bears some resemblance to the smoothing lemma in [19]. 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 [24] 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 [19], 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. [19] showed that if $\mathbf{C}$ belongs to a $\beta$-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[3], 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 $\mu = 1/2 - 1/\mathsf{poly}(n)$. 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

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

[3] More generally, as an end result [19] proves the $n^{O(\lambda)}$-hardness of LPN at noise rate $1/2 - 2^{-\Omega(\lambda)}$ for tunable parameter $\lambda = \omega(1)$, see Remark 2 for discussions.

hardness assumption to hold and recall the 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

Prior to our main work, we give a worst-case hardness result for LPN over large fields, which was introduced in [36] and used in various works, e.g., [3, 5, 16, 17, 25, 30, 37, 51]. Informally, the large-field LPN extends the original LPN to a prime field $\mathbb{F}_p$ with a generalized Bernoulli distribution $\mathcal{B}_{r,p}$, which samples a random element from $\mathbb{F}_p$ with probability $r$ and sets to 0 with probability $1-r$. We show that the hardness of large-field LPN with noise $r = 1 - \Omega(1/\alpha p)$ is implied by that of LWE with the same dimension $n$ and modulus $p$ and parameter $\alpha p$ for the discrete Gaussian distribution. In composition with known worst-case to average-case reductions for LWE, this ensures worst-case hardness for LPN with field size $p \geq \mathsf{poly}(n)$ and high noise rate $r = 1 - \Omega(1/\sqrt{n})$. To our best knowledge, this result doesn't seem to be known previously despite a simple proof. However, similar to the end result of [19], it establishes worst-case hardness guarantee only for LPN whose noise is inversely polynomial ($\Omega(1/\sqrt{n})$ more precisely) close to uniform.

Next we start our investigation on the original LPN (over the binary field). We consider the promise version of NCP on two classes of binary linear codes, i.e., balanced code considered in [19] and (a relaxed form of) independent code. Informally, a $\beta$-balanced $[m, n]$ code is a strengthened form of $[m, n, m(1/2-\beta)]$ code with maximal distance $m(1/2+\beta)$, 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 $\mu \stackrel{\mathsf{def}}{=} \Pr[\mathbf{x}^\mathsf{T}\mathbf{r} = 1] = 1/2 - 2^{-\Theta(\frac{w}{m}d)}$ 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 [19] 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 [2, 10] 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 [42, 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 (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 $(T,\epsilon,q)$-hardness of LPN refers to that no algorithm of time $T$ can solve LPN of $q$ samples with probability better than $\epsilon$. The constant-noise LPN with sub-exponential hardness already implies efficient symmetric-key cryptographic applications, and we further discuss possibilities of going beyond $\mathsf{minicrypt}$[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 [52, 53], 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 3.7). 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

[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 $\delta$. In general, the hardness of NCP (resp., LPN) is insensitive to codeword length $m$ (resp., sample complexity $q$).

[5] $\mathsf{minicrypt}$ is Impagliazzo's [35] hypothetical world where one-way functions exist but public-key cryptography does not.

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 3.7 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 $\bigoplus_{i=1}^m \mathbf{t}_i$.
- $\mathcal{B}_\mu^q \stackrel{\mathsf{def}}{=} \underbrace{\mathcal{B}_\mu \times \cdots \times \mathcal{B}_\mu}_{q}$, where $\mathcal{B}_\mu$ is Bernoulli distribution with parameter $\mu$.

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 $X$. $X \sim Y$ denotes that $X$ and $Y$ are identically distributed. 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 $\mathsf{SD}(X,Y) \stackrel{\mathsf{def}}{=} \frac{1}{2}\sum_x |\Pr[X = x] - \Pr[Y = x]|$. We say that $X$ and $Y$ are $\varepsilon$-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 [19], where $n$ is the size of message (secret to be decoded) and $m$ is codeword length.

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

**Definition 3 (minimum/maximum distance).** *The minimum (resp., maximum) distance of a binary linear code $\mathcal{C}$ refers to* $\min_{\mathbf{x}\neq\mathbf{y}\in\mathcal{C}}\{|\mathbf{x}-\mathbf{y}|\}$ *(resp.,* $\max_{\mathbf{x},\mathbf{y}\in\mathcal{C}}\{|\mathbf{x}-\mathbf{y}|\}$*). A linear $[m,n]$-code with minimum distance $d$ is called a $[m,n,d]$-code.*

A $\beta$-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 and only if its minimum dual distance is at least $k+1$ (i.e., its generator matrix has $k$-wise independent columns). 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 4 (balanced code).** *A binary linear $[m,n]$ code $\mathcal{C}\subseteq\mathbb{F}_2^m$ is $\beta$-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 5 (independent code).** *For a binary linear $[m,n]$ code $\mathcal{C}\subseteq\mathbb{F}_2^m$,*

- *$\mathcal{C}$ is $k$-independent iff. 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 $(k,\zeta)$-independent iff. $\forall i\in\{\frac{k}{2},\frac{k}{2}+1,\ldots,k\}:\Pr[\mathbf{Cr}=\mathbf{0}:\mathbf{r}\leftarrow R_i^m]\leq 2^{-n}(1+\zeta)$.*

*The latter relaxes the independence condition by only enforcing it for $i\in[k/2,k]$ (instead of for all $i\in(0,k]$) and even for $i\in[k/2,k]$ a slackness of $\zeta$ is allowed, in the spirit of almost universal hash functions [49]. Note that there is nothing special with the cut-off point $k/2$, which can be replaced with $\delta k$ for any constant $0<\delta<1$ without affecting our results asymptotically.*

The following lemmas assert 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). In other words, it is very likely that a random matrix is both balanced and independent at the same time. The proof of Lemma 1 follows a simple probabilistic argument (already given in [19]) while as for the proof of Lemma 2 we exploit the pairwise independence in order to apply the Chebyshev's inequality. We refer interested readers to Appendix A and Remark 3 for its proof and discussions. A similar result with $k\approx n/2$ was stated in [21, Theorem 6].

**Lemma 1 (Existence of balanced code [19]).** *A random binary linear $[n,m]$-code is $\beta$-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 $\beta$-balanced with probability $1-2^{-\Omega(n)}$.*

*Remark 1 (existence vs. abundance).* Lemma 1 states that $\beta\geq 2\sqrt{n/m}$ ensures the overwhelming abundance rather than the mere existence of balanced codes. We remark that the difference is not substantial, e.g., for any arbitrarily small

$\varepsilon > 0$ by setting $\beta \geq \sqrt{\frac{4(n+1+\varepsilon\log(e))}{(\log e)m}} \approx 1.66\sqrt{n/m}$ we derive a corollary of Lemma 1 that $\beta$-balanced [$n$,$m$]-code exists with a fraction of at least

$$1 - 2^{n+1}e^{-\frac{\beta^2 m}{4}} \geq 1 - e^{-\varepsilon} \approx \varepsilon \ .$$

The above is essentially the Gilbert-Varshamov bound that asserts the existence of certain codes[6], and it is almost tight for binary linear codes [12].

**Lemma 2 (Existence of independent code).** *A random binary linear [m,n] code $\mathcal{C}$ is (k, $\zeta$)-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 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 $\beta$-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* $(k,\zeta)$*-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 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 [6].

[6] Strictly speaking, Gilbert-Varshamov bound concerns with the existence of a code with minimum distance $m(1/2-\beta)$ while the balanced code we consider requires minimum/maximum distance $m(1/2\mp\beta)$ at the same time, where the difference can be omitted due to the symmetry of binomial coefficient $\binom{m}{m(1/2\mp\beta)}$ centered on $m/2$.

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

**Lemma 3 (Unique decoding of binary 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 LPN

## 3.1 Worst-case hardness for large-field LPN

Denote with $\mathsf{LWE}_{n,p,\alpha}$ and $\mathsf{LPN}_{n,r}(\mathbb{F}_p)$ the LWE problem and the large-field LPN problem respectively, both of dimension $n$ and over prime modulus $p$, where the LWE's noise follows the discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z},\alpha p}$ of standard deviation parameter $\alpha p$, and the LPN's noise distribution returns a random element over $\mathbb{F}_p$ with probability $r$, and is set to 0 otherwise.

**Lemma 4 (LWE implies high-noise LPN over $\mathbb{F}_p$).** *Assume that $\mathsf{LWE}_{n,p,\alpha}$ with prime $p$, $\alpha = o(1)$,$\alpha p = \omega(\log n)$ is hard, then $\mathsf{LPN}_{n,r}(\mathbb{F}_p)$ with $r = 1 - \Omega(\frac{1}{\alpha p})$ is hard.*

*Proof.* Every LWE sample $(\mathbf{a}_i,\langle \mathbf{a}_i, \mathbf{s}\rangle + e_i)$ can be transformed into an LPN sample $(\mathbf{a}_i',\langle \mathbf{a}_i', \mathbf{s}\rangle + e_i')$ (over the same field) by multiplying with a random $m_i \xleftarrow{\$} \mathbb{F}_p \setminus \{0\}$, where $\mathbf{a}_i'$ is the scalar-vector product $m_i\mathbf{a}_i$, and $e_i' = m_i e_i$. For any $e_i \neq 0$ we have $(\mathbf{a}_i',e_i')$ is uniformly distributed over $\mathbb{F}_p^n \times (\mathbb{F}_p \setminus \{0\})$, and for $e_i = 0$ it is uniform over $\mathbb{F}_p^n \times \{0\}$. Thus, overall $(\mathbf{a}_i',e_i')$ is an $\mathsf{LPN}_{n,r}(\mathbb{F}_p)$ sample with

$$1 - r = \Pr[e_i = 0] - \frac{1 - \Pr[e_i = 0]}{p-1} \ge \Omega(1/\alpha p) - \frac{2}{p} \ge \Omega(1/\alpha p)\ .$$

Lemma 4 puts no lower bounds on the size of $p$, and recall that the LPN problem becomes a special case of LWE for $p = 2$ (for which no reductions are needed).

However, in order for the LWE to be quantumly reducible from worst-case lattice problems, we need $q = \mathsf{poly}(n)$ and $\alpha p = \Omega(\sqrt{n})$. The reduction can be made classical at the cost of either a much larger modulus $q \geq 2^{n/2}$ or relying on a non-standard variant of $\mathsf{GapSVP}$ [46].

**Theorem 2 ( [47]).** *For any $p \leq \mathsf{poly}(n)$, any $\alpha p \geq 2\sqrt{n}$ and $0 < \alpha < 1$, solving the (decisional) $\mathsf{LWE}_{n,p,\alpha}$ problem is at least as hard as quantumly solving $\mathsf{GapSVP}_\gamma$ and $\mathsf{SIVP}_\gamma$ on arbitrary $n$-dimensional lattices, for some $\gamma = \tilde{O}(n/\alpha)$.*

To summarize, based on the (quantum or even classical) worst-case hardness of lattice problems, we establish up to $2^{O(n)}$-(average-case)-hardness of large-field LPN for modulus $p \geq \mathsf{poly}(n)$ and noise rate $r = 1 - \Omega(1/\sqrt{n})$. Next, we will revisit [19] and show worst-case to average-case reductions for constant-noise LPN (over the binary field), which is the main focus of this work.

### 3.2 The worst-case to average-case reduction from [19]

Brakerski et al. [19] 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 ( [19]).** *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 4, the conversion produces $q$ LPN samples of noise rate $\mu$ 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 2 (possibilities and limitations).* Other possible parameter choices are also discussed in [19], 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 $\mu$ (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 due to the high noise rate. In fact, the barriers are inherent in its smoothing lemma Theorem 4. Informally, assume that NCP at noise rate $\frac{w}{m} = \frac{\log n \cdot \lambda}{n}$ is $n^{O(\lambda)}$-hard[7] on $\beta$-balanced code,

[7] We recall the known attacks [2,10] of time complexity $2^{O(\frac{w}{m}n)}$ on NCP of noise rate $\frac{w}{m}$.

then the LPN of noise rate $\mu = \frac{1}{2} - \frac{1}{2}(1 - \frac{2w}{m})^d$ is (at most) $n^{O(\lambda)}$-hard provided that $(\frac{2w}{m} + \beta)^d < 2^{-n/2}$. Therefore, we need to set $\lambda = \omega(1)$ for the worst-case hardness assumption to hold. Further, regardless of the value of $\beta$ it requires $d = \Omega(n/\log n)$ to make $(\frac{2w}{m} + \beta)^d < 2^{-n/2}$. This lower bounds the noise rate of LPN, i.e., $\mu = \frac{1}{2} - \frac{1}{2}(1 - \frac{2w}{m})^d = 1/2 - 2^{-\Omega(\lambda)}$. Raising the value of $\lambda$ brings better hardness, but at the same time it makes the noise of LPN closer to uniform (and hence renders the result less interesting). A reasonable compromise seems to let $\lambda = \log n$ which was the main choice of [19].

**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 4 (W/A-case reduction via code smoothing [19]).** *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* $\beta$*-balanced* $[m, n]$ *code and* $O(mnq)$ *accounts for the complexity of Algorithm* 1*.*

The authors of [19] proved the above smoothing lemma using harmonic analysis. We give an alternative proof via Vazirani's XOR lemma [31, 50]. 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 [19] after unrolling out the proof of the XOR lemma.

**Lemma 5 (Vazirani's XOR lemma [31, 50]).** *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 4. 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 $R_{d-j,m-w}$.

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

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 $\mu$ by the following:

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

### 3.3 On the Non-triviality of Code Smoothing

As discussed in Remark 2, the worst-case to average-case reduction in [19] may only give rise to the $n^{O(\lambda)}$-hardness of LPN on noise rate $\mu = 1/2 - 2^{-\Omega(\lambda)}$. Ideally, the dependency of $\mu$ on $\lambda$ would be removed such that the noise rate of LPN $\mu$ can be kept constant while assigning a large value to $\lambda$ to enjoy sub-exponential hardness for LPN. This will be goal of this paper.

Before we proceed, it is worth to repeat what we pointed out in the introduction that a better smoothing lemma is non-trivial without new ideas. The possibilities of smoothing linear binary codes can be investigated existentially using a probabilistic argument. The code smoothing lemma, as stated in eq: 1, can be seen as deterministic randomness extractor from Bernoulli-like distributions. 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 $\mathsf{SD}\Big((\mathbf{C},\mathbf{Cr},\mathbf{x}^{\mathsf{T}}\mathbf{r}),(\mathbf{C},\mathbf{U}_n,\mathbf{x}^{\mathsf{T}}\mathbf{r})\Big)\le 2^{-n}$ 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 $\mathsf{SD}\Big((\mathbf{Cr},\mathbf{x}^{\mathsf{T}}\mathbf{r}),(\mathbf{U}_n,\mathbf{x}^{\mathsf{T}}\mathbf{r})\Big)\le 2^{-n/2}$. 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 eq: 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 code/distribution 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).

### 3.4 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 eq: 1 of Theorem 4) and the values of $\beta$, $k$, and $\zeta$ are chosen to ensure the existence of respective codes (Lemma 1 and Lemma 2 ).

**Theorem 5 (W/A-reduction for $\beta$-balanced codes).** *Assume that the* $\mathsf{balNCP}_{n,m,w,\beta}$ *is $(T,\epsilon)$-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 $m$ and $d$ satisfying $d\log(m/d)\ge 4n$.*

**Theorem 6 (W/A-reduction for independent codes).** *Assume that the* $\mathsf{indNCP}_{n,m,w,k,\zeta}$ *is $(T,\epsilon)$-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 $m$ and $d$ satisfying $d\log(m/d)\ge 7n$.*

*Proof sketch.* The proofs of Theorem 5 and Theorem 6 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 9 and Lemma 12). While replacing $R_{d,m}$ with $\mathcal{B}_{\frac{d}{m}}$ seems equivalent in terms of the resulting noise rate $\mu$ (almost same as eq: 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 $\delta$ that allow for a wider range of parameter choices. □

**A comparison with [19].** With appropriate parameter assignment to Theorem 5, we obtain comparable results to [19] (see Theorem 3). Following [19], we consider balanced code with noise rate $w/m = \log^2 n/n$. As explained in Remark 1, while $\beta \geq 2\sqrt{n/m}$ ensures the overwhelming abundance of the balanced code, the existence condition does not impose much less, i.e., $\beta \geq 1.66\sqrt{n/m}$. It is convenient to fix $\beta = 2\sqrt{n/m}$ as larger values for $\beta$ can only lead to larger $d$ and renders LPN's noise $\mu$ closer to uniform. We give the comparison in Table 1 with various values for $m \geq n^{1+\epsilon}$. Note that the NCP is hard up to $T = n^{O(\log n)}$ due to known attacks, and the reduction requires $T' = T - O(nmq) > 0$, so here we let $m = \mathsf{poly}(n)$, and $q = \mathsf{poly}(n)$. In [19] the constraint on $d$ is implied by eq: 1, i.e.,

$$2^{\frac{n+1}{2}} \cdot (\frac{2w}{m} + \beta)^d = 2^{\frac{n+1}{2}} \cdot (\frac{2\log^2 n}{n} + \beta)^d = \mathsf{negl}(n) \tag{3}$$

while Theorem 5 explicitly sets $d\log(m/d) = 4n$. Substituting $d$ into the noise rate of LPN, which is roughly $\mu \approx 1/2 - e^{-\frac{2w}{m}d + O(1)}$ in both cases, yields

$$\mu \approx \begin{cases} 1/2 - n^{\frac{-2.88\log n}{(\log m - \log n)}} & \text{for } \frac{n^3}{\log^4 n} > m \geq n^{1+\epsilon} \\ 1/2 - n^{-1.44} & \text{for } m \geq \frac{n^3}{\log^4 n} \end{cases}$$

for [19], and $\mu \approx 1/2 - n^{\frac{-11.54\log m}{(\log m - \log n)}}$ for $m \geq n^{1+\epsilon}$ in our case. As we can see from Table 1, our result is slightly (by a factor of 4 in the exponent) worse than [19] for $m < n^3$, and the gap decreases from $m \geq n^3$. Our result starts to show its advantage for $m \geq n^9$. In other words, [19] stays at $\mu \approx 1/2 - n^{-1.4}$ and ceases to improve for $m \geq n^3$. This is because for $m \geq n^3$ it is $2\log^2 n/n$ (instead of $\beta \leq 2\sqrt{n/m} \leq 2/n$) that dominates the term in eq: 3, and thus one can no longer trade $\beta$ for better $\mu$ regardless how small $\beta$ is.

**Table 1.** Restate Theorem 5 and its analogue in [19] as "$T$-wc-hardness of "NCP$(n, m, \frac{w}{m})$ on $\beta$-balanced code implies $T'$-ac-hardness of LPN$(n, q, \mu)$" for $m \in \{n^{1.2}, n^2, \ldots, 100\}$, where $\frac{w}{m} = \frac{\log^2 n}{n}$, $\beta = 2\sqrt{n/m}$, $T' = T - O(nmq)$, $q = \mathsf{poly}(n)$.

| $m$ | LPN's noise rate $\mu$ from [19] (see Theorem 3) | LPN's noise rate $\mu$ from our Theorem 5 |
|---|---|---|
| $n^{1.2}$ | $\frac{1}{2} - n^{-14}$ | $\frac{1}{2} - n^{-58}$ |
| $n^2$ | $\frac{1}{2} - n^{-3}$ | $\frac{1}{2} - n^{-12}$ |
| $n^3$ | $\frac{1}{2} - n^{-1.4}$ | $\frac{1}{2} - n^{-6}$ |
| $n^6$ | $\frac{1}{2} - n^{-1.4}$ | $\frac{1}{2} - n^{-2.3}$ |
| $n^9$ | $\frac{1}{2} - n^{-1.4}$ | $\frac{1}{2} - n^{-1.4}$ |
| $n^{10}$ | $\frac{1}{2} - n^{-1.4}$ | $\frac{1}{2} - n^{-1.3}$ |
| $n^{100}$ | $\frac{1}{2} - n^{-1.4}$ | $\frac{1}{2} - n^{-0.1}$ |

Our result admits a wider range of trade-offs between $m$ and $\mu$. More importantly, when $m$ goes beyond $\mathsf{poly}(n)$ it enables to guarantee sub-exponential hardness for constant-noise LPN. In particular, we now assume that there exists constant $\varepsilon$ 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 7 (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** $1/2 \le c < 1$**:** $\mathsf{LPN}_{n,\mu,q}$ *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 5 and Theorem 6 (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. □

### 3.5 Smoothing balanced codes

Our smoothing lemma benefits from Lemma 7 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 that is coordinate-wise independent. We first introduce Lemma 6 based on which Lemma 7 is built.

**Lemma 6.** *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*

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

*where* $e_1 \xleftarrow{\$} \mathcal{B}_a$ $(0 \le a \le 1/2)$ *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 7.** *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*

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

*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 $\mathbf{C}\mathbf{r} = \mathbf{C}_\mathbf{x}\mathbf{r}_\mathbf{x} + \bigoplus_{i=1}^{w} e_i\mathbf{c}_i$ where $e_i \leftarrow \mathcal{B}_a$ and $\mathbf{c}_i$ is the $i$-th column vector of $\mathbf{C}\setminus\mathbf{C}_\mathbf{x}$ (i.e., the columns of $\mathbf{C}$ that are excluded from $\mathbf{C}_\mathbf{x}$). By applying Lemma 6 $w$ times we get

$$\mathsf{SD}(\mathbf{C}\mathbf{r}, \mathbf{U}_n)\ \geq\ (1-2a)^w \cdot \mathsf{SD}(\mathbf{C}_\mathbf{x}\mathbf{r}_\mathbf{x}, \mathbf{U}_n)\ .$$

We need the following corollary of two-source extractors to prove the smoothing lemma. Recall that two-source extractor distills almost uniform randomness from pair-wise independent sources $\mathbf{b}^\mathsf{T}$ and $\mathbf{r}$, while Corollary 1 shows that the result holds even when $\mathbf{b}^\mathsf{T}$ is fixed (has no entropy at all) as long as certain conditioned are met.

**Lemma 8 (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*

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

**Corollary 1.** *For random variable $\mathbf{r}$ and distribution $D$ defined over $\mathbb{F}_2^m$ and $\mathbb{F}_2$ respectively, define set $\mathcal{B}_{D,\mathbf{r}} \stackrel{\mathsf{def}}{=} \{\mathbf{b}^\mathsf{T} : \mathbf{b}^\mathsf{T}\mathbf{r} \sim D\}$, where $\mathbf{H}_\infty(\mathbf{r}) = k_r$, and $|\mathcal{B}_{D,\mathbf{r}}| \geq 2^{k_b}$. Then, for any $\mathbf{b}^\mathsf{T} \in \mathcal{B}_{D,\mathbf{r}}$ it holds that*

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

*Proof.* Fix an arbitrary $\mathbf{b}^\mathsf{T} \in \mathcal{B}_{D,\mathbf{r}}$, and let $\mathbf{b}'^\mathsf{T}$ be a random variable that is uniform over $\mathcal{B}_{D,\mathbf{r}}$, we have

$$\mathsf{SD}(\mathbf{b}^\mathsf{T}\mathbf{r},\ U_1) = \mathsf{SD}(D, U_1) = \mathsf{SD}\Big(\ (\mathbf{b}'^\mathsf{T}, \mathbf{b}'^\mathsf{T}\mathbf{r}),\ (\mathbf{b}'^\mathsf{T}, U_1)\ \Big) \leq 2^{-(\frac{k_b+k_r-m}{2}+1)}\ ,$$

where the equalities are simply by the definitions of $\mathcal{B}_{D,\mathbf{r}}$ and $\mathbf{b}'^\mathsf{T}$, and the inequality follows from the two source extractor lemma below.

**Lemma 9 (Smoothing lemma for balanced codes).** *Let $\beta \le 2\sqrt{n/m}$, $d\log(m/d) \ge 4n$, $d = O(n)$, and let $\mathbf{C} \in \mathbb{F}_2^{n\times m}$ be any generator matrix for a $\beta$-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*

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

*Proof.* The noise rate $\mu$ directly follows from the Piling-up lemma.

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

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. The third and fourth inequalities follow from Corollary 1 based on two-source extractors. In particular, let $\mathbf{b}'^\mathsf{T}$ be a random variable uniformly drawn from $\mathcal{R}^m_{\frac{(1-\beta)m}{2}}$, i.e, the set of all values with the same Hamming weight as $\mathbf{b}^\mathsf{T}$. We observe that $\mathbf{r}' \sim R^m_j$ implies that every $\mathbf{b}_1^\mathsf{T}$ and $\mathbf{b}_2^\mathsf{T}$ with $|\mathbf{b}_1^\mathsf{T}| = |\mathbf{b}_2^\mathsf{T}|$ must satisfy $\mathbf{b}_1^\mathsf{T}\mathbf{r}' \sim \mathbf{b}_2^\mathsf{T}\mathbf{r}'$ and therefore $\mathsf{SD}(\mathbf{b}^\mathsf{T}\mathbf{r}',\ U_1) = \mathsf{SD}\Big(\ (\mathbf{b}'^\mathsf{T}, \mathbf{b}'^\mathsf{T}\mathbf{r}'),\ (\mathbf{b}'^\mathsf{T}, U_1)\ \Big)$. This allows to apply the strong two-source extractor, where Fact 2 is used to estimate the entropy of $\mathbf{b}'^\mathsf{T}$, i.e., $\log\binom{m}{\frac{(1-\beta)m}{2}}$. Finally, we set $\beta = 2\sqrt{n/m}$, $d\log(m/d) = 4n$ and sufficiently small $\delta$ to complete the proof. Following the proof of Theorem 4, 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 7.

As stated in Lemma 10, it is not hard to see a lower bound on smoothing any binary linear code (i.e., not just the balanced code considered above) with respect to $\mathbf{r} \leftarrow \mathcal{B}^m_{\frac{d}{m}}$. This means that our smoothing lemmas (Lemma 9 and Lemma 12) are optimal (up to some constant factor in the exponent) for $\mu \le 1/2 - 2^{-O(d)}$.

**Lemma 10 (Lower bound on code smoothing).** *For any $\mathbf{C} \in \mathbb{F}_2^{n\times m}$, for any $\mathbf{x} \in \mathbb{F}_2^m$ and $\mathbf{r} \leftarrow \mathcal{B}^m_{\frac{d}{m}}$ with $\frac{d}{m} = o(1)$ it holds that*

$$\mathsf{SD}\Big((\mathbf{Cr}, \mathbf{x}^\mathsf{T}\mathbf{r})\ ,\ (\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})\ \Big) \ge 2^{-O(d)} \quad .$$

*Proof.* Denote the first row of $\mathbf{C}$ by $\mathbf{c}_1^\mathsf{T}$

$$\mathsf{SD}\Big((\mathbf{Cr}, \mathbf{x}^\mathsf{T}\mathbf{r})\ ,\ (\mathbf{U}_n, \mathbf{x}^\mathsf{T}\mathbf{r})\ \Big) \ge \mathsf{SD}(\mathbf{c}_1^\mathsf{T}\mathbf{r}, \mathbf{U}_1) = \frac{(1-\frac{2d}{m})^{|\mathbf{c}_1^\mathsf{T}|}}{2} \ge 2^{-O(d)} \quad ,$$

where the equality is the piling-up lemma, and the last inequality is due to $|\mathbf{c}_1^\mathsf{T}| \le m$ and $1 - x = 2^{-O(x)}$ for $x = o(1)$.

### 3.6 Smoothing Independent Codes

The proof of the smoothing lemma relies on following Lemma 11, which is abstracted out from the leftover hash lemma (see Appendix A for its proof).

**Lemma 11 (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*

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

**Lemma 12 (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*

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

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

*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 11,

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

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

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

where for constant $0 < \Delta < 1$ variable $\mathbf{\ddot{r}} \stackrel{\mathsf{def}}{=} \mathbf{r}_1 - \mathbf{r}_2$ follows a convex combination of $R^m_{2d(1-\delta)(1-\Delta)}$, $R^m_{2d(1-\delta)(1-\Delta)+1}$, $\ldots$, $R^m_{2d(1-\delta)}$ whose weights lie in between[8]

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

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

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

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

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

and $\mathsf{SD}(\mathbf{Cr}\ ,\ \mathbf{U}_n) \le 2^{-\Omega(d)}$. The rest follow the same steps as in the proof of Lemma 9.

### 3.7 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 [52, 53], 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.[9] We explain in details below.

**Theorem 8 ( [53]).** *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 $t^2 \le q \le T = 2^{\frac{8\mu t}{\ln 2(1-2\mu)}}$. For*

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

[9] 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 $q \ge \mathsf{poly}(n)$ due to the sample-preserving reduction [6].

*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* $\mathsf{Expand}$ *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 [53, 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* $T' = 2^{\frac{4\mu t}{\ln 2(1-2\mu)}-1}$

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

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 $\mu$ 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 [53] 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.

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 9). 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 [52] already minimized the hardness needed for LPN to construct PKE, and also used the parameter switching technique. We restate the main results of [52] below.

**Theorem 9 ( [52]).** *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 10 below, and we informally state our reduction results (Theorem 5 and Theorem 6) in Lemma 13. In order for Lemma 13 to compose with Theorem 10, 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 needed by Theorem 10, i.e., $\mu = 1/2 - 2^{-(\log n)^\delta}$ for any constant $0 < \delta < 1$.

**Theorem 10 ( [43]).** *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 13 (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 5 and Theorem 6) 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 first show that the hardness of high-noise large-field LPN is reducible from the worst-case hardness of lattice problems via a simple reduction from LWE to LPN over the same modulus. We then 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, improving upon the work of [19]. 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.

## Acknowledgement

Yu Yu, the corresponding author, was supported by the National Key Research and Development Program of China (Grant Nos. 2020YFA0309705 and 2018YFA0704701) and the National Natural Science Foundation of China (Grant Nos. 61872236 and 61971192). Jiang Zhang is supported by the National Natural Science Foundation of China (Grant Nos. 62022018, 61932019), the National Key Research and Development Program of China (Grant No. 2018YFB0804105). This work is also supported by the Major Program of Guangdong Basic and Applied Research (Grant No. 2019B030302008), Shandong Provincial Key Research and Development Program (Major Scientific and Technological Innovation Project, Grant No. 2019JZZY010133), Shandong Key Research and Development Program (Grant No. 2020ZLYS09) and in part by the Anhui Initiative in Quantum Information Technologies under Grant No. AHY150100.

## References

1. Alekhnovich, M.: More on average case vs approximation complexity. In: 44th FOCS. pp. 298–307. IEEE Computer Society Press, Cambridge, MA, USA (Oct 11–14, 2003). https://doi.org/10.1109/SFCS.2003.1238204
2. Alon, N., Panigrahy, R., Yekhanin, S.: Deterministic approximation algorithms for the nearest codeword problem. In: Algebraic Methods in Computational Complexity (2009)
3. Applebaum, B., Avron, J., Brzuska, C.: Arithmetic cryptography: Extended abstract. In: Roughgarden, T. (ed.) ITCS 2015. pp. 143–151. ACM, Rehovot, Israel (Jan 11–13, 2015). https://doi.org/10.1145/2688073.2688114

4. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 2009). https://doi.org/10.1007/978-3-642-03356-8_35
5. Applebaum, B., Damgård, I., Ishai, Y., Nielsen, M., Zichron, L.: Secure arithmetic computation with constant computational overhead. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 223–254. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2017). https://doi.org/10.1007/978-3-319-63688-7_8
6. Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography with constant input locality. Journal of Cryptology **22**(4), 429–469 (Oct 2009). https://doi.org/10.1007/s00145-009-9039-0
7. Arora, S., Babai, L., Stern, J., Sweedyk, Z.: The hardness of approximate optima in lattices, codes, and systems of linear equations. Journal of Computer and System Sciences **54**(2), 317–331 (1997)
8. Arora, S., Babai, L., Stern, J., Sweedyk, Z.: The hardness of approximate optima in lattices, codes, and systems of linear equations. Journal of Computer and System Sciences **54**(2), 317–331 (1997)
9. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 520–536. Springer, Heidelberg, Germany, Cambridge, UK (Apr 15–19, 2012). https://doi.org/10.1007/978-3-642-29011-4_31
10. Berman, P., Karpinski, M.: Approximating minimum unsatisfiability of linear equations. In: Eppstein, D. (ed.) 13th SODA. pp. 514–516. ACM-SIAM, San Francisco, CA, USA (Jan 6–8, 2002)
11. Bernstein, D.J., Lange, T., Peters, C.: Smaller decoding exponents: Ball-collision decoding. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 743–760. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2011). https://doi.org/10.1007/978-3-642-22792-9_42
12. Blinovsky, V.M.: Proof of tightness of Varshamov - Gilbert bound for binary codes. CoRR **abs/1606.01592** (2016)
13. Blum, A., Furst, M.L., Kearns, M.J., Lipton, R.J.: Cryptographic primitives based on hard learning problems. In: Stinson, D.R. (ed.) CRYPTO'93. LNCS, vol. 773, pp. 278–291. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 22–26, 1994). https://doi.org/10.1007/3-540-48329-2_24
14. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. In: 32nd ACM STOC. pp. 435–440. ACM Press, Portland, OR, USA (May 21–23, 2000). https://doi.org/10.1145/335305.335355
15. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer, Heidelberg, Germany, Copenhagen, Denmark (May 11–15, 2014). https://doi.org/10.1007/978-3-642-55220-5_30
16. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y.: Compressing vector OLE. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018. pp. 896–912. ACM Press, Toronto, ON, Canada (Oct 15–19, 2018). https://doi.org/10.1145/3243734.3243868

17. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Rindal, P., Scholl, P.: Efficient two-round OT extension and silent non-interactive secure computation. In: ACM CCS 2019. pp. 291–308. ACM Press (2019). https://doi.org/10.1145/3319535.3354255
18. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC. pp. 575–584. ACM Press, Palo Alto, CA, USA (Jun 1–4, 2013). https://doi.org/10.1145/2488608.2488680
19. Brakerski, Z., Lyubashevsky, V., Vaikuntanathan, V., Wichs, D.: Worst-case hardness for LPN and cryptographic hashing via code smoothing. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part III. LNCS, vol. 11478, pp. 619–635. Springer, Heidelberg, Germany, Darmstadt, Germany (May 19–23, 2019). https://doi.org/10.1007/978-3-030-17659-4_21
20. Cash, D., Kiltz, E., Tessaro, S.: Two-round man-in-the-middle security from LPN. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A, Part I. LNCS, vol. 9562, pp. 225–248. Springer, Heidelberg, Germany, Tel Aviv, Israel (Jan 10–13, 2016). https://doi.org/10.1007/978-3-662-49096-9_10
21. Chen, H., Cramer, R., Goldwasser, S., de Haan, R., Vaikuntanathan, V.: Secure computation from random error correcting codes. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 291–310. Springer, Heidelberg, Germany, Barcelona, Spain (May 20–24, 2007). https://doi.org/10.1007/978-3-540-72540-4_17
22. David, B., Dowsley, R., Nascimento, A.C.A.: Universally composable oblivious transfer based on a variant of LPN. In: Gritzalis, D., Kiayias, A., Askoxylakis, I.G. (eds.) CANS 14. LNCS, vol. 8813, pp. 143–158. Springer, Heidelberg, Germany, Heraklion, Crete, Greece (Oct 22–24, 2014). https://doi.org/10.1007/978-3-319-12280-9_10
23. Dodis, Y., Kiltz, E., Pietrzak, K., Wichs, D.: Message authentication, revisited. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 355–374. Springer, Heidelberg, Germany, Cambridge, UK (Apr 15–19, 2012). https://doi.org/10.1007/978-3-642-29011-4_22
24. Döttling, N.: Low noise LPN: KDM secure public key encryption and sample amplification. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 604–626. Springer, Heidelberg, Germany, Gaithersburg, MD, USA (Mar 30 – Apr 1, 2015). https://doi.org/10.1007/978-3-662-46447-2_27
25. Döttling, N., Ghosh, S., Nielsen, J.B., Nilges, T., Trifiletti, R.: TinyOLE: Efficient actively secure two-party computation from oblivious linear function evaluation. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 2263–2276. ACM Press, Dallas, TX, USA (Oct 31 – Nov 2, 2017). https://doi.org/10.1145/3133956.3134024
26. Dumer, I., Micciancio, D., Sudan, M.: Hardness of approximating the minimum distance of a linear code. In: 40th FOCS. pp. 475–485. IEEE Computer Society Press, New York, NY, USA (Oct 17–19, 1999). https://doi.org/10.1109/SFFCS.1999.814620
27. Feldman, V., Gopalan, P., Khot, S., Ponnuswami, A.K.: New results for learning noisy parities and halfspaces. In: 47th FOCS. pp. 563–574. IEEE Computer Society Press, Berkeley, CA, USA (Oct 21–24, 2006). https://doi.org/10.1109/FOCS.2006.51
28. Fischer, J.B., Stern, J.: An efficient pseudo-random generator provably as secure as syndrome decoding. In: Maurer, U.M. (ed.) EUROCRYPT'96. LNCS, vol. 1070,

pp. 245–255. Springer, Heidelberg, Germany, Saragossa, Spain (May 12–16, 1996). https://doi.org/10.1007/3-540-68339-9_22

29. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 169–178. ACM Press, Bethesda, MD, USA (May 31 – Jun 2, 2009). https://doi.org/10.1145/1536414.1536440
30. Ghosh, S., Nielsen, J.B., Nilges, T.: Maliciously secure oblivious linear function evaluation with constant overhead. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 629–659. Springer, Heidelberg, Germany, Hong Kong, China (Dec 3–7, 2017). https://doi.org/10.1007/978-3-319-70694-8_22
31. Goldreich, O.: Three XOR-lemmas - an exposition. In: Studies in Complexity and Cryptography, pp. 248–272 (2011)
32. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: 21st ACM STOC. pp. 25–32. ACM Press, Seattle, WA, USA (May 15–17, 1989). https://doi.org/10.1145/73007.73010
33. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC. pp. 545–554. ACM Press, Palo Alto, CA, USA (Jun 1–4, 2013). https://doi.org/10.1145/2488608.2488677
34. Hopper, N.J., Blum, M.: Secure human identification protocols. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 52–66. Springer, Heidelberg, Germany, Gold Coast, Australia (Dec 9–13, 2001). https://doi.org/10.1007/3-540-45682-1_4
35. Impagliazzo, R.: A personal view of average-case complexity. In: Structure in Complexity Theory Conference. pp. 134–147 (1995)
36. Ishai, Y., Prabhakaran, M., Sahai, A.: Secure arithmetic computation with no honest majority. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 294–314. Springer, Heidelberg, Germany (Mar 15–17, 2009). https://doi.org/10.1007/978-3-642-00457-5_18
37. Jain, A., Lin, H., Sahai, A.: Indistinguishability obfuscation from well-founded assumptions. Cryptology ePrint Archive, Report 2020/1003 (2020), `https://eprint.iacr.org/2020/1003`
38. Jain, A., Krenn, S., Pietrzak, K., Tentes, A.: Commitments and efficient zero-knowledge proofs from learning parity with noise. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 663–680. Springer, Heidelberg, Germany, Beijing, China (Dec 2–6, 2012). https://doi.org/10.1007/978-3-642-34961-4_40
39. Juels, A., Weis, S.A.: Authenticating pervasive devices with human protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 293–308. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2005). https://doi.org/10.1007/11535218_18
40. Katz, J., Shin, J.S.: Parallel and concurrent security of the HB and HB+ protocols. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 73–87. Springer, Heidelberg, Germany, St. Petersburg, Russia (May 28 – Jun 1, 2006). https://doi.org/10.1007/11761679_6
41. Kiltz, E., Pietrzak, K., Cash, D., Jain, A., Venturi, D.: Efficient authentication from hard learning problems. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 7–26. Springer, Heidelberg, Germany, Tallinn, Estonia (May 15–19, 2011). https://doi.org/10.1007/978-3-642-20465-4_3
42. Kirchner, P., Fouque, P.A.: An improved BKW algorithm for LWE with applications to cryptography and lattices. In: Gennaro, R., Robshaw, M.J.B. (eds.)

CRYPTO 2015, Part I. LNCS, vol. 9215, pp. 43–62. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 2015). https://doi.org/10.1007/978-3-662-47989-6_3

43. Lyubashevsky, V.: 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). pp. 378–389 (2005)
44. Lyubashevsky, V., Masny, D.: Man-in-the-middle secure authentication schemes from LPN and weak PRFs. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 308–325. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2013). https://doi.org/10.1007/978-3-642-40084-1_18
45. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 107–124. Springer, Heidelberg, Germany, Seoul, South Korea (Dec 4–8, 2011). https://doi.org/10.1007/978-3-642-25385-0_6
46. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 333–342. ACM Press, Bethesda, MD, USA (May 31 – Jun 2, 2009). https://doi.org/10.1145/1536414.1536461
47. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC. pp. 84–93. ACM Press, Baltimore, MA, USA (May 22–24, 2005). https://doi.org/10.1145/1060590.1060603
48. Stern, J.: A method for finding codewords of small weight. In: Coding Theory and Applications, 3rd International Colloquium. pp. 106–113 (1988)
49. Stinson, D.R.: 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)
50. Vazirani, U.V.: Randomness, Adversaries and Computation (Random Polynomial Time). Ph.D. thesis (1986), aAI8718194
51. Weng, C., Yang, K., Katz, J., Wang, X.: Wolverine: Fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits. Cryptology ePrint Archive, Report 2020/925 (2020), `https://eprint.iacr.org/2020/925`
52. Yu, Y., Zhang, J.: Cryptography with auxiliary input and trapdoor from constant-noise LPN. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 214–243. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016). https://doi.org/10.1007/978-3-662-53018-4_9
53. Yu, Y., Zhang, J., Weng, J., Guo, C., Li, X.: Collision resistant hashing from sub-exponential learning parity with noise. In: ASIACRYPT 2019, Part II. pp. 3–24. LNCS, Springer, Heidelberg, Germany (Dec 2019). https://doi.org/10.1007/978-3-030-34621-8_1

## A Proofs Omitted

*Proof of Lemma 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 $k/2 \leq i \leq k$,

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

where $N \stackrel{\mathsf{def}}{=} |\mathcal{R}_i^m| \geq \binom{m}{k/2}$, the second inequality is by Chebyshev, and the equality is due to the following: denote $z = \sum_{\mathbf{r}\in\mathcal{R}_i^m} z_{\mathbf{C},\mathbf{r}}$ and $\mu = \mathbb{E}[z]$ and therefore

$$
\begin{aligned}
Var[z] &= \mathbb{E}[(z-\mu)^2] \\
&= \mathbb{E}[z^2] - 2\mu\mathbb{E}[z] + \mu^2 \\
&= \mathbb{E}[z^2] - \mu^2 \\
&= \mathbb{E}[z^2] - N^2 2^{-2n} ,
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}[z^2] &= \mathbb{E}\Big[(z_1 + z_2 + \ldots + z_N)^2\Big] \\
&= \mathbb{E}\Big[\sum_{u\neq v} z_u \cdot z_v\Big] + \mathbb{E}\Big[\sum_u z_u^2\Big] \\
&= \sum_{u\neq v} \mathbb{E}[z_u]\cdot\mathbb{E}[z_v] + \sum_u 2^{-n} \\
&= 2^{-2n}(N^2 - N) + N2^{-n} = N^2 2^{-2n} + N2^{-n}(1-2^{-n}) .
\end{aligned}
$$

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

*Remark 3 (Why not $i \in (0, k/2)$).* Note that the above considers only $i \geq k/2$. 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 3.* Let $\mathbf{s} \stackrel{\mathsf{def}}{=} \mathbf{s}_1 - \mathbf{s}_2$ and $\mathbf{x} \stackrel{\mathsf{def}}{=} \mathbf{x}_1 - \mathbf{x}_2$. 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 $\{\mathbf{x} \in \mathbb{F}_2^m \colon |\mathbf{x}| \leq 2w\}$ 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 11.* We denote $\mathcal{S} \stackrel{\text{def}}{=} \mathbb{F}_2^n$ and $p_s = \Pr[h(\mathbf{r}) = s]$.

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

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

## B Inequalities, Theorems and Lemmas

**Lemma 14 (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 15 (Chebyshev's inequality).** *Let $Y$ be any random variable (taking real values) with expectation $\mu$ and standard deviation $\sigma$ (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 16 (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 $0 < \Delta_2 < 1$,*

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

**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 $\log(n!) = \log\left(O(\sqrt{n}(\frac{n}{e})^n)\right) = \frac{1}{2}\log n + n\log n - n\log e + O(1)$ and for the second one we have

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

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 17 (Sample-preserving reduction [6]).** *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 18 (Goldreich-Levin Theorem [32]).** *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 $\mathbf{r} \xleftarrow{\$} \mathbb{F}_2^n$.*