Rigorous Foundations for Dual Attacks in Coding Theory

Charles Meyer-Hilfiger, Jean-Pierre Tillich

Rigorous Foundations for Dual Attacks in Coding Theory

Charles Meyer-Hilfiger and Jean-Pierre Tillich

Project COSMIQ, Inria de Paris,
charles.meyer-hilfiger@inria.fr,jean-pierre.tillich@inria.fr

Abstract. Dual attacks aiming at decoding generic linear codes have been found recently to outperform for certain parameters information set decoding techniques which have been for 60 years the dominant tool for solving this problem and choosing the parameters of code-based cryptosystems. However, the analysis of the complexity of these dual attacks relies on some unproven assumptions that are not even fully backed up with experimental evidence. These dual attacks can actually be viewed as the code-based analogue of dual attacks in lattice based cryptography. Here too, dual attacks have been found out those past years to be strong competitors to primal attacks and a controversy has emerged whether similar heuristics made for instance on the independence of certain random variables really hold. We will show that the dual attacks in coding theory can be studied by providing in a first step a simple alternative expression of the fundamental quantity used in these dual attacks. We then show that this expression can be studied without relying on independence assumptions whatsoever. This study leads us to discover that there is indeed a problem with the latest and most powerful dual attack proposed in [CDMT22a] and that for the parameters chosen in this algorithm there are indeed false candidates which are produced and which are not predicted by the analysis provided there which relies on independence assumptions. We then suggest a slight modification of this algorithm consisting in a further verification step, analyze it thoroughly, provide experimental evidence that our analysis is accurate and show that the complexity claims made in [CDMT22a] are indeed valid for this modified algorithm. This approach provides a simple methodology for studying rigorously dual attacks which could turn out to be useful for further developing the subject.

1 Introduction

1.1 The decoding problem and methods for solving it

Code-based cryptography is based on the hardness of decoding generic linear codes which in the binary case corresponds to

Problem 1 (decoding a linear code). *Let $\mathscr{C}$ be a binary linear code over $\mathbb{F}_2$ of dimension k and length n,* i.e. *a subspace of $\mathbb{F}_2^n$ of dimension k. We are given $\mathbf{y} = \mathbf{c} + \mathbf{e}$ where $\mathbf{c} \in \mathscr{C}$ and $\mathbf{e}$ is an error vector of Hamming weight $|\mathbf{e}| = t$ and we wish to recover $\mathbf{c}$ and $\mathbf{e}$.*

Equivalently, this problem can also be viewed as solving an underdetermined linear system with a weight constraint. Indeed, we can associate to a subspace $\mathscr{C}$ of dimension k of $\mathbb{F}_2^n$ a binary $(n-k) \times n$ matrix $\mathbf{H}$ (also called a *parity-check* matrix of the code) whose kernel defines $\mathscr{C}$, namely $\mathscr{C} = \{\mathbf{x} \in \mathbb{F}_2^n : \mathbf{H}\mathbf{x}^\intercal = \mathbf{0}\}$. The decoding problem is equivalent to find an $\mathbf{e}$ of Hamming weight t such that $\mathbf{H}\mathbf{e}^\intercal = \mathbf{s}^\intercal$ where $\mathbf{s}$ is the *syndrome* of $\mathbf{y}$ with respect to $\mathbf{H}$, *i.e.* $\mathbf{s}^\intercal = \mathbf{H}\mathbf{y}^\intercal$. This can be verified by observing that if there exists $\mathbf{c} \in \mathscr{C}$ and $\mathbf{e}$ such that $\mathbf{y} = \mathbf{c} + \mathbf{e}$ then $\mathbf{H}\mathbf{y}^\intercal = \mathbf{H}(\mathbf{c}+\mathbf{e})^\intercal = \mathbf{H}\mathbf{c}^\intercal + \mathbf{H}\mathbf{e}^\intercal = \mathbf{H}\mathbf{e}^\intercal$.

This problem has been studied for many years [Pra62, Ste88, Dum91, Bar97, FS09, BLP11, MMT11, BJMM12, MO15, BM17] and until recently the best way to solve this problem has been based on variations/improvements on the information set decoding (ISD) algorithm of [Pra62]. They are basically all of exponential complexity in the codelength n for a fixed code rate $R \stackrel{\triangle}{=} \frac{k}{n}$, and error rate $\tau \stackrel{\triangle}{=} \frac{t}{n}$: correcting t errors in a binary linear code of length n with the aforementioned algorithms has a cost of $2^{\alpha n(1+o(1))}$ where $\alpha = \alpha(R, \tau)$ is a constant depending on the algorithm which is used. All the efforts that have been spent on this problem have only managed to decrease slightly this exponent α. This exponent is the key for estimating the security level of any code-based cryptosystem. This was the case until [CDMT22a] found a way to improve greatly another decoding strategy, statistical decoding [Jab01], and this despite the fact that this strategy is far from being competitive as shown in [Ove06, DT17a]. Indeed, [CDMT22a] showed that when performing a modification of statistical decoding already suggested in [DT17b, §8], there is a dramatic improvement which allows to beat significantly the best ISD algorithms in the low rate regime (say $R < 0.3$). Basically the idea is to split the support in two pieces and reduce decoding to an LPN problem that is solved with standard Fourier techniques.

1.2 Reduction to LPN

It can be argued that [CDMT22a] is more than just a variation of the original statistical decoding algorithm, it is also a rather different way of approaching the decoding problem. As in [Jab01] the first step consists in producing low-weight parity-check equations, i.e. vectors that are in the row span of a parity-check matrix of the code and that are of low Hamming weight. But the way these parity-check equations are used is different (and even their form is different as we will explain shortly). Basically they are used to *reduce* the decoding problem to an LPN problem with as many samples (or oracle calls) as there are parity-check equations. Recall that the LPN problem is defined as follows

Problem 2 (LPN problem). *Let $\mathbf{s} \in \mathbb{F}_2^s$ be a secret vector and let $\tau \in [0,1]$ be a parameter. Let $\mathcal{O}_{\mathbf{s},\tau}(\cdot)$ be an oracle which, on a call, output:*

$$(\mathbf{a}, \langle \mathbf{s}, \mathbf{a} \rangle + e)$$

where $\mathbf{a}$ is uniformly distributed on $\mathbb{F}_2^s$ and e is distributed according to a Bernoulli of parameter τ. Moreover $\mathbf{a}$ and e are independent. We have access to $\mathcal{O}_{\mathbf{s},\tau}(\cdot)$ and want to find $\mathbf{s}$.

The notation $\langle \mathbf{x}, \mathbf{y} \rangle$ stands for the scalar product $\sum_{i=1}^{n} x_i y_i$ performed over $\mathbb{F}_2$ between two binary vectors $\mathbf{x} = (x_i)_{1 \le i \le n}$ and $\mathbf{y} = (y_i)_{1 \le i \le n}$. The idea behind the reduction comes by noticing that given $\mathbf{y} = \mathbf{c} + \mathbf{e}$ the noisy codeword we wish to decode and $\mathbf{h} \in \mathscr{C}^{\perp}$ a parity-check of $\mathscr{C}$ we have that

$$\langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}, \mathbf{h} \rangle.$$

Now, by considering $\mathscr{P}$ and $\mathscr{N}$ two complementary parts of the positions we can write that

$$\langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}, \mathbf{h} \rangle = \langle \mathbf{e}_{\mathscr{P}}, \mathbf{h}_{\mathscr{P}} \rangle + \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle$$

where $\mathbf{x}_{\mathscr{I}}$ denotes for a vector $\mathbf{x} = (x_i)_{1 \le i \le n}$ and a subset $\mathscr{I}$ of indices the vector $(x_i)_{i \in \mathscr{I}}$. And thus $\langle \mathbf{y}, \mathbf{h} \rangle$ can be seen as an LPN sample given as follows:

$$(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \quad \text{with} \quad \begin{cases} \mathbf{a} \stackrel{\triangle}{=} \mathbf{h}_{\mathscr{P}} \\ \mathbf{s} \stackrel{\triangle}{=} \mathbf{e}_{\mathscr{P}} \\ e \stackrel{\triangle}{=} \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle \end{cases} \tag{1.1}$$

where $\mathbf{e}_{\mathscr{P}}$ is the secret of our LPN problem, the samples come from $\mathbf{h}_{\mathscr{P}}$ and the noise is given by $\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle$. We expect that the noise term $e = \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle = \sum_{i \in \mathscr{N}} \mathbf{e}_i \mathbf{h}_i$ is more biased toward zero as $w \stackrel{\triangle}{=} |\mathbf{h}_{\mathscr{N}}|$ and $|\mathbf{e}_{\mathscr{N}}|$ decreases so that we need less samples to solve the LPN problem.

The idea of RLPN is to compute a set $\mathscr{H}$ of N parity checks of $\mathscr{C}$ of weight w on $\mathscr{N}$ to get N LPN samples and then solve the LPN problem to recover $\mathbf{e}_{\mathscr{P}}$. To that extent, [CDMT22a] makes the assumption that the samples (1.1) match exactly the framework of the LPN Problem 2.

1.3 Dual Attacks and a Controversy

Statistical decoding [Jab01] or its variant, namely RLPN decoding, fall both into the category of what can be called a *dual attack* which means here a decoding algorithm that computes in a first step low weight codewords in the dual code (i.e. the vector space spanned by the rows of a parity-check matrix of the code) and then computes the inner products of the received word $\mathbf{y}$ with those parity-check to infer some information about the error $\mathbf{e}$. These methods can be viewed as the coding theoretic analogue of the dual attacks in lattice based cryptography [MR09] and like in coding theory were shown after a sequence of improvements [Alb17, EJK20, GJ21, MAT22] to be able of being competitive with primal attacks, and the crucial improvement came from similar techniques, namely by a splitting strategy. However, it was noticed in [CDMT22a, §3.4] that the i.i.d. Bernoulli model implied by the LPN model for the $\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle$'s is not always accurate but it was conjectured there that the discrepancy between this ideal model and experiments was not severe enough to impact the asymptotic analysis of the decoding based on this model. In lattice based cryptography, the dual attacks were strongly questioned recently in [DP23] by showing that similar assumptions made for analyzing dual attacks were in contradiction with some

theorems in certain regimes or with well-tested heuristics in some other regimes. This raises the issue of really having a theoretical analysis of dual attacks on which we can rely on, not only to have faith on the predictions made with it on dual attacks in code or lattice-based cryptography but also to develop the topic.

It should be noted that this work has already begun in [CDMT22a, Prop. 3.1], where the basic assumption of statistical decoding is proved. Indeed, the very first task is to estimate the "noise" term $\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle$, or simply $\langle \mathbf{e}, \mathbf{h} \rangle$ where $\mathbf{e}$ is of a certain weight and $\mathbf{h}$ a parity-check equation of some weight too. The estimation of the bias $\varepsilon \stackrel{\triangle}{=} \mathrm{bias}(\langle \mathbf{e}, \mathbf{h} \rangle)$ is crucial, the reason being that $1/\varepsilon^2$ is the quantity that governs the complexity of statistical decoding (we need to collect that many samples in order to distinguish random scalar products from scalar products of this form). We have used here the following notation for a binary random variable X:

$$\mathrm{bias}(X) \stackrel{\triangle}{=} \mathbb{P}(X = 0) - \mathbb{P}(X = 1).$$

In [Jab01, Ove06, DT17b] this was done by assuming that $\mathrm{bias}(\langle \mathbf{e}, \mathbf{h} \rangle)$ is the same as the bias of $\langle \mathbf{e}, \mathbf{h}' \rangle$ where $\mathbf{h}'$ is uniformly distributed among the words of the same weight as $\mathbf{h}$. The point is that the bias of the latter can be computed by using Krawtchouk polynomials (see §2.3)

$$\mathrm{bias}\left(\langle \mathbf{e}, \mathbf{h}' \rangle\right) = \frac{K_w^{(n)}\left(|\mathbf{e}|\right)}{\binom{n}{w}}, \tag{1.2}$$

where $|\cdot|$ stands for the Hamming weight and $w = |\mathbf{h}'|$. [CDMT22a, Prop. 3.1] proved that for a random code $\mathscr{C}$ we typically have $\mathrm{bias}(\langle \mathbf{e}, \mathbf{h} \rangle) = \mathrm{bias}(\langle \mathbf{e}, \mathbf{h}' \rangle)(1 + o(1))$ as soon as the number of parity-check equations of weight w is large enough, namely it requires that this quantity is of the form $\omega\big(1/\varepsilon^2\big)$.

1.4 Our Contribution

A formula for the bias. If we simplify a little bit, a Krawtchouk polynomial is essentially decreasing in the weight $|\mathbf{e}|$ [1] and therefore decoding by using a bunch of scalar products $\langle \mathbf{h}, \mathbf{y} \rangle$ is really finding the $\mathbf{x} \in \mathbb{F}_2^s$ such that the $\langle \mathbf{h}, \mathbf{y} \rangle + \langle \mathbf{h}_{\mathscr{P}}, \mathbf{x} \rangle$'s take the value 0 the most often. Indeed, if $\mathbf{x} = \mathbf{e}_{\mathscr{P}}$ then $\langle \mathbf{h}, \mathbf{y} \rangle + \langle \mathbf{h}_{\mathscr{P}}, \mathbf{x} \rangle = \langle \mathbf{h}, \mathbf{y} \rangle + \langle \mathbf{h}_{\mathscr{P}}, \mathbf{e} \rangle = \langle \mathbf{h}_{\mathscr{N}}, \mathbf{e}_{\mathscr{N}} \rangle$ by using that $\langle \mathbf{e}, \mathbf{h} \rangle = \langle \mathbf{e}_{\mathscr{P}}, \mathbf{h}_{\mathscr{P}} \rangle + \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle$. In such a case, we could expect that $\langle \mathbf{h}_{\mathscr{N}}, \mathbf{e}_{\mathscr{N}} \rangle$ is biased towards 0. For the wrong choice of $\mathbf{x}$, we expect that $\langle \mathbf{h}, \mathbf{y} \rangle + \langle \mathbf{h}_{\mathscr{P}}, \mathbf{x} \rangle$ behaves essentially like a Bernoulli random variable of parameter $\frac{1}{2}$ which is independent from $\langle \mathbf{h}_{\mathscr{N}}, \mathbf{e}_{\mathscr{N}} \rangle$. In a nutshell, the RLPN decoder of [CDMT22a] is based on this heuristic together

[1] It is decreasing in the interval $[0, n/2 - \sqrt{w(n-w)}]$ and then even if there are fluctuations (the polynomial has zeros) it behaves like $\mathrm{poly}\,(n) \sin(\alpha) e^{\beta n}$ with an exponent β which is decreasing with $t = |\mathbf{e}|$ and where Proposition 2 shows that there are nearby weights t' and w' for t and w respectively for which the exponential term captures the behavior of $K_{w'}(t')$.

with a fast Fourier transform trick which allows to compute the weights of the vectors $\mathbf{v}(\mathbf{x}) = (\langle \mathbf{h}, \mathbf{y} \rangle + \langle \mathbf{h}_{\mathscr{P}}, \mathbf{x} \rangle)_{\mathbf{h} \in \mathscr{H}}$ all at once for all $\mathbf{x} \in \mathbb{F}_2^s$ where $\mathscr{H}$ is the set of parity-checks we use for decoding. The analysis of the decoder in [CDMT22a] uses basically the aforementioned heuristic on the behavior of the $\mathbf{v}(\mathbf{x})$'s.

However, we have found that we have to be more careful than this for the model of the random variable $\mathrm{bias}(\langle \mathbf{h}, \mathbf{y} \rangle + \langle \mathbf{h}_{\mathscr{P}}, \mathbf{x} \rangle)$ when $\mathbf{h}$ is chosen uniformly at random among the parity-check equations of weight w on $\mathscr{N}$. Our first contribution is indeed to give a general proposition which allows to get a clear picture about the bias of these random variables

Proposition 1. *Let $\mathscr{C}$ be an $[n,k]$-linear code. Let $\mathbf{y} = \mathbf{c} + \mathbf{e}$ be the noisy codeword we want to decode. Let $\mathscr{P}$ and $\mathscr{N}$ be two complementary subsets of $[\![1,n]\!]$ of size s and $n-s$ respectively and such that the restriction $\mathscr{C}_{\mathscr{P}}$ of the codewords of $\mathscr{C}$ to $\mathscr{P}$ has full dimension s. Then there is a unique linear map $\mathbf{R}$ such that $\mathbf{h}_{\mathscr{P}}^\top = \mathbf{R}\mathbf{h}_{\mathscr{N}}^\top$ for any parity-check $\mathbf{h}$. Let $\mathbf{x}$ be a vector of $\mathbb{F}_2^s$ and let $\mathbf{h}$ be sampled uniformly at random among the set $\mathscr{H}_{\mathscr{N}}(w)$ of parity-checks of $\mathscr{C}$ that are of weight w on $\mathscr{N}$. We have*

$$\mathrm{bias}\left(\langle \mathbf{y}, \mathbf{h} \rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle\right) = \frac{1}{2^{k-s}\left|\mathscr{H}_{\mathscr{N}}(w)\right|} \sum_{\mathbf{c}^{\mathscr{N}} \in \mathscr{C}^{\mathscr{N}}} K_w^{(n-s)}\left(\left|(\mathbf{x} + \mathbf{e}_{\mathscr{P}})\mathbf{R} + \mathbf{e}_{\mathscr{N}} + \mathbf{c}^{\mathscr{N}}\right|\right), \tag{1.3}$$

where $\mathscr{C}^{\mathscr{N}}$ is $\mathscr{C}$ shortened in $[\![1,n]\!] \setminus \mathscr{N} = \mathscr{P}$, i.e. $\mathscr{C}^{\mathscr{N}} = \{\mathbf{c}_{\mathscr{N}} : \mathbf{c} \in \mathscr{C},\ \mathbf{c}_{\mathscr{P}} = \mathbf{0}\}$.

A simple corollary of this result (by taking $\mathscr{P} = \emptyset$) is also that

Corollary 1. *Let $\mathscr{C}$ be an $[n,k]$-linear code. Let w be an integer in $[\![0,n]\!]$ and let $\mathbf{e}$ be a vector of $\mathbb{F}_2^n$ sampled uniformly at random among the set $\mathscr{H}(w)$ of parity-checks of $\mathscr{C}$ of weight w. Then*

$$\mathrm{bias}\left(\langle \mathbf{e}, \mathbf{h} \rangle\right) = \frac{1}{2^k\left|\mathscr{H}(w)\right|} \sum_{\mathbf{c} \in \mathscr{C}} K_w^{(n)}\left(|\mathbf{c} + \mathbf{e}|\right). \tag{1.4}$$

This gives a formula of the bias used in standard statistical decoding which holds for *all codes* and not only for *most* codes as is the case of the aforementioned Proposition 3.1 in [CDMT22a]. An analysis of this sum for random codes allows to easily recover this proposition as $2^k\left|\mathscr{H}(w)\right| \approx \binom{n}{w}$ and the sum is easily seen to be dominated by the term $K_w^{(n)}(|\mathbf{e}|)$ in this case. The nice thing about this corollary is that it allows in principle to get a handle on this bias for more specific class of codes.

Analysis of RLPN decoding and a simple correction. Proposition 1 is instrumental in analyzing the RLPN decoder of [CDMT22a]. We will show that the heuristic made for the behavior of $\mathbf{v}(\mathbf{x})$ does not hold, and for the choice of parameters made in the RLPN decoder, with overwhelming probability there exists a bunch of $\mathbf{x}$ in $\mathbb{F}_2^s$ such that $\langle \mathbf{h}, \mathbf{y} \rangle + \langle \mathbf{h}_{\mathscr{P}}, \mathbf{x} \rangle$ is more biased toward zero than $\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle$, the error term associated to the secret vector $\mathbf{e}_{\mathscr{P}}$ we wish to recover. In such a case, $\mathbf{e}_{\mathscr{P}}$ is not the solution to the LPN problem anymore.

This shows that the RLPN decoding algorithm does not work as expected in [CDMT22a]. Fortunately the set of $\mathbf{x}$ for which $\langle \mathbf{h}, \mathbf{y} \rangle + \langle \mathbf{h}_{\mathscr{P}}, \mathbf{x} \rangle$ is more biased toward zero than $\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle$ can be bounded rigorously by using Proposition 1 and turns out to be moderate enough that a simple correction of the RLPN decoder works. It suffices to keep all candidates $\mathbf{x}$ whose $\text{bias}(\langle \mathbf{h}, \mathbf{y} \rangle + \langle \mathbf{h}_{\mathscr{P}}, \mathbf{x} \rangle)$ is above a certain threshold for which we also expect to keep $\mathbf{e}_{\mathscr{P}}$ and check if the candidate is correct by decoding a code in which s positions are removed (those belonging to $\mathscr{P}$) and for which the s positions of the error on which we bet (corresponding to $\mathbf{x}$) allow to decrease by s the dimension of the code. This problem is much easier to solve than the original decoding problem. It can for instance be solved by simple ISD algorithms. This variant of the RLPN decoder can be analyzed rigorously thanks to the aforementioned bound on the number of candidates. It turns out that for the parameters chosen in the RLPN decoder of [CDMT22a], at least in the region where it beats the best ISD decoding algorithms, the number of those candidates times the complexity of an ISD solver on the reduced problem is not more than producing the set of parity-checks of weight w on $\mathscr{N}$ and computing the set of candidates by the fast Fourier technique used in [CDMT22a]. From this we actually infer that the complexity result given in [CDMT22a] actually holds for this variant of the RLPN decoder.

2 Notation and Preliminaries

2.1 Notation

Set, vector and matrix notation. $[\![a, b]\!]$ indicates the closed integer interval between a and b. $\mathbb{F}_2$ is the binary field. $|E|$ is the cardinality of a finite set E. Vectors are indicated by lowercase bold letters $\mathbf{x}$ and matrices by uppercase bold letters $\mathbf{A}$. For a vector $\mathbf{x} = (x_i)_{1 \leq i \leq n}$ and $\mathcal{I} \subset [\![1, n]\!]$, $\mathbf{x}_{\mathcal{I}}$ is given by $\mathbf{x}_{\mathcal{I}} = (x_i)_{i \in \mathcal{I}}$ and $|\mathbf{x}|$ stands for the Hamming weight of $\mathbf{x}$. $\mathcal{S}_w^n \stackrel{\triangle}{=} \{\mathbf{x} \in \mathbb{F}_2^n \ : |\mathbf{x}| = w\}$ is the Hamming sphere of weight w of $\mathbb{F}_2^n$.

Probability and entropy. When $\mathcal{D}$ is a probability distribution we write that $\mathbf{X} \sim \mathcal{D}$ to specify that $\mathbf{X}$ is distributed according to $\mathcal{D}$. More simply, when $\mathscr{H}$ is a set we write that $\mathbf{h} \hookleftarrow \mathscr{H}$ to specify that $\mathbf{h}$ is a random variable uniformly distributed over $\mathscr{H}$. Let $h_2(x) \stackrel{\triangle}{=} -x \log_2(x) - (1-x) \log_2(1-x)$ be the binary entropy and h_2^{-1} its inverse on $[0, \frac{1}{2}]$.

Fourier transform. Let $f : \mathbb{F}_2^n \to \mathbb{R}$ be a function. We define the Fourier transform $\widehat{f} : \mathbb{F}_2^n \to \mathbb{R}$ of f as $\widehat{f}(\mathbf{x}) = \sum_{\mathbf{u} \in \mathbb{F}_2^n} f(\mathbf{u}) (-1)^{\langle \mathbf{x}, \mathbf{u} \rangle}$.

Landau and asymptotic notation. For real valued functions defined over $\mathbb{R}$ or $\mathbb{N}$ we define $o()$, $O()$, $\Omega()$, $\Theta()$, in the usual way. We write that $f = \omega(g)$ when f dominates g asymptotically; that is when $\lim_{x \to \infty} \frac{|f(x)|}{g(x)} = \infty$. We use the

less common notation $\widetilde{O}()$, where $f = \widetilde{O}(g)$ means that $f(x) = O\Big(g(x)\log^k g(x)\Big)$ for some k. We will use this for functions which have an exponential behaviour, say $g(x) = e^{\alpha x}$, in which case $f(x) = \widetilde{O}(g(x))$ means that $f(x) = O(P(x)g(x))$ where P is a polynomial in x. We write that $f(n) = \mathrm{poly}(n)\, g(n)$ when there exist two reals d_1 and d_2 such that $g(n)\, n^{d_2} < f(n) < g(n)\, n^{d_1}$.

2.2 Linear codes

Definition 1. *(Binary linear code and dual code) A binary linear code $\mathscr{C}$ of length n and dimension k is a linear subspace of $\mathbb{F}_2^n$ of dimension k. We say that $\mathscr{C}$ is an $[n,k]$ linear code. We call $R = \frac{k}{n}$ the rate of the code. We denote by $\mathscr{C}^\perp$ the subspace of the vectors orthogonal to $\mathscr{C}$,* i.e. $\mathscr{C}^\perp \stackrel{\triangle}{=} \{\mathbf{d} \in \mathbb{F}_2^n : \langle \mathbf{c}, \mathbf{d}\rangle = 0,\ \forall \mathbf{c} \in \mathscr{C}\}$.

We will also use the following notions of

Definition 2. *(punctured code) For a code $\mathscr{C}$ and a subset $\mathcal{I}$ of code positions, we denote by $\mathscr{C}_\mathcal{I}$ the punctured code obtained from $\mathscr{C}$ by keeping only the positions in I, i.e:*

$$\mathscr{C}_\mathcal{I} = \{\mathbf{c}_\mathcal{I} : \mathbf{c} \in \mathscr{C}\}.$$

Definition 3. *(shortened code) For a code $\mathscr{C}$ and a subset I of code positions, we denote by $\mathscr{C}^\mathcal{I}$ the shortened code obtained as follows:*

$$\mathscr{C}^\mathcal{I} = \{\mathbf{c}_\mathcal{I} : \mathbf{c} \in \mathscr{C} \text{ such that } \mathbf{c}_{[\![1,n]\!]\setminus\mathcal{I}} = \mathbf{0}\}.$$

It is readily seen that for any code $\mathscr{C}$ and any subset of positions $\mathscr{I}$ we have

$$(\mathscr{C}_\mathscr{I})^\perp = \left(\mathscr{C}^\perp\right)^\mathscr{I} \quad \text{and} \quad \left(\mathscr{C}^\mathscr{I}\right)^\perp = \left(\mathscr{C}^\perp\right)_\mathscr{I}. \tag{2.1}$$

2.3 Krawtchouk polynomial

We recall here some properties about Krawtchouk polynomial that will be useful in the article. Many useful properties can be found in [KS21, §2.2]

Definition 4. *(Krawtchouk polynomial) We define the Krawtchouk polynomial $K_w^{(n)}$ of degree w and of order n as $K_w^{(n)}(X) \stackrel{\triangle}{=} \sum_{j=0}^{w} (-1)^j \binom{X}{j}\binom{n-X}{w-j}$.*

The following fact is well known, it gives an alternate expression of the Krawtchouk polynomial (see for instance [vL99, Lemma 5.3.1]) :

Fact 3. *For any $\mathbf{a} \in \mathbb{F}_2^n$,*

$$K_w^{(n)}(|\mathbf{x}|) = \sum_{\mathbf{y}\in\mathbb{F}_2^n : |\mathbf{y}=w} (-1)^{\langle \mathbf{x},\mathbf{y}\rangle}. \tag{2.2}$$

We recall here the summary of some known results about Krawtchouk polynomials made in [CDMT22a].

Proposition 2. *[CDMT22a, Prop. 3.5, Prop. 3.6]*

1. Value at 0. *For all* $0 \leq w \leq n$, $K_w^{(n)}(0) = \binom{n}{w}$.
2. Reciprocity. *For all* $0 \leq t, w \leq n$, $\binom{n}{t} K_w^{(n)}(t) = \binom{n}{w} K_t^{(n)}(w)$.
3. Roots. *The polynomials* $K_w^{(n)}$ *have* w *distinct roots which lie in the interval* $\left[\!\left[n/2 - \sqrt{w(n-w)}, n/2 + \sqrt{w(n-w)}\right]\!\right]$. *The distance between roots is at least* 2 *and at most* $o(n)$.
4. Magnitude in and out the root region. *Let* τ *and* ω *be two reals in* $[0,1]$. *Let* $\omega^\perp \stackrel{\triangle}{=} \frac{1}{2} - \sqrt{\omega(1-\omega)}$, *and let* $z \stackrel{\triangle}{=} \frac{1-2\tau-\sqrt{D}}{2(1-\omega)}$ *where* $D \stackrel{\triangle}{=} (1-2\tau)^2 - 4\omega(1-\omega)$. *Define* $\widetilde{\kappa}(\tau,\omega) \stackrel{\triangle}{=} \begin{cases} \tau \log_2(1-z) + (1-\tau)\log_2(1+z) - \omega \log_2 z & \text{if } \tau \in [0, \omega^\perp] \\ \frac{1-h(\tau)+h(\omega)}{2} & \text{otherwise.} \end{cases}$
   - *4.1. If* $\tau \leq \frac{1}{2} - \sqrt{\omega(1-\omega)}$, *then for all* t *and* w *such that* $\lim_{n\to\infty} \frac{t}{n} = \tau$ *and* $\lim_{n\to\infty} \frac{w}{n} = \omega$ *we have* $K_w^{(n)}(t) = 2^{n(\widetilde{\kappa}(\tau,\omega)+o(1))}$.
   - *4.2. If* $\tau > \frac{1}{2} - \sqrt{\omega(1-\omega)}$, *then there exists* $t(n)$ *and* $w(n)$ *such that* $\lim_{n\to\infty} \frac{t}{n} = \tau$, $\lim_{n\to\infty} \frac{w}{n} = \omega$ *and* $\left|K_w^{(n)}(t)\right| = 2^{n(\widetilde{\kappa}(\tau,\omega)+o(1))}$.

3 An expression for the bias of $\langle \mathbf{y}, \mathbf{h} \rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle$

In this section, we give the basic tool for studying the fundamental quantity manipulated by the RLPN algorithm, namely the bias $\langle \mathbf{y}, \mathbf{h} \rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle$. Let us first rewrite this expression as a scalar product of a vector depending on $\mathbf{x}$ and $\mathbf{e}$ with the restriction $\mathbf{h}_{\mathscr{N}}$ of $\mathbf{h}$ to $\mathscr{N}$. This is given by

Lemma 1. *Let* $\mathscr{C}$ *be an* $[n,k]$, $\mathbf{c} \in \mathscr{C}$ *and* $\mathbf{y} = \mathbf{c} + \mathbf{e}$ *be a noisy codeword. Let* $\mathscr{P}$ *and* $\mathscr{N}$ *be complementary subsets of* $[\![1,n]\!]$ *of size* s *and* $n-s$ *respectively and suppose that* $\dim \mathscr{C}_{\mathscr{P}} = s$. *Then there exists a unique linear map* $\mathbf{R} : \mathbb{F}_2^{n-s} \to \mathbb{F}_2^s$ *such that for all* $\mathbf{h} \in \mathscr{C}^\perp$:

$$\mathbf{h}_{\mathscr{P}}{}^\intercal = \mathbf{R}\, \mathbf{h}_{\mathscr{N}}{}^\intercal. \tag{3.1}$$

Moreover, we have that for all $\mathbf{h} \in \mathscr{C}^\perp$ *and* $\mathbf{x} \in \mathbb{F}_2^s$:

$$\langle \mathbf{y}, \mathbf{h} \rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle = \langle \mathbf{x}\mathbf{R} + \mathbf{e}_{\mathscr{P}}\mathbf{R} + \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle. \tag{3.2}$$

Proof. First, let us show (3.1). Suppose w.l.o.g. that $\mathscr{P} = [\![1,s]\!]$ and $\mathscr{N} = [\![s+1,n]\!]$. Let $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ be a generator matrix of $\mathscr{C}$. Because $\mathscr{C}_{\mathscr{P}}$ is of dimension s there exists an invertible $\mathbf{J} \in \mathbb{F}_2^{k\times k}$ such that

$$\mathbf{JG} = \begin{pmatrix} \mathbf{Id}_s & \mathbf{R} \\ \mathbf{0}_{k-s} & \mathbf{R}' \end{pmatrix}$$

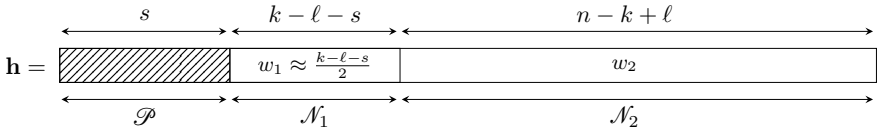
where $\mathbf{R} \in \mathbb{F}_2^{s\times(n-s)}$ and $\mathbf{R}' \in \mathbb{F}_2^{(k-s)\times(n-s)}$. $\mathbf{JG}$ is another generator matrix for $\mathscr{C}$. Therefore for any $\mathbf{h} \in \mathscr{C}^\perp$ we have $\mathbf{JG}\,\mathbf{h}^\intercal = \mathbf{0}$. Since $\mathbf{JG}\mathbf{h}^\intercal = \mathbf{h}_{\mathscr{P}}{}^\intercal + \mathbf{R}\mathbf{h}_{\mathscr{N}}{}^\intercal$, this gives (3.1). It is readily seen that $\mathbf{R}$ seen as a linear map is unique.

Now, let us prove (3.2). Recall that $\langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}_{\mathscr{P}}, \mathbf{h}_{\mathscr{P}} \rangle + \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle$. Thus

$$\begin{aligned}
\langle \mathbf{y}, \mathbf{h} \rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle &= \langle \mathbf{e}_{\mathscr{P}} + \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle + \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle \\
&= \langle \mathbf{e}_{\mathscr{P}} + \mathbf{x}, \left(\mathbf{R}\, \mathbf{h}_{\mathscr{N}}{}^{\mathsf{T}} \right)^{\mathsf{T}} \rangle + \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle \quad \text{(Using (3.1))} \\
&= \langle (\mathbf{e}_{\mathscr{P}} + \mathbf{x})\, \mathbf{R}, \mathbf{h}_{\mathscr{N}} \rangle + \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle \\
&= \langle (\mathbf{e}_{\mathscr{P}} + \mathbf{x})\, \mathbf{R} + \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle.
\end{aligned}$$

□

The RLPN decoder of [CDMT22a] produces in its first step many parity-check equations of weight w on $\mathscr{N}$ by using standard techniques for producing low-weight codewords in a code. More elaborate techniques actually produce parity-check equations which have unbalanced weights, say $\mathscr{N}$ is partitioned into two sets $\mathscr{N}_1$ and $\mathscr{N}_2$ and $\mathbf{h}$ is of weight w_i on $\mathscr{N}_i$ for $i \in \{1, 2\}$ and $w = w_1 + w_2$. Generally if we are looking for parity-check equations of an $[n, k]$ code $\mathscr{C}$, we look for codewords in the dual $\mathscr{C}^{\perp}$ which is of dimension $n - k$, we generally choose $\mathscr{N}_2$ of size slightly larger than $n - k$, say $|\mathscr{N}_2| = n - k + \ell$ and try to minimize the weight w_2 as much as we can on this part, whereas the rest of the parity on $\mathscr{N}$ (i.e. the $\mathscr{N}_1$ part) is just computed linearly from the $\mathbf{h}_{\mathscr{N}_2}$ and is of weight w_2 close to $|\mathscr{N}_1|/2$ (see picture below).



Our main tool for studying the RLPN decoder is now the following expression of the bias by using Lemma 1 and duality with the Poisson formula

Proposition 3. *Let $\mathscr{C}$ be an $[n, k]$ code. Let $\mathbf{y} = \mathbf{c} + \mathbf{e}$ be a noisy codeword (where $\mathbf{c} \in \mathscr{C}$). Let $\mathscr{P}$, $\mathscr{N}_1$ and $\mathscr{N}_2$ be fixed complementary subsets of $[\![1, n]\!]$ of size s, $k - \ell - s$ and $n - k + \ell$ respectively and such that $\dim \mathscr{C}_{\mathscr{P}} = s$. We denote $\mathscr{N} \triangleq \mathscr{N}_1 \cup \mathscr{N}_2$. Then there exists a unique linear map $\mathbf{R} : \mathbb{F}_2^{n-s} \to \mathbb{F}_2^s$ such that for all $\mathbf{h} \in \mathscr{C}^{\perp}$, we have $\mathbf{h}_{\mathscr{P}}{}^{\mathsf{T}} = \mathbf{R}\, \mathbf{h}_{\mathscr{N}}{}^{\mathsf{T}}$. Let $\mathbf{x}$ be a fixed vector of $\mathbb{F}_2^s$ and let $\mathbf{h}$ be a vector sampled uniformly at random among the set $\widetilde{\mathscr{H}}$ of parity-checks of $\mathscr{C}$ that are of weight w_1 on $\mathscr{N}_1$ and w_2 on $\mathscr{N}_2$. We have that:*

$$\operatorname{bias}\left(\langle \mathbf{y}, \mathbf{h} \rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle \right) = \frac{1}{2^{k-s}\, |\widetilde{\mathscr{H}}|} \sum_{\mathbf{c}^{\mathscr{N}} \in \mathscr{C}^{\mathscr{N}}} K_{w_1}^{(k-\ell-s)}(|\mathbf{c}_1|) K_{w_2}^{(n-k+\ell)}(|\mathbf{c}_2|), \tag{3.3}$$

where $\mathbf{c}_1 \triangleq \left((\mathbf{x} + \mathbf{e}_{\mathscr{P}})\mathbf{R} + \mathbf{e}_{\mathscr{N}} + \mathbf{c}^{\mathscr{N}} \right)_{\mathscr{N}_1}$ and $\mathbf{c}_2 \triangleq \left((\mathbf{x} + \mathbf{e}_{\mathscr{P}})\mathbf{R} + \mathbf{e}_{\mathscr{N}} + \mathbf{c}^{\mathscr{N}} \right)_{\mathscr{N}_2}$.

Proof. Let us notice that

$$\begin{aligned}
\text{bias}\left(\langle \mathbf{y}, \mathbf{h}\rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}}\rangle\right) &= \frac{1}{|\widetilde{\mathscr{H}}|} \sum_{\mathbf{h}\in\widetilde{\mathscr{H}}} (-1)^{\langle \mathbf{y},\mathbf{h}\rangle + \langle \mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle} \\
&= \frac{1}{|\widetilde{\mathscr{H}}|} \sum_{\mathbf{h}_{\mathscr{N}}:\mathbf{h}\in\widetilde{\mathscr{H}}} (-1)^{\langle (\mathbf{x}+\mathbf{e}_{\mathscr{P}})\mathbf{R}+\mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}}\rangle} \quad \text{(by Lemma 1)} \\
&= \frac{1}{|\widetilde{\mathscr{H}}|} \sum_{\mathbf{h}_{\mathscr{N}}\in(\mathscr{C}^{\perp})_{\mathscr{N}}} f(\mathbf{h}_{\mathscr{N}}) \\
\text{where } \ f(\mathbf{z}) &\stackrel{\triangle}{=} (-1)^{\langle (\mathbf{x}+\mathbf{e}_{\mathscr{P}})\mathbf{R}+\mathbf{e}_{\mathscr{N}}, \mathbf{z}\rangle} \mathbf{1}_{\{|\mathbf{z}_{\mathscr{N}_1}|=w_1,\, |\mathbf{z}_{\mathscr{N}_2}|=w_2\}}.
\end{aligned}$$

Thanks to (2.1), we have that $(\mathscr{C}^{\perp})_{\mathscr{N}} = \left(\mathscr{C}^{\mathscr{N}}\right)^{\perp}$. By using the Poisson formula (see [MS86, Lemma 2, Ch. 5. §2]), together with $\dim(\mathscr{C}^{\mathscr{N}}) = k - s$, we get

$$\frac{1}{|\widetilde{\mathscr{H}}|} \sum_{\mathbf{h}_{\mathscr{N}}\in(\mathscr{C}^{\mathscr{N}})^{\perp}} f(\mathbf{h}_{\mathscr{N}}) = \frac{1}{2^{k-s}|\widetilde{\mathscr{H}}|} \sum_{\mathbf{c}^{\mathscr{N}}\in\mathscr{C}^{\mathscr{N}}} \widehat{f}(\mathbf{c}^{\mathscr{N}}). \tag{3.4}$$

Let $g(\mathbf{z}_{\mathscr{N}}) \stackrel{\triangle}{=} \mathbf{1}_{\{|\mathbf{z}_{\mathscr{N}_1}|=w_1,\, |\mathbf{z}_{\mathscr{N}_2}|=w_2\}}$. We have

$$\begin{aligned}
\widehat{g}(\mathbf{u}_{\mathscr{N}}) &= \sum_{\mathbf{z}_{\mathscr{N}}:|\mathbf{z}_{\mathscr{N}_1}|=w_1,\, |\mathbf{z}_{\mathscr{N}_2}|=w_2} (-1)^{\langle \mathbf{u}_{\mathscr{N}_1}, \mathbf{z}_{\mathscr{N}_1}\rangle + \langle \mathbf{u}_{\mathscr{N}_2}, \mathbf{z}_{\mathscr{N}_2}\rangle} && (3.5) \\
&= \sum_{\mathbf{z}_{\mathscr{N}_1}\in\mathcal{S}_{w_1}^{k-\ell-s}} (-1)^{\langle \mathbf{u}_{\mathscr{N}_1}, \mathbf{z}_{\mathscr{N}_1}\rangle} \sum_{\mathbf{z}_{\mathscr{N}_2}\in\mathcal{S}_{w_2}^{n-k+\ell}} (-1)^{\langle \mathbf{u}_{\mathscr{N}_2}, \mathbf{z}_{\mathscr{N}_2}\rangle} && (3.6) \\
&= K_{w_1}^{(k-\ell-s)}(|\mathbf{u}_{\mathscr{N}_1}|) K_{w_2}^{(n-k+\ell)}(|\mathbf{u}_{\mathscr{N}_2}|) \quad \text{(by (3))}. && (3.7)
\end{aligned}$$

We finish the proof by noticing that from the definition of f we have

$$\widehat{f}(\mathbf{c}^{\mathscr{N}}) = \widehat{g}(\mathbf{c}^{\mathscr{N}} + (\mathbf{x} + \mathbf{e}_{\mathscr{P}})\mathbf{R} + \mathbf{e}_{\mathscr{N}}).$$

□

Proposition 1 is a special case of this proposition when $\mathscr{N}_2 = \mathscr{N}$ and $\mathscr{N}_1 = \emptyset$.

4 A corrected RLPN algorithm

As we show in Section 5, the RLPN decoding algorithm proposed in [CDMT22a] together with the choice of parameters made there has exponentially many candidates that pass the same test as the right candidate $\mathbf{e}_{\mathscr{P}}$. As already explained in the introduction, there is a very simple way to tackle this issue by simply continuing as if a candidate is the right one. If it is the right candidate, further decoding will succeed and if not, decoding will fail. This leads to the following algorithm.

Algorithm 4.1. corrected RLPN decoder

Input: $\mathbf{y}$, t, $\mathscr{C}$ an $[n,k]$-code
Parameters: s, w_1, w_2, u_1, u_2
Output: $\mathbf{e}$ such that $|\mathbf{e}| = t$ and $\mathbf{y} - \mathbf{e} \in \mathscr{C}$.

1: **function** RLPNDECODE($\mathbf{y}$, $\mathscr{C}$, t)
2: **while** True **do**
3: $(\mathscr{P}, \mathscr{N}_1, \mathscr{N}_2) \xleftarrow{\$} \{\mathscr{P} \sqcup \mathscr{N}_1 \sqcup \mathscr{N}_2 = [\![1, n]\!] \text{ such that } \#\mathscr{P} = s, \#\mathscr{N}_1 = k-\ell-s, \#\mathscr{N}_2 = n-k+\ell\}$
4: $\mathscr{H} \leftarrow \text{CREATE}(N, w_1, \mathscr{N}_1, w_2, \mathscr{N}_2)$
5: $\widehat{f_{\mathbf{y},\mathscr{H}}} \leftarrow \text{FFT}(f_{\mathbf{y},\mathscr{H}})$
6: $\mathcal{S} \leftarrow \{\mathbf{x} \in \mathcal{S}^s_{t-u} : \widehat{f_{\mathbf{y},\mathscr{H}}}(\mathbf{x}) > \frac{\delta}{2} N\}$ ▷ $\delta = \frac{K_{w_1}^{(k-\ell-s)}(u_1)}{\binom{k-\ell-s}{w_1}} \frac{K_{w_2}^{(n-k+\ell)}(u_2)}{\binom{n-k+\ell}{w_2}}$.
7: **if** $|\mathcal{S}| < N^{\max}_{\text{candi}}$ **then**
8: **for** $\mathbf{x} \in \mathcal{S}$ **do**
9: $\mathbf{e}' \leftarrow \text{SOLVE-SUBPROBLEM}(\mathscr{C}, \mathscr{N}_1 \cup \mathscr{N}_2, \mathbf{y}, \mathbf{x}, u)$
10: **if** $\mathbf{e}' \neq \perp$ **then**
11: **return** $\mathbf{e}$ such that $\mathbf{e}_{\mathscr{P}} = \mathbf{x}$ and $\mathbf{e}_{\mathscr{N}} = \mathbf{e}'$
12: **end if**
13: **end for**
14: **end if**
15: **end while**
16: **return** $\perp$ ▷ If no solution found return "fail".
17: **end function**

This algorithm contains the following ingredients:

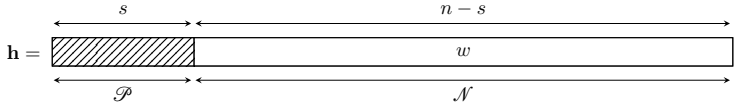
- Line 2. An outer loop repeated until the solution is found. Each time new complementary subsets $\mathscr{P}, \mathscr{N}_1$ and $\mathscr{N}_2$ of $[\![1, n]\!]$ are chosen with the hope of having $\mathbf{e}_{\mathscr{N}_1}$ and $\mathbf{e}_{\mathscr{N}_2}$ of unusually low weight u_1 and u_2 respectively.
- Line 4. A routine $\text{CREATE}(N, w_1, \mathscr{N}_1, w_2, \mathscr{N}_2)$ creating a set $\mathscr{H}$ of parity-check equations uniformly sampled among the $\mathbf{h} \in \mathscr{C}^\perp$ such that $|\mathbf{h}_{\mathscr{N}_1}| = w_1$ and $|\mathbf{h}_{\mathscr{N}_2}| = w_2$. We do not specify how this function is realized here: this is explained in [CDMT22a, §4, §5]. In practice w_1 is chosen as $\frac{k-\ell-s}{2} + o(n)$ (and so is u_1) where the term in $o(n)$ is such that 4.2 of Proposition 2 applies.
- Line 5. A routine $\text{FFT}(f_{\mathbf{y},\mathscr{H}})$ that computes the fast Fourier transform $\widehat{f_{\mathbf{y},\mathscr{H}}}$ of the function $f_{\mathbf{y},\mathscr{H}}$ from $\mathbb{F}_2^s$ to $\mathbb{R}$ defined as $f_{\mathbf{y},\mathscr{H}}(\mathbf{a}) = \sum_{\substack{\mathbf{h} \in \mathscr{H} \\ \mathbf{h}_{\mathscr{P}} = \mathbf{a}}} (-1)^{\langle \mathbf{y}, \mathbf{h} \rangle}$ (and 0 if there is no such $\mathbf{h}$).
- Line 6. We compute a set of candidates $\mathcal{S} = \{\mathbf{x} \in \mathcal{S}^s_{t-u_1-u_2} : \widehat{f_{\mathbf{y},\mathscr{H}}}(\mathbf{x}) > \frac{\delta}{2} N\}$. We expect that if $|\mathbf{e}_{\mathscr{N}_1}| = u_1$ and $|\mathbf{e}_{\mathscr{N}_2}| = u_2$ then $\mathbf{e}_{\mathscr{P}} \in \mathcal{S}$. This set of candidates will also contain other vectors which are false positives.
- A routine $\text{SOLVE-SUBPROBLEM}(\mathscr{C}, \mathscr{N}, \mathbf{y}, \mathbf{x}, u)$ that allows us to verify if a candidate $\mathbf{x} \in \mathcal{S}$ is a false positive or $\mathbf{e}_{\mathscr{P}}$. This routine solves a decoding problem on the code $\mathscr{C}$ shortened on $\mathscr{N}$, namely $\mathscr{C}^{\mathscr{N}}$. More specifically, supposing without loss of generality that $\mathscr{P} = [\![1, s]\!]$, $\mathscr{N} = \mathscr{N}_1 \cup \mathscr{N}_2 = [\![s+1, n]\!]$ and $\mathscr{C}_{\mathscr{P}}$ is of full rank dimension s we can compute $\mathbf{G}$ a generator matrix

of $\mathscr{C}$ of the form $\mathbf{G} = \begin{pmatrix} \mathbf{Id}_s & \mathbf{R} \\ \mathbf{0}_{k-s} & \mathbf{R}' \end{pmatrix}$ by applying a partial Gaussian elimination on a generator matrix of $\mathscr{C}$. Then Solve-SubProblem$(\mathscr{C}, \mathscr{N}, \mathbf{y}, \mathbf{x}, u)$ decodes at distance u the word $\mathbf{y}' \stackrel{\triangle}{=} \mathbf{y}_{\mathscr{N}} - (\mathbf{y}_{\mathscr{P}} - \mathbf{x})\mathbf{R}$ onto the code $\mathscr{C}^{\mathscr{N}}$ of generator matrix $\mathbf{R}'$. This routine is expected to return $\mathbf{e}_{\mathscr{N}}$ if $\mathbf{x} = \mathbf{e}_{\mathscr{P}}$ and to fail (return $\perp$) in case $\mathbf{x} \neq \mathbf{e}_{\mathscr{P}}$.

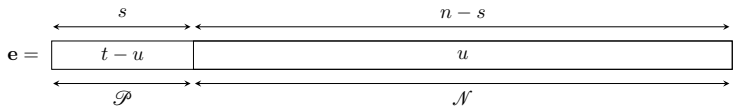
There are two changes with respect to RLPN:

- Line 6, 8. Originally RLPN took $\mathbf{x} \in \mathbb{F}_2^s$ that maximizes $\widehat{f_{\mathbf{y},\mathscr{H}}}(\mathbf{x})$ and checked if $\mathbf{x}$ met the threshold $\widehat{f_{\mathbf{y},\mathscr{H}}}(\mathbf{x}) > \frac{\delta}{2}N$. If the threshold was met, then it asserted that $\mathbf{x}$ equals $\mathbf{e}_{\mathscr{P}}$. In the corrected RLPN we (i) only consider the vectors of $\mathbb{F}_2^s$ of weight $t-u$, (ii) compute the set $\mathcal{S}$ of all vectors $\mathbf{x} \in \mathbb{F}_2^s$ of weight $t-u$ such that $\widehat{f_{\mathbf{y},\mathscr{H}}}(\mathbf{x}) > \frac{\delta}{2}N$. We then solve the resulting decoding problem on $\mathscr{C}^{\mathscr{N}}$ for each $\mathbf{x} \in \mathcal{S}$. Notice here that it can happen, depending on the parameters that $\mathcal{S}$ is of exponential size.
- Line 7. We discard $\mathcal{S}$ if it is bigger than $N_{\text{candi}}^{\max}$. This number is chosen to be of order $\max\left(\widetilde{O}(\mathbb{E}[\mathcal{S}]), 1\right)$ so that a proportion $1 - o(1)$ of iterations passes the test and a proportion $o(1)$ of iterations is discarded. This allows us to bound the number of calls made to Solve-SubProblem. This step is mainly useful for the proof of complexity of the algorithm in §5.

This algorithm is easily transformed into its simplified and less efficient counterpart spliting the vectors in two parts $\mathscr{P}$ and $\mathscr{N}$ ([CDMT22a, §3]). More precisely the "simplified" corrected RLPN is taking $\ell = k - s$ leading to $\mathscr{N}_1 = \emptyset$, $w_1 = 0$, $u_1 = 0$. To simplify notation we define $\mathscr{N} \stackrel{\triangle}{=} \mathscr{N}_2$, $w \stackrel{\triangle}{=} w_2$, $u \stackrel{\triangle}{=} u_2$. Thus in this simplified version we compute parity-checks $\mathbf{h} \in \mathscr{C}^{\perp}$ with this shape:



and make the bet that the error vector $\mathbf{e}$ (of weight t) has this shape:



5 Analysis of corrected RLPN when the support is split in two parts

Algorithm 4.1 is analyzed here in its simplest form, namely when the support is split in two parts $\mathscr{P}$ and $\mathscr{N}$. In all this section, we call $\widetilde{\mathscr{H}}$ the set of *all* parity-checks of weight w on $\mathscr{N}$, as such the set $\mathscr{H}$ (Line 4 of Algorithm 4.1) is a random subset of $\widetilde{\mathscr{H}}$ of size N elements, i.e. we use the following notation

Notation 4.

$$\widetilde{\mathscr{H}} \triangleq \{\mathbf{h} \in \mathscr{C}^{\perp} \,:\, |\mathbf{h}_{\mathscr{N}}| = w\} \tag{5.1}$$

$$N \triangleq |\mathscr{H}| \quad \textit{(number of parity-check equations used in Alg. 4.1)}, \tag{5.2}$$

$$\delta \triangleq \frac{K_w^{(n-s)}(u)}{\binom{n-s}{w}}. \tag{5.3}$$

By noticing that $\left(\mathscr{C}^{\perp}\right)_{\mathscr{N}}$ is an $[n-s, n-k]$-linear code we have the following useful fact regarding the distribution of the size of $\widetilde{\mathscr{H}}$ when $\mathscr{C}$ is a random $[n,k]$-linear code.

Fact 5. *[Bar97, Lem. 1.1, §1.3]*

$$\mathbb{E}_{\mathscr{C}}\left(\left|\widetilde{\mathscr{H}}\right|\right) = \frac{\binom{n-s}{w}}{2^{k-s}}, \tag{5.4}$$

$$\mathbf{Var}_{\mathscr{C}}\left(\left|\widetilde{\mathscr{H}}\right|\right) \leq \frac{\binom{n-s}{w}}{2^{k-s}}. \tag{5.5}$$

We assume from now on that the parameters meet the RLPN constraints so that the algorithm of [CDMT22a] works, namely that k, w, s, u and N verify:

Parameter constraint 6. *The expected size of $\widetilde{\mathscr{H}}$ (the number of parity-checks of weight w on $\mathscr{N}$) and N verify*

$$\frac{\binom{n-s}{w}}{2^{k-s}} = \frac{f(n)}{\delta^2} \quad \textit{where } f(n) = \omega(n^5), \tag{5.6}$$

$$N = \frac{g(n)}{\delta^2} \quad \textit{where } g(n) = \omega(n). \tag{5.7}$$

Remark 1. Note that (5.6) is just slightly stronger than the constraint given in the original analysis [CDMT22a, Prop. 3.9] where the right-hand term is $\frac{\omega(n)}{\delta^2}$. This is needed here in Proposition 5 in the complexity analysis.

5.1 Correctness of Algorithm 4.1

In this section we show that we expect to find the error $\mathbf{e}$ of weight t when the bet on $\mathbf{e}_{\mathscr{N}}$ is valid, *i.e.* $|\mathbf{e}_{\mathscr{N}}| = u$. In this case, we show that we expect with high probability that $\mathbf{e}_{\mathscr{P}} \in \mathcal{S}$. Then, calling Solve-SubProblem on $\mathbf{e}_{\mathscr{P}}$ returns the rest of the secret vector $\mathbf{e}_{\mathscr{N}}$. It is readily seen that the probability over the choice of $\mathscr{P}$ and $\mathscr{N}$ complementary in $[\![1, n]\!]$ of size s and $n-s$ respectively that the bet is valid is given by $P_{\text{Succ}} = \frac{\binom{s}{t-u}\binom{n-s}{u}}{\binom{n}{t}}$. More formally,

Proposition 4. *(Correctness) After at most $N_{\text{iter}} = \omega\left(\frac{\binom{s}{t-u}\binom{n-s}{u}}{\binom{n}{t}}\right)$ executions of the outer loop (Line 2) of Algorithm 4.1, the algorithm outputs with probability $1-o(1)$ over the choices of $\mathscr{C}$ and $\mathscr{H}$, an $\mathbf{e} \in \mathbb{F}_2^n$ of weight t such that $\mathbf{y}-\mathbf{e} \in \mathscr{C}$.*

It is essentially the same proof as in [CDMT22a] but without using any assumptions. The proof is given in §C.1 of the appendix.

5.2 Complexity analysis of Algorithm 4.1

This subsection gives the complexity analysis of the corrected RLPN algorithm when applied to a random $[n,k]$-code $\mathscr{C}$.

Estimating the expected number of candidates. The key part of this analysis relies on estimating the size of $\mathcal{S}$ defined by

$$\mathcal{S} \triangleq \{\mathbf{x} \in \mathcal{S}_{t-u}^{s} \ : \widehat{f_{\mathbf{y},\mathscr{H}}}(\mathbf{x}) > \frac{\delta}{2} N\}.$$

The final formula for the upper-bound of $\mathbb{E}_{\mathscr{C},\mathscr{H}}(|\mathcal{S}|)$ is given in Proposition 5. We give here an outline of the proof.

Step 1. Firstly, by definition $\widehat{f_{\mathbf{y},\mathscr{H}}}(\mathbf{x}) \triangleq \sum_{\mathbf{h}\in\mathscr{H}} (-1)^{\langle \mathbf{y},\mathbf{h}\rangle + \langle \mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle}$ and $N \triangleq |\mathscr{H}|$ thus it is readily seen that we have the following fact

Fact 7.

$$\widehat{f_{\mathbf{y},\mathscr{H}}}(\mathbf{x}) = N \operatorname{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}(\langle \mathbf{y},\mathbf{h}\rangle + \langle \mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle).$$

Notice how the distribution of $\operatorname{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}(\langle \mathbf{y},\mathbf{h}\rangle + \langle \mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle)$ is independent of $\mathbf{x}$ as long as $\mathbf{x} \neq \mathbf{e}_{\mathscr{P}}$ thus we can write using the linearity of the expectation that

$$\begin{aligned}\mathbb{E}_{\mathscr{C},\mathscr{H}}(|\mathcal{S} \setminus \{\mathbf{e}_{\mathscr{P}}\}|) &\leq \binom{s}{t-u} \mathbb{P}_{\mathscr{C},\mathscr{H}}\left(\widehat{f_{\mathbf{y},\mathscr{H}}}(\mathbf{x}) > N\delta/2\right) \\ &= \binom{s}{t-u} \mathbb{P}_{\mathscr{C},\mathscr{H}}(\operatorname{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}(\langle \mathbf{y},\mathbf{h}\rangle + \langle \mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle) > \delta/2).\end{aligned}$$

We then relate $\mathbb{P}_{\mathscr{C},\mathscr{H}}(\operatorname{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}(\langle \mathbf{y},\mathbf{h}\rangle + \langle \mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle) > \delta/2)$ with the probability that the bias is above $\delta/4$ when we choose $\mathbf{h}$ uniformly at random among the whole set $\widetilde{\mathscr{H}}$ of parity-check equations to get

$$\mathbb{E}_{\mathscr{C},\mathscr{H}}(|\mathcal{S} \setminus \{\mathbf{e}_{\mathscr{P}}\}|) \leq \binom{s}{t-u} O\left(\mathbb{P}_{\mathscr{C}}\left(\operatorname{bias}_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}(\langle \mathbf{y},\mathbf{h}\rangle + \langle \mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle) > \frac{\delta}{4}\right)\right). \tag{5.8}$$

Step 2. The point is now that Proposition 1 can be invoked to get

$$\operatorname{bias}_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}(\langle \mathbf{y},\mathbf{h}\rangle + \langle \mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle) = \frac{1}{2^{k-s}|\widetilde{\mathscr{H}}|} \sum_{i=0}^{n-s} N_i K_w^{(n-s)}(i) \tag{5.9}$$

where N_i is the number of codewords in the code $(\mathbf{x}+\mathbf{e}_{\mathscr{P}})\mathbf{R} + \mathbf{e}_{\mathscr{N}} + \mathscr{C}^{\mathscr{N}}$ of weight i. The crucial argument is now used, it is a centering trick based on the fact that $\sum_{i=0}^{n-s} \bar{N}_i K_w^{(n-s)}(i) = 0$ where $\bar{N}_i \triangleq \mathbb{E}(N_i)$. This is used to upper-bound the probability appearing in (5.8) by

$$O(n) \max_{i=0\ldots n-s} \mathbb{P}_{\mathscr{C}}\left(|N_i - \mathbb{E}_{\mathscr{C}}(N_i)| > \left|\frac{K_w^{(n-s)}(u)}{K_w^{(n-s)}(i)}\right| \frac{1}{\Theta(n)}\right).$$

Step 3. We then bound the previous probability by distinguishing two cases for $i \in [\![0, n-s]\!]$:
(i) when i is not too big compared to u ($i < u + o(n)$) the previous probability is upper-bounded by $\mathbb{E}_{\mathscr{C}}(N_i) = \frac{\binom{n-s}{i}}{2^{n-k}}$;
(ii) otherwise the previous probability is bounded by some $2^{-\omega(n)}$.
Unfortunately, the known tail bounds about the weight enumerators of a random code will be insufficient to prove our result. Thus, to make our analysis tractable we will model the weight enumerators as Poisson variables, namely:

Assumption 8.

$$N_i \sim \text{Poisson}\left(\frac{\binom{n-s}{i}}{2^{n-k}}\right).$$

Experimental results concerning the validity of this assumption can be found in Appendix D. All in all, we prove the following proposition giving an upper bound on $\mathbb{E}(|\mathcal{S}|)$.

Proposition 5. *We have under Assumption 8 that:*

$$\mathbb{E}_{\mathscr{C},\mathscr{H}}(|\mathcal{S} \setminus \{\mathbf{e}_{\mathscr{P}}\}|) = \widetilde{O}\left(\binom{s}{t-u}\frac{\binom{n-s}{u}}{2^{n-k}}\right). \tag{5.10}$$

This proposition is proved in Appendix §C.3. Note that there is also in Appendix §C.2 a simple lower bound on this quantity which is of the right order and which gives an insight where this bound comes from.

Complexity of the algorithm. The complexity of this variant of RLPN decoding follows on the spot from the formula of the complexity of the original RLPN algorithm [CDMT22a, Prop. 3.10]) where we add an extra cost (Line 10 to 14) per iteration which is equal to $|\mathcal{S}|$ times the cost of a call to Solve-SubProblem. From this we readily deduce that the original RLPN original complexity claims hold as long as

Proposition 6. *Let $T_{solve\text{-}subproblem}(n-s, k-s, u)$ be the complexity of* Solve-SubProblem *for decoding u errors in an $[n-s, k-s]$ linear code. Suppose that*

$$\max\left(\binom{s}{t-u}\frac{\binom{n-s}{u}}{2^{n-k}}, 1\right) T_{solve\text{-}subproblem}(n-s, k-s, u) \leq \widetilde{O}(2^s) \tag{5.11}$$

Then, under Model 8, the expected complexity of the corrected RLPN algorithm is a $\widetilde{O}()$ of the RLPN claimed complexity made in [CDMT22a, Prop 3.10].

The original asymptotic parameters [2] of RLPN [CDMT22a, §4.1] satisfy for all rates the asymptotic counterpart of (5.11) when Solve-SubProblem uses

[2] `https://github.com/tillich/RLPNdecoding/blob/master/supplementaryMaterial/RLPN_Dumer89.csv`

Dumer's decoder [Dum91] recalled in Appendix B. This can be verified from the supplementary material GitHub page [3]. We give in Appendix §C.4 further results about the complexity of Algorithm 4.1.

6 Concluding Remarks

We have provided here tools for removing all the independence assumptions used for analyzing the most recent (and powerful) dual attack in coding theory [CDMT22a]. Even if this decoder does not work as predicted in [CDMT22a] we provide here an analysis of the number of candidates in $\mathcal{S}$ provided by the slightly modified Algorithm 4.1 which shows that under a very mild assumption on the N_i's (Assumption 8) the original complexity claims are indeed valid for the modified algorithm. Assumption 8 is only used to show large deviation results for the number of codewords N_i of weight i in a random code. Proving rigourous tail bounds on N_i would then give a complete proof of the complexity of Algorithm 4.1. It is also clear that the fundamental tool, the duality result (Proposition 1) carries over straightforwardly for studying dual attacks in lattice based cryptography and so does part of our approach which removed the independence assumptions.

Acknowledgement

We would like to express our thanks to Thomas Debris-Alazard for the insightful discussions. The work of C. Meyer-Hilfiger was funded by the French Agence de l'innovation de défense and by Inria. The work of J-P. Tillich was funded by the French Agence Nationale de la Recherche through the France 2023 ANR project ANR-22-PETQ-0008 PQ-TLS.

References

Alb17. Martin R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 103–129, 2017.

Bar97. Alexander Barg. Complexity issues in coding theory. *Electronic Colloquium on Computational Complexity*, October 1997.

BEL10. V. Blinovsky, U. Erez, and S. Litsyn. Weight distribution moments of random linear/coset codes. *Designs, Codes and Cryptography*, 57:127–138, 2010.

[3] `https://github.com/meyer-hilfiger/Rigorous-Foundations-for-Dual-Attacks-in-Coding-Theory`

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

BLP11. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: ball-collision decoding. In *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *LNCS*, pages 743–760, 2011.

BM17. Leif Both and Alexander May. Optimizing BJMM with Nearest Neighbors: Full Decoding in $2^{2/21n}$ and McEliece Security. In *WCC Workshop on Coding and Cryptography*, September 2017.

CDMT22a. Kevin Carrier, Thomas Debris-Alazard, Charles Meyer-Hilfiger, and Jean-Pierre Tillich. Statistical decoding 2.0: Reducing decoding to LPN. In *Advances in Cryptology - ASIACRYPT 2022*, LNCS. Springer, 2022.

CDMT22b. Kevin Carrier, Thomas Debris-Alazard, Charles Meyer-Hilfiger, and Jean-Pierre Tillich. Statistical decoding 2.0: Reducing decoding to LPN. supplementary material: Asymptotic parameters and complexities, 2022. https://github.com/tillich/RLPNdecoding/tree/master/supplementaryMaterial.

DP23. Léo Ducas and Ludo N. Pulles. Does the dual-sieve attack on learning with errors even work? *IACR Cryptol. ePrint Arch.*, page 302, 2023.

DT17a. Thomas Debris-Alazard and Jean-Pierre Tillich. Statistical decoding. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT 2017*, pages 1798–1802, Aachen, Germany, June 2017.

DT17b. Thomas Debris-Alazard and Jean-Pierre Tillich. Statistical decoding. preprint, January 2017. arXiv:1701.07416.

Dum86. Ilya Dumer. On syndrome decoding of linear codes. In *Proceedings of the 9th All-Union Symp. on Redundancy in Information Systems, abstracts of papers (in russian), Part 2*, pages 157–159, Leningrad, 1986.

Dum91. Ilya Dumer. On minimum distance decoding of linear codes. In *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory*, pages 50–52, Moscow, 1991.

EJK20. Thomas Espitau, Antoine Joux, and Natalia Kharchenko. On a dual/hybrid approach to small secret LWE - A dual/enumeration technique for learning with errors and application to security estimates of FHE schemes. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *Progress in Cryptology - INDOCRYPT 2020 - 21st International Conference on Cryptology in India, Bangalore, India, December 13-16, 2020, Proceedings*, volume 12578 of *Lecture Notes in Computer Science*, pages 440–462. Springer, 2020.

FS09. Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In M. Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 88–105. Springer, 2009.

GJ21. Qian Guo and Thomas Johansson. Faster dual lattice attacks for solving LWE with applications to CRYSTALS. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part IV*, volume 13093 of *Lecture Notes in Computer Science*, pages 33–62. Springer, 2021.

Jab01. Abdulrahman Al Jabri. A statistical decoding algorithm for general linear block codes. In Bahram Honary, editor, *Cryptography and coding. Pro-*

ceedings of the 8th IMA International Conference, volume 2260 of *LNCS*, pages 1–8, Cirencester, UK, December 2001. Springer.

KS21. Naomi Kirshner and Alex Samorodnitsky. A moment ratio bound for polynomials and some extremal properties of krawchouk polynomials and hamming spheres. *IEEE Trans. Inform. Theory*, 67(6):3509–3541, 2021.

LM19. N. Linial and J. Mosheiff. On the weight distribution of random binary linear codes. *Random Structures and Algorithms*, 56:5–36, 2019.

MAT22. MATZOV. Report on the Security of LWE: Improved Dual Lattice Attack, April 2022.

MMT11. Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $O(2^{0.054n})$. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 107–124. Springer, 2011.

MO15. Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 203–228. Springer, 2015.

MR09. Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

MS86. Florence J. MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes*. North–Holland, Amsterdam, fifth edition, 1986.

Ove06. Raphael Overbeck. Statistical decoding revisited. In Reihaneh Safavi-Naini Lynn Batten, editor, *Information security and privacy : 11th Australasian conference, ACISP 2006*, volume 4058 of *LNCS*, pages 283–294. Springer, 2006.

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

Ste88. Jacques Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *LNCS*, pages 106–113. Springer, 1988.

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

Appendix

A Complexity of [Dum86] to compute low weight parity-checks

We give here the asymptotic complexity of one of the method devised in [CDMT22a, §4.1] to produce low weight parity-checks.

Proposition 7. *Asymptotic complexity of Dumer's [Dum86] method to compute low weight parity-checks. Let* $R \stackrel{\triangle}{=} \lim_{n\to\infty} \frac{k}{n}, \omega \stackrel{\triangle}{=} \lim_{n\to\infty} \frac{w}{n}$. *The asymptotic time and space complexities of Dumer's [Dum86] method to compute and store* $2^{n\,(h_2(\omega)-R+o(1))}$ *parity-checks are in* $2^{n\,(\alpha+o(1))}$ *where* $\alpha \stackrel{\triangle}{=} \max\left(\frac{h(\omega)}{2}, h_2(\omega)-R\right)$.

B [Dum91] ISD Decoder

Proposition 8 (Asymptotic time complexity of ISD Decoder [Dum91]). *Let* $R \stackrel{\triangle}{=} \lim_{n\to\infty} \frac{k}{n}, \tau \stackrel{\triangle}{=} \lim_{n\to\infty} \frac{t}{n}$ *and suppose that* $\tau \leq h_2^{-1}(1-R)$. *Let* ℓ *and* w *be two (implicit) parameters of the algorithm and define* $\lambda \stackrel{\triangle}{=} \lim_{n\to\infty} \frac{\ell}{n}, \omega \stackrel{\triangle}{=} \lim_{n\to\infty} \frac{w}{n}$. *The time and space complexities of [Dum91] decoder to decode a word at distance* t *in an* $[n,k]$ *linear code are given by* $2^{n\,(\alpha+o(1))}$ *and* $2^{n\,(\beta+o(1))}$ *respectively where*

$$\alpha \stackrel{\triangle}{=} \pi + \max\left(\frac{R+\lambda}{2} h_2\left(\frac{\omega}{R+\lambda}\right), (R+\lambda)\, h_2\left(\frac{\omega}{R+\lambda}\right) - \lambda\right), \tag{B.1}$$

$$\pi \stackrel{\triangle}{=} h_2(\tau) - (1-R-\lambda) h_2\left(\frac{\tau-\omega}{1-R-\lambda}\right) - (R+\lambda) h_2\left(\frac{\omega}{R+\lambda}\right), \tag{B.2}$$

$$\beta \stackrel{\triangle}{=} \frac{R+\lambda}{2} h_2\left(\frac{\omega}{R+\lambda}\right). \tag{B.3}$$

Moreover λ *and* ω *must verify the following constraints:*

$$0 \leq \lambda \leq 1-R, \qquad \max(R+\lambda+\tau-1, 0) \leq \omega \leq \min(\tau, R+\lambda).$$

C Proofs and Results Corresponding to Section 5

C.1 Proof of Proposition 4

The aim of this subsection is to prove

Proposition 4. *(Correctness) After at most* $N_{\text{iter}} = \omega\left(\frac{\binom{s}{t-u}\binom{n-s}{u}}{\binom{n}{t}}\right)$ *executions of the outer loop (Line 2) of Algorithm 4.1, the algorithm outputs with probability* $1-o(1)$ *over the choices of* $\mathscr{C}$ *and* $\mathscr{H}$, *an* $\mathbf{e} \in \mathbb{F}_2^n$ *of weight* t *such that* $\mathbf{y}-\mathbf{e} \in \mathscr{C}$.

It is readily seen that when $N_{\text{iter}} = \omega\left(\frac{\binom{s}{t-u}\binom{n-s}{u}}{\binom{n}{t}}\right)$ there exists, with probability $1-o(1)$, an iteration such that $|\mathbf{e}_{\mathscr{N}}| = u$. Let us consider such an iteration and show that $\mathbf{e}_{\mathscr{P}} \in \mathcal{S}$ with high probability. Recall that

$$\mathcal{S} \stackrel{\triangle}{=} \{\mathbf{x} \in \mathcal{S}^s_{t-u} \ : \widehat{f_{\mathbf{y},\mathscr{H}}}(\mathbf{x}) > \frac{\delta}{2} N\}$$

and that from Fact 7 we have that

$$\widehat{f_{\mathbf{y},\mathscr{H}}}(\mathbf{x}) = N \operatorname{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}\left(\langle \mathbf{y}, \mathbf{h}\rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}}\rangle\right).$$

Now, using the fact that

$$\langle \mathbf{y}, \mathbf{h}\rangle + \langle \mathbf{e}_{\mathscr{P}}, \mathbf{h}_{\mathscr{P}}\rangle = \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}}\rangle,$$

we only have to lower bound the term

$$\mathbb{P}_{\mathscr{C},\mathscr{H}}\left(\operatorname{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}\left(\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}}\rangle\right) > \frac{\delta}{2}\right)$$

to lower bound the probability that $\mathbf{e}_{\mathscr{P}}$ belong to $\mathcal{S}$. The only known results we have regarding the bias of $\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}}\rangle$ is when $\mathbf{h}$ is sampled in $\widetilde{\mathscr{H}}$ (see [CDMT22a, Prop. 3.1] and proposition 1). But, because $\mathscr{H}$ is a random subset of N (where N is lower bounded in Parameter constraint 6) elements of the set $\widetilde{\mathscr{H}}$, the distribution of $\operatorname{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}\left(\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}}\rangle\right)$ is relatively close to the distribution of $\operatorname{bias}_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}\left(\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}}\rangle\right)$. Namely, we have:

Lemma 2. *For any constant $c > 0$:*

$$\mathbb{P}_{\mathscr{C},\mathscr{H}}\left(\left|\operatorname{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}\left(\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}}\rangle\right) - \operatorname{bias}_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}\left(\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}}\rangle\right)\right| \geq \delta\, c\right) \leq 2^{-\omega(n)}$$

Proof. The proof is available in the eprint version of the paper. □

And thus, as a corollary we get the following lower bound on our probability:

Corollary 2. *We have that*

$$\mathbb{P}_{\mathscr{C},\mathscr{H}}\left(\operatorname{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}\left(\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}}\rangle\right) > \frac{\delta}{2}\right) \geq (1-e^{-\omega(n)})\, \mathbb{P}_{\mathscr{C}}\left(\operatorname{bias}_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}\left(\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}}\rangle\right) > \frac{\delta}{1.5}\right).$$

Now, recall that we supposed that the iteration considered is such that $|\mathbf{e}_{\mathscr{N}}| = u$. Moreover, from Condition (5.7) in Parameter constraint 6 we have that $N = \omega(\frac{n}{\delta^2})$. Thus, a direct application of Proposition [CDMT22a, Prop. 3.1] gives us that with probability $1-o(1)$ over the choice of $\mathscr{C}$, we have

$$\operatorname{bias}_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}\left(\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}}\rangle\right) = \delta\,(1+o(1)).$$

And thus we have that:

$$\mathbb{P}_{\mathscr{C}}\left(\operatorname{bias}_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}\left(\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}}\rangle\right) > \frac{\delta}{1.5}\right) = 1 - o(1).$$

This concludes the proof that $\mathbf{e}_{\mathscr{P}}$ belongs to $\mathcal{S}$ with probability $1-o(1)$ which in turns proves the correctness of Proposition 4.

C.2 A Simple Lower Bound

It turns out that we could easily compute a lower bound on the size of $\mathcal{S}$ using Lemma 1 altogether with a slight adaptation of Proposition [CDMT22a, Prop. 3.1]. Indeed, recall that in Lemma 1 we proved that for a parity-check $\mathbf{h}$:

$$\langle \mathbf{y}, \mathbf{h} \rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle = \langle (\mathbf{x} + \mathbf{e}_{\mathscr{P}}) \mathbf{R} + \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle, \tag{C.1}$$

$$= \langle (\mathbf{x} + \mathbf{e}_{\mathscr{P}}) \mathbf{R} + \mathbf{e}_{\mathscr{N}} + \mathbf{c}^{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle, \quad \forall \mathbf{c}^{\mathscr{N}} \in \mathscr{C}^{\mathscr{N}}, \tag{C.2}$$

where in the last line we used the fact that $\mathbf{h}_{\mathscr{N}} \in \left(\mathscr{C}^{\mathscr{N}}\right)^{\perp}$. Thus if there exists $\mathbf{c}^{\mathscr{N}} \in \mathscr{C}^{\mathscr{H}}$ such that

$$\left| (\mathbf{x} + \mathbf{e}_{\mathscr{P}}) \mathbf{R} + \mathbf{e}_{\mathscr{N}} + \mathbf{c}^{\mathscr{N}} \right| \leq u. \tag{C.3}$$

then with high probability

$$\mathrm{bias}_{\mathbf{h} \hookleftarrow \widetilde{\mathscr{H}}} \left(\langle (\mathbf{x} + \mathbf{e}_{\mathscr{P}}) \mathbf{R} + \mathbf{e}_{\mathscr{N}} + \mathbf{c}^{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle \right) \geq \delta(1 + o(1)).$$

As a matter of fact, from Parameter constraint 6 we have that $N = \omega\left(\frac{n}{\delta^2}\right)$, and, because $K_w^{(n-s)}$ is decreasing in this range, we can use a slight adaptation of Proposition [CDMT22a, Prop. 3.1] to show this point. And thus with high probability $\mathbf{x} \in \mathcal{S}$. We can give a lower bound on the number of $\mathbf{x}$ verifying Condition (C.3) by counting the number of codewords of weight lower than u in the following code:

$$\{ (\mathbf{x} + \mathbf{e}_{\mathscr{P}}) \mathbf{R} + \mathbf{e}_{\mathscr{N}} + \mathbf{c}^{\mathscr{N}} \ : (\mathbf{x}, \mathbf{c}^{\mathscr{N}}) \in \mathcal{S}_{t-u}^{s} \times \mathscr{C}^{\mathscr{N}} \}.$$

This is a random code of length $n - s$ and with a maximum size of $\left|\mathcal{S}_{t-u}^{s}\right| \left|\mathscr{C}^{\mathscr{N}}\right|$. Thus we expect that there are at most

$$\widetilde{O}\left(\left|\mathcal{S}_{t-u}^{s}\right| \left|\mathscr{C}^{\mathscr{N}}\right| \frac{\binom{n-s}{u}}{2^{n-s}} \right) = \widetilde{O}\left(\binom{s}{t-u} \frac{\binom{n-s}{u}}{2^{n-k}} \right)$$

codewords of weight lower than u in the previous code, giving us a lower bound on the expected size of $\mathcal{S}$. This lower-bound actually matches up to polynomial terms the upper-bound appearing in Proposition 5.

C.3 Proof of Proposition 5

Let us first recall Proposition 5

Proposition 5. *We have under Assumption 8 that:*

$$\mathbb{E}_{\mathscr{C},\mathscr{H}} \left(|\mathcal{S} \setminus \{\mathbf{e}_{\mathscr{P}}\}| \right) = \widetilde{O}\left(\binom{s}{t-u} \frac{\binom{n-s}{u}}{2^{n-k}} \right). \tag{5.10}$$

To ease up our analysis, we will suppose in the following that the predicate $P\left(\widetilde{\mathscr{H}}\right)$ defined as:

$$P\left(\widetilde{\mathscr{H}}\right) \text{ : " } \left|\widetilde{\mathscr{H}}\right| \geq \frac{\mathbb{E}_{\mathscr{C}}\left(\left|\widetilde{\mathscr{H}}\right|\right)}{2}\text{"} \tag{C.4}$$

is true and we will only compute the value of $\mathbb{E}_{\mathscr{C},\mathscr{H}}\left(|\mathcal{S}| \;\middle|\; P\left(\widetilde{\mathscr{H}}\right)\right)$. We can show using the Bienaymé-Tchebychev inequality that $P\left(\widetilde{\mathscr{H}}\right)$ is true with probability $1-o(1)$. However, contrary to the previous supposition that $\mathscr{C}_{\mathscr{P}}$ is of full rank dimension s, in general, we have no way of verifying in polynomial time if $P\left(\widetilde{\mathscr{H}}\right)$ is true or not, thus we cannot just simply restart the algorithm from Line 2 if it is not true. The strategy we adopt is to bound the complexity of each iteration of corrected RLPN regardless of the value of the predicate $P\left(\widetilde{\mathscr{H}}\right)$, this is done by discarding the iterations that are such that the size of $\mathcal{S}$ is greater than a certain threshold (Line 7 of Algorithm 4.1). The correctness of our algorithm is not impacted by this as the probability that $P\left(\widetilde{\mathscr{H}}\right)$ is verified is in $1-o(1)$ and the threshold will be chosen such that, when $P\left(\widetilde{\mathscr{H}}\right)$ is verified the set $\mathcal{S}$ meets the threshold with probability $1-o(1)$. More specifically, the threshold $N_{\text{candi}}^{\max}$ is chosen as

$$n\,\mathbb{E}_{\mathscr{C},\mathscr{H}}\left(|\mathcal{S}| \;\middle|\; P\left(\widetilde{\mathscr{H}}\right)\right),$$

and, we can show using Markov inequality that this treshold is met with probability $1-o(1)$ for the iterations such that $P\left(\widetilde{\mathscr{H}}\right)$ is true. Thus in what follows we will only compute the value of $\mathbb{E}_{\mathscr{C},\mathscr{H}}\left(|\mathcal{S}| \;\middle|\; P\left(\widetilde{\mathscr{H}}\right)\right)$ and, to simplify notation we just write it as $\mathbb{E}_{\mathscr{C},\mathscr{H}}\left(|\mathcal{S}|\right)$. We are ready now to prove Proposition 5.

Step 1. It is readily seen that by linearity of the expected value and from the fact that the distribution of $\text{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}\left(\langle\mathbf{y},\mathbf{h}\rangle+\langle\mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle\right) > \frac{\delta}{2}$ does not depend on $\mathbf{x}$ as long as $\mathbf{x}\neq\mathbf{e}_{\mathscr{P}}$ we have

Fact 9.

$$\mathbb{E}_{\mathscr{C},\mathscr{H}}\left(|\mathcal{S}\setminus\{\mathbf{e}_{\mathscr{P}}\}|\right) \leq \binom{s}{t-u}\mathbb{P}_{\mathscr{C},\mathscr{H}}\left(\text{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}\left(\langle\mathbf{y},\mathbf{h}\rangle+\langle\mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle\right) > \frac{\delta}{2}\right).$$

The only known results we have regarding the bias is when $\mathbf{h}$ is sampled in $\widetilde{\mathscr{H}}$ (see [CDMT22a, Prop. 3.1] and Proposition 1). But we have the following slight generalization of Lemma 2 which essentially tells us that the distribution of the bias when $\mathbf{h}$ is sampled in $\mathscr{H}$ is close to the bias when $\mathbf{h}$ is sampled in $\widetilde{\mathscr{H}}$.

Lemma 3. *For any constant $c>0$:*

$$\mathbb{P}_{\mathscr{C},\mathscr{H}}\left(\left|\text{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}\left(\langle\mathbf{y},\mathbf{h}\rangle+\langle\mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle\right) - \text{bias}_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}\left(\langle\mathbf{y},\mathbf{h}\rangle+\langle\mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle\right)\right| \geq \delta\, c\right) \leq 2^{-\omega(n)}.$$

As a direct corollary we get the following bound

Corollary 3. *We have that*

$$\mathbb{P}_{\mathscr{C},\mathscr{H}}\left(\mathrm{bias}_{\mathbf{h}\hookleftarrow\mathscr{H}}\left(\langle\mathbf{y},\mathbf{h}\rangle+\langle\mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle\right)>\frac{\delta}{2}\right)=O\left(\mathbb{P}_{\mathscr{C}}\left(\mathrm{bias}_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}\left(\langle\mathbf{y},\mathbf{h}\rangle+\langle\mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle\right)>\frac{\delta}{4}\right)\right).$$

Step 2. Thus we are now interested in upper bounding $\mathbb{P}_{\mathscr{C}}\left(\mathrm{bias}_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}\left(\langle\mathbf{y},\mathbf{h}\rangle+\langle\mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle\right)>\frac{\delta}{4}\right)$. A first step is given in the next Lemma 4. This lemma uses the fact that we can write using Proposition 1 the former bias as

$$\mathrm{bias}_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}\left(\langle\mathbf{y},\mathbf{h}\rangle+\langle\mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle\right)=\frac{1}{2^{k-s}\left|\widetilde{\mathscr{H}}\right|}\sum_{i=0}^{n-s}N_i\left(\mathscr{C}^{\mathscr{N}}+\left(\mathbf{x}+\mathbf{e}_{\mathscr{P}}\right)\mathbf{R}+\mathbf{e}_{\mathscr{N}}\right)K_w^{(n-s)}(i)$$

where $N_i()$ is the number of codeword of weight i in a code, namely:

$$N_i\left(\mathscr{C}^{\mathscr{N}}+\left(\mathbf{x}+\mathbf{e}_{\mathscr{P}}\right)\mathbf{R}+\mathbf{e}_{\mathscr{N}}\right)\triangleq\left|\left(\mathscr{C}^{\mathscr{N}}+\left(\mathbf{x}+\mathbf{e}_{\mathscr{P}}\right)\mathbf{R}+\mathbf{e}_{\mathscr{N}}\right)\bigcap\mathcal{S}_i^{n-s}\right|.$$

We define, for simplicity:

Notation 10.

$$N_i\triangleq N_i\left(\mathscr{C}^{\mathscr{N}}+\left(\mathbf{x}+\mathbf{e}_{\mathscr{P}}\right)\mathbf{R}+\mathbf{e}_{\mathscr{N}}\right),\tag{C.5}$$

$$\bar{N}_i\triangleq\mathbb{E}_{\mathscr{C}}\left(N_i\right).\tag{C.6}$$

Recall that $\mathscr{C}^{\mathscr{N}}+\left(\mathbf{x}+\mathbf{e}_{\mathscr{P}}\right)\mathbf{R}+\mathbf{e}_{\mathscr{N}}$ is a coset of the $[n-s,k-s]$ linear code $\mathscr{C}^{\mathscr{N}}$ thus it is readily seen that we have

Fact 11. *[Bar97, Lem. 1.1, §1.3]*

$$\bar{N}_i=\frac{\binom{n-s}{i}}{2^{n-k}},\tag{C.7}$$

$$\mathbf{Var}_{\mathscr{C}}\left(N_i\right)\leq\frac{\binom{n-s}{i}}{2^{n-k}}.\tag{C.8}$$

The following lemma essentially says that we can study only the dominant term in the previous sum to bound the tail distribution of the bias. The key trick will be to use Krawtchouk polynomials orthogonality with the measure $\mu(i)=\binom{n-s}{i}$ so that we gain a factor $\bar{N}_iK_w^{(n-s)}(i)$ in our expressions

Lemma 4. *We have that*

$$\mathbb{P}_{\mathscr{C}}\left(bias_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}\left(\langle\mathbf{y},\mathbf{h}\rangle-\langle\mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle\right)>\frac{\delta}{4}\right)\leq n\max_{i=0\ldots n-s}\mathbb{P}_{\mathscr{C}}\left(\left|N_i-\bar{N}_i\right|>\left|\frac{K_w^{(n-s)}(u)}{K_w^{(n-s)}(i)}\right|\frac{1}{8\left(n-s+1\right)}\right)$$

Proof. By using Proposition 1 we derive that

$$\underset{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}{\text{bias}}\left(\langle \mathbf{y},\mathbf{h}\rangle - \langle \mathbf{x},\mathbf{h}_{\mathscr{P}}\rangle\right) = \frac{1}{2^{k-s}|\widetilde{\mathscr{H}}|}\sum_{i=0}^{n-s} N_i\, K_w^{(n-s)}(i)\,.$$

From the orthogonality of Krawtchouk polynomials relatively to the measure $\mu(i) = \binom{n-s}{i}$ [MS86, Ch. 5. §7. Theorem 16] we have:

$$\sum_{i=0}^{n-s}\binom{n-s}{i} K_w^{(n-s)}(i) = 0.$$

And thus, altogether with fact 11 we have that

$$\frac{1}{2^{k-s}|\widetilde{\mathscr{H}}|}\sum_{i=0}^{n-s} \bar{N}_i\, K_w^{(n-s)}(i) = 0.$$

And thus,

$$\frac{1}{2^{k-s}|\widetilde{\mathscr{H}}|}\sum_{i=0}^{n-s} N_i\, K_w^{(n-s)}(i) = \frac{1}{2^{k-s}|\widetilde{\mathscr{H}}|}\sum_{i=0}^{n-s} \left(N_i - \bar{N}_i\right)\, K_w^{(n-s)}(i)\,. \tag{C.9}$$

Moreover, the event

$$\frac{1}{2^{k-s}|\widetilde{\mathscr{H}}|}\sum_{i=0}^{n-s} \left(N_i - \bar{N}_i\right)\, K_w^{(n-s)}(i) > \frac{\delta}{4}$$

implies that it exists $i \in [\![0, n-s]\!]$ such that

$$\frac{1}{2^{k-s}|\widetilde{\mathscr{H}}|}\left(N_i - \bar{N}_i\right)\, K_w^{(n-s)}(i) > \frac{\delta}{4\,(n-s+1)}. \tag{C.10}$$

Thus we get:

$$\begin{aligned}
&\mathbb{P}_{\mathscr{C}}\left(\frac{1}{2^{k-s}|\widetilde{\mathscr{H}}|}\sum_{i=0}^{n-s} \left(N_i - \bar{N}_i\right)\, K_w^{(n-s)}(i) > \frac{\delta}{4}\right)\\
&\leq \mathbb{P}_{\mathscr{C}}\left(\bigvee_{i=0}^{n-s}\left(\frac{1}{2^{k-s}|\widetilde{\mathscr{H}}|}\left(N_i - \bar{N}_i\right)\, K_w^{(n-s)}(i) > \frac{\delta}{4\,(n-s+1)}\right)\right) \quad \text{(using } C.10\text{)}\\
&\leq \sum_{i=0}^{n-s}\mathbb{P}_{\mathscr{C}}\left(\frac{1}{2^{k-s}|\widetilde{\mathscr{H}}|}\left(N_i - \bar{N}_i\right)\, K_w^{(n-s)}(i) > \frac{\delta}{4\,(n-s+1)}\right) \quad \text{(union bound)}\\
&\leq (n-s+1)\max_{i=0\ldots(n-s)}\mathbb{P}_{\mathscr{C}}\left(\frac{1}{2^{k-s}|\widetilde{\mathscr{H}}|}\left(N_i - \bar{N}_i\right)\, K_w^{(n-s)}(i) > \frac{\delta}{4\,(n-s+1)}\right)\\
&\leq (n-s+1)\max_{i=0\ldots(n-s)}\mathbb{P}_{\mathscr{C}}\left(\left|N_i - \bar{N}_i\right| > \left|\frac{\delta\, 2^{k-s}\,|\widetilde{\mathscr{H}}|}{K_w^{(n-s)}(i)}\right|\frac{1}{4\,(n-s+1)}\right)
\end{aligned}$$

Now we get our result using the fact that $\delta = \frac{K_w^{(n-s)}(u)}{\binom{n-s}{w}}$ (Equation (5.3) of Parameter constraint 6) altogether with the fact that we supposed in (C.4) that $|\widetilde{\mathscr{H}}| > \frac{1}{2}\,\mathbb{E}_{\mathscr{C}}\left(|\widetilde{\mathscr{H}}|\right)$ and that from Fact 5 we have that $\mathbb{E}_{\mathscr{C}}\left(|\widetilde{\mathscr{H}}|\right) = \frac{\binom{n-s}{w}}{2^{k-s}}$. □

Step 3. In this step we want to upper bound

$$\mathbb{P}_{\mathscr{C}}\left(\left|N_i - \bar{N}_i\right| > \left|\frac{K_w^{(n-s)}(u)}{K_w^{(n-s)}(i)}\right| \frac{1}{8\,(n-s+1)}\right)$$

First we give a useful lemma which essentially tells us that the right term in the probability is always greater than $\sqrt{\mathbf{Var}_{\mathscr{C}}\left(N_i\right)} = \sqrt{\bar{N}_i}$.

Lemma 5. *We have that*

$$\bar{N}_i\, f(n) < \left(\frac{K_w^{(n-s)}(u)}{K_w^{(n-s)}(i)}\right)^2$$

Proof. The proof is available in the eprint version of the paper. □

We want to obtain an exponential bound on the previous probability

$$\mathbb{P}_{\mathscr{C}}\left(\left|N_i - \bar{N}_i\right| > \left|\frac{K_w^{(n-s)}(u)}{K_w^{(n-s)}(i)}\right| \frac{1}{8\,(n-s+1)}\right).$$

But, very few results are known about the N_i's, or, more generally the weight distribution of a random affine code. The first two moments of N_i are known with Fact 11. Some higher moments are studied in [LM19, BEL10], but in general, there is no known expressions for all the higher moments of the weight distribution of a random linear code. Furthermore, up to our knowledge no exponential tail bound on $N_i - \bar{N}_i$ exists. Thus, we are left to bound the previous probability using only the expected value and the variance of N_i by using Bienaymé-Tchebychev second order bound (which is the best bound we can get for a generic random variable using only its first two moments) which is given by:

$$\mathbb{P}_{\mathscr{C}}\left(\left|N_i - \bar{N}_i\right| > p(n)\sqrt{\mathbf{Var}\left(N_i\right)}\right) \leq \frac{1}{p(n)^2}$$

Now the problem is that the previous Lemma 5 is tight for some $i \approx \frac{n-s}{2}$, namely we have:

$$\bar{N}_i\, f(n) = \mathrm{poly}(n)\left(\frac{K_w^{(n-s)}(u)}{K_w^{(n-s)}(i)}\right)^2 \tag{C.11}$$

And thus if f is big enough we have that

$$\left|\frac{K_w^{(n-s)}(u)}{K_w^{(n-s)}(i)}\right| \frac{1}{8\,(n-s+1)} = \mathrm{poly}(n)\,\sqrt{\mathbf{Var}\left(N_i\right)}.$$

As a result we can only get a polynomial bound using the Bienaymé-Tchebychev inequality. The fact that equation (C.11) holds can be seen with the fact that Krawtchouk polynomials attain their ℓ_2 norm [KS21, Prop. 2.15] (regarding the measure $\mu(i) = \frac{\binom{n-s}{i}}{2^{n-s}}$), up to a polynomial factor for certain i close to $(n-s)/2$, namely we have

$$\left(K_w^{(n-s)}(i)\right)^2 = \mathrm{poly}(n)\binom{n-s}{w}2^{n-s}.$$

All in all, to be able to use tighter bounds, we decide to model in Assumption 8 the weight distributions N_i as Poisson variables of parameters

$$\mathbb{E}(N_i) = \frac{\binom{n-s}{i}}{2^{n-k}}.$$

We ran extensive experimentations to show that the distribution of the bias remains unchanged is we replace the weight distribution N_i by the former model. See Section D for the experimental results. We are now ready to give the two following tail bounds for $N_i - \bar{N}_i$. We first give a bound for the i's that are relatively small compared to u, namely when $i < u + O(\log n)$ (which corresponds to the case $\mathrm{poly}(n) K_w^{(n-s)}(u) < K_w^{(n-s)}(i)$). Then, we prove a second bound using our model for the i's that are relatively big compared to u (which corresponds to the case $\mathrm{poly}(n) K_w^{(n-s)}(u) > K_w^{(n-s)}(i)$).

Lemma 6. *Define $\epsilon \triangleq 1/4$. We have:*

$$\textit{If} \quad \left|\frac{K_w^{(n-s)}(u)}{K_w^{(n-s)}(i)}\right| \leq n^{2+\epsilon} \quad \textit{then} \quad \mathbb{P}_{\mathscr{C}}\left(\left|N_i - \bar{N}_i\right| > \left|\frac{K_w^{(n-s)}(u)}{K_w^{(n-s)}(i)}\right| \frac{1}{8(n-s+1)}\right) \leq \frac{\binom{n-s}{i}}{2^{n-k}}. \tag{C.12}$$

Moreover, under Assumption 8 we have:

$$\textit{If} \quad \left|\frac{K_w^{(n-s)}(u)}{K_w^{(n-s)}(i)}\right| > n^{2+\epsilon} \quad \textit{then} \quad \mathbb{P}_{\mathscr{C}}\left(\left|N_i - \bar{N}_i\right| > \left|\frac{K_w^{(n-s)}(u)}{K_w^{(n-s)}(i)}\right| \frac{1}{8(n-s+1)}\right) \leq 2^{-\omega(n)}. \tag{C.13}$$

Proof. The proof is available in the eprint version of the paper. □

Lemma 7. *We have under Assumption 8 that:*

$$\mathbb{P}_{\mathscr{C}}\left(bias_{\mathbf{h}\hookleftarrow\widetilde{\mathscr{H}}}(\langle \mathbf{y}, \mathbf{h}\rangle - \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}}\rangle) > \frac{\delta}{4}\right) \leq \widetilde{O}\left(\frac{\binom{n-s}{u}}{2^{n-k}}\right) \tag{C.14}$$

Proof. The proof is available in the eprint version of the paper. □

And thus, fact 9 altogether with Lemma 7 proves Proposition 5 which gives the expected size of $\mathcal{S}$.

C.4 Further Results on the Complexity of Algorithm 4.1

Let us give here the complexity of Algorithm 4.1 when the support is split in two parts $\mathscr{P}$ and $\mathscr{N}$.

Proposition 9. *The expected space and time complexities of the "2-split" version of Algorithm 4.1 to decode an $[n,k]$-linear code at distance t are given by*

$$\textbf{Space: } O(S_{eq}+2^s+S')\,, \quad \textbf{Time: } \widetilde{O}\left(\frac{T_{eq}+T_{\mathrm{FFT}}}{P_{succ}}\right)+\widetilde{O}\left(\max\left(1,\frac{S}{P_{succ}}\right)\,T'(n-s,k-s,u)\right).$$

Where

- $P_{succ}=\frac{\binom{s}{t-u}\binom{n-s}{u}}{\binom{n}{t}}$ *is the probability over $\mathscr{N}$ that $|\mathbf{e}_{\mathscr{N}}|=u$,*
- S_{eq}, T_{eq} *are respectively the space and time complexities of* Create*(N,w,$\mathscr{P}$) for computing N parity-checks of weight w on $\mathscr{N}$,*
- $T_{\mathrm{FFT}}=O(2^s)$ *is the time complexity of the fast Fourier Transform,*
- $S=\widetilde{O}\left(\binom{s}{t-u}\frac{\binom{n-s}{u}}{2^{n-k}}\right)$ *is the average number of candidates in the set $\mathcal{S}$*
- S', T' *are respectively the space and time complexities of* Solve-SubProblem *to decode an $[n-s,k-s]$ code at distance u.*

and where N the number of LPN samples and the parameters s, u and w are such that:

$$N<\frac{\binom{n-s}{w}}{2^{k-s}} \quad \textit{and} \quad N=\omega\left(n^5\left(\frac{\binom{n-s}{w}}{K_w^{(n-s)}(u)}\right)^2\right).$$

Notice here that the only change in the complexity of Algorithm 4.1 compared to the original RLPN algorithm (Proposition 3.10 of [CDMT22a]) is that we added a term

$$\widetilde{O}\left(\max\left(1,\frac{S}{\mathrm{P_{succ}}}\right)\,T'(n-s,k-s,u)\right)$$

in the complexity. We now give the asymptotic complexity of the corrected RLPN algorithm. Note that the techniques used to compute low-weight parity-checks [Dum86, CDMT22a] compute in fact all the parity checks of weight w on $\mathscr{N}$. Thus, to simplify the next proposition we will simply replace the parameter N by the expected number of parity-checks of weight w on $\mathscr{N}$ that is $\frac{\binom{n-s}{w}}{2^{k-s}}$.

Proposition 10. *Define*

$$R\stackrel{\triangle}{=}\lim_{n\to\infty}\frac{k}{n},\quad \tau\stackrel{\triangle}{=}\lim_{n\to\infty}\frac{t}{n},\quad \sigma\stackrel{\triangle}{=}\lim_{n\to\infty}\frac{s}{n},\quad \omega\stackrel{\triangle}{=}\lim_{n\to\infty}\frac{w}{n},\quad \mu\stackrel{\triangle}{=}\lim_{n\to\infty}\frac{u}{n}.$$

The number of LPN samples N is chosen to be equal to the total number of parity-checks of weight w on $\mathscr{N}$, namely $N=2^{n\,(\nu_{\mathrm{eq}}+o(1))}$ where $\nu_{\mathrm{eq}}\stackrel{\triangle}{=}(1-\sigma)\,h_2\left(\frac{\omega}{1-\sigma}\right)-(R-\sigma)$. Then, the time complexity of the RLPN-decoder to decode an $[n,k]$-linear

code at distance t is given by $2^{n\,(\alpha+o(1))}$ and the space complexity is $2^{n\,(\alpha_{\text{space}}+o(1))}$ where

$$\begin{aligned}
\alpha &\triangleq \max\left((1-\sigma)\,\gamma\left(\frac{R-\sigma}{1-\sigma},\frac{\omega}{1-\sigma}\right)+\pi,\sigma+\pi,\max\left(\chi+\pi,0\right)+(1-\sigma)\alpha'\left(\frac{R-\sigma}{1-\sigma},\frac{\mu}{1-\sigma}\right)\right),\\
\pi &\triangleq h\left(\tau\right)-\sigma h\left(\frac{\tau-\mu}{\sigma}\right)-(1-\sigma)h\left(\frac{\mu}{1-\sigma}\right),\\
\chi &\triangleq \sigma h\left(\frac{\tau-\mu}{\sigma}\right)+(1-\sigma)h\left(\frac{\mu}{1-\sigma}\right)-(1-R),\\
\alpha_{\text{space}} &\triangleq \max\left((1-\sigma)\alpha'_{\text{space}}\left(\frac{R-\sigma}{1-\sigma},\frac{\mu}{1-\sigma}\right),(1-\sigma)\,\gamma_{\text{space}}\left(\frac{R-\sigma}{1-\sigma},\frac{\omega}{1-\sigma}\right)\right).
\end{aligned}$$

And where

- *the time complexity of* Create *to compute all parity-checks of relative weight τ' of a code of rate R' and length n' is given by $2^{n'\,\gamma(R',\tau')}$ and the space complexity is $2^{n'\,\gamma_{\text{space}}(R',\tau')}$.*
- *The time complexity of* Solve-SubProblem *to decode a code of rate R' and length n' at relative distance τ' is given by $2^{n'\,\alpha'(R',\tau')}$. Its space complexity is $2^{n'\,\alpha'_{\text{space}}(R',\tau')}$.*

Moreover, σ, μ, and ω are non-negative and such that

$$\sigma\leq R,\quad \tau-\sigma\leq\mu\leq\tau,\quad \omega\leq 1-\sigma,$$

$$\nu_{\text{eq}}\geq 2\,(1-\sigma)\left(h_2\left(\frac{\omega}{1-\sigma}\right)-\widetilde{\kappa}\left(\frac{\omega}{1-\sigma},\frac{\mu}{1-\sigma}\right)\right),$$

where $\widetilde{\kappa}$ is the function defined in Proposition 2.

The only added term here compared to the original RLPN asymptotic complexity exponent is

$$\max\left(\chi+\pi,0\right)+(1-\sigma)\alpha'\left(\frac{R-\sigma}{1-\sigma},\frac{\mu}{1-\sigma}\right).$$

For simplicity, we make the choice here to use [Dum91] ISD-decoder as the routine Solve-SubProblem to solve the relevant decoding problem. Thus, we simply replace α' in Proposition 10 by the asymptotic complexity of [Dum91] ISD-decoder given in Proposition 8 of Section B. One could use [Dum86] to compute parity-checks and thus replace γ by the exponent given in Proposition 7 or use some more involved methods as described in [CDMT22a, §5].

D Experimental Results Regarding the Poisson Model

In this section we give experimental evidence for our claims which rely on the Poisson Model 8. Specifically, we show that the experimental distribution of the

bias of $\langle \mathbf{y}, \mathbf{h} \rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle$ coincides with the theoretical distribution of the random variable X obtained by replacing the N_i's in Proposition 3:

$$\mathrm{bias}_{\mathbf{h} \hookleftarrow \widetilde{\mathscr{H}}} \left(\langle \mathbf{y}, \mathbf{h} \rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle \right) = \frac{1}{2^{k-s} \left| \widetilde{\mathscr{H}} \right|} \sum_{i=0}^{n} N_i \left(\mathscr{C}^{\mathscr{N}} + \left(\mathbf{x} + \mathbf{e}_{\mathscr{P}} \right) \mathbf{R} + \mathbf{e}_{\mathscr{N}} \right) K_w^{(n-s)} (i)$$

by independent Poisson variables $\widetilde{N_i}$ of parameter $\mathbb{E}_{\mathscr{C}} \left(\left| N_i \left(\mathscr{C}^{\mathscr{N}} + \left(\mathbf{x} + \mathbf{e}_{\mathscr{P}} \right) \mathbf{R} + \mathbf{e}_{\mathscr{N}} \right) \right| \right) = \frac{\binom{n-s}{i}}{2^{n-k}}$. That is

$$X \stackrel{\triangle}{=} \frac{1}{2^{k-s} \left| \widetilde{\mathscr{H}} \right|} \sum_{i=0}^{n} \widetilde{N_i} K_w^{(n-s)} (i) . \tag{D.1}$$

The independence assumption on the $\widetilde{N_i}$'s was just made to be able to compute numerically the distribution of X.

Now, as far as we are aware, there is no simple way to derive a closed expression for the probability distribution of $\mathrm{bias}_{\mathbf{h} \hookleftarrow \widetilde{\mathscr{H}}} \left(\langle \mathbf{y}, \mathbf{h} \rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle \right)$ and $\mathrm{bias}_{\mathbf{h} \hookleftarrow \mathscr{H}} \left(\langle \mathbf{y}, \mathbf{h} \rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle \right)$ under the previous model. As such, we simply experimentally computed it using Monte-Carlo method.

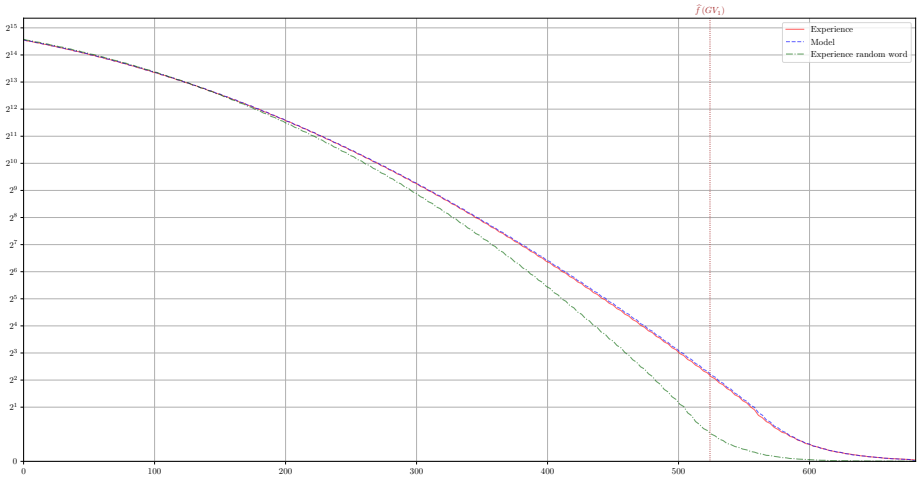
As a side note, so that our quantity are intuitively interpreted as outputs of the corrected RLPN algorithm we in fact compare the distribution of the bias but normalized by a factor $\binom{s}{t-u}$, indeed we have:

Fact 12. *If* $\mathbf{y}$ *is a random word of* $\mathbb{F}_2^n$ *then for any real* T *we have*

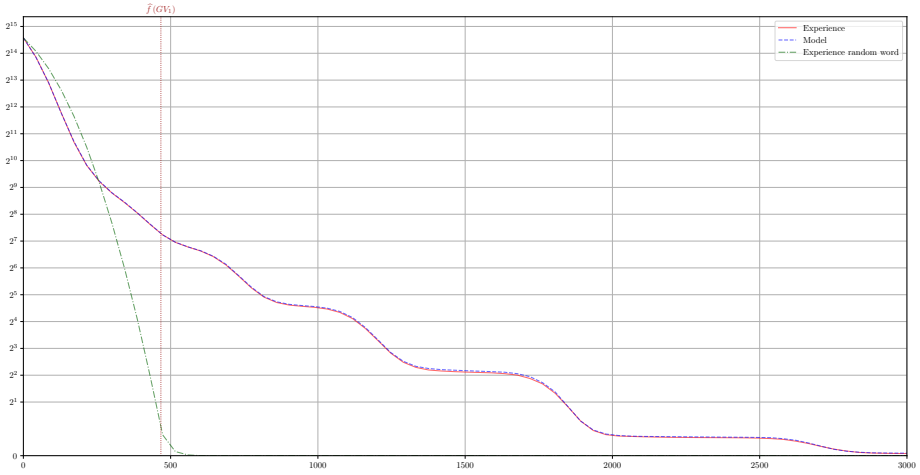
$$\mathbb{E}_{\mathscr{C}} \left(\left| \{ \mathbf{x} \in \mathcal{S}_{t-u}^{s} \ : \widehat{f_{\mathbf{y}, \mathscr{H}}} (\mathbf{x}) \geq T \} \right| \right) = \binom{s}{t-u} \mathbb{P}_{\mathscr{C}, \mathscr{H}} \left(\mathrm{bias}_{\mathbf{h} \hookleftarrow \mathscr{H}} \left(\langle \mathbf{y}, \mathbf{h} \rangle + \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle \right) \geq T \right) .$$

Some experimental results are summed up in figure D.1. More figures can be found in the supplementary material GitHub page [4]. Note that the parameters considered are such that we have unusually large Fourier coefficients compared to what we would expect if the original LPN model made in [CDMT22a] was to hold. It is readily seen that if the LPN model of [CDMT22a] was to hold the full curve in red should roughly match the green dash-dotted one. However, as [CDMT22a] already noticed in Section 3.4, and as we also notice here, it is not the case. The dashed blue curve represents the tail coefficients given by Fact 12 under this independent Poisson Model. We see that our model very well predicts the experimental distribution of the Fourier coefficients.

[4] `https://github.com/meyer-hilfiger/Rigorous-Foundations-for-Dual-Attacks-in-Coding-Theory`



(a) $[s, k, n, w, u, t] = [18, 24, 2000, 2, 873, 882]$ and $N = 2000$



(b) $[s, k, n, w, u, t] = [18, 24, 60, 5, 1, 10]$ and $N = 2000$

Fig. D.1: Expected size of the set $\{\mathbf{x} \in \mathcal{S}_{t-u}^{s} \ : \widehat{f_{\mathbf{y},\mathscr{H}}}(\mathbf{x}) \geq T\}$ as a function of T for two different parameters.

- Experimentally in corrected RLPN.
- Theoretically under the independent Poisson Model (D.1).
- Theoretically under LPN model, that is, if we supposed that the LPN samples produced by RLPN followed exactly the framework of Problem 2.

We define $\widehat{f}(GV_1) \triangleq N - 2\, d_{\mathrm{GV}}\left(N, \log_2\left(\binom{s}{t-u}\right)\right)$ corresponding to the highest theoretical Fourier coefficient under the LPN model.