Worst and average case hardness of decoding via smoothing bounds

Thomas Debris-Alazard, Nicolas Resch

WORST AND AVERAGE CASE HARDNESS OF DECODING VIA SMOOTHING BOUNDS

THOMAS DEBRIS–ALAZARD [1,2] AND NICOLAS RESCH[3]

Abstract. In this work, we consider the worst and average case hardness of the decoding problems that are the basis for code-based cryptography. By a decoding problem, we consider inputs of the form $(\mathbf{G}, \mathbf{mG} + \mathbf{t})$ for a matrix $\mathbf{G}$ that generates a code and a noise vector $\mathbf{t}$, and the algorithm's goal is to recover $\mathbf{m}$.

We consider a natural strategy for creating a reduction to an average-case problem: from our input we simulate a *Learning Parity with Noise* (LPN) oracle, where we recall that LPN is essentially an average-case decoding problem where there is no a priori lower bound on the rate of the code. More formally, the oracle $\mathcal{O}_{\mathbf{x}}$ outputs independent samples of the form $\langle \mathbf{x}, \mathbf{a} \rangle + e$, where $\mathbf{a}$ is a uniformly random vector and e is a noise bit. Such an approach is (implicit in) the previous worst-case to average-case reductions for coding problems (Brakerski *et al* Eurocrypt 2019, Yu and Zhang CRYPTO 2021).

To analyze the effectiveness of this reduction, we use a *smoothing bound* derived recently by (Debris-Alazard *et al* IACR Eprint 2022), which quantifies the simulation error of this reduction. It is worth noting that this latter work crucially use a bound, known as *the second linear programming bounds*, on the weight distribution of the code generated here by $\mathbf{G}$. Our approach, which is Fourier analytic in nature, applies to any smoothing distribution (so long as it is *radial*); for our purposes, the best choice appears to be Bernoulli (although for the analysis it is most effective to study the uniform distribution over a sphere, and subsequently translate the bound back to the Bernoulli distribution by applying a truncation trick). Our approach works naturally when reducing from a worst-case instance, as well as from an average-case instance.

While we are unable to improve the parameters of the worst-case to average-case reductions of Brakerski *et al* or Yu and Zhang, we think that our work highlights two important points. Firstly, in analyzing the average-case to average-case reduction we run into inherent limitations of this reduction template. Essentially, it appears hopeless to reduce to an LPN instance for which the noise rate is more than inverse-polynomially biased away from uniform. We furthermore uncover a surprising weakness in the second linear programming bound: we observe that it is essentially useless for the regime of parameters where the rate of the code is inverse polynomial in the block-length. By highlighting these shortcomings, we hope to stimulate the development of new techniques for reductions between cryptographic decoding problems.

1. Introduction

Security of code-based cryptography. In the last twenty years, cryptography based on the hardness of decoding noisy linear equations has seen an explosion of interest, in part due to the powerful primitives it allows one to construct, as well as its plausible security against quantum attacks. Lattice-based cryptography, based on the presumed hardness of the *learning with errors* (LWE) problem, has been instrumental in the construction of many powerful cryptographic primitives. The maturity of these constructions is evidenced by the confidence NIST has shown in their

[1] Inria

[2] Laboratoire LIX, École Polytechnique, Institut Polytechnique de Paris, 1 rue Honoré d'Estienne d'Orves, 91120 Palaiseau Cedex

[3] University of Amsterdam, The Netherlands

E-mail addresses: thomas.debris@inria.fr, n.a.resch@uva.nl.

The work of TDA was funded by the French Agence Nationale de la Recherche through ANR JCJC COLA (ANR-21-CE39-0011).

security: in the third round of the NIST standardization process for post-quantum schemes, 3 of the 4 selected schemes were lattice-based.

A main reason why lattice-based cryptography is viewed as more mature than code-based cryptography is the preponderance of security reductions between various lattice problems. Of particular note are the worst-case to average-case reductions [Reg05, LPR10, LS15, PRS17, RSW18, PMS21], which imply that the security of these cryptographic schemes can be conjectured relying only on the worst-case hardness of well-studied lattice problems like SVP or SIS. However, until recently, no such worst-case to average-case reduction was known for coding-theoretic problems.

Motivated by this state-of-affairs, Brakerski, Lyubashevsky, Vaikuntanathan and Wichs [BLVW19] (and subsequently Yu and Zhang [YZ21]) gave a worst-case to average-case reduction for decoding problems. Namely, given an algorithm solving the average-case LPN problem they succeed in solving the worst-case decoding problems for codes. However, it is worth mentioning that in both cases, additional assumptions are required on the codes in order to derive the reductions. In [BLVW19], the code is required to be balanced (*i.e.* not only are there no non-zero codewords of weight at most d, but also no codewords of weight greater than $n-d$). The reductions in [YZ21] either require the code to be balanced, or that the code has good dual distance. Furthermore, the arguments in these papers rely on a specific choice of smoothing distribution (in the sequel, we will precisely define what we mean by a smoothing distribution).

The decoding problems. Before proceeding, we would like to discuss the decoding problems we study in this work. First, there is the *worst-case decoding problem* $\mathsf{wDP}(n,k,d,t)$, which is the following task: given a generator matrix $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ for a code $\mathcal{C} \subseteq \mathbb{F}_2^n$ of dimension k and minimum distance $\geq d$, and a vector of the form $\mathbf{xG}+\mathbf{e}$ where $\mathbf{e} \in \mathbb{F}_2^n$ has weight t, output the codeword $\mathbf{xG} \in \mathcal{C}$. We further denote by $\mathsf{wBalDP}(n,k,t,d)$ the variant where the code is assumed to be d-balanced.

The natural average-case version of this problem is what we call the *average-case decoding problem* $\mathsf{aDP}(n,k,t)$: the only difference is that $\mathbf{G}$, $\mathbf{x}$ and $\mathbf{e}$ are sampled uniformly at random from their respective domains. This is the average-case problem that is the basis for code-based cryptography. Our goal will be to understand for what parameters wDP reduces to aDP. We will also consider reductions between aDP problems, what we term an average to average-case reduction.

The *learning parity with noise* problem $\mathsf{LPN}(k,\tau)$ is quite similar to the average-case decoding problem (indeed, many works consider them equivalent); however, as we discuss below, we find that it is valuable to distinguish them. In the LPN problem, one is given access to an oracle $\mathcal{O}_{\mathbf{x},\tau}^{\mathsf{LPN}}$ holding a uniformly random secret vector $\mathbf{x} \leftarrow \mathbb{F}_2^k$ which, when queried, responds with a sample of the form $(\mathbf{a}, \langle \mathbf{a},\mathbf{x}\rangle + b)$ where $\mathbf{a} \leftarrow \mathbb{F}_2^k$ and $b \leftarrow \mathsf{Ber}(-\log_2(1-2\tau))$.[1] That is, an algorithm solving the LPN problem is given an a priori unbounded number of samples from the oracle, unlike in the aDP problem where the number of samples is fixed ahead of time. Further, note that the error in aDP does not act independently on each coordinate (as is necessarily the case in LPN). We advocate viewing the LPN as a useful intermediate problem between wDP and aDP: the reduction we analyze goes from wDP to LPN, which we complement with a straightforward reduction from LPN to aDP which does not change the noise-rate. However, we remark that a reduction from aDP to LPN which does not change the noise-rate is not obvious to us: it seems difficult to simulate the uniform sphere error vector of the aDP instance from independent Bernoulli samples.

[1] Namely $\mathbb{P}(b=1) = \frac{1}{2}\left(1-2^{\log_2(1-2\tau)}\right) = \tau$. We will explain later why we choose this, admittedly non-standard, parametrization.

Smoothing bounds. The main technical tools underlying the wDP to LPN reductions are *smoothing bounds*, which are also ubiquitous in reductions from lattice-based cryptography. Informally, a smoothing bound is a demonstration that a certain noise distribution looks uniform when the noise is reduced modulo a code, even if the distribution of the noise is quite far from uniform. More precisely, we are given a noise vector $\mathbf{r}$ distributed over the ambient space $\mathbb{F}_2^n$ and a code $\mathcal{C} \subseteq \mathbb{F}_2^n$ and we hope to show that the statistical distance between $\mathbf{r} \mod \mathcal{C}$ and the uniform distribution over $\mathbb{F}_2^n/\mathcal{C}$ is very small. We further try to ensure that the average weight of $\mathbf{r}$ is as small as possible.

We briefly explain why smoothing bounds play such an important role in reducing wDP to LPN. The general strategy is to argue that, given a worst-case input $(\mathbf{G}, \mathbf{xG} + \mathbf{e})$ one can simulate an oracle $\mathcal{O}^{\mathsf{LPN}}_{\mathbf{x},\tau'}$ for some τ' related to $\tau = -\log_2\left(1 - 2\frac{|\mathbf{e}|}{n}\right)$. The natural approach is to choose $\mathbf{r}$ according to a smoothing distribution and output $\mathbf{rG}^\top$ and $\langle \mathbf{r}, \mathbf{xG} + \mathbf{v}\rangle = \langle \mathbf{rG}^\top, \mathbf{x}\rangle + \langle \mathbf{r}, \mathbf{e}\rangle$. If we set $\mathbf{a} = \mathbf{rG}^\top$ and we pretend that $\mathbf{a}$ is uniformly random and that the bit $b = \langle \mathbf{r}, \mathbf{e}\rangle$ is independent of $\mathbf{a}$, then we would have exactly produced LPN samples; that is, we would have perfectly simulated the oracle $\mathcal{O}^{\mathsf{LPN}}_{\mathbf{x},\tau'}$ if τ' is the Bernoulli parameter of the bit $\langle \mathbf{r}, \mathbf{e}\rangle$. Of course, this is not completely true, but a smoothing bound precisely guarantees that the simulation error is not too large. Furthermore, if we can keep the expected weight of the smoothing distribution $\mathbf{r}$ small that will hopefully prevent the parameter τ' from being too large: in other words, it will allow us to reduce to the "easiest" LPN problem possible.

In [BLVW19] smoothing bounds are obtained via a Fourier-analytic argument, while in [YZ21] they are derived via the leftover hash lemma. We view our work as an effort to develop the Fourier-analytic approach more fully.

1.1. **Our Results.** We derive a worst-case to average-case reduction, under the assumption that the input for the worst-case instance defines a balanced code. Here is an informal special case of our result.

Theorem 1.1 (Special case of Theorem 7.7)**.** *Let $n, n_a \in \mathbb{N}$. Suppose there exists an algorithm solving $\mathsf{aDP}(n_a, O(n^{2/3}), 1/2 - O(1/n^{0.34}))$ with non-negligible success probability. Then there exists an algorithm solving $\mathsf{wBalDP}(n, O(n^{2/3}), O(n^{1/2}\log^2(n)), O(n^{2/3}/\log(n)))$ with negligible failure probability.*

While the noise-rate $1/2 - O(1/n^{0.34})$ might appear to be very large, we note that the average-case problem to which we reduce is a well-defined problem in the sense that with overwhelming probability there will be a unique solution. That is, at least information-theoretically $\mathsf{aDP}(n_a, O(n^{2/3}), 1/2 - O(1/n^{0.34}))$ can be solved, so the hypothesis of the reduction is not immediately vacuous. We further comment that we could reduce to a problem with noise rate *e.g.* $1/2 - O\left(2^{-n^{0.5}}\right)$, but in this case the code length of the average-case instance n_a would need to be subexponential in n for the problem to have a unique solution (in other words, it is "too" hard).

While our result is qualitatively similar to that of [BLVW19], we remark that our Fourier-analytic argument is flexible enough to apply to any smoothing distribution. In comparison, [BLVW19] only provides a meaningful argument for the random walk distribution. In particular, we are able to provide a bound for the Bernoulli distribution, essentially by relating it to the uniform distribution over a sphere.[(2)]

In our analysis of the worst-case to average-case reduction, we also observe a surprising weakness in the second linear programming bound [MRRW77], which provides the best known upper bounds

[(2)] This idea of relating the Bernoulli distribution to the uniform distribution over a sphere in this way was also used in [DDRT22].

on the number of codewords of a given weight (at least for weights close to the minimum distance of the code). In fact, [DDRT22] showed how these bounds can be used to provide extremely strong smoothing bounds for codes when the rate of the code we are smoothing is bounded away from 0. However, these bounds are completely useless when one is considering codes of rate $o(1)$, as we necessarily do in our reductions. Indeed, this setting is very natural from a cryptographic perspective, corresponding to the regime of parameters relevant for the LPN assumption. Our discussion highlights this limitation. It further demonstrates that an effective bound for the $o(1)$ rate regime could allow us to remove the (somewhat unnatural) assumption of balancedness that we are forced to employ.

We also consider average to average-case reductions, not only for its inherent interest (*i.e.* , the interest in seeing how we can trade off different parameters and keep the problem hard), but also to help us set our expectations for the strongest worst-case to average-case reduction we could hope for. In fact, the best parameters for the average-case to average-case reduction that we find are qualitatively identical to the parameters that we can achieve for the worst-case to average-case reduction. This leads us to suspect that there is an inherent limitation to the smoothing-based reductions that we, and prior works [BLVW19, YZ21], study. Quantifying a genuine barrier seems like an achievable task that we leave for further work.

1.2. **Our Techniques.** The starting point for our work is a recent paper by Debris-Alazard, Ducas, Resch and Tillich [DDRT22] which proved a new smoothing bound for codes. Their approach provably smoothes "most codes" optimally: that is, for the vast majority of matrices $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ they show that when $\mathbf{r}$ is sampled uniformly at random amongst vectors of weight $\approx w$ we have $\mathbf{r}\mathbf{G}^\top$ is statistically close to uniform, where w is the Gilbert-Varshamov (GV) bound; namely, the w satisfying $2^k = \binom{n}{w}$.[3] Alternatively, one can say that the smoothing bound works as soon as $\mathbf{r}$ has min-entropy k, which is necessary[4] for any distribution that is statistically close to the uniform distribution on $\mathbb{F}_2^k$.

Now, recall what we must do to create a reduction from a decoding problem (either wDP or aDP) to LPN: given the input $(\mathbf{G}, \mathbf{y} = \mathbf{m}\mathbf{G} + \mathbf{t})$, we must show that samples of the form $(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{y} \rangle)$ look like genuine LPN samples. That is, they should look like $(\mathbf{a}, \langle \mathbf{a}, \mathbf{m} \rangle + e)$ where $\mathbf{a}$ is uniformly random and e is a noise bit.[5] Now, the argument of [DDRT22] indeed shows that $\mathbf{r}\mathbf{G}^\top$ looks like a uniform vector $\mathbf{a}$, but it does not say anything about the bit $\langle \mathbf{r}, \mathbf{y} \rangle = \langle \mathbf{r}\mathbf{G}^\top, \mathbf{m} \rangle + \langle \mathbf{r}, \mathbf{t} \rangle$. In particular, there are dependencies between the simulation of the noise bit $\langle \mathbf{r}, \mathbf{t} \rangle$ and the simulation of the uniform vector $\mathbf{r}\mathbf{G}^\top$. Our first step is to generalize the Fourier-analytic smoothing bound argument of [DDRT22] to show that these correlations are statistically negligible once the smoothing distribution $\mathbf{r}$ is able to smooth.

We emphasize that this idea of simulating an LPN sample as we do is (to the best of our knowledge) due to [BLVW19] (although similar ideas have certainly appeared earlier in the lattice-theoretic literature). Furthermore, [BLVW19, YZ21] also both analyze this reduction by proving a smoothing bound.

The bound that we have on the simulation error involves two quantities: first, the Fourier transform of the smoothing distribution; secondly, the number of codewords from $\mathcal{C} = \{\mathbf{m}\mathbf{G} : \mathbf{m} \in \mathbb{F}_2^k\}$ of weight ℓ (summed over all $\ell = 1, \ldots, n$), and the translation of the code $\mathcal{C} + \mathbf{t}$. We are thus left

[3] We remark that in [DDRT22], what the authors refer to as smoothing a code $\mathcal{C}$ would correspond in our terminology to smoothing the code *dual* to $\mathcal{C}$. See the preliminaries for the definition of the dual code $\mathcal{C}^*$.

[4] Technically, the distribution must only be *close* to having min-entropy k; however, if one has a distribution which is close to having min-entropy $n - k$ which smoothes, then there is indeed a smoothing distribution of min-entropy $n - k$ which also smoothes.

[5] We should also rerandomize the secret/message vector $\mathbf{m}$, but we elide this detail for this discussion.

with two tasks: first, choosing the smoothing distribution in such a way that we have good control over its Fourier transform; secondly, applying bounds on the number of codewords (or, translated codewords) of a given weight. We now consider both of these tasks in turn.

Choice of smoothing distribution. Letting $\mathbf{r}$ denote a random vector sampled according to our smoothing distribution μ, to minimize the contribution from the Fourier transform of μ to our smoothing bound, it turns out that it is best to choose it so that $|\mathbf{r}|$ is as concentrated as possible[(6)]. For this reason, a seemingly optimal choice is to choose $\mathbf{r}$ to be uniformly distributed over a Hamming sphere. And, indeed, for our average-case to average-case reduction (*i.e.* when we are tasked with smoothing a random code) this choice of distribution provides a strong result.

However, this distribution has a drawback when it comes to smoothing worst-case codes: it is inherently limited to smoothing codes that are assumed to be *balanced*. To give a sense of why we cannot allow high weight codewords, suppose $\mathbf{1} \in \mathcal{C}$ where $\mathbf{1}$ denotes the all-1's vector. Thus, we have $\mathbf{xG} = \mathbf{1}$ for some non-zero $\mathbf{x} \in \mathbb{F}_2^k$. We would then have that $\langle \mathbf{rG}^\top, \mathbf{x} \rangle = \langle \mathbf{r}, \mathbf{xG} \rangle = \langle \mathbf{r}, \mathbf{1} \rangle = w \mod 2$. That is, we have an easy way of distinguishing $\mathbf{rG}^\top$ from uniform: we compute its inner-product with $\mathbf{x}$ and check if it equals $w \mod 2$. More generally, we can find distinguishing attacks when the code has a very high weight codeword, showing that $\mathbf{rG}^\top$ is not close to uniform.

Now, there is another natural choice of distribution that will not run into this problem: we could sample $\mathbf{r}$ as the concatenation of n independent Bernoulli random variables. However, this distribution is not as directly amenable to the Fourier analytic techniques for proving smoothing bounds,[(7)] and the bound is correspondingly weaker.

To rectify this, we follow an approach suggested in [DDRT22]: essentially due to the strong concentration of the Bernoulli distribution (*i.e.* the Chernoff bound) one can show that a Bernoulli distribution is statistically indistinguishable from a "truncated" version, where we sample the Bernoulli vector $\mathbf{r}$ conditioned on $|\mathbf{r}|$ being roughly its expectation $\mathbb{E}(|\mathbf{r}|)$. This truncated version can then be written as a convex combination of uniform sphere distribution all of which we can show have nice smoothing properties. The conclusion is that the original Bernoulli distribution must have effectively smoothed as well, as it is statistically indistinguishable from a distribution which smoothes. We thereby provide an effective smoothing argument for the Bernoulli distribution. This opens the door to proving a worst-case to average-case reduction for codes without this balancedness assumption; however, as we elucidate below, we are lacking appropriate bounds on the number of codewords of a given weight.

Lastly, we suspect that the flexibility offered by our Fourier-based argument, which applies to a broader class of distributions than [BLVW19], will be useful in other reductions. In particular, as mentioned in [YZ21] the Bernoulli distribution has the pleasant property of being defined independently on coordinates. We foresee future applications for our analysis in theoretical analyses of decoding problems.

Bounds on codewords (or their translates) of a given weight. The second ingredient in our smoothing bound are the quantities $N_\ell(\mathcal{C})$ and $N_\ell(\mathbf{t} + \mathcal{C} \setminus \{\mathbf{t}\})$ which respectively denote the cardinality of the following sets:

$$\{\mathbf{c} \in \mathcal{C} : |\mathbf{c}| = \ell\} \quad \text{and} \quad \{\mathbf{y} \in \mathbf{t} + \mathcal{C} \setminus \{\mathbf{t}\} : |\mathbf{y}| = \ell\} \ .$$

It is straightforward to bound these quantities for *random* codes, so we obtain in this way an analysis of our average-case to average-case reduction. However, for arbitrary codes, we need

[(6)] Here and throughout, $|\mathbf{x}|$ denotes the Hamming weight of a vector $\mathbf{x}$ defined as $\#\{i \in [\![1, n]\!] : x_i \neq 0\}$.

[(7)] Informally, the fact that the distribution is not as concentrated causes the Fourier transforms to be a bit larger.

to use fairly deep results from coding theory. In particular, we apply the best known bounds on *spherical* codes, which are codes in which (a) every codeword has a given weight ℓ, and (b) all codewords are at distance at least d apart. Note that if $\mathcal{C}$ is an $[n, k, d]$ code (namely a subspace of $\mathbb{F}_2^n$ of dimension k and minimum distance d), then in particular $\{\mathbf{c} \in \mathcal{C} : |\mathbf{c}| = \ell\}$ and $\{\mathbf{y} \in \mathbf{t} + \mathcal{C} \backslash \{\mathbf{t}\} : |\mathbf{y}| = \ell\}$ are spherical codes. We assume that the input code has minimum distance d achieving the Gilbert-Varshamov (GV) bound, which is the best minimum distance we could hope for: that is, we assume $\binom{n}{d} \approx 2^{n-k}$.

Now, for the regime of parameters relevant to us it appears that the best bound to use is the second linear programming bound [MRRW77] (see also the discussion around Lemma 1 of [ACKL05]). However, much to our surprise, when the rate of $\mathcal{C}$ is $o(1)$ the bound we derive is completely useless! In fact, we are better off upper-bounding $N_\ell(\mathcal{C})$ and $N_\ell(\mathbf{t} + \mathcal{C} \setminus \{\mathbf{t}\})$ by $2^k = |\mathcal{C}|$.[8] As we are unable to derive a stronger bound than 2^k for $N_\ell(\mathcal{C}) + N_\ell(\mathbf{t} + \mathcal{C} \setminus \{\mathbf{t}\})$, we are required to assume the worst-case code is balanced: for ℓ close to n, while it is intuitive to suspect that the number of codewords/translated codewords of weight ℓ is small enough to compensate for the commensurately larger value of $\widehat{\mu}(\ell)$,[9] we cannot actually establish this.

Unfortunately, we are inherently limited to considering codes of rate $o(1)$ in our reductions. Indeed, when considering this reduction template starting from a constant rate code, the magnitude (*i.e.* the expected weight $\mathbb{E}(|\mathbf{r}|)$) of the smoothing distribution would have to be so large that the noise rate of the bit $\langle \mathbf{r}, \mathbf{t} \rangle$ (which simulates the noise of the LPN sample) would end up being of the form $1/2 - 2^{-\Omega(n)}$. It is hopeless to solve such an LPN instance with subexponentially many samples: indeed, exponentially many samples are needed to even uniquely determine the solution!

We note these bounds are very effective when smoothing codes of constant rate: this was a main conclusion of [DDRT22]. For this reason, we found it natural to suspect that they would be useful for providing reductions between decoding problems, but unfortunately this turns out not to be the case. We hope that our work highlights this inefficiency, and motivates the challenge of improving these bounds for this "cryptographic" regime of parameters (which has not been studied in depth by the information-theoretic community).

1.3. **Organisation.** We begin in Section 2 with necessary preliminaries, and then provide separately in Section 3 the definitions of the decoding problems we study. The general framework for the reductions are provided in Section 4 while the Fourier-analytic smoothing bound generalizing [DDRT22] is provided in Section 5. We then instantiate the reduction, first providing an average to average-case reduction in Section 6, and subsequently a worst to average-case reduction in Section 7.

2. Preliminaries: Basic Notation, Linear Codes and Fourier Transform

2.1. **General Notation.** The notation $x \stackrel{\text{def}}{=} y$ means that x is defined to be equal to y. Given a set $\mathscr{S}$, its indicator function is denoted $1_{\mathscr{S}}$. For a finite set $\mathscr{S}$, we denote by $\sharp\mathscr{S}$ its cardinality. Vectors will be written with bold letters (such as $\mathbf{x}$) and matrices will be denoted by bold uppercase letters (such as $\mathbf{X}$). By default, vectors are viewed as *row* vectors. In what follows, $\mathscr{S}_t$ will denote the sphere of radius t around $\mathbf{0}$ in $\mathbb{F}_2^n$ for the Hamming weight $|\cdot|$ which is defined as

$$\forall \mathbf{x} \in \mathbb{F}_2^n, \quad |\mathbf{x}| \stackrel{\text{def}}{=} \sharp \{i \in [\![1, n]\!] : x_i \neq 0\}.$$

[8] We note that, implicitly, [BLVW19] is using the same weak bound; see display equation (6) of the proof of Lemma 3.1.

[9] That is, the Fourier transform of the smoothing distribution evaluated at any vector of weight ℓ.

The size of any sphere of radius w is $\binom{n}{w}$ and we have $\frac{1}{n}\log_2\binom{n}{w} = h(w/n)(1+o(1))$ where h denotes the binary-entropy, namely

$$h(x) \stackrel{\text{def}}{=} -x\log_2(x) - (1-x)\log_2(1-x).$$

In this article, we wish to emphasize on which probability space the probabilities or the expectations are taken. We will denote by a subscript the random variable specifying the associated probability space over which the probabilities or expectations are taken. For instance the probability $\mathbb{P}_X(E)$ of the event E is taken over Ω the probability space over which the random variable X is defined.

The *statistical distance* between two discrete probability distributions f and g over a same discrete space $\mathscr{S}$ is defined as:

$$\Delta(f,g) \stackrel{\text{def}}{=} \frac{1}{2}\sum_{x\in\mathscr{S}} |f(x) - g(x)|.$$

Furthermore, the statistical distance between two discrete random variables X, Y, of probability distribution f, g, is defined as $\Delta(X,Y) \stackrel{\text{def}}{=} \Delta(f,g)$. Statistical distance enjoys many interesting properties. Among others, it cannot increase by applying a function φ:

$$\Delta(\varphi(X),\varphi(Y)) \leq \Delta(X,Y) \quad (\textit{data processing inequality.}) \tag{1}$$

The function φ can be randomized, but its internal randomness has to be independent of X and Y for the data processing inequality to hold. In particular, it implies that the "success" probability of any algorithm $\mathscr{A}$ for inputs distributed according to X in one case or Y in the other, can only differ by at most $\Delta(X,Y)$. Furthermore, when $(X_1,\ldots,X_r)$ and $(Y_1,\ldots,Y_r)$ are two sequences of random variables such that the X_i's (respectively the Y_i's) are pairwise independent, then

$$\Delta((X_1,\ldots,X_r),(Y_1,\ldots,Y_r)) \leq \sum_{i=1}^{r}\Delta(X_i,Y_i). \tag{2}$$

In the sequel, we write $X \leftarrow \mathscr{D}$ to denote that X is a random variable distributed according to $\mathscr{D}$. Given a finite set $\mathscr{E}$, the notation $X \leftarrow \mathscr{E}$ means X is uniformly distributed over $\mathscr{E}$. A Bernoulli random variable $X \leftarrow \mathsf{Ber}(\omega)$ *of parameter* $\omega \in \mathbb{R}_+$ is any binary random variable such that

$$\mathbb{P}(X=1) = \frac{1}{2}\left(1 - 2^{-\omega}\right).$$

Finally, $X \leftarrow \mathsf{Ber}(\omega)^{\otimes n}$ means that $X \stackrel{\text{def}}{=} (X_1,\ldots,X_n)$ where the X_i's are independent and identically distributed Bernoulli random variables of parameter ω.

Remark 2.1. *Notice that when* $X \leftarrow \mathsf{Ber}(-\log_2(1-2\tau))^{\otimes n}$*, then*

$$\mathbb{E}(|X|) = \tau n.$$

This notation may seem surprising, but it is in fact more convenient for our setting. The rationale behind this choice is that in our reduction we strongly need to "focus" in the neighbourhood of $1/2$. This notation has also the following advantage: a calculation shows that given two independent random variables $X \leftarrow \mathsf{Ber}(\omega_1)$ and $Y \leftarrow \mathsf{Ber}(\omega_2)$, then $X + Y \leftarrow \mathsf{Ber}(\omega_1+\omega_2)$. More generally we have the following lemma (often called the piling-up lemma)

Lemma 2.2. *Let* $\mathbf{r} \leftarrow \mathsf{Ber}(\omega)^{\otimes n}$ *and* $\mathbf{t} \in \mathbb{F}_2^n$*. Then,*

$$\langle \mathbf{r}, \mathbf{t}\rangle \leftarrow \mathsf{Ber}(\omega|\mathbf{t}|)\ . \tag{3}$$

A distribution over $\mathbb{F}_2^n$ is called *radial* if the probability mass it assigns to a vector $\mathbf{x}$ depends only on its Hamming weight $|\mathbf{x}|$. Note that $\mathsf{Ber}(\omega)^{\otimes n}$ is indeed radial.

2.2. **Linear Codes.** In this paper we will deal exclusively with binary linear codes, namely subspaces of $\mathbb{F}_2^n$ (equipped with the Hamming weight) for some positive integer n. An $[n,k]$-code $\mathcal{C}$ is defined as a dimension k subspace of $\mathbb{F}_2^n$. The rate of $\mathcal{C}$ is $\frac{k}{n}$. Its minimal distance is given by

$$\begin{aligned} d_{\min}(\mathcal{C}) &\stackrel{\text{def}}{=} \min\{|\mathbf{c}-\mathbf{c}'| : \mathbf{c},\ \mathbf{c}' \in \mathcal{C} \text{ and } \mathbf{c} \neq \mathbf{c}'\} \\ &= \min\{|\mathbf{c}| : \mathbf{c} \in \mathcal{C} \text{ and } \mathbf{c} \neq \mathbf{0}\}. \end{aligned}$$

The number of vectors of Hamming weight t in a set $\mathscr{E} \subseteq \mathbb{F}_2^n$ will be denoted by $N_t(\mathscr{E})$, namely

$$N_t(\mathscr{E}) \stackrel{\text{def}}{=} \sharp\{\mathbf{x} \in \mathscr{E} : |\mathbf{x}| = t\}.$$

In particular, $N_t(\mathcal{C})$ denotes the number of codewords $\mathbf{c} \in \mathcal{C}$ of Hamming weight t.

To represent an $[n,k]$-code $\mathcal{C}$ we will take any basis of it, namely a set of k linearly independent vectors $\mathbf{g}_1, \ldots, \mathbf{g}_k \in \mathcal{C}$, and form the matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ whose rows are the $\mathbf{g}_i$'s. Then $\mathcal{C}$ can be written as

$$\mathcal{C} = \left\{\mathbf{x}\mathbf{G} : \mathbf{x} \in \mathbb{F}_q^k\right\}.$$

Conversely, any matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ of rank k defines a code with the previous representation. Such a matrix $\mathbf{G}$ is usually called a *generator matrix of* $\mathcal{C}$.

The dual of a code $\mathcal{C}$ is defined as $\mathcal{C}^* \stackrel{\text{def}}{=} \{\mathbf{c}^* \in \mathbb{F}_2^n : \forall \mathbf{c} \in \mathcal{C},\ \langle \mathbf{c}, \mathbf{c}^*\rangle = 0\}$ where $\langle\cdot,\cdot\rangle$ denotes the standard inner product on $\mathbb{F}_2^n$ defined as $\langle \mathbf{x}, \mathbf{y}\rangle \stackrel{\text{def}}{=} \sum_{i=1}^n x_i y_i \in \mathbb{F}_2$.

2.3. **Fourier Transform over $\mathbb{F}_2^\ell$ and $\mathbb{F}_2^n/\mathcal{C}^* \times \mathbb{F}_2$.** We will use, as a fundamental tool, the Fourier transform of functions taking their values in $\mathbb{C}$ and which are defined over the ambient space $\mathbb{F}_2^\ell$ or the coset-space $\mathbb{F}_2^n/\mathcal{C}^* \times \mathbb{F}_2$ for some $[n,k]$-code $\mathcal{C}$. We give in the sequel notation and recall basic results about the Fourier transform. Everything could have been stated directly with finite groups and the duality theory. However, we have preferred to present the Fourier transform in our particular use cases for the unfamiliar reader. This presentation will be useful only for the proof of Proposition 5.1; it can be safety skipped before reading the proof of this proposition.

Fourier Analysis over Coset-functions. We consider here functions over cosets of $\mathcal{C}^*$ (where $\mathcal{C}$ is an $[n,k]$-code), and an additional field element, namely functions $f : \mathbb{F}_2^n/\mathcal{C}^* \times \mathbb{F}_2 \to \mathbb{C}$. The scalar product between two such functions f, g is defined as

$$\langle f, g\rangle_c \stackrel{\text{def}}{=} \frac{1}{2^{k+1}} \sum_{\substack{\mathbf{x}\in\mathbb{F}_2^n/\mathcal{C}^* \\ \alpha\in\mathbb{F}_2}} f(\mathbf{x},\alpha)\overline{g(\mathbf{x},\alpha)}\ .$$

Then, the norm of a function f is defined as $|f|_{2,c} \stackrel{\text{def}}{=} \sqrt{\langle f, f\rangle_c}$. An orthonormal basis for the above scalar product is given by the following characters (one can verify they are well-defined as functions $\mathbb{F}_2^n/\mathcal{C}^* \times \mathbb{F}_2 \to \mathbb{C}$):

$$(\chi_{(\mathbf{c},\alpha)})_{(\mathbf{c},\alpha)\in\mathcal{C}\times\mathbb{F}_2^n} \quad \text{where} \quad \chi_{(\mathbf{c},\alpha)}(\mathbf{y},\beta) \stackrel{\text{def}}{=} (-1)^{\langle \mathbf{c},\mathbf{y}\rangle + \alpha\,\beta}\ .$$

The Fourier transform of $f : \mathbb{F}_2^n/\mathcal{C}^* \times \mathbb{F}_2 \to \mathbb{C}$ is the following function

$$\widetilde{f} : (\mathbf{c},\alpha) \in \mathcal{C} \times \mathbb{F}_2 \longmapsto \langle f, \chi_{(\mathbf{c},\alpha)}\rangle_c \in \mathbb{C}\ .$$

Notice that it is a function defined over $\mathcal{C} \times \mathbb{F}_2$. Then, using that characters $(\chi_{(\mathbf{c},\alpha)})_{(\mathbf{c},\alpha)\in\mathcal{C}}$ form an orthonormal basis we have the following "Fourier" decomposition

$$f = \sum_{\substack{\mathbf{c}\in\mathcal{C} \\ \alpha\in\mathbb{F}_2}} \langle f, \chi_{(\mathbf{c},\alpha)}\rangle_c\ \chi_{(\mathbf{c},\alpha)} = \sum_{\substack{\mathbf{c}\in\mathcal{C} \\ \alpha\in\mathbb{F}_2}} \widetilde{f}(\mathbf{c},\alpha)\ \chi_{\mathbf{c},\alpha}\ .$$

The Parseval formula is a simple consequence of the above decomposition, namely

$$|f|_{2,\mathrm{c}} = \sqrt{\sum_{\substack{\mathbf{c}\in\mathcal{C}\\ \alpha\in\mathbb{F}_2}} |\widetilde{f}(\mathbf{c},\alpha)|^2} \ . \tag{4}$$

We will also need the Fourier transform of functions defined over the ambient space $\mathbb{F}_2^\ell$.

Fourier Analysis over Ambient Space Functions. Let us consider now functions $f : \mathbb{F}_2^\ell \to \mathbb{C}$. The scalar product between two such functions f, g is defined as

$$\langle f, g\rangle \stackrel{\text{def}}{=} \frac{1}{2^\ell} \sum_{\mathbf{x}\in\mathbb{F}_2^\ell} f(\mathbf{x})\overline{g(\mathbf{x})} \ .$$

The Fourier transform is then simply defined as

$$\widehat{f} : \mathbf{x} \in \mathbb{F}_2^\ell \longmapsto \langle f, \chi_{\mathbf{x}}\rangle \in \mathbb{C} \quad \text{where} \quad \chi_{\mathbf{x}}(\mathbf{y}) \stackrel{\text{def}}{=} (-1)^{\langle \mathbf{x},\mathbf{y}\rangle} \ .$$

Given any function $f : \mathbb{F}_2^n \times \mathbb{F}_2 \to \mathbb{C}$, it will be useful to consider its periodization over $\mathcal{C}^* \subseteq \mathbb{F}_2^n$, which is defined as

$$\forall (\mathbf{x},\alpha) \in \mathbb{F}_2^n \times \mathbb{F}_2, \quad f^{|\mathcal{C}^*}(\mathbf{x},\alpha) \stackrel{\text{def}}{=} \frac{1}{2^{n-k}} \sum_{\mathbf{c}^*\in\mathcal{C}^*} f(\mathbf{x}+\mathbf{c}^*,\alpha) \ . \tag{5}$$

It is easily verified that $f^{|\mathcal{C}^*}$ gives a well-defined function over cosets, namely $f^{|\mathcal{C}^*} : \mathbb{F}_2^n/\mathcal{C}^* \times \mathbb{F}_2 \to \mathbb{C}$. Furthermore, a classical result asserts that periodizing over $\mathcal{C}^*$ and then applying the Fourier transform is the same as applying the Fourier transform and then considering the restriction over $\mathcal{C}$. This result is formalized in our case in the following lemma.

Lemma 2.3. *Given any function $f : \mathbb{F}_2^n \times \mathbb{F}_2 \to \mathbb{C}$, we have*

$$\widetilde{(f^{|\mathcal{C}^*})} = (\widehat{f})_{|\mathcal{C}\times\mathbb{F}_2}$$

where $(\widehat{f})_{|\mathcal{C}\times\mathbb{F}_2}$ denotes the function $\widehat{f}$ restricted to $\mathcal{C}\times\mathbb{F}_2$.

3. Decoding Problem(s)

In this work we are concerned with providing reductions between average and worst-case decoding problems. We now formally define them.

In what follows $k(n)$, $d(n)$ and $t(n)$ will be functions taking their values in $[\![0, n]\!]$. They will denote respectively the dimension, minimum distance of the considered input code and the decoding distance. To simplify notation, since n (*i.e.* the length of the considered code) is clear here from the context, we will drop the dependency on n and simply write k, d and t.

Problem 3.1 (Worst-Case Decoding Problem - $\mathsf{wDP}(n,k,t,d)$)**.**

- Input: $(\mathbf{G}, \mathbf{y} \stackrel{def}{=} \mathbf{xG}+\mathbf{e})$ *where* $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ *is a generator matrix of an* $[n,k]$*-code whose minimum distance is* d*,* $\mathbf{x} \in \mathbb{F}_2^k$*, and* $\mathbf{e} \in \mathbb{F}_2^n$ *of Hamming weight* t*.*
- Output: $\mathbf{xG}$.

The average version of this problem, which is the basis for the security of most code-based cryptosystems [McE78, Ste93, CFS01, Ale03, DST19, FJR22], is simply the case where the inputs chosen uniformly over their domain. Namely:

Problem 3.2 (Average Decoding Problem - $\mathsf{aDP}(n,k,t)$)**.**

- Input: $(\mathbf{G}, \mathbf{y} \stackrel{def}{=} \mathbf{xG}+\mathbf{e})$ *where* $\mathbf{G} \leftarrow \mathbb{F}_2^{k\times n}$*,* $\mathbf{x} \leftarrow \mathbb{F}_2^k$*, and* $\mathbf{e} \leftarrow \mathscr{S}_t$
- Output: $\mathbf{xG}$.

Remark 3.3. *One can consider a natural decisional variant of this problem, wherein the task is to distinguish samples of the form* $\mathbf{xG} + \mathbf{e}$ *(for* $\mathbf{x}$, $\mathbf{G}$ *and* $\mathbf{e}$ *sampled as above) and vectors* $\mathbf{b}$ *sampled uniformly from* $\mathbb{F}_2^n$*. It was shown by Fischer and Stern* [FS96] *that these problems are polynomially equivalent.*

Any (probabilistic) algorithm $\mathscr{A}$ is said to solve $\mathsf{aDP}(n, k, t)$ with *success probability* ε if given as input $\mathbf{G} \leftarrow \mathbb{F}_2^{k\times n}$ – viewed as the generator matrix of a code $\mathcal{C} \subseteq \mathbb{F}_2^n$ – and a noisy codeword $\mathbf{c} + \mathbf{e}$ where $\mathbf{c} \leftarrow \mathcal{C}$ and $\mathbf{e} \leftarrow \mathscr{S}_t$, it outputs with probability ε the "right" $\mathbf{c}$, namely

$$\varepsilon = \mathbb{P}_{\mathbf{G},\mathbf{c},\mathbf{e}}\left(\mathscr{A}(\mathbf{G}, \mathbf{c} + \mathbf{e}) = \mathbf{c}\right).$$

LPN: a special case of aDP. In the cryptographic literature, a problem closely related to aDP and referred to as Learning Parity with Noise (LPN) is often considered.

Definition 3.4 (LPN-oracle)**.** *Let* $k \in \mathbb{N}$, $\tau \in [0, 1/2)$ *and* $\mathbf{x} \in \mathbb{F}_2^k$*. We define the* $\mathsf{LPN}(k, \tau)$*-oracle* $\mathscr{O}_{\mathbf{x},\tau}^{\mathsf{LPN}}$ *as follows: on a call it outputs* $(\mathbf{a}, \langle \mathbf{x}, \mathbf{a}\rangle + e)$ *where* $\mathbf{a} \leftarrow \mathbb{F}_2^k$ *and* $e \leftarrow \mathsf{Ber}(-\log_2(1 - 2\tau))$*.*

Problem 3.5 (Learning with Parity Noise Problem - $\mathsf{LPN}(k, \tau)$)**.**

- Input: $\mathscr{O}_{\mathbf{x},\tau}^{\mathsf{LPN}}$ *be an* $\mathsf{LPN}(k, \tau)$*-oracle parametrized by* $\mathbf{x} \in \mathbb{F}_2^k$ *which has been chosen uniformly at random.*
- Output: $\mathbf{x}$.

It turns out that LPN is an easier problem than aDP as shown in the following proposition.

Proposition 3.6. *Suppose there exists an algorithm* $\mathscr{A}$ *solving* $\mathsf{aDP}(n, k, t)$ *with success probability* ε *and running in time* T*. Then, there exists an algorithm solving* $\mathsf{LPN}\left(k, \frac{t}{n}\right)$ *(making* n *queries) with success probability* $\Omega\left(\frac{\varepsilon}{\sqrt{t}}\right)$ *and running in time* $O(T + n)$*.*

Proof. Let $\mathscr{O}_{\mathbf{x},\tau}^{\mathsf{LPN}}$ be the input of the considered $\mathsf{LPN}(k, \tau)$. The algorithm to find $\mathbf{x}$ is as follows. First, it makes n queries to $\mathscr{O}_{\mathbf{x},\tau}^{\mathsf{LPN}}$ which outputs the sequence

$$\langle \mathbf{x}, \mathbf{a}_1\rangle + e_1, \ \ldots, \ \langle \mathbf{x}, \mathbf{a}_n\rangle + e_n.$$

These n samples are rewritten as $\mathbf{xG} + \mathbf{e}$ where columns of $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ are the $\mathbf{a}_i$'s and $\mathbf{e} \stackrel{\text{def}}{=} (e_1, \ldots, e_n)$. Notice that $\mathbf{G} \leftarrow \mathbb{F}_2^{k\times n}$ and $\mathbf{e} \leftarrow \mathsf{Ber}(-\log_2\left(1 - 2\frac{t}{n}\right))^{\otimes n}$. At last, the algorithm outputs $\mathbf{x}'\mathbf{G} - \mathscr{A}\left(\mathbf{G}, (\mathbf{x} + \mathbf{x}')\,\mathbf{G} + \mathbf{e}\right)$ with $\mathbf{x}' \leftarrow \mathbb{F}_2^k$ (from which we can easily recover $\mathbf{x}$ as $\mathbf{G}$ has full rank with overwhelming probability). Therefore, its probability of success is given by

$$\begin{aligned}
\mathbb{P}_{\mathbf{G},\mathbf{x}',\mathbf{e}}\left(\mathscr{A}(\mathbf{G}, (\mathbf{x} + \mathbf{x}')\,\mathbf{G} + \mathbf{e}) = \mathbf{x} + \mathbf{x}'\right) &= \mathbb{P}_{\mathbf{G},\mathbf{x}',\mathbf{e}}\left(\mathscr{A}(\mathbf{G}, \mathbf{x}'\mathbf{G} + \mathbf{e}) = \mathbf{x}'\right) \\
&\geq \mathbb{P}_{\mathbf{G},\mathbf{x}',\mathbf{e}}\left(\mathscr{A}(\mathbf{G}, \mathbf{x}'\mathbf{G} + \mathbf{e}) = \mathbf{x}' \mid |\mathbf{e}| = t\right)\mathbb{P}_{\mathbf{e}}\left(|\mathbf{e}| = t\right) \\
&= \varepsilon\,\mathbb{P}_{\mathbf{e}}\left(|\mathbf{e}| = t\right) \qquad (6)
\end{aligned}$$

where in the last line we used the following remark: if $\mathbf{e} \leftarrow \mathsf{Ber}(\omega)^{\otimes n}$ conditioned on $|\mathbf{e}| = t$, then $\mathbf{e}$ is uniformly distributed over words of weight t.

Now, by definition of $\mathbf{e} \leftarrow \mathrm{Ber}\left(-\log_2\left(1 - 2\frac{t}{n}\right)\right)^{\otimes n}$,

$$\mathbb{P}_{\mathbf{e}}\left(|\mathbf{e}| = t\right) = \binom{n}{t}\left(\frac{t}{n}\right)^t\left(1 - \frac{t}{n}\right)^{n-t} = \binom{n}{t}2^{-nh\left(\frac{t}{n}\right)} = \Omega\left(\frac{1}{\sqrt{t}}\right)$$

where in the last equality we used Stirling's formula. It concludes the proof by plugging this in Equation (6). □

Remark 3.7. *Proposition 3.6 shows that* LPN *is easier than* aDP*. In fact, the proof shows that* LPN *roughly corresponds to a particular average decoding instance. That is, solving* $\mathsf{LPN}(k, \tau)$ *with* N *samples amounts to solving an average decoding problem whose rate and noise rate are given by*

k/N and τ/N. But the number of samples N is a priori unlimited, therefore LPN *really amounts to solving* aDP *where the code rate can be chosen arbitrarily close to* 0.

Separate worst and average case notation. In this article, when considering reductions between decoding problems, we often need to emphasize if we are considering the worst or average case version. For this reason, in the sequel, all notation with a subscript "w" and "a" will denote respectively the worst and the average version of the decoding problem. For instance, n_w will denote the length of the input code in wDP while n_a will denote the length of random codes considered in aDP.

4. Reduction(s)

The following general theorem, inspired by [BLVW19][Theorem 4.1], shows that from any algorithm solving the average decoding problem, we can build an algorithm solving a given decoding instance, namely from $(\mathbf{G}, \mathbf{mG}+\mathbf{t})$ it recovers $\mathbf{m}$.

Theorem 4.1. *Let $n_w, t_w \in \mathbb{N}$, $\mathscr{D}$ be some radial distribution over $\mathbb{F}_2^{n_w}$ and $\mathbf{r} \leftarrow \mathscr{D}$. Suppose that (for some $\omega \in \mathbb{R}_+$)*

$$\langle \mathbf{r}, \mathbf{t} \rangle \leftarrow \mathsf{Ber}(\omega) \tag{7}$$

where $\mathbf{t}$ is an arbitrary vector of $\mathbb{F}_2^{n_w}$ such that $|\mathbf{t}| = t_w$[10]*.*
Let $n_a, t_a, k \in \mathbb{N}$ and $(\mathbf{G}, \mathbf{mG}+\mathbf{t})$ with $\mathbf{G} \in \mathbb{F}_2^{k \times n_w}$, $\mathbf{m} \in \mathbb{F}_2^k$, and $|\mathbf{t}| = t_w$. Suppose that there exists an algorithm $\mathscr{A}$ which solves $\mathsf{aDP}(n_a, k, t_a)$ with success probability ε and with n_a, t_a verifying

$$-\log_2\left(1-2\frac{t_a}{n_a}\right) = \omega \iff \frac{t_a}{n_a} = \frac{1}{2}\left(1-2^{-\omega}\right). \tag{8}$$

Then, there exists an algorithm which takes as input $(\mathbf{G}, \mathbf{mG}+\mathbf{t})$ and which outputs $\mathbf{m}$ in time $O(T+n_a)$ with probability (over its internal randomness and not the choice of $\mathbf{G}$, $\mathbf{m}$ and $\mathbf{t}$) bigger than

$$\Omega\left(\frac{\varepsilon}{\sqrt{t_a}}\right) - n_a\, \Delta\Big(\left(\mathbf{rG}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big)$$

where $\mathbf{a} \leftarrow \mathbb{F}_2^k$ and $e \leftarrow \mathsf{Ber}(\omega)$.

Proof. Let $\mathbf{y} \stackrel{\text{def}}{=} \mathbf{mG}+\mathbf{t}$. The starting point of the proof is the observation that from any input of a decoding problem we can build some LPN-oracle. Given

$$\Big(\mathbf{G}, \mathbf{y} \stackrel{\text{def}}{=} \mathbf{mG}+\mathbf{t}\Big) \in \mathbb{F}_2^{k \times n_w} \times \mathbb{F}_2^{n_w}$$

we can design the oracle $\mathscr{O}_{\mathbf{x}_0}$ (defined in Figure 1) where $\mathbf{x_0}$ has been chosen uniformly at random from $\mathbb{F}_2^k$.

Oracle $\mathscr{O}_{\mathbf{x}_0}$:
Sample: $\mathbf{r} \leftarrow \mathscr{D}$
Return: $\left(\mathbf{rG}^\top, \langle \mathbf{y}+\mathbf{x}_0\mathbf{G}, \mathbf{r}\rangle\right)$

Figure 1. Oracle $\mathscr{O}_{\mathbf{x}_0}$.

[10] Notice that $\mathbb{P}_{\mathbf{r}}(\langle \mathbf{r}, \mathbf{t}\rangle = 1)$ is only function of $|\mathbf{t}|$ and $\mathscr{D}$ as this latter distribution is radial.

Notice that oracle $\mathscr{O}_{\mathbf{x}_0}$ outputs LPN–like samples of the form:

$$\mathscr{O}_{\mathbf{x}_0} : (\mathbf{a}', \langle \mathbf{s}, \mathbf{a}' \rangle + e') \quad \text{where} \quad \begin{cases} \mathbf{s} \stackrel{\text{def}}{=} \mathbf{m} + \mathbf{x}_0 \\ \mathbf{a}' \stackrel{\text{def}}{=} \mathbf{r}\mathbf{G}^\top \\ e' \stackrel{\text{def}}{=} \langle \mathbf{t}, \mathbf{r} \rangle. \end{cases} \tag{9}$$

The random variable e' follows a Bernoulli distribution of parameter $\omega = -\log_2\left(1 - 2\frac{t_a}{n_a}\right)$ (see Equations (7) and (8)). Furthermore, $\mathbf{s}$ is uniformly random over $\mathbb{F}_2^k$ as $\mathbf{x}_0$ is. Therefore, $\mathscr{O}_{\mathbf{x}_0}$ outputs $\mathsf{LPN}\left(k, \frac{t_a}{n_a}\right)$-like samples. However notice that it does not provide genuine LPN samples (that is the reason why we said LPN-*like*) since $\mathbf{r}\mathbf{G}^\top$ is a priori not uniformly distributed and is correlated with e'.

It remains now to solve this LPN-like problem to recover $\mathbf{m}+\mathbf{x}_0$ and then $\mathbf{m}$ as $\mathbf{x}_0$ is known. Using Proposition 3.6, we turn $\mathscr{A}$, which solves $\mathsf{aDP}(n_a, k, t_a)$, into an algorithm $\mathscr{B}$ solving $\mathsf{LPN}\left(k, \frac{t_a}{n_a}\right)$ with n_a queries to its input oracle, running in time $O(T + n_a)$ and whose probability of success is a $\Omega\left(\frac{\varepsilon}{\sqrt{t_a}}\right)$. Then we feed oracle $\mathscr{O}_{\mathbf{x}_0}$ to $\mathscr{B}$. However, we cannot directly analyse its success probability as $\mathscr{O}_{\mathbf{x}_0}$ is not a valid $\mathsf{LPN}\left(k, \frac{t_a}{n_a}\right)$-oracle. Thanks to the data processing inequality (see (1)), replacing the sample $(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}\mathbf{G}^\top, \mathbf{m} + \mathbf{x}_0 \rangle + \langle \mathbf{t}, \mathbf{r} \rangle)$ by a genuine $\mathsf{LPN}\left(k, \frac{t_a}{n_a}\right)$-sample $(\mathbf{a}, \langle \mathbf{a}, \mathbf{m} + \mathbf{x}_0 \rangle + e)$ changes the probabilities by at most the additive term

$$\Delta\left(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t} \rangle\right), (\mathbf{a}, e)\right), \quad \text{where} \quad e \leftarrow \mathrm{Ber}\left(-\log_2\left(1 - 2\frac{t_a}{n_a}\right)\right).$$

Therefore, when running $\mathscr{B}$ with the oracle $\mathscr{O}_{\mathbf{x}_0}$, its probability of success will be given by its probability of success when given the ideal version of $\mathscr{O}_{\mathbf{x}_0}$, namely $\mathscr{O}^{\mathsf{LPN}}_{\mathbf{x}_0+\mathbf{m}}$ (see Definition 3.4), up to the above statistical distance times n_a which is the number of queries (see Equation (2)). This concludes the proof. □

In order to instantiate this theorem, it remains now to show how

$$\Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t} \rangle\right), (\mathbf{a}, e)\Big)$$

can be made negligible. In the next section we upper-bound (by stating a *smoothing bound*) this statistical distance by some function of the distribution of $\mathbf{r}$ and the weight distribution of $\mathcal{C} = \{\mathbf{m}\mathbf{G} : \mathbf{m} \in \mathbb{F}_2^k\}$, the code with generator matrix $\mathbf{G}$.

5. Smoothing Bound

The following proposition states our smoothing bound for the worst to average case reduction in terms of the Fourier transforms of the relevant distribution. We remark that the argument is an adaptation of a similar argument used recently in [DDRT22]. Furthermore, notice that *no* hypothesis is made on the distribution used for smoothing and on the code to smooth. In particular, our bound does not require the code to be "balanced", or apply only to a particular distribution. This is in contrast to the smoothing bound of [BLVW19], which (a) only applies to the discrete random walk [BLVW19, Definition 3.1] and (b) could not hope to give a meaningful bound unless the code is balanced.

Proposition 5.1. *Let μ denote the probability density function of $\mathbf{r} \in \mathbb{F}_2^n$, which we assume to be radial. Let $\mathbf{t} \in \mathbb{F}_2^n$ be an arbitrary vector of $\mathbb{F}_2^n$ such that $|\mathbf{t}| = t$. Suppose that*

$$\langle \mathbf{r}, \mathbf{t} \rangle \leftarrow \mathsf{Ber}(\omega)$$

for some $\omega \in \mathbb{R}_+$. Let $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ be the generator matrix of an $[n,k]$-code $\mathcal{C}$. Then,

$$\Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big) \leq 2^{n-1} \sqrt{\sum_{\ell \geq d_{\min}(\mathcal{C})} N_\ell(\mathcal{C})\widehat{\mu}(\ell)^2 + \sum_{\ell \geq 0} N_\ell(\mathcal{C} + \mathbf{t}\backslash\{\mathbf{t}\})\widehat{\mu}(\ell)^2}$$

where $\mathbf{a} \leftarrow \mathbb{F}_2^k$ and $e \leftarrow \mathsf{Ber}(\omega)$.

Proof. Let f (*resp.* g) be the probability density function of $\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right)$ (*resp.* $(\mathbf{a}, e)$). To upper-bound $\Delta(f,g)$, let us start by showing how computations can be done in $\mathbb{F}_2^n/\mathcal{C}^* \times \mathbb{F}_2$ instead of $\mathbb{F}_2^k \times \mathbb{F}_2$. It will be more suitable for the Fourier analysis. Define $h : \mathbb{F}_2^n \times \mathbb{F}_2 \to [0,1]$ as

$$h(\mathbf{x}, b) \stackrel{\text{def}}{=} \mathbb{P}_{\mathbf{r}}\left(\mathbf{r} = \mathbf{x}, \langle \mathbf{r}, \mathbf{t}\rangle = b\right) \ .$$

Notice that (see Definition 5)

$$\begin{aligned} 2^{n-k}\, h^{|\mathcal{C}^*}(\mathbf{x}, b) &= \sum_{\mathbf{c}^* \in \mathcal{C}^*} \mathbb{P}_{\mathbf{r}}\left(\mathbf{r} = \mathbf{x} + \mathbf{c}^*, \langle \mathbf{r}, \mathbf{t}\rangle = b\right) \\ &= \mathbb{P}_{\mathbf{r}}\left(\mathbf{r} - \mathbf{x} \in \mathcal{C}^*, \langle \mathbf{r}, \mathbf{t}\rangle = b\right) \\ &= \mathbb{P}_{\mathbf{r}}\left(\mathbf{r}\mathbf{G}^\top = \mathbf{x}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle = b\right) \\ &= f(\mathbf{x}\mathbf{G}^\top, b) \ . \end{aligned}$$

Using that $\mathbf{G}$ has rank k, we can map $\mathbf{u} \in \mathbb{F}_2^k$ to some $\mathbf{x}(\mathbf{u}) \in \mathbb{F}_2^n$ such that $\mathbf{x}(\mathbf{u})\mathbf{G}^\top = \mathbf{u}$. Note that $(\mathbf{x}(\mathbf{u}))_{\mathbf{u}\in\mathbb{F}_2^k}$ is a canonical set of representatives of $\mathbb{F}_2^n/\mathcal{C}^*$. In particular, $2^{n-k}h^{|\mathcal{C}^*}(\mathbf{x}(\mathbf{u}), b) = f(\mathbf{u}, b)$, where we recall that $h^{|\mathcal{C}^*}$ is naturally defined as a function with domain $\mathbb{F}_2^n/\mathcal{C}^* \times \mathbb{F}_2$ (see Equation (5)). Therefore,

$$2\ \Delta(f,g) = \sum_{\substack{\mathbf{u}\in\mathbb{F}_2^k \\ b\in\mathbb{F}_2}} |g(\mathbf{u}, b) - f(\mathbf{u}, b)| = \sum_{\substack{\mathbf{x}\in\mathbb{F}_2^n/\mathcal{C}^* \\ b\in\mathbb{F}_2}} \left|u(\mathbf{x}, b) - 2^{n-k} h^{|\mathcal{C}^*}(\mathbf{x}, b)\right|$$

where u denotes probability density function of $(\mathbf{y}, e)$ with $\mathbf{y} \leftarrow \mathbb{F}_2^n/\mathcal{C}^*$ and $e \leftarrow \mathsf{Ber}(\omega)$. Therefore, using the Cauchy-Schwarz inequality and then the Parseval identity for coset functions (see Subsection 2.3) we obtain

$$\begin{aligned} 2\Delta(f,g) &\leq \sqrt{2^{k+1}} \sqrt{\sum_{\substack{\mathbf{x}\in\mathbb{F}_2^n/\mathcal{C}^* \\ b\in\mathbb{F}_2}} \left(u(\mathbf{x}, b) - 2^{n-k}\ h^{|\mathcal{C}^*}(\mathbf{x}, b)\right)^2} \\ &= \sqrt{2^{k+1}}\ \sqrt{2^{k+1}}\ \left|u - 2^{n-k}\ h^{|\mathcal{C}^*}\right|_{2,c} \\ &= 2^{k+1} \sqrt{\sum_{\substack{\mathbf{c}\in\mathcal{C} \\ b\in\mathbb{F}_2}} \left(\widetilde{u}(\mathbf{c}) - 2^{n-k}\ \widetilde{h^{|\mathcal{C}^*}}(\mathbf{c})\right)^2} \\ &= 2^{k+1} \sqrt{\sum_{\substack{\mathbf{c}\in\mathcal{C} \\ b\in\mathbb{F}_2}} \left(\widetilde{u}(\mathbf{c}, b) - 2^{n-k}\ \widehat{h}(\mathbf{c}, b)\right)^2} \end{aligned} \tag{10}$$

where in the last line we used Lemma 2.3.

Let us compute now $\widetilde{u}$. For any $\mathbf{c} \in \mathcal{C}$ and $\alpha \in \mathbb{F}_2$, we have the following computation:

$$\begin{aligned}
\widetilde{u}(\mathbf{c},\alpha) &= \frac{1}{2^{k+1}} \sum_{\substack{\mathbf{x}\in\mathbb{F}_2^n/\mathcal{C}^* \\ \beta\in\mathbb{F}_2}} u(\mathbf{x},\beta)\chi_{(\mathbf{c},\alpha)}(\mathbf{x},\beta) \\
&= \frac{1}{2^{k+1}} \left(\sum_{\mathbf{x}\in\mathbb{F}_2^n/\mathcal{C}^*} \frac{\frac{1}{2}(1-2^{-\omega})}{2^k}\, \chi_{(\mathbf{c},\alpha)}(\mathbf{x},1) + \sum_{\mathbf{x}\in\mathbb{F}_2^n/\mathcal{C}^*} \frac{\frac{1}{2}(1+2^{-\omega})}{2^k}\, \chi_{(\mathbf{c},\alpha)}(\mathbf{x},0) \right) \\
&= \begin{cases} \frac{1}{2^{k+1}} & \text{if } (\mathbf{c},\alpha) = (\mathbf{0},0) \\ \frac{2^{-\omega}}{2^{k+1}} & \text{if } (\mathbf{c},\alpha) = (\mathbf{0},1) \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

Let us compute now $\widehat{h}$. For any $\mathbf{c} \in \mathcal{C}$ and $\alpha \in \mathbb{F}_2$,

$$\begin{aligned}
\widehat{h}(\mathbf{c},\alpha) &= \frac{1}{2^{n+1}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_2^n \\ \beta\in\mathbb{F}_2}} h(\mathbf{y},\beta)\chi_{(\mathbf{c},\alpha)}(\mathbf{y},\beta) \\
&= \frac{1}{2^{n+1}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_2^n \\ \beta\in\mathbb{F}_2}} \mathbb{P}_{\mathbf{r}}\left(\mathbf{r}=\mathbf{y}, \langle \mathbf{r},\mathbf{t}\rangle = \beta\right) (-1)^{\langle \mathbf{c},\mathbf{y}\rangle} (-1)^{\alpha\beta} \\
&= \frac{1}{2^{n+1}} \sum_{\substack{\beta\in\mathbb{F}_2 \\ \mathbf{y}\in\mathbb{F}_2^n \,:\, \langle \mathbf{y},\mathbf{t}\rangle = \beta}} \mathbb{P}_{\mathbf{r}}\left(\mathbf{r}=\mathbf{y}\right) (-1)^{\langle \mathbf{c},\mathbf{y}\rangle} (-1)^{\alpha\beta} \\
&= \frac{1}{2^{n+1}} \sum_{\beta\in\mathbb{F}_2} (-1)^{\alpha\beta}\, \frac{1}{2} \left(\sum_{\mathbf{y}\in\mathbb{F}_2^n} \mathbb{P}_{\mathbf{r}}(\mathbf{r}=\mathbf{y})(-1)^{\langle \mathbf{c},\mathbf{y}\rangle} + (-1)^{\beta} \sum_{\mathbf{y}\in\mathbb{F}_2^n} \mathbb{P}_{\mathbf{r}}(\mathbf{r}=\mathbf{y})(-1)^{\langle \mathbf{y},\mathbf{t}\rangle} (-1)^{\langle \mathbf{c},\mathbf{y}\rangle} \right) \\
&= \frac{1}{4} \sum_{\beta\in\mathbb{F}_2} (-1)^{\alpha\beta} \left(\widehat{\mu}(\mathbf{c}) + (-1)^{\beta} \widehat{\mu}(\mathbf{c}+\mathbf{t}) \right) . \qquad (11)
\end{aligned}$$

In particular,

$$\widehat{h}(\mathbf{0},0) = \frac{1}{4} \sum_{\beta\in\mathbb{F}_2} \left(\widehat{\mu}(\mathbf{0}) + (-1)^{\beta} \widehat{\mu}(\mathbf{t}) \right) = \frac{1}{2}\, \widehat{\mu}(\mathbf{0}) = \frac{1}{2^{n+1}}$$

where in the last line we used that μ is a probability distribution. Therefore,

$$2^{n-k}\, \widehat{h}(\mathbf{0},0) = \frac{1}{2^{k+1}} = \widetilde{u}(\mathbf{0},0) \ . \qquad (12)$$

Furthermore,

$$\begin{aligned}
\widehat{h}(\mathbf{0},1) &= \frac{1}{4}\sum_{\beta\in\mathbb{F}_2}\left((-1)^\beta\widehat{\mu}(\mathbf{0})+\widehat{\mu}(\mathbf{t})\right)\\
&= \frac{1}{2}\,\widehat{\mu}(\mathbf{t})\\
&= \frac{1}{2^{n+1}}\sum_{\mathbf{y}\in\mathbb{F}_2^n}\mu(\mathbf{y})(-1)^{\langle\mathbf{y},\mathbf{t}\rangle}\\
&= \frac{1}{2^{n+1}}\left(\sum_{\mathbf{y}\in\mathbb{F}_2^n}\mu(\mathbf{y}) - 2\sum_{\substack{\mathbf{y}\in\mathbb{F}_2^n\\ \langle\mathbf{y},\mathbf{t}\rangle=1}}\mu(\mathbf{y})\right)\\
&= \frac{1}{2^{n+1}}\left(1-2\,\mathbb{P}_{\mathbf{r}}\left(\langle\mathbf{r},\mathbf{t}\rangle=1\right)\right)\\
&= \frac{2^{-\omega}}{2^{n+1}}\ .
\end{aligned}$$

Therefore

$$2^{n-k}\,\widehat{h}(\mathbf{0},1) = \frac{2^{-\omega}}{2^{k+1}} = \widetilde{u}(\mathbf{0},1)\ . \tag{13}$$

Plugging Equations (12), (13) and (11) in (10), we obtain

$$\begin{aligned}
2\Delta(f,g) &\le 2^{k+1}\sqrt{\sum_{\substack{\mathbf{c}\in\mathcal{C}\setminus\{\mathbf{0}\}\\ \alpha\in\mathbb{F}_2}}\left(2^{n-k-2}\sum_{\beta\in\mathbb{F}_2}(-1)^{\alpha\beta}\left(\widehat{\mu}(\mathbf{c})+(-1)^\beta\widehat{\mu}(\mathbf{c}+\mathbf{t})\right)\right)^2}\\
&= 2^n\sqrt{\sum_{\mathbf{c}\in\mathcal{C}\setminus\{\mathbf{0}\}}\widehat{\mu}(\mathbf{c})^2+\sum_{\mathbf{c}\in\mathcal{C}\setminus\{\mathbf{0}\}}\widehat{\mu}(\mathbf{c}+\mathbf{t})^2}\\
&= 2^n\sqrt{\sum_{\ell\ge d_{\min}(\mathcal{C})}N_\ell(\mathcal{C})\widehat{\mu}(\ell)^2+\sum_{\ell\ge 0}N_\ell(\mathcal{C}+\mathbf{t}\backslash\{\mathbf{t}\})\widehat{\mu}(\ell)^2}
\end{aligned}$$

which concludes the proof. □

6. Instantiating the Reduction: Average-to-Average Case

Armed with our smoothing bound from Proposition 5.1 and the general template for reductions from Theorem 4.1, we consider in the following two sections specific choices for the noise distribution and derive reductions between decoding problems.

Our upper-bound of Proposition 5.1 involves the weight distribution of the code $\mathcal{C}$ and its translations by $\mathbf{t}$, namely $(N_\ell(\mathcal{C}))_{\ell\ge d_{\min}(\mathcal{C})}$ and $(N_\ell(\mathbf{t}+\mathcal{C}\setminus\{\mathbf{t}\}))_{\ell\ge 0}$. To understand how our bound behaves for a given distribution μ, we will start with the case of *random* codes as in [DDRT22]. The expected values for the above N_ℓ is well known in this case. This will lead us to estimate our bound for almost all codes and gives us some hints about the best distribution to choose for our smoothing bound in the worst case. Furthermore this gives a sense of the best sorts of trade-off between parameters that we can achieve with the reduction. In that case Theorem 4.1 becomes:

Theorem 6.1. *Let $n_w, t_w \in \mathbb{N}$, $\mathscr{D}$ be some radial distribution over $\mathbb{F}_2^{n_w}$ and $\mathbf{r}\leftarrow\mathscr{D}$. Suppose that (for some $\omega\in\mathbb{R}_+$) $\langle\mathbf{r},\mathbf{t}\rangle\leftarrow\mathsf{Ber}(\omega)$ where $\mathbf{t}$ is an arbitrary vector of $\mathbb{F}_2^{n_w}$ such that $|\mathbf{t}|=t_w$.*

Suppose that there exists an algorithm $\mathscr{A}$ which solves $\mathsf{aDP}\left(n_a,k,t_a\stackrel{def}{=}\frac{n_a}{2}(1-2^{-\omega})\right)$ in time T with success probability ε. Then, there exists an algorithm which solves $\mathsf{aDP}(n_w,k,t_w)$ in time

$O(T + n_a)$ *with probability at least*

$$\left(\Omega\left(\frac{\varepsilon}{\sqrt{t_a}}\right) - n_a\, \mathbb{E}_{\mathbf{G}}\left(\Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big)\right)\right)\left(1 - 2^{-\Omega(n_w - k)}\right).$$

where $\mathbf{a} \leftarrow \mathbb{F}_2^k$, $e \leftarrow \mathsf{Ber}(\omega)$ *and* $\mathbf{G}$ *is picked uniformly at random among* $k \times n_w$ *binary matrices of rank* k.

Proof. By Theorem 4.1, there exists an algorithm which takes as input $(\mathbf{G}, \mathbf{m}\mathbf{G} + \mathbf{t})$ (where $\mathbf{G} \in \mathbb{F}_2^{k \times n_w}$ with rank k and $|\mathbf{t}| = t_w$) and which outputs $\mathbf{m}$ in time $O\left(T + n_a\right)$ with probability (over its internal randomness and not the choice of $\mathbf{G}$, $\mathbf{m}$ and $\mathbf{t}$) bigger than $\Omega\left(\frac{\varepsilon}{\sqrt{t_a}}\right) - n_a\ \Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big)$ where $\mathbf{a} \leftarrow \mathbb{F}_2^k$ and $e \leftarrow \mathsf{Ber}(\omega)$. Therefore, its success probability when $\mathbf{G}$ and $\mathbf{t}$ are uniformly distributed over $k \times n$ matrices of rank k and $\mathscr{S}_{t_w}$ is bigger than

$$\begin{aligned}\mathbb{E}_{\mathbf{G},\mathbf{t}}\left(\Omega\left(\frac{\varepsilon}{\sqrt{t_a}}\right) - n_a\ \Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big)\right) &= \Omega\left(\frac{\varepsilon}{\sqrt{t_a}}\right) - n_a\, \mathbb{E}_{\mathbf{G},\mathbf{t}}\left(\Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big)\right) \\ &= \Omega\left(\frac{\varepsilon}{\sqrt{t_a}}\right) - n_a\, \mathbb{E}_{\mathbf{G}}\left(\Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big)\right)\end{aligned}$$

where in the last equality we used that $\langle \mathbf{r}, \mathbf{t}\rangle$ is only function of $|\mathbf{t}| = t_w$ and the radial distribution $\mathscr{D}$ over $\mathbf{r}$. To conclude the proof, we must change the distribution of $\mathbf{G} \in \mathbb{F}_2^{k \times n_w}$ from being uniformly distributed over rank k matrices by the uniform distribution over $\mathbb{F}_2^{k \times n_w}$ (to obtain the correct input distribution for aDP). But the probability that $\mathbf{G}$ is not full rank is a $2^{-\Omega(n_w - k)}$ which concludes the proof. □

In what follows, we use the following probabilistic model $\mathcal{C}_{n,k}$ for sampling $[n, k]$-codes. We sample uniformly at random $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ conditioned on having rank k. This defines a generator matrix for a code $\mathcal{C} = \{\mathbf{m}\mathbf{G} : \mathbf{m} \in \mathbb{F}_2^k\}$.

Lemma 6.2. *For* $\mathcal{C}$ *chosen according to* $\mathcal{C}_{n,k}$

$$\forall \ell > 0,\ \mathbb{E}_{\mathbf{G}}\left(N_\ell(\mathcal{C})\right) = \frac{(2^k - 1)\binom{n}{\ell}}{2^n} \quad \textit{and} \quad \forall \ell \geq 0,\ \mathbb{E}_{\mathbf{G}}\left(N_\ell(\mathbf{t} + \mathcal{C} \setminus \{\mathbf{t}\})\right) = \frac{(2^k - 1)\binom{n}{\ell}}{2^{n-k}}.$$

The proof of this lemma follows from classical facts about random codes. For the sake of completeness, this proof can be found in Appendix A.

The above estimation combined with Proposition 5.1 enables us to upper-bound the expectation (over $\mathcal{C}$) of the statistical distance.

Proposition 6.3. *Let* μ *denote the probability density function of* $\mathbf{r} \in \mathbb{F}_2^n$. *Let* $\mathbf{t} \in \mathbb{F}_2^n$ *be an arbitrary vector of* $\mathbb{F}_2^n$ *such that* $|\mathbf{t}| = t$. *Suppose that* $\langle \mathbf{r}, \mathbf{t}\rangle \leftarrow \mathsf{Ber}(\omega)$ *for some* $\omega \in \mathbb{R}_+$. *Let* $\mathbf{G}$ *be any generator matrix of* $\mathcal{C}$ *which is drawn according to* $\mathcal{C}_{n,k}$. *Then*

$$\mathbb{E}_{\mathbf{G}}\left(\Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big)\right) \leq 2^n \sqrt{\sum_{\ell \geq 0} \frac{\binom{n}{\ell}}{2^{n-k}}\, \widehat{\mu}(\ell)^2}. \tag{14}$$

where $\mathbf{a} \leftarrow \mathbb{F}_2^k$ *and* $e \leftarrow \mathsf{Ber}(\omega)$.

Proof. By using Proposition 5.1, we obtain:

$$\begin{aligned}\mathbb{E}_{\mathcal{C}}\left(\Delta\Big(\big(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\big), (\mathbf{a}, e)\Big)\right) &\leq \mathbb{E}_{\mathcal{C}}\left(2^{n-1}\sqrt{\sum_{\ell \geq d_{\min}(\mathcal{C})} N_\ell(\mathcal{C})\widehat{\mu}(\ell)^2 + \sum_{\ell \geq 0} N_\ell(\mathcal{C}+\mathbf{t}\backslash\{\mathbf{t}\})\widehat{\mu}(\ell)^2}\right)\\ &\leq 2^{n-1}\sqrt{\mathbb{E}_{\mathcal{C}}\left(\sum_{\ell \geq d_{\min}(\mathcal{C})} N_\ell(\mathcal{C})\widehat{\mu}(\ell)^2 + \sum_{\ell \geq 0} N_\ell(\mathcal{C}+\mathbf{t}\backslash\{\mathbf{t}\})\widehat{\mu}(\ell)^2\right)}\\ &= 2^n\sqrt{\sum_{\ell \geq 0}\frac{\binom{n}{\ell}}{2^{n-k}}\,\widehat{\mu}(\ell)^2}\end{aligned}$$

where in the second inequality we used Jensen's inequality and in the last line we used the linearity of the expectation and Lemma 6.2. It concludes the proof. □

Notice that our upper-bound is agnostic to the specific choice of error distribution μ, allowing us to apply it with different error distributions and compare the results. To compare different (radial) distributions, we advocate parametrizing them by the expected weight. That is, we quantify the magnitude of a vector $\mathbf{r}$ by $r = \mathbb{E}(|\mathbf{r}|)$. We consider a smoothing bound to be more effective for the reduction if for the smoothed distribution to be close to uniform we require a smaller lower-bound on r. Indeed it will imply that the noise $\langle \mathbf{r}, \mathbf{t}\rangle$ will be smaller for this choice (i.e., if $\langle \mathbf{r}, \mathbf{t}\rangle \leftarrow \mathsf{Ber}(\omega)$, it will imply we can choose ω smaller), leading us to reduce to an "easier" average decoding problem.

We will investigate two natural choices of distribution:

- $\mu_{\mathrm{Ber},\rho}$, the Bernoulli distribution $\mathsf{Ber}(\rho)^{\otimes n_w}$:

$$\forall \mathbf{x} \in \mathbb{F}_2^{n_w}, \quad \mu_{\mathrm{Ber},\rho}(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{2^{n_w}}\,(1-2^{-\rho})^{|\mathbf{x}|}\,(1+2^{-\rho})^{n_w-|\mathbf{x}|}.$$

- μ_r, the uniform distribution over the Hamming sphere of radius r in $\mathbb{F}_2^{n_w}$:

$$\forall \mathbf{x} \in \mathbb{F}_2^{n_w}, \quad \mu_r(x) \stackrel{\text{def}}{=} \frac{1_{\mathscr{S}_r}}{\binom{n_w}{r}}.$$

Let us start with the case of the Bernoulli distribution $\mu_{\mathrm{Ber},\rho}$, as this case requires the simplest computations (and which may seem more natural for smoothing).

6.1. **Bernoulli distribution.** A computation shows that $\mu_{\mathrm{Ber},\rho}(\mathbf{x}) = \frac{1}{2^{n_w}}2^{-\rho|\mathbf{x}|}$. The bound in in (14) therefore becomes

$$2^{n_w}\sqrt{\sum_{\ell=0}^{n_w}\frac{\binom{n_w}{\ell}}{2^{n_w-k}}\,2^{-2n_w}\,2^{-2\rho\ell}} = \sqrt{\frac{1}{2^{n_w-k}}\,(1+2^{-2\rho})^{n_w}}.$$

In order to guarantee that this bound is negligible, we have to choose ρ such that

$$\rho > \rho^\star \stackrel{\text{def}}{=} -\frac{1}{2}\log_2\left(2^{1-k/n_w}-1\right). \tag{15}$$

Notice that $\rho^\star$ has two behaviours according to $k/n_w = \Theta(1)$ or $k/n_w = o(1)$. More precisely,

- If $k/n_w = \Theta(1)$, then

$$\rho^\star = \Theta(1). \tag{16}$$

- If $k/n_w = o(1)$, then

$$\rho^\star = -\frac{1}{2}\log_2\left(2\left(1-k/n_w(\ln(2)+o(1))\right)-1\right) = k/n_w(1+o(1)). \tag{17}$$

We now analyze the noise of the average decoding problem to which we reduce. That is, we look at the parameter ω for which $\langle \mathbf{r}, \mathbf{t} \rangle \leftarrow \mathsf{Ber}(\omega)$ where $|\mathbf{t}| = t_w$ and $\mathbf{r} \leftarrow \mathsf{Ber}(\rho^\star)^{\otimes n_w}$. According to Lemma 2.2, $\omega = \rho^* t_w$, *i.e.*,

$$\langle \mathbf{r}, \mathbf{t} \rangle \leftarrow \mathsf{Ber}(\rho^\star t_w). \tag{18}$$

We split the analysis into three cases, depending on the regime of the parameters n_w, t_w and k.

First Case. We first consider the following parameter regime:

$$\frac{k}{n_w} = \Theta(1) \text{ and } \frac{t_w}{n_w} = \Theta(1) \, . \tag{19}$$

Informally, this corresponds to the cast that the worst case decoding problem $\mathsf{wDP}(n_w, k, t_w, d)$ is as difficult as possible. Unfortunately, according to Equations (16) and (18), we have $\langle \mathbf{r}, \mathbf{t} \rangle \leftarrow \mathsf{Ber}(\Theta(n_w))$. Therefore, we reduce this problem to an average decoding problem with noise rate, *i.e.* $\frac{t_a}{n_a} = \mathbb{E}_{\mathbf{r}} \left(\langle \mathbf{t}, \mathbf{r} \rangle \right)$, *exponentially* close to $1/2$. To our dissatisfaction, this problem is thus extremely hard: in particular, n_a would have to be exponential in n_w for aDP to have a unique solution.

One way to mitigate this is to choose t_w as a $o(n_w)$; this subsequently decreases the noise rate of $\langle \mathbf{r}, \mathbf{t} \rangle$. We investigate this setting now.

Second Case. Let us choose now consider this parameter regime:

$$\frac{k}{n_w} = \Theta(1) \text{ and } \frac{t_w}{n_w} = o(1). \tag{20}$$

Therefore, as $\langle \mathbf{r}, \mathbf{t} \rangle \leftarrow \mathsf{Ber}(\Theta(t_w))$, it seems that we need to choose t_w as a $O(\log_2(n_w))$ to reach a noise rate $\frac{t_a}{n_a} = 1/2 - 1/\mathsf{poly}(n_w)$ in the average decoding problem. However, this choice is a real disaster for the reduction: decoding a code of length n_w at distance $O\left(\log_2(n_w)\right)$ can be done in polynomial time (using for instance Prange algorithm [Pra62]). That is, we would be reducing an *easy* worst-case problem to the average-case decoding problem; this says nothing about the hardness of the average-case problem.

We therefore conclude that the only way to reach an error rate $1/2 - 1/\mathsf{poly}(n_w)$ is to decrease $\rho^\star$ as given in Equation (15). In particular, we are led to choose $k/n_w = o(1)$ for which $\rho^\star \approx k/n_w$.

Third Case. At last, let us consider the case

$$\frac{k}{n_w} = o(1). \tag{21}$$

Using Equation (17) we obtain $\langle \mathbf{r}, \mathbf{t} \rangle \leftarrow \mathrm{Ber}\left(t_w \, \frac{k}{n_w}(1 + o(1)) \right)$. It means that we need (to reach the noise rate $1/2 - 1/\mathsf{poly}(n_w)$) to choose parameters as

$$t_w \, \frac{k}{n_w} = O\left(\log_2(n_w)\right). \tag{22}$$

Unfortunately, this choice of parameters is also dramatic for the reduction. Notice that necessarily $t_w = o(n_w)$ as otherwise k will be too small, enabling an exhaustive search to decode in polynomial time. But decoding a code (with Prange's algorithm [Pra62]) of length n_w, dimension k_w at distance $t_w = o(n_w)$ costs in first approximation (see [CS16])

$$2^{\Theta\left(t_w \frac{k_w}{n_w}\right)}$$

which is polynomial when parameters verify Equation (22).

In conclusion, it seems impossible to use in the reduction the Bernoulli distribution with our smoothing bound and reach noise rate $1/2 - 1/\mathsf{poly}(n_w)$. However notice that this difficulty arises from the fact that we have shown that the Bernoulli distribution only smoothes at $\rho^\star \approx k/n_w$, which is not small enough. Fortunately, some recent work [DDRT22] has precisely shown that the Bernoulli smoothes at a much smaller distance. To show this, [DDRT22] suggested considering instead the uniform distribution over a sphere. We analyze this choice in the following subsection.

Furthermore, by using a "truncating argument", [DDRT22] has shown how to relate the uniform distribution over a sphere to the Bernoulli distribution, essentially showing that they smoothe equally well. We will follow this procedure in Subsection 6.3.

6.2. **Uniform Distribution over the Hamming sphere of radius r.** Recall that $\mu_r = \frac{1_{\mathscr{S}_r}}{\binom{n_w}{r}}$ denotes the uniform distribution over the sphere of radius r. The Fourier transform of this distribution involves Krawtchouk polynomials.

We denote $K_r(X) \stackrel{\text{def}}{=} \sum_{j=0}^{r} (-1)^j \binom{X}{j}\binom{n-X}{r-j}$ the Krawtchouk polynomial of order r. The following fact allows to relate K_r with $\widehat{\mu_r}$ (see for instance [vL99, Lem. 3.5.1, §3.5])

Fact 6.4. *For any $\mathbf{t} \in \mathscr{S}_t$,*

$$\sum_{\mathbf{x}\in\mathscr{S}_r} (-1)^{\langle \mathbf{t},\mathbf{x}\rangle} = K_r(t). \tag{23}$$

We deduce that $\widehat{\mu_r}(\ell) = 2^{-n_w} \frac{K_r(\ell)}{\binom{n_w}{r}}$. The bound in (14) therefore becomes

$$2^{n_w} \sqrt{\sum_{\ell=0}^{n_w} \frac{\binom{n_w}{\ell}}{2^{n_w-k}}\, 2^{-2n_w} \frac{K_r(\ell)^2}{\binom{n_w}{r}^2}} = \sqrt{\frac{2^k}{\binom{n_w}{r}^2} \sum_{\ell=0}^{n_w} \frac{\binom{n_w}{\ell}}{2^{n_w}} K_r(\ell)^2}\ .$$

Applying the fact that $\left(K_r/\sqrt{\binom{n_w}{r}}\right)_{0\le r\le n}$ forms an orthonormal basis of radial functions with respect to the binomial measure [Lev95, Corollary 2.3], the above sum equals

$$\sqrt{\frac{2^k}{\binom{n_w}{r}}}\ . \tag{24}$$

In order to guarantee that the above is negligible we need to choose r so that $h(r/n_w) = (1+\eta)k/n_w$ for any arbitrarily small constant $\eta > 0$, *i.e.* so that r/n_w is greater than the relative Gilbert-Varshamov (GV) bound of codes of dimension $n_w - k$, namely

$$\rho^*_{\mathrm{GV}} \stackrel{\text{def}}{=} h^{-1}\left(\frac{k}{n_w}\right)$$

with $h^{-1} : [0,1] \to [0,1/2]$ being the inverse of the binary entropy function h.

Remark 6.5. *This value of radius $n_w\rho^*_{\mathrm{GV}}$ is optimal: clearly, the support size of an error distribution ensuring that $\mathbf{r}\mathbf{G}^\top$ is uniform, where $\mathbf{G} \in \mathbb{F}_2^{k\times n_w}$, must exceed $\sharp\mathbb{F}_2^k$. Thus, we cannot expect to smooth with errors in the sphere $\mathscr{S}_r$ if its volume is smaller than 2^k.*

Let us now assume that $\mathsf{Ber}(\rho)^{\otimes n_w}$ is well approximated by the uniform distribution over a sphere of radius $\frac{n}{2}(1-2^{-\rho})$ (it will be stated formally rigorously proved in Proposition 6.6). In that case the above discussion has shown that to smooth it is enough to choose ρ as

$$\rho = -\log_2(1-2\rho^*_{\mathrm{GV}}) \iff \frac{1}{2}\left(1-2^{-\rho}\right) = \rho^*_{\mathrm{GV}}$$

As above, if one wants a noise rate $\langle \mathbf{r},\mathbf{t}\rangle$ to be some $1/2 - 1/\mathsf{poly}(n_w)$, one as to choose parameters such that $\rho = -\log_2(1-2\rho^*_{\mathrm{GV}}) = o(1)$, in particular $k/n_w = o(1)$ as $\rho^*_{\mathrm{GV}} = h^{-1}(k/n_w)$ and

$h^{-1}(0) = 0$. Therefore in that case, it is enough for smoothing to choose ρ as

$$2\rho^*_{\mathrm{GV}}(1+o(1)) = \frac{2}{\log_2\left(\frac{n_w}{k}\right)}\frac{k}{n_w}(1+o(1))$$

where we used the expansion $h^{-1}(\varepsilon) \underset{\varepsilon\to 0}{=} \varepsilon/\log_2(1/\varepsilon)(1+o(1))$.

Observe that the noise rate we obtain is much smaller than before. To reach the noise rate $1/2 - 1/\mathsf{poly}(n_w)$ we need to choose parameters such that

$$\frac{k}{n_w} = o(1) \quad \text{and} \quad \frac{1}{\log_2\left(\frac{n_w}{k}\right)}\frac{k}{n_w}\, t_w = O\left(\log_2(n_w)\right) \ . \tag{25}$$

If one compares this to Equation (22), there is an extra $1/\log_2(n_w/k_w)$ term, enabling us to choose a larger t_w for which Equation (25) is satisfied. We are thus able to reach a noise rate $1/2 - 1/\mathsf{poly}(n_w)$ with a larger t_w.

The above choice of parameters for the reduction is this time non-trivial. In that case the cost of Prange's algorithm [Pra62] is given by

$$2^{\Theta\left(t_w \frac{k_w}{n_w}\right)} = 2^{\Theta(\log_2(n_w)\log_2(n_w/k))} = n_w^{\Theta(\log_2(n_w/k))}$$

which is super-polynomial as $k/n_w = o(1)$.

6.3. **From the uniform distribution over a sphere to the Bernoulli distribution.** The above discussion has shown that applying directly our proof template with the Bernoulli does not lead to a "useful" reduction contrary to the uniform distribution over a sphere. It may seem surprising at first sight but it arises from the fact its Fourier transform is not as concentrated as that of the uniform sphere distribution. However, one can notice, by using Chernoffs bound, that a Bernoulli is concentrated on vectors whose weight lies in a width εn interval around its expected weight. Therefore, outside of this interval, the contribution of the Bernoulli to the statistical distance is negligible. Taking advantage of this, one can show that a Bernoulli distribution has the same smoothing properties as the uniform distribution over a sphere. More precisely we have the following proposition, which is a variation of [DDRT22, Prosposition 3.9].

Proposition 6.6. *Let $\mathbf{t} \in \mathbb{F}_2^n$, $\varepsilon > 0$, $\rho \in \mathbb{R}_+$ and $p \overset{def}{=} \frac{1}{2}\left(1 - 2^{-\rho}\right)$. Let $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ be the generator matrix of an $[n,k]$-code. Then,*

$$\Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big) \leq \sum_{r=(1-\varepsilon)np}^{(1+\varepsilon)np} \Delta\Big(\left(\mathbf{r}_r\mathbf{G}^\top, \langle \mathbf{r}_r, \mathbf{t}\rangle\right), (\mathbf{a}, e_r)\Big) + 2^{-\Omega(n)}$$

where $\mathbf{r} \leftarrow \mathsf{Ber}(\rho)^{\otimes n}$, $\mathbf{a} \leftarrow \mathbb{F}_2^k$, $e \leftarrow \mathsf{Ber}(\rho|\mathbf{t}|)$, $\mathbf{r}_r \leftarrow \mathscr{S}_r$ and the e_r's are distributed as the $\langle \mathbf{r}_r, \mathbf{t}\rangle$'s.

Proof. See Appendix B. □

In other words, the Bernoulli inherits the smoothing properties of the uniform distribution over a sphere. We can thus fruitfully apply the Bernoulli distribution and obtain an effective average-to-average case reduction.

Proposition 6.7. *Let $\mathbf{t} \in \mathbb{F}_2^{n_w}$, $\varepsilon, \eta > 0$ and $\rho \in \mathbb{R}_+$ be such that $(1-\varepsilon)\frac{1}{2}(1-2^{-\rho}) \geq (1+\eta)h^{-1}\left(\frac{k}{n_w}\right)$. Let $\mathbf{r} \leftarrow \mathsf{Ber}(\rho)$, $\mathbf{a} \leftarrow \mathbb{F}_2^k$, $e \leftarrow \mathsf{Ber}(\rho|\mathbf{t}|)$ and $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ be sampled uniformly at random among matrices of size $k \times n_w$ and rank k. Then,*

$$\mathbb{E}_{\mathbf{G}}\left(\Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big)\right) = 2^{-\Omega(n_w)} \ .$$

The same result holds if $\mathbf{r}$ is instead uniformly distributed over a sphere of radius $n_w(1+\eta)h^{-1}\left(\frac{k}{n_w}\right)$.

Proof. Let us show that the result holds for $\mathbf{r}_r \leftarrow \mu_r$ where μ_r denotes the uniform distribution over a sphere of radius $(1+\eta)h^{-1}\left(\frac{k}{n_w}\right) \leq r \leq 1/2$. To conclude the proof it will just remain to use Proposition 6.6.

We have shown in Equation (24) that

$$\mathbb{E}_{\mathbf{G}}\left(\Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big)\right) \leq \sqrt{\frac{2^k}{\binom{n_w}{r}}}$$

Recall that $\binom{n}{r} = 2^{n_w h(r/n)(1+o(1))}$ where h denotes the binary entropy. But $(1+\eta)h^{-1}\left(\frac{k}{n_w}\right) \leq r \leq 1/2$ and h is an increasing function. Therefore the above upper-bound is a $2^{-\Omega(n_w)}$. This concludes the proof. □

6.4. **Instantiating the distribution.** We are now ready to instantiate the reduction in the average-to-average case. We propose a set of parameters to reduce an average decoding problem of super-polynomial hardness into an average decoding problem whose noise rate is given by some $1/2 - 1/\mathsf{poly}(n_w)$. We were motivated by the fact it corresponds to the choice of parameters for which the decoding problem into which we reduce is the easiest as possible. Furthermore, this result is insightful as it gives (up to some constant) parameters for which the worst-to-average case reduction holds.

Theorem 6.8. *Let $\varepsilon, \eta \in (0,1)$, $C > 0$ and $n_w, k, t_w \in \mathbb{N}$ be such that (for some constant)*

$$\frac{k}{n_w} = o(1) \quad \text{and} \quad \frac{2}{\ln(2)}\,\frac{1+\eta}{1-\varepsilon}\,\frac{1}{\log_2\left(\frac{n_w}{k}\right)}\,\frac{k}{n_w}\,t_w = C\log_2(n_w) \tag{26}$$

Suppose that there exists an algorithm $\mathscr{A}$ which solves $\mathsf{aDP}\left(n_a, k, \frac{(1-\varepsilon)n_a}{2}\left(1 - 1/n_w^{C(1+o(1))}\right)\right)$ with success probability ε. Then there exists an algorithm which solves $\mathsf{aDP}(n_w, k, t_w)$ with probability bigger than

$$\Omega\left(\frac{\varepsilon}{\sqrt{n_a}}\right) - n_a\, 2^{-\Omega(n_w)}.$$

This theorem is a consequence of Theorem 6.1 and Proposition 6.7.

Proof. Let $\rho \in \mathbb{R}_+$ be such that

$$(1-\varepsilon)\,\frac{1}{2}(1-2^{-\rho}) = (1+\eta)h^{-1}\left(\frac{k}{n_w}\right) \iff \rho = -\log_2\left(1 - 2\,\frac{1+\eta}{1-\varepsilon}\,h^{-1}\left(\frac{k}{n_w}\right)\right) \tag{27}$$

and $\mathbf{r} \leftarrow \mathsf{Ber}(\rho)^{\otimes n_w}$. According to Proposition 6.7 we have

$$\mathbb{E}_{\mathbf{G}}\left(\Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big)\right) = 2^{-\Omega(n_w)}\ .$$

Recall that $\langle \mathbf{r}, \mathbf{t}\rangle \leftarrow \mathsf{Ber}(\rho t_w)$ (see Lemma 2.2). Therefore, according to Theorem 6.1, if there exists an algorithm which solves $\mathsf{aDP}\left(n_a, k, \frac{n_a}{2}\left(1 - 2^{-\rho t_w}\right)\right)$ in time T then there exists an algorithm which solves $\mathsf{aDP}(n_w, k, t_w)$ in time $O(T + n_a)$ with probability bigger than

$$\Omega\left(\frac{\varepsilon}{\sqrt{\frac{n_a}{2}\left(1 - 2^{-\rho t_w}\right)}}\right) - n_a\, 2^{-\Omega(n_w)}.$$

Using that $k/n_w = o(1)$ and Equation (27) we have the following computation

$$\begin{aligned}
\rho t_w &= -\log_2\left(1 - 2\,\frac{1+\eta}{1-\varepsilon}\,h^{-1}\left(\frac{k}{n_w}\right)\right) t_w \\
&= \frac{2}{\ln(2)}\,\frac{1+\eta}{1-\varepsilon}\,\frac{1}{\log_2\left(\frac{n_w}{k}\right)}\,\frac{k}{n_w}\,t_w(1+o(1)) \\
&= C(1+o(1))\log_2(n_w)
\end{aligned}$$

where in the last line we used the assumption of Theorem 6.8, in particular Equation (26). Therefore,

$$\frac{1}{2}\left(1-2^{-\rho t_w}\right)=\frac{1}{2}\left(1-1/n_w^{C(1+o(1))}\right)$$

which concludes the proof. □

7. Instantiating the Reduction: Worst-to-Average Case

Our aim in this section is to instantiate the worst-to-average case reduction. To this end we need to choose parameters such that our smoothing bound is negligible. Recall that our upper-bound of Proposition 5.1 involves $(N_\ell(\mathcal{C}))_{\ell\geq d_{\min}(\mathcal{C})}$ and $(N_\ell(\mathbf{t}+\mathcal{C}\setminus\{\mathbf{t}\}))_{\ell\geq 0}$ which respectively denote the cardinality of the following sets:

$$\{\mathbf{c}\in\mathcal{C}:\,|\mathbf{c}|=\ell\}\quad\text{and}\quad\{\mathbf{y}\in\mathbf{t}+\mathcal{C}\backslash\{\mathbf{t}\}:\,|\mathbf{y}|=\ell\}\,.$$

Notice that both of these sets are formed by words of Hamming weight ℓ and being at distance at least $d\stackrel{\text{def}}{=}d_{\min}(\mathcal{C})$. These sets are thus *spherical codes* of radius ℓ and minimum distance d.

To derive smoothing bounds for a fixed code $\mathcal{C}$ our first attempt consists of choosing the best known upper-bound on the size of spherical codes. This bound is known as the *second linear programming bound* [MRRW77].

Lemma 7.1 ([ACKL05, Lemma 1])**.** *Let $A(n,d,\ell)$ denote the maximal possible number of vectors in $\mathbb{F}_2^n$ of Hamming weight ℓ, being at Hamming distance at least d. Let*

$$R(\delta,\lambda)\stackrel{def}{=}\limsup_{n\to+\infty}\frac{1}{n}\log_2 A(n,\delta n,\lambda n).$$

Then, if $\delta\leq 2\lambda(1-\lambda)$

$$R(\delta,\lambda)\leq h\left(\frac{1}{2}\left(1-\sqrt{1-\left(\sqrt{4\lambda(1-\lambda)-\delta(2-\delta)}-\delta\right)^2}\right)\right)\,.$$

Remark 7.2. *One can notice that* [ACKL05, Lemma 1, *b*)] *gives another bound on $R(\delta,\lambda)$ for the case where $\delta>2\lambda(1-\lambda)$. However, in our application, we are are most interested in the case $\lambda\approx(1-\delta)$.*

To instantiate the reduction we need first to fix an $[n_w,k]$-code $\mathcal{C}$ of minimum distance δn_w. A natural choice for δ is given by the relative *Gilbert-Varshamov* bound $h^{-1}\left(1-\frac{k}{n_w}\right)$ which appears ubiquitously in the coding-theoretic literature: amongst other contexts, it arises as the (expected) relative minimum distance of a random code of dimension k and length n_w (see for instance [BF02, §C]). However, thanks to our experience with random codes, we need to choose k/n_w as a $o(1)$ for the reduction to be useful. But surprisingly this choice is dramatic: it turns out that if one uses the Lemma 7.1 to upper-bound $N_{\lambda n}(\mathcal{C})$ and $(N_{\lambda n}(\mathbf{t}+\mathcal{C}\setminus\{\mathbf{t}\}))$ for these parameters and $\lambda=1-\delta$, one gets an upper-bound worse than $\frac{1}{n_w}\log_2(\mathcal{C})=\frac{k}{n_w}$.

Lemma 7.3. *Suppose that $\delta=h^{-1}(1-\varepsilon)$ where $\varepsilon=o(1)$. Let $\lambda=1-\delta$. Then,*

$$h\left(\frac{1}{2}\left(1-\sqrt{1-\left(\sqrt{4\lambda(1-\lambda)-\delta(2-\delta)}-\delta\right)^2}\right)\right)=-\varepsilon\log_2\varepsilon(1+o(1))$$

Proof. First, let us use that $h^{-1}(1-\varepsilon) = \frac{1}{2} - f(\varepsilon)$ where $f(\varepsilon) = \sqrt{\varepsilon}(1+o(1))$. Therefore,

$$\begin{aligned}
4\lambda(1-\lambda) - \delta(2-\delta) &= 4\left(\frac{1}{2}+f(\varepsilon)\right)\left(\frac{1}{2}-f(\varepsilon)\right) - \left(\frac{1}{2}-f(\varepsilon)\right)\left(\frac{3}{2}+f(\varepsilon)\right) \\
&= 1 - 4f(\varepsilon)^2 - \frac{3}{4} - \frac{f(\varepsilon)}{2} + \frac{3f(\varepsilon)}{2} + f(\varepsilon)^2 \\
&= \frac{1}{4} + f(\varepsilon) - 3f(\varepsilon)^2 \\
&= \frac{1}{4}\left(1 + 4\sqrt{\varepsilon}(1+o(1))\right)
\end{aligned}$$

where in the last line we used the definition of f. Therefore,

$$\begin{aligned}
\left(\sqrt{4\lambda(1-\lambda)-\delta(2-\delta)} - \delta\right)^2 &= \left(\frac{1}{2}\sqrt{1+4\sqrt{\varepsilon}(1+o(1))} - 1/2 + \sqrt{\varepsilon}(1+o(1))\right)^2 \\
&= \left(2\sqrt{\varepsilon}(1+o(1))\right)^2 \\
&= 4\varepsilon(1+o(1))
\end{aligned}$$

Let us continue the computation

$$\sqrt{1 - \left(\sqrt{4\lambda(1-\lambda)-\delta(2-\delta)} - \delta\right)^2} = \sqrt{1-4\varepsilon(1+o(1))} = 1 - 2\varepsilon(1+o(1))$$

which leads to

$$\begin{aligned}
h\left(\frac{1}{2}\left(1 - \sqrt{1 - \left(\sqrt{4\lambda(1-\lambda)-\delta(2-\delta)} - \delta\right)^2}\right)\right) &= h\left(\varepsilon(1+o(1))\right) \\
&= -\varepsilon \log_2 \varepsilon \, (1+o(1))
\end{aligned}$$

It concludes the proof. □

The conclusion from the above lemma is the following: using the second linear programming bound to upper-bound $(N_\ell(\mathcal{C}))_{\ell \geq d_{\min}(\mathcal{C})}$ and $(N_\ell(\mathbf{t}+\mathcal{C}\setminus\{\mathbf{t}\}))_{\ell \geq 0}$ is no better than using the trivial bound of 2^k. That is, the best upper-bound on the number of codewords of weight ℓ is just the total number of codewords!

Even though this bound is trivial we can still successfully recover the same parameters as [BLVW19] for a worst-to-average case reduction. In fact, [BLVW19] is also implicitly using this weak bound on the number of codewords of a given weight (see [BLVW19, Equation (6)]).

As this bound is quite crude, we will be forced to assume our code is *balanced* as in [BLVW19, Corollary 4.2].

Definition 7.4 (Balanced code)**.** *An $[n,k]$-code is δ-balanced if its minimum distance is at least δn and all the codewords have Hamming weight at most $(1-\delta)n$. That is, for all $\mathbf{x} \in \mathcal{C}\setminus\{\mathbf{0}\}$,*

$$\delta n \leq |\mathbf{x}| \leq (1-\delta)n \ .$$

We won't be able to reduce wDP into aDP but only sub-instances for which the input code is supposed to be balanced. More precisely we will use the following worst-case problem

Problem 7.5 (Worst-Case Balanced Decoding Problem - $\mathsf{wBalDP}(n,k,t,d)$)**.**

- Input: $(\mathbf{G}, \mathbf{y} \stackrel{def}{=} \mathbf{xG}+\mathbf{e})$ *where $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ is a generator matrix of an $[n,k]$-code δ-balanced code, $\mathbf{x} \in \mathbb{F}_2^k$, and $\mathbf{e} \in \mathbb{F}_2^n$ of Hamming weight t.*
- Output: $\mathbf{xG}$.

We remark that a standard argument, detailed in [BLVW19, Lemma 2.2], shows that balanced codes with our required parameters do indeed exist.

Proposition 7.6. *Let $\mathbf{G} \in \mathbb{F}_2^{k\times n_w}$ be the generator matrix of an $[n_w, k]$-code which is δ-balanced with $1/2 \geq \delta \geq h^{-1}\left(1-\frac{k}{n_w}\right) \geq C$ for some constant $C > 0$. Let $\mathbf{t} \in \mathbb{F}_2^{n_w}$, suppose that $\frac{|\mathbf{t}|}{n_w} = o(1)$. Let $\varepsilon, \eta \in (0,1)$ and $\rho \in \mathbb{R}_+$ be such that $(1-\varepsilon)\frac{1}{2}(1-2^{-\rho}) \geq (1+\eta)h^{-1}\left(2\frac{k}{n_w} + D\frac{|\mathbf{t}|}{n_w}\right)$ for some constant D large enough. Then*

$$\Delta\Big(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big) = 2^{-\Omega(n_w)}$$

where $\mathbf{r} \leftarrow \mathsf{Ber}(\rho)^{\otimes n}$, $\mathbf{a} \leftarrow \mathbb{F}_2^k$ and $e \leftarrow \mathsf{Ber}(\rho|\mathbf{t}|)$.
The same result holds if $\mathbf{r}$ is instead uniformly distributed over a sphere of radius greater than $(1+\eta)h^{-1}\left(2\frac{k}{n_w} + D\frac{|\mathbf{t}|}{n_w}\right)$.

Proof. Let us show that the result holds for $\mathbf{r} \leftarrow \mu_r$ where μ_r denotes the uniform distribution over a sphere of radius $r \geq n(1+\eta)h^{-1}\left(2\frac{k}{n_w} + D\frac{|\mathbf{t}|}{n_w}\right)$. To conclude the proof it will just remain to use Proposition 6.6. Recall that $\widehat{\mu_r}(\ell) = 2^{-n_w}\frac{K_r(\ell)}{\binom{n_w}{r}}$ and

$$\sum_{\ell=0}^{n} \frac{K_r(\ell)^2}{\binom{n_w}{r}} \frac{\binom{n}{\ell}}{2^{n_w}} = 1. \tag{28}$$

Let $d \stackrel{\text{def}}{=} \delta n$. Notice that $d \leq d_{\min}(\mathcal{C})$. The bound given in Proposition 5.1 becomes

$$\begin{aligned}
&\frac{1}{2}\sqrt{\sum_{\ell \geq d_{\min}(\mathcal{C})} N_\ell(\mathcal{C})\frac{K_r(\ell)^2}{\binom{n_w}{r}^2} + \sum_{\ell \geq 0} N_\ell(\mathcal{C}+\mathbf{t}\backslash\{\mathbf{t}\})\frac{K_r(\ell)^2}{\binom{n_w}{r}^2}} \\
&\leq \frac{1}{2}\sqrt{\sum_{\ell=d}^{n_w-d} 2^k \frac{K_r(\ell)^2}{\binom{n_w}{r}^2} + \sum_{\ell=d-|\mathbf{t}|}^{n_w-d+|\mathbf{t}|} 2^k \frac{K_r(\ell)^2}{\binom{n_w}{r}^2}} \\
&\leq \frac{1}{2}\sqrt{2 \max_{d-|\mathbf{t}| \leq \ell \leq n_w-d+|\mathbf{t}|} \left\{\frac{2^{n_w}}{\binom{n_w}{\ell}}\right\} \sum_{\ell=d-|\mathbf{t}|}^{n_w-d+|\mathbf{t}|} 2^k \frac{\binom{n_w}{\ell}}{2^{n_w}} \frac{K_r(\ell)^2}{\binom{n_w}{r}^2}} \\
&\leq \sqrt{\frac{1}{2}\,\frac{2^{n_w}}{\binom{n_w}{d-|\mathbf{t}|}}\,\frac{2^k}{\binom{n_w}{r}}}
\end{aligned}$$

where in the second lined we used that $\mathcal{C}$ is δ-balanced and in the last line we used Equation (28). Recall now that $\binom{n_w}{r} = 2^{\Theta(n)h(r/n_w)}$. The above upper-bound becomes

$$\sqrt{\frac{1}{2}\,\frac{2^{n_w}}{\binom{n_w}{d-|\mathbf{t}|}}\,\frac{2^k}{\binom{n}{r}}} = 2^{\Theta(n_w)\left(1-h\left(\frac{d-|\mathbf{t}|}{n_w}\right)+\frac{k}{n_w}-h\left(\frac{r}{n_w}\right)\right)}\ . \tag{29}$$

Using that $\tau \stackrel{\text{def}}{=} \frac{|\mathbf{t}|}{n_w} = o(1)$ and $\delta = \frac{d}{n_w}$ is such that $\delta \leq 1/2$ and $\delta \geq h^{-1}\left(1-\frac{k}{n_w}\right) \geq C > 0$ we have

$$h\left(\frac{d-|\mathbf{t}|}{n_w}\right) = h\left(\delta - \tau\right) = h(\delta) + O(\tau) \geq 1 - \frac{k}{n_w} - O(\tau)\ . \tag{30}$$

Furthermore, $r \geq n_w(1+\eta)h^{-1}\left(2\frac{k}{n_w} + D\frac{|\mathbf{t}|}{n_w}\right)$ which implies that

$$h\left(\frac{r}{n_w}\right) \geq 2\frac{k}{n_w} + D\tau\ . \tag{31}$$

Therefore, using Equations (30) and (31), we obtain

$$1 - h\left(\frac{d-|\mathbf{t}|}{n_w}\right) + \frac{k}{n_w} - h\left(\frac{r}{n_w}\right) \leq 2\frac{k}{n_w} + O(\tau) - 2\frac{k}{n_w} - D\tau < 0$$

assuming D is a sufficiently large constant. In other words the upper-bound given in Equation (29) is $2^{-\Omega(n_w)}$ which concludes the proof. □

We are now ready to instantiate the worst to average-case reduction. We propose parameters for which the average case decoding problem to which we reduce has a noise rate $1/2 - 1/\mathsf{poly}(n_w)$. However, this set of parameters is such that the worst case decoding problem is "only" super-polynomially hard. Many other sets of parameters can be proposed; for instance, one could aim for noise rate $1/2 - 2^{-o(n)}$ and assume sub-exponential hardness of the worst case problem.

Theorem 7.7. *Let $\varepsilon, \eta, C > 0$ and $n_w, k, t_w \in \mathbb{N}$ be such that*

$$\frac{k}{n_w} = o(1), \quad \frac{t_w}{n_w} = o\left(\frac{k}{n_w}\right) \quad \text{and} \quad \frac{4}{\ln(2)}\,\frac{1+\eta}{1-\varepsilon}\,\frac{1}{\log_2\left(\frac{n_w}{k}\right)}\,\frac{k}{n_w}\,t_w = C\log_2(n_w)\ . \tag{32}$$

Suppose that there exists an algorithm $\mathscr{A}$ which solves $\mathsf{aDP}\left(n_a, k, \frac{(1-\varepsilon)n_a}{2}\left(1 - 1/n_w^{C(1+o(1))}\right)\right)$ with success probability ε. Then there exists an algorithm which solves $\mathsf{wBalDP}(n_w, k, t_w, \delta n_w)$, where $\delta \geq h^{-1}\left(1 - \frac{k}{n_w}\right)$, and with probability bigger than

$$\Omega\left(\frac{\varepsilon}{\sqrt{n_a}}\right) - n_a\, 2^{-\Omega(n_w)}.$$

Proof. Let $(\mathbf{G}, \mathbf{mG} + \mathbf{t})$ be an instance of $\mathsf{wBalDP}(n_w, k, t_w, \delta n_w)$. Let $\rho \in \mathbb{R}_+$ be such that

$$(1-\varepsilon)\,\frac{1}{2}(1 - 2^{-\rho}) = (1+\eta)h^{-1}\left(2\frac{k}{n_w} + D\frac{t}{n_w}\right)$$
$$\iff \rho = -\log_2\left(1 - 2\,\frac{1+\eta}{1-\varepsilon}\,h^{-1}\left(2\frac{k}{n_w} + D\frac{t}{n_w}\right)\right) \tag{33}$$

for some constant $D > 0$ large enough and $\mathbf{r} \leftarrow \mathsf{Ber}(\rho)^{\otimes n_w}$. According to Proposition 7.6 we have

$$\Delta\Big(\left(\mathbf{rG}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big) = 2^{-\Omega(n_w)}$$

Notice now that $\langle \mathbf{r}, \mathbf{t}\rangle \leftarrow \mathsf{Ber}(\rho t_w)$ (see Lemma 2.2). Therefore, according to Theorem 4.1, if there exists an algorithm which solves $\mathsf{aDP}\left(n_a, k, \frac{n_a}{2}\left(1 - 2^{-\rho t_w}\right)\right)$ then there exists an algorithm which recovers $\mathbf{m}$ from $\mathbf{mG} + \mathbf{t}$ with probability bigger than

$$\Omega\left(\frac{\varepsilon}{\sqrt{\frac{n_a}{2}\left(1 - 2^{-\rho t_w}\right)}}\right) - n_a\, 2^{-\Omega(n_w)}.$$

Using that $k/n_w = o(1)$, $t_w/n_w = o(k/n_w)$ and Equation (33) we have the following computation:

$$\begin{aligned}
\rho t_w &= -\log_2\left(1 - 2\,\frac{1+\eta}{1-\varepsilon}\,h^{-1}\left(2\frac{k}{n_w} + D\frac{t}{n_w}\right)\right) t_w \\
&= \frac{4}{\ln(2)}\,\frac{1+\eta}{1-\varepsilon}\,\frac{1}{\log_2\left(\frac{n_w}{k}\right)}\,\frac{k}{n_w}\,t_w(1+o(1)) \\
&= C(1+o(1))\log_2(n_w)
\end{aligned}$$

where in the last line we used the assumption of Theorem 6.8, in particular Equation (7.7). Therefore,

$$\frac{1}{2}\left(1 - 2^{-\rho t_w}\right) = \frac{1}{2}\left(1 - 1/n_w^{C(1+o(1))}\right)$$

In other words, from an algorithm solving $\mathsf{aDP}\left(n_a, k, \frac{n_a}{2}\left(1-2^{-\rho t_w}\right)\right)$, we can recover $\mathbf{m}$ from any instance $(\mathbf{G}, \mathbf{mG}+\mathbf{t})$ of $\mathsf{wBalDP}(n_w, k, t_w, \delta n_w)$ with probability bigger than $\Omega\left(\frac{\varepsilon}{\sqrt{n_a}}\right) - n_a 2^{-\Omega(n_w)}$ which concludes the proof. □

A set of parameters. One can apply Theorem 7.7 for instance with the following set of parameters

$$\frac{k}{n_w} = \frac{1}{n_w^C} \quad \text{and} \quad \frac{t}{n_w} = \frac{\log_2(n_w)^2}{n_w^{1-C}}$$

with $C < 1/2$.

References

[ACKL05] Alexei E. Ashikhmin, Gérard D. Cohen, Michael Krivelevich, and Simon Litsyn. Bounds on distance distributions in codes of known size. *IEEE Trans. Inf. Theory*, 51(1):250–258, 2005.

[Ale03] Alekhnovich, Michael. More on Average Case vs Approximation Complexity. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 298–307. IEEE Computer Society, 2003.

[BF02] Alexander Barg and G. David Forney. Random codes: Minimum distances and error exponents. *IEEE Trans. Inf. Theory*, 48(9):2568–2573, 2002.

[BLVW19] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. Worst-case hardness for LPN and cryptographic hashing via code smoothing. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*, volume 11478 of *LNCS*, pages 619–635. Springer, 2019.

[CFS01] Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 157–174, Gold Coast, Australia, 2001. Springer.

[CS16] Rodolfo Canto-Torres and Nicolas Sendrier. Analysis of information set decoding for a sub-linear error weight. In *Post-Quantum Cryptography 2016*, LNCS, pages 144–161, Fukuoka, Japan, February 2016.

[DDRT22] Thomas Debris-Alazard, Léo Ducas, Nicolas Resch, and Jean-Pierre Tillich. Smoothing codes and lattices: Systematic study and new bounds. *CoRR*, abs/2205.10552, 2022.

[DST19] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. Wave: A new family of trapdoor one-way preimage sampleable functions based on codes. In *Advances in Cryptology - ASIACRYPT 2019*, volume 11921 of *LNCS*, pages 21–51, Kobe, Japan, December 2019. Springer.

[FJR22] Thibauld Feneuil, Antoine Joux, and Matthieu Rivain. Syndrome decoding in the head: Shorter signatures from zero-knowledge proofs. *IACR Cryptol. ePrint Arch.*, page 188, 2022.

[FS96] Jean-Bernard Fischer and Jacques Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In Ueli Maurer, editor, *Advances in Cryptology - EUROCRYPT'96*, volume 1070 of *LNCS*, pages 245–255. Springer, 1996.

[Lev95] Vladimir I. Levenshtein. Krawtchouk polynomials and universal bounds for codes and designs in hamming spaces. *IEEE Trans. Inf. Theory*, 41(5):1303–1321, 1995.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology - EUROCRYPT2010*, volume 6110 of *LNCS*, pages 1–23. Springer, 2010.

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75:565–599, 2015.

[McE78] Robert J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.

[MRRW77] R. J. McEliece, E. R. Rodemich, H. Rumsey, and L. R. Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Trans. Inform. Theory*, 23(2):157–166, 1977.

[PMS21] Alice Pellet-Mary and Damien Stehlé. On the hardness of the NTRU problem. In *Asiacrypt 2021 - 27th Annual International Conference on the Theory and Applications of Cryptology and Information Security*, Advances in Cryptology – ASIACRYPT 2021. Lecture Notes in Computer Science, vol 13090., Singapore, Singapore, December 2021.

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

[PRS17] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 461–473, 2017.

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

[RSW18] Miruna Rosca, Damien Stehlé, and Alexandre Wallet. On the ring-LWE and polynomial-LWE problems. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 146–173. Springer, 2018.

[Ste93] Jacques Stern. A new identification scheme based on syndrome decoding. In D.R. Stinson, editor, *Advances in Cryptology - CRYPTO'93*, volume 773 of *LNCS*, pages 13–21. Springer, 1993.

[vL99] Jacobus Hendricus van Lint. *Introduction to coding theory*. Graduate texts in mathematics. Springer, 3rd edition edition, 1999.

[YZ21] Yu Yu and Jiang Zhang. Smoothing out binary linear codes and worst-case sub-exponential hardness for LPN. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part III*, volume 12827 of *LNCS*, pages 473–501. Springer, 2021.

Appendix A. Proof of Lemma 6.2

In what follows $\mathbf{G}$ is uniformly distributed over $k \times n$ binary matrices of rank k and is the generator matrix of the $[n,k]$-code $\mathcal{C}$.

Proof of Lemma A. First, notice that for any $\mathbf{m} \in \mathbb{F}_2^k \backslash \{\mathbf{0}\}$, $\mathbf{mG}$ is uniformly distributed as $\mathbf{G}$ has rank k. Furthermore, we have

$$\forall \ell > 0,\ N_\ell(\mathcal{C}) = \sum_{\mathbf{m} \in \mathbb{F}_2^k \backslash \{\mathbf{0}\}} \mathbb{1}_{\{|\mathbf{mG}|=\ell\}} \quad \text{and} \quad \forall \ell \geq 0,\ N_\ell(\mathbf{t}+\mathcal{C}) = \sum_{\mathbf{m} \in \mathbb{F}_2^k \backslash \{\mathbf{0}\}} 1_{\{|\mathbf{mG}+\mathbf{t}|=\ell\}}$$

where $\mathbb{1}_{\{\mathscr{E}\}}$ denotes the indicator function of the event $\mathscr{E}$. To conclude the proof it remains to use the linearity of the expectation and using that $\mathbf{mG}$ is uniform when $\mathbf{m} \neq 0$. □

Appendix B. Proof of Proposition 6.6

Our aim in this section is to prove the following proposition

Proposition 6.6. *Let $\mathbf{t} \in \mathbb{F}_2^n$, $\varepsilon > 0$, $\rho \in \mathbb{R}_+$ and $p \overset{def}{=} \frac{1}{2}\left(1 - 2^{-\rho}\right)$. Let $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ be the generator matrix of an $[n,k]$-code. Then,*

$$\Delta\Big(\left(\mathbf{rG}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\Big) \leq \sum_{r=(1-\varepsilon)np}^{(1+\varepsilon)np} \Delta\Big(\left(\mathbf{r}_r\mathbf{G}^\top, \langle \mathbf{r}_r, \mathbf{t}\rangle\right), (\mathbf{a}, e_r)\Big) + 2^{-\Omega(n)}$$

where $\mathbf{r} \leftarrow \mathsf{Ber}(\rho)^{\otimes n}$, $\mathbf{a} \leftarrow \mathbb{F}_2^k$, $e \leftarrow \mathsf{Ber}(\rho|\mathbf{t}|)$, $\mathbf{r}_r \leftarrow \mathscr{S}_r$ and the e_r's are distributed as the $\langle \mathbf{r}_r, \mathbf{t}\rangle$'s.

Roughly speaking, this proposition is a consequence of the fact that a Bernoulli distribution concentrates Hamming weights over a small number of slices close to the expected weight and, on each slice the Bernoulli distribution is uniform.

Let,

$$p \overset{\text{def}}{=} \frac{1}{2}\left(1 - 2^{-\rho}\right)$$

Recall that $\mu_{\text{Ber},\rho}$ denotes the Bernoulli distribution $\mathsf{Ber}(\rho)^{\otimes n_w}$,

$$\forall \mathbf{x} \in \mathbb{F}_2^{n_w}, \quad \mu_{\text{Ber},\rho}(\mathbf{x}) \overset{\text{def}}{=} \frac{1}{2^n}\left(1 - 2^{-\rho}\right)^{|\mathbf{x}|}\left(1 + 2^{-\rho}\right)^{n-|\mathbf{x}|}.$$

Let us introduce the truncated Bernoulli distribution over words of Hamming weight $[(1-\varepsilon)pn, (1+\varepsilon)pn]$ for some $\varepsilon > 0$, namely

$$\mu_{\text{TrBer},\rho}(\mathbf{x}) \overset{\text{def}}{=} \begin{cases} \frac{1}{Z}\,\mu_{\text{Ber},\rho}(\mathbf{x}) & \text{if } |\mathbf{x}| \in [(1-\varepsilon)pn, (1+\varepsilon)pn] \\ 0 & \text{otherwise.} \end{cases} \tag{34}$$

where

$$Z \overset{\text{def}}{=} \sum_{|\mathbf{y}|=(1-\varepsilon)np}^{(1+\varepsilon)np} \mu_{\text{Ber},\rho}(\mathbf{y}) \tag{35}$$

is the probability normalizing constant. Furthermore, by Chernoff's bound

$$Z = 1 - 2^{-\Omega(n)}. \tag{36}$$

Let f_{Ber} and f_{TrBer} be the distribution of $(\mathbf{rG}^\top, \langle \mathbf{r}, \mathbf{t}\rangle)$ and $(\mathbf{r}_{\text{Tr}}\mathbf{G}^\top, \langle \mathbf{r}_{\text{Tr}}, \mathbf{t}\rangle)$ where $\mathbf{r}$ and $\mathbf{r}_{\text{Tr}}$ are distributed according to $\mu_{\text{Ber},\rho}$ and $\mu_{\text{TrBer},\rho}$.

Let g_{Ber} and g_{TrBer} be the distribution of $(\mathbf{a}, e)$ and $(\mathbf{a}, e_{\text{Tr}})$ where $\mathbf{a} \leftarrow \mathbb{F}_2^k$ and e is distributed according to $\langle \mathbf{r}, \mathbf{t}\rangle$ and $\langle \mathbf{r}_{\text{Tr}}, \mathbf{t}\rangle$.

Lemma B.1. *We have*

$$\Delta\left(g_{\text{Ber}}, g_{\text{TrBer}}\right) = 2^{-\Omega(n)} \quad \textit{and} \quad \Delta\left(f_{\text{Ber}}, f_{\text{TrBer}}\right) = 2^{-\Omega(n)}$$

Proof. Let us start by proving the first equality. Notice that in distributions g_{Ber} and g_{TrBer} the first component follows the same (uniform) distribution. Furthermore, components are independent. It shows that

$$\Delta\left(g_{\mathrm{Ber}}, g_{\mathrm{TrBer}}\right) = \Delta\left(\langle \mathbf{r}, \mathbf{t}\rangle, \langle \mathbf{r}_{\mathrm{Tr}}, \mathbf{t}\rangle\right) \leq \Delta(\mu_{\mathrm{Ber},\rho}, \mu_{\mathrm{TrBer},\rho})$$

where the last inequality is the data processing inequality. We have now the following computation

$$\begin{aligned}
2\Delta\left(\mu_{\mathrm{Ber},\rho}, \mu_{\mathrm{TrBer},\rho}\right) &= \sum_{\mathbf{x}\in\mathbb{F}_2^n} \left|\mu_{\mathrm{Ber},\rho}(\mathbf{x}) - \mu_{\mathrm{TrBer},\rho}(\mathbf{x})\right| \\
&= \sum_{|\mathbf{x}|\in[(1-\varepsilon)np,(1+\varepsilon)np]} \left|\mu_{\mathrm{Ber},\rho}(\mathbf{x}) - \mu_{\mathrm{TrBer},\rho}(\mathbf{x})\right| + \sum_{|\mathbf{x}|\notin[(1-\varepsilon)np,(1+\varepsilon)np]} \left|\mu_{\mathrm{Ber},\rho}(\mathbf{x})\right| \\
&= 2^{-\Omega(n)} \left(\sum_{|\mathbf{x}|\in[(1-\varepsilon)np,(1+\varepsilon)np]} \left|\mu_{\mathrm{Ber},\rho}(\mathbf{x})\right| \right) + 2^{-\Omega(n)} \quad \text{(Equations (34) and (36))} \\
&= 2^{-\Omega(n)} \qquad (37)
\end{aligned}$$

where in the last line we used that $\mu_{\mathrm{Ber},\rho}$ is a probability distribution.

Focusing now on the second equality ($\mathbf{r}$ and $\mathbf{r}_{\mathrm{Tr}}$ are distributed according to $\mu_{\mathrm{Ber},\rho}$ and $\mu_{\mathrm{TrBer},\rho}$):

$$\begin{aligned}
\Delta\left(f_{\mathrm{Ber}}, f_{\mathrm{TrBer}}\right) &= \sum_{\substack{\mathbf{m}\in\mathbb{F}_2^k \\ b\in\{0,1\}}} \left|\mathbb{P}_{\mathbf{r}}\left(\mathbf{r}\mathbf{G}^\top = \mathbf{m}, \langle \mathbf{r}, \mathbf{t}\rangle = b\right) - \mathbb{P}_{\mathbf{r}_{\mathrm{Tr}}}\left(\mathbf{r}_{\mathrm{Tr}}\mathbf{G}^\top = \mathbf{m}, \langle \mathbf{r}_{\mathrm{Tr}}, \mathbf{t}\rangle = b\right)\right| \\
&= \sum_{\substack{\mathbf{x}\in\mathbb{F}_2^n/\mathcal{C}^* \\ b\in\{0,1\}}} \left|\mathbb{P}_{\mathbf{r}}\left(\mathbf{r}\mathbf{G}^\top = \mathbf{x}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle = b\right) - \mathbb{P}_{\mathbf{r}_{\mathrm{Tr}}}\left(\mathbf{r}_{\mathrm{Tr}}\mathbf{G}^\top = \mathbf{x}\mathbf{G}^\top, \langle \mathbf{r}_{\mathrm{Tr}}, \mathbf{t}\rangle = b\right)\right| \\
&= \sum_{\substack{\mathbf{x}\in\mathbb{F}_2^n/\mathcal{C}^* \\ b\in\{0,1\}}} \left|\mathbb{P}_{\mathbf{r}}\left(\mathbf{r} \in \mathbf{x} + \mathcal{C}^*, \langle \mathbf{r}, \mathbf{t}\rangle = b\right) - \mathbb{P}_{\mathbf{r}_{\mathrm{Tr}}}\left(\mathbf{r}_{\mathrm{Tr}} \in \mathbf{x} + \mathcal{C}^*, \langle \mathbf{r}_{\mathrm{Tr}}, \mathbf{t}\rangle = b\right)\right| \\
&= \sum_{\substack{\mathbf{x}\in\mathbb{F}_2^n/\mathcal{C}^* \\ b\in\{0,1\}}} \left| \sum_{\mathbf{c}^*\in\mathcal{C}^*} \mathbb{P}_{\mathbf{r}}\left(\mathbf{r} = \mathbf{x} + \mathbf{c}^*, \langle \mathbf{r}, \mathbf{t}\rangle = b\right) - \mathbb{P}_{\mathbf{r}_{\mathrm{Tr}}}\left(\mathbf{r}_{\mathrm{Tr}} = \mathbf{x} + \mathbf{c}^*, \langle \mathbf{r}_{\mathrm{Tr}}, \mathbf{t}\rangle = b\right)\right| \\
&\leq \sum_{\substack{\mathbf{x}\in\mathbb{F}_2^n/\mathcal{C}^* \\ b\in\{0,1\}}} \sum_{\mathbf{c}^*\in\mathcal{C}^*} \left|\mathbb{P}_{\mathbf{r}}\left(\mathbf{r} = \mathbf{x} + \mathbf{c}^*, \langle \mathbf{r}, \mathbf{t}\rangle = b\right) - \mathbb{P}_{\mathbf{r}_{\mathrm{Tr}}}\left(\mathbf{r}_{\mathrm{Tr}} = \mathbf{x} + \mathbf{c}^*, \langle \mathbf{r}_{\mathrm{Tr}}, \mathbf{t}\rangle = b\right)\right| \\
&= \Delta\left(\left(\mathbf{r}, \langle \mathbf{r}, \mathbf{t}\rangle\right), \left(\mathbf{r}_{\mathrm{Tr}}, \langle \mathbf{r}_{\mathrm{Tr}}, \mathbf{t}\rangle\right)\right) \\
&\leq \Delta\left(\mathbf{r}, \mathbf{r}_{\mathrm{Tr}}, \right) = \Delta\left(\mu_{\mathrm{Ber},\rho}, \mu_{\mathrm{TrBer},\rho}\right) = 2^{-\Omega(n)}
\end{aligned}$$

where in the last line we used Equation (37) and the data processing inequality. □

Lemma B.2. *We have*

$$\Delta(f_{\mathrm{Ber}}, g_{\mathrm{Ber}}) \leq \Delta(f_{\mathrm{TrBer}}, g_{\mathrm{TrBer}}) + 2^{-\Omega(n)}$$

Proof. By using the triangular inequality

$$\Delta(f_{\mathrm{Ber}}, g_{\mathrm{Ber}}) \leq \Delta(f_{\mathrm{Ber}}, f_{\mathrm{TrBer}}) + \Delta(f_{\mathrm{TrBer}}, g_{\mathrm{TrBer}}) + \Delta(g_{\mathrm{TrBer}}, g_{\mathrm{Ber}})$$

Applying Lemma B.1 concludes the proof. □

The following lemma is a basic property of the statistical distance.

Lemma B.3. *For any distribution $(f_i)_{1\leq i\leq n}$ and $(g_i)_{1\leq i\leq n}$ we have*

$$\Delta\left(\sum_{i=1}^n \lambda_i f_i, \sum_{i=1}^n \lambda_i g_i\right) \leq \sum_{i=1}^n \lambda_i\, \Delta(f_i, g_i)$$

where the λ_i's are positive and sum to one.

We are now ready to prove Proposition 6.6.

Proof of Proposition 6.6. First, by Lemma B.2 we have

$$\Delta(f_{\mathrm{Ber}}, g_{\mathrm{Ber}}) \leq \Delta(f_{\mathrm{TrBer}}, g_{\mathrm{TrBer}}) + 2^{-\Omega(n)} \tag{38}$$

Notice that,

$$\mu_{\mathrm{TrBer},\rho} = \sum_{r=(1-\varepsilon)np}^{(1+\varepsilon)np} \frac{1}{Z} \binom{n}{r} p^r (1-p)^{n-r} \mu_r$$

where μ_r denotes the uniform distribution over $\mathscr{S}_r$. Therefore we can decompose f_{TrBer} and g_{TrBer} as following convex combination

$$f_{\mathrm{Ber}} = \sum_{r=(1-\varepsilon)np}^{(1+\varepsilon)np} \lambda_r u_r \quad \text{and} \quad g_{\mathrm{TrBer}} = \sum_{r=(1-\varepsilon)np}^{(1+\varepsilon)np} \lambda_r v_r$$

where $\lambda_r \stackrel{\mathrm{def}}{=} \frac{1}{Z} \binom{n}{r} p^r (1-p)^{n-r}$, and u_r, v_r denote respectively the distribution of $\left(\mathbf{r}_r \mathbf{G}^\top, \langle \mathbf{r}_r, \mathbf{t} \rangle\right)$ and $(\mathbf{a}, e)$ with $\mathbf{r}_r \leftarrow \mathscr{S}_r$, $\mathbf{a} \leftarrow \mathbb{F}_2^k$ and e_r being distributed as $\langle \mathbf{r}_r, \mathbf{t} \rangle$. By using Lemma B.3 we obtain:

$$\begin{aligned} \Delta(f_{\mathrm{TrBer}}, g_{\mathrm{TrBer}}) &\leq \sum_{r=(1-\varepsilon)np}^{(1+\varepsilon)np} \lambda_r \Delta\left(u_r, v_r\right) \\ &\leq \sum_{r=(1-\varepsilon)np}^{(1+\varepsilon)np} \Delta\left(u_r, v_r\right) \end{aligned} \tag{39}$$

where in the last line we used that the λ_r's are smaller than one. To conclude the proof we plug Equation (39) in (38). □