Statistical Decoding

Thomas Debris-Alazard, Jean-Pierre Tillich

Statistical Decoding

Thomas Debris-Alazard, Jean-Pierre Tillich
Inria, SECRET Project, 2 Rue Simone Iff 75012 Paris Cedex, France
Email: {thomas.debris,jean-pierre.tillich}@inria.fr

***Abstract*—The security of code-based cryptography relies primarily on the hardness of generic decoding with linear codes. The best generic decoding algorithms are all improvements of an old algorithm due to Prange: they are known under the name of information set decoding techniques (ISD). A while ago a generic decoding algorithm which does not belong to this family was proposed: statistical decoding. It is a randomized algorithm that requires the computation of a large set of parity-check equations of moderate weight. We solve here several open problems related to this decoding algorithm. We give in particular the asymptotic complexity of this algorithm, give a rather efficient way of computing the parity-check equations needed for it inspired by ISD techniques and give a lower bound on its complexity showing that when it comes to decoding on the Gilbert-Varshamov bound it can never be better than Prange's algorithm.**

I. Introduction

Code-based cryptography relies crucially on the hardness of decoding generic linear codes. This problem has been studied for a long time and despite many efforts on this issue [1], [2], [4], [8], [9], [11], [12] the best algorithms for solving this problem [2], [9] are exponential in the number of errors that have to be corrected: correcting t errors in a binary linear code of length n has with the aforementioned algorithms a cost of $2^{ct(1+o(1))}$ where c is a constant depending of the code rate R and the algorithm. All the efforts that have been spent on this problem have only managed to decrease slightly this exponent c. Let us emphasize that this exponent is the key for estimating the security level of any code-based cryptosystem.

All the aforementioned algorithms can be viewed as a refinement of the original Prange algorithm [11] and are actually all referred to as ISD algorithms. There is however an algorithm that does not rely at all on Prange's idea and does not belong to the ISD family: statistical decoding proposed first by Al Jabri in [7] and improved a little bit by Overbeck in [10]. Later on, [5] proposed an iterative version of this algorithm. It is essentially a two-stage algorithm, the first step consisting in computing an exponentially large number of parity-check equations of the smallest possible weight w, and then from these parity-check equations the error is recovered by some kind of majority voting based on these parity-check equations.

However, even if the study made by R. Overbeck in [10] lead to the conclusion that this algorithm did not allow better attacks on the cryptosystems he considered, he did not propose an asymptotic formula of its complexity that would have allowed to conduct a systematic study of the performances of this algorithm. Such an asymptotic formula has been proposed in [5] through a simplified analysis of statistical decoding, but as we will see this analysis does not capture accurately the complexity of statistical decoding. Moreover both papers did not assess in general the complexity of the first step of the algorithm which consists in computing a large set of parity-check equations of moderate weight.

The primary purpose of this paper is to clarify this matter by giving three results. First, we give a rigorous asymptotic study of the exponent c of statistical decoding by relying on asymptotic formulas for Krawtchouk polynomials [6]. The number of equations which are needed for this method turns out to be remarkably simple for a large set of parameters (see Theorem 1). For instance when we consider the hardest instances of the decoding problem which correspond to the case where the number of errors is equal to the Gilbert-Varshamov bound, then essentially our results indicate that we have to take *all* possible parity-checks of a given weight (when the code is assumed to be random) to perform statistical decoding. This asymptotic study also allows to conclude that the modeling of iterative statistical decoding made in [5] is too optimistic. Second, inspired by ISD techniques, we propose a rather efficient method for computing a huge set of parity-check equations of rather low weight. Finally, we give a lower bound on the complexity of this algorithm that shows that it can not improve upon Prange's algorithm for the hardest instances of decoding.

This lower bound follows by observing that the number P_w of the parity-check equations of weight w that are needed for the second step of the algorithm is clearly a lower-bound on the complexity of statistical decoding. What we actually prove in the last part of the paper is that irrelevant of the way we obtain these parity-check equations in the first step, the lower bound on the complexity of statistical decoding coming from the infinitum of these P_w's is always larger than the complexity of the Prange algorithm for the hardest instances of decoding.

This paper is given without proofs, they are given in the full version that is on `arxiv` [3].

II. Notation

As our study will be asymptotic, we neglect polynomial factors and use the following notation:

Notation 1. Let $f, g : \mathbb{N} \to \mathbb{R}$, we write $f = \tilde{O}(g)$ iff there exists a polynomial P such that $f = O(Pg)$.

Moreover, we will often use the classical result $\binom{n}{w} = \tilde{O}\left(2^{nH\left(\frac{w}{n}\right)}\right)$ where H denotes the binary entropy.

978-1-5090-4096-4/17/$31.00 ©2017 IEEE

III. Statistical Decoding

In the whole paper we consider the computational decoding problem which we define as follows:

Problem 1. *Given a binary linear code of length n of rate R, a word $y \in \mathbb{F}_2^n$ at distance t from the code, find a codeword x such that $d_H(x,y) = t$ where d_H denotes the Hamming distance.*

Generally we will specify the code by an arbitrary generator matrix G and we will denote by $\mathrm{CSD}(G,t,y)$ a specific instance of this problem. We will be interested as is standard in cryptography in the case where $G \in \mathbb{F}_2^{Rn\times n}$ *is supposed to be random.*

The idea behind statistical decoding may be described as follows. We first compute a very large set $\mathcal{S}$ of parity-check equations of some weight w and compute all scalar products $\langle y,h\rangle$ (scalar product is modulo 2) for $h \in \mathcal{S}$. It turns out that if we consider only the parity-checks involving a given code position i the scalar products have a probability of being equal to 1 which depends on whether there is an error in this position or not. Therefore counting the number of times when $\langle y,h\rangle = 1$ allows to recover the error in this position.

Let us analyze now this algorithm more precisely. To make this analysis tractable we will need to make a few simplifying assumptions. The first one we make is the same as the one made by R. Overbeck in [10], namely that

Assumption 1. The distribution of the $\langle y,h\rangle$'s when h is drawn uniformly at random from the dual codewords of weight w is approximated by the distribution of $\langle y,h\rangle$ when h is drawn uniformly at random among the words of weight w.

A much simpler model is given in [5] and is based on modeling the distribution of the $\langle y,h\rangle$'s as the distribution of $\langle y,h\rangle$ where the coordinates of h are i.i.d. and distributed as a Bernoulli variable of parameter w/n. This presents the advantage of making the analysis of statistical decoding much simpler and allows to analyze more refined versions of statistical decoding. However as we will show, this is an oversimplification and leads to an over-optimistic estimation of the complexity of statistical decoding. The following notation will be useful.

Notation 2.
· $S_w \stackrel{\triangle}{=} \{x \in \mathbb{F}_2^n : w_H(x) = w\}$ where w_H is the Hamming weight;
· $S_{w,i} \stackrel{\triangle}{=} \{x \in S_w : x_i = 1\}$;
· $\mathcal{H}_w \stackrel{\triangle}{=} \mathscr{C}^\perp \cap S_w$;
· $\mathcal{H}_{w,i} \stackrel{\triangle}{=} \mathscr{C}^\perp \cap S_{w,i}$;
· $\mathscr{B}(p)$ denotes the Bernoulli distribution of parameter p ;
· $h \sim S_{w,i}$ means we pick h uniformly at random in $S_{w,i}$.

We start the analysis of statistical decoding by computing the following probabilities which approximate the true probabilities we are interested in (which correspond to choosing h uniformly at random in $\mathcal{H}_{w,i}$ and not in $S_{w,i}$) under Assumption 1:

$$q_w^+ = \mathbb{P}_{h\sim S_{w,i}}\left(\langle e,h\rangle = 1\right) \text{ when } e_i = 1$$
$$q_w^- = \mathbb{P}_{h\sim S_{w,i}}\left(\langle e,h\rangle = 1\right) \text{ when } e_i = 0.$$

These probabilities are readily seen to be equal to

$$q_w^+ = \frac{\sum_{j \text{ even}}^{w-1} \binom{t-1}{j}\binom{n-t}{w-1-j}}{\binom{n-1}{w-1}},\ q_w^- = \frac{\sum_{j \text{ odd}}^{w-1} \binom{t}{j}\binom{n-t-1}{w-1-j}}{\binom{n-1}{w-1}}$$

We define the biases ε_0 and ε_1 of statistical decoding by

$$q_w^- = \frac{1}{2} + \varepsilon_0 \ ;\ q_w^+ = \frac{1}{2} + \varepsilon_1$$

It will turn out, and this is essential, that $\varepsilon_0 \neq \varepsilon_1$. We can use these biases "as a distinguisher". Statistical decoding is nothing but a statistical hypothesis testing algorithm distinguishing between two hypotheses :

$$\mathscr{H}_0 \ :\ e_i = 0 \quad ; \quad \mathscr{H}_1 \ :\ e_1 = 1$$

based on computing the random variable V_i^m for m uniform and independent draws of vectors in $\mathscr{H}_{w,i}$:

$$V_i^m = \sum_{k=1}^{m} s\langle y,h^k\rangle \in \mathbb{Z} \text{ where } s \stackrel{\triangle}{=} \mathrm{sgn}(\varepsilon_0 - \varepsilon_1).$$

We have $\langle y,h^k\rangle \sim \mathscr{B}(1/2+\varepsilon_j)$ according to $\mathscr{H}_j$. In order to apply the following proposition, we make the following assumption:

Assumption 2. $\langle y,h^k\rangle$ are independent variables.

With these assumptions we can prove that

Proposition 1. *Under $\mathscr{H}_j$, we have:*

$$\mathbb{P}\left(|V_i^m - s\cdot m(1/2+\varepsilon_j)| \geq m\frac{|\varepsilon_1-\varepsilon_0|}{2}\right) \leq 2\cdot 2^{-m\cdot\frac{(\varepsilon_1-\varepsilon_0)^2}{2\ln(2)}}$$

To take our decision we proceed as follows: if $V_i^m < s\cdot\frac{m}{2}(1+\varepsilon_1+\varepsilon_0)$, we choose $\mathscr{H}_1$ and $\mathscr{H}_0$ if not. For the cases of interest to us (namely w and t linear in n) the bias $\varepsilon_1-\varepsilon_0$ is an exponentially small function of the code-length n and it is obviously enough to choose m to be of order $O\left(\frac{\log n}{(\varepsilon_1-\varepsilon_0)^2}\right)$ to be able to make good decisions on all n positions simultaneously.

On the optimality of the statistical test. All the arguments used for distinguishing both hypotheses are very crude and this raises the question whether a better test exists. It turns out that in the regime of interest to us, namely t and w linear in n, the term $\tilde{O}\left(\frac{1}{(\varepsilon_1-\varepsilon_0)^2}\right)$ is of the right order. Indeed our statistical test amounts actually to the Neymann-Pearson test (with a threshold in this case which is not necessarily in the middle, i.e. equal to $s\cdot m\frac{1+\varepsilon_0+\varepsilon_1}{2}$). In the case of interest to us, the bias between both distributions $\varepsilon_1-\varepsilon_0$ is exponentially small in n and Chernoff's bound captures accurately the large deviations of the random variable V_i^m. Finer knowledge about the hypotheses $\mathscr{H}_0$ and $\mathscr{H}_1$ does not make a difference. For instance even using the a priori probability $\mathbb{P}(e_i = 1) = \frac{t}{n}$

will not change the number of the tests that are needed: it will still be $\tilde{O}\left(\frac{1}{(\varepsilon_1-\varepsilon_0)^2}\right)$ when t and w are linear in n.

Statistical decoding is a randomized algorithm which uses the previous distinguisher. As we just noted, this distinguisher needs $\tilde{O}\left(\frac{1}{(\varepsilon_1-\varepsilon_0)^2}\right)$ parity-check equations of weight w to work. This number obviously depends on w, R and t and we use the notation:

Notation 3. $P_w \triangleq \frac{1}{(\varepsilon_1-\varepsilon_0)^2}$.

Now we have two frameworks to present statistical decoding. We can consider the computation of $\tilde{O}(P_w)$ parity-check equations as a pre-computation or to consider it as a part of the algorithm. To consider the case of pre-computation, simply remove Line 4 of Algorithm 1 and consider the $\mathcal{S}_i$'s as an additional input to the algorithm. $\texttt{ParityCheckComputation}_w$ will denote an algorithm which for an input G, i outputs $\tilde{O}(P_w)$ vectors of $\mathscr{H}_{w,i}$.

Algorithm 1 `DecoStat` : **Statistical Decoding**

1: *Input* : $G \in \mathbb{F}_2^{Rn\times n}, y = xG + e \in \mathbb{F}_2^n, w \in \mathbb{N}$
2: *Output* : e */*Error Vector*/*
3: **for** $i = 1 \cdots n$ **do**
4: $\quad \mathcal{S}_i \leftarrow \texttt{ParityCheckComputation}_w(G, i)$
5: $\quad V_i \leftarrow 0$
6: $\quad$ **for all** $h \in \mathcal{S}_i$ **do**
7: $\quad\quad V_i \leftarrow V_i + s \cdot \langle y, h\rangle$
8: $\quad$ **if** $V_i < s \cdot P_w \frac{1+\varepsilon_1+\varepsilon_0}{2}$ **then**
9: $\quad\quad e_i \leftarrow 1$
10: $\quad$ **else**
11: $\quad\quad e_i \leftarrow 0$
12: **return** e

Clearly statistical decoding complexity is given by

- When the $\mathcal{S}_i$'s are already stored and computed: $\tilde{O}\left(P_w\right)$;
- When the $\mathcal{S}_i$'s have to be computed: $\tilde{O}\left(P_w + |PCC_w|\right)$ where $|PCC_w|$ stands for the complexity of the call $\texttt{ParityCheckComputation}_w$.

The following quantities will be helpful in quantifying this complexity.

Notation 4.

- $\omega \triangleq \frac{w}{n}$; $\tau \triangleq \frac{t}{n}$;
- $\pi(\omega,\tau) \triangleq \underline{\lim}_{n\to+\infty} \frac{1}{n}\log_2 P_w$;
- $\pi^{complete}(\omega,\tau) \triangleq \underline{\lim}_{n\to+\infty} \frac{1}{n}\max\left(\log_2 P_w, \log_2 |\texttt{PCC}_w|\right)$.

One of our main result of this article is that we can evaluate very precisely $\pi(\omega,\tau)$ by expressing the biases in terms of Krawtchouk polynomials and then use asymptotic formulas for Krawtchouk polynomials [6] and some auxiliary results to derive the following theorem.

Theorem 1 (Asymptotic complexity of statistical decoding)**.** *$\pi(\omega,\tau)$ is equal to*

- $2\omega\log_2(r) - 2\tau\log_2(1-r) - 2(1-\tau)\log_2(1+r) + 2H(\omega)$ *if* $\tau \in \left(0, \frac{1}{2} - \sqrt{\omega-\omega^2}\right)$ *where r is the smallest root of* $(1-\omega)X^2 - (1-2\tau)X + \omega = 0$.
- $H(\omega) + H(\tau) - 1$ *if* $\tau \in \left(\frac{1}{2} - \sqrt{\omega-\omega^2}, \frac{1}{2}\right)$.

This statement allows to perform a systematic study of statistical decoding. Let us start by considering the hardest case for decoding which corresponds to the Gilbert-Varshamov bound (it is the largest distance where we can still expect to recover with good probability the right error) $\tau_{\text{DGV}} = H^{-1}(1-R)$. For $\omega \le \frac{1}{2} - \sqrt{\tau_{\text{DGV}} - \tau_{\text{DGV}}^2}$ it is readily verified that we are in the second case of Theorem 1. Therefore $\pi(\omega, \tau_{\text{DGV}}) = H(\omega) + 1 - R - 1 = H(\omega) - R$. Notice that this corresponds precisely to $\lim_{n\to\infty}\frac{1}{n}\log_2 \bar{a}_w^{\perp}$ where $\bar{a}_w^{\perp}$ is the expected number of parity-check equations of weight w in the code we want to decode. In other words, if we want to decode up to the Gilbert Varshamov distance we have to take all possible codewords of weight w (and even this is actually not enough due to the polynomial factors in the number of such parity-checks).

This theorem also shows that the simplified model for parity-check equations considered in [5] where the parity-check equations are binary words obtained by drawing their coordinates independently at random from a Bernoulli distribution of parameter w/n is significantly different from the constant weight model of weight w. In this case, we have $\pi(\omega,\tau) = -2\tau\log_2(1-2\omega)$. The two exponents are compared on Figure 1 as a function of the rate R with $\tau = H^{-1}(1-R)$ and $\omega = R/2$. As we see, there is a huge difference. The problem with the model chosen in [5] is that it is a very favorable model for statistical decoding. To the best of our knowledge there are no efficient algorithms for producing such parity-checks when $\omega \le R/2$. Note that even such an algorithm were to exist, selecting appropriately only one weight would not change the exponential complexity of the algorithm (for more details see the full version of the paper). In other words, in order to study statistical decoding we may restrict ourselves, as we do here, to considering only one weight and not a whole range of weights.

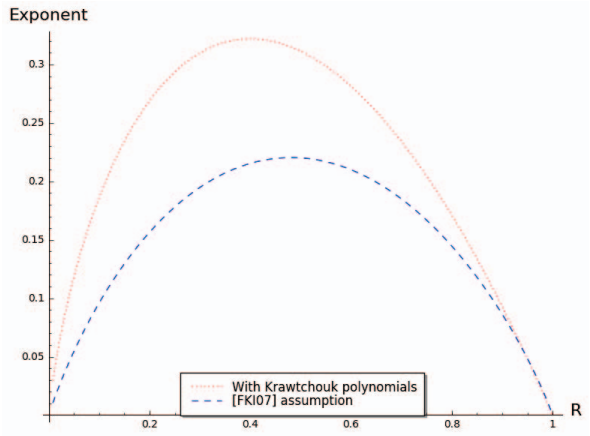

Figure 1: Comparisons of the complexities

As we are now able to give a formula for $\pi(\omega,\tau)$ we come back to the algorithm $\texttt{ParityCheckComputation}_w$

in order to estimate $\pi^{complete}(\omega, \tau)$. There is an easy way of producing parity-check equations of moderate weight by Gaussian elimination. This is given in Algorithm 2 that provides a method for finding parity-check equations of weight $w = \frac{Rn}{2}$ of an $[n, Rn]$ random code. Gaussian elimination (`GElim`) of an $Rn \times n$ matrix G_0 consists in finding U ($Rn \times Rn$ and non-singular) such that: $UG_0 = [I_{Rn}|G']$. $L_j(G)$ denotes the $j-$th row of G in Algorithm 2.

Algorithm 2 $\texttt{ParityCheckComputation}_{Rn/2}$

Input : $G \in \mathbb{F}_2^{Rn \times n}, i \in \mathbb{N}$
Output : $\mathcal{S}_i$ /*$P_{Rn/2}$ *parity-check equations**/
$\mathcal{S}_i \leftarrow [\,]$
while $|\mathcal{S}_i| < P_{Rn/2}$ **do**
  $P \leftarrow$ random $n \times n$ permutation matrix
  $[I_{Rn}|G'] \leftarrow \texttt{GElim}(GP)$ and if it fails return to line 5
  $H \leftarrow [G'^T|I_{n(1-R)}]$ /*Parity-Check check matrix of the code*/
  for $j = 1$ to $n(1-R)$ **do**
    if $L_j(H)_i = 1$ and $w_H(L_j(H)) = Rn/2$ **then**
      $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{L_j(H)P^T\}$
return $\mathcal{S}$

Algorithm 2 is a randomized algorithm. Randomness comes from the choice of the permutation P. Vectors returned by this algorithm have a weight of $Rn/2$ and it is clear that $\texttt{ParityCheckComputation}_{Rn/2}$ returns $P_{Rn/2}$ parity-check equations of weight $Rn/2$ in time $\tilde{O}\left(P_{Rn/2}\right)$. Now we set $\tau = H^{-1}(1-R)$. This relative weight, corresponds to the Gilbert-Varshamov bound. It is usually used to measure the efficiency of decoding algorithms because it corresponds to the hardest instance of the decoding problem as explained before. It is then clear that with this algorithm we have

$$\pi^{complete}(Rn/2, \tau) = \pi(Rn/2, \tau). \tag{1}$$

We call this the "naive statistical decoding complexity". Exponents (as a function of R) of Prange's ISD and statistical decoding are given in Figure 2. As we see the difference is huge. This version of statistical decoding can not be considered as an improvement over information set decoding algorithms.

IV. Improvements and limitations of statistical decoding

A. Lower bound on the complexity

By definition, statistical decoding needs $\tilde{O}\left(P_w\right)$ parity-check equations of weight w to work. Its complexity is therefore always greater than $\tilde{O}\left(P_w\right)$. Recall that the expected number of parity-check equations of weight w in an $[n, Rn]$ random binary linear code is $\frac{\binom{n}{w}}{2^{Rn}}$. Obviously if w is too small there are not enough equations for statistical decoding to work, we namely need that

$$P_w \leq \frac{\binom{n}{w}}{2^{Rn}}.$$

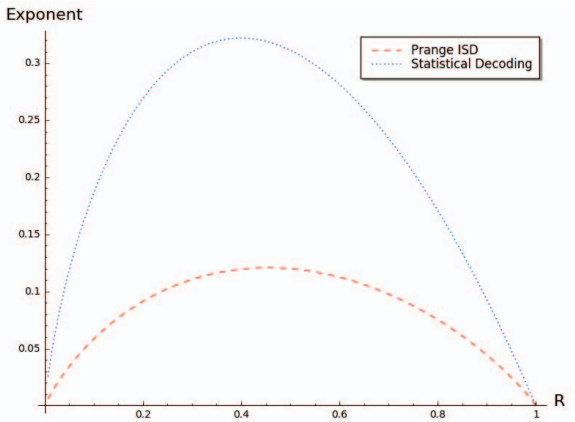

Figure 2: Asymptotic exponents of Prange ISD and naive statistical Decoding for $\tau = H^{-1}(1-R)$ et $\omega = R/2$

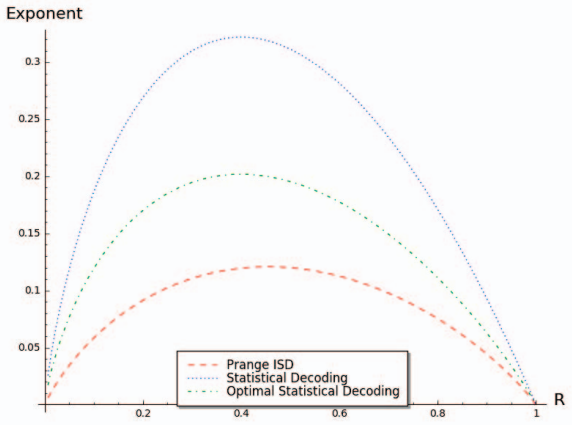

Figure 3: Asymptotic exponents of Prange ISD, naive statistical decoding and optimal/optimistic statistical decoding for $\tau = H^{-1}(1-R)$

The minimum $\omega_0(R, \tau)$ such that this holds is clearly given by

$$H\left(\omega_0(R, \tau)\right) - R = \pi(\omega_0(R, \tau), \tau)$$

So $\omega_0(R, \tau)$ gives the minimal relative weight such that asymptotically the number of parity-check equations needed for decoding is exactly the number of parity-check equations of weight $w_0(R, \tau)$ in the code, where $w_0(R, \tau) \stackrel{\triangle}{=} \omega_0(R, \tau)n$. In other words the asymptotic exponent of statistical decoding is always lower-bounded by $\pi(\omega_0(R, \tau), \tau)$.

Thanks to Figure 3 which compares Prange's ISD, statistical decoding with parity-check equations of relative weight $R/2$ and $\omega_0(R, \tau)$ with $\tau = H^{-1}(1-R)$, we clearly see on the one hand that there is some room of improving upon naive statistical decoding based on parity-check equations of weight $Rn/2$, but on the other hand that even with the best improvement upon statistical decoding we might hope for, we will still be above the most naive information set decoding algorithm, namely Prange's algorithm.

B. An improvement close to the complexity lower bound

The goal of this subsection is to present an improvement to the computation of parity-check equations and to give its asymptotic complexity. R. Overbeck in [10, Sec. 4] showed how to compute parity-check equations thanks to Stern's algorithm. We are going to use this algorithm too. However, whereas Overbeck used many iterations of this algorithm to produce a few parity-check equations of small weight, we observe that this algorithm produces in a natural way during its execution a large number of parity-check equations of relative weight smaller than $R/2$. We will analyze this process here and show how to choose parameters in order to get parity-check equations in amortized time $\tilde{O}(1)$. The algorithm we are interested in is given by:

Algorithm 3 `DumerFusion`

Input : $G \in \mathbb{F}_2^{Rn \times n}, l, r$.
Output : $\mathcal{S}$ /**subset of* $\mathscr{H}_w$**/*
$\mathcal{S} \leftarrow [\,]$ /**Empty list*/*
$\mathcal{T} \leftarrow [\,]$ /**Hash table*/*
$P \leftarrow$ random $n \times n$ permutation matrix
Find $U \in \mathbb{F}_2^{Rn \times Rn}$ non-singular such that $UGP = \begin{bmatrix} I_{Rn-l} & G_1 \\ 0 & G_2 \end{bmatrix}$
Partition G_2 as $[G_2^{(1)}|G_2^{(2)}]$ where $G_2^{(i)} \in \mathbb{F}_2^{l \times \left(\frac{n(1-R)+l}{2}\right)}$
for all $e_1 \in \mathbb{F}_2^{(n(1-R)+l)/2}$ of weight $r/2$ **do**
    $x \leftarrow G_2^{(1)} e_1^T$
    $\mathcal{T}[x] \leftarrow \mathcal{T}[x] \cup \{e_1\}$
for all $e_2 \in \mathbb{F}_2^{(n(1-R)+l)/2}$ of weight $r/2$ **do**
    $x \leftarrow G_2^{(2)} e_2^T$
    for all $e_1 \in \mathcal{T}[x]$ **do**
        $e \leftarrow (e_1, e_2)$
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{(eG_1^T, e)P^T\}$

In order to study this algorithm asymptotically, we introduce the relative parameters: $\rho = \frac{r}{n}$ and $\lambda = \frac{l}{n}$. We have many strategies with the choice of ρ and λ. In the following theorem we give three constraints on these parameters which we find relevant.

Theorem 2. *With λ and ρ satisfying the constraints (i) $\omega_0(R,\tau) \leq \rho + \frac{R-\lambda}{2}$, (ii) $\rho = (1-R+\lambda)H^{-1}\left(\frac{2\lambda}{1-R+\lambda}\right)$, (iii) $\lambda \leq \pi(\rho + \frac{R-\lambda}{2}, \tau)$ we have:*

$$\pi^{complete}(\rho + (R-\lambda)/2, \tau) = \pi(\rho + (R-\lambda)/2, \tau)$$

In order to get the optimal statistical decoding complexity we minimize $\pi(\rho + (R-\lambda)/2, \tau)$ (with $\pi(\rho + (R-\lambda)/2, \tau)$ given by Theorem 1) under constraints (i), (ii) and (iii). The exponent of statistical decoding with this strategy is given in Figure 4. As we see, `DumerFusion` with our strategy allows statistical decoding to be optimal for rates close to 0. We can further improve `DumerFusion` with ideas of [8] and [2], however the analysis would be much more involved and would not allow to break the barrier of the lower bound on the complexity of statistical decoding given in the previous subsection. Nevertheless these improvements lead to a larger range of rates where we reach the complexity of optimal statistical decoding. Another way of improving statistical decoding consists in considering iterative decoding techniques. However this does not change the exponent in the complexity of the algorithm (for more details see [3]).

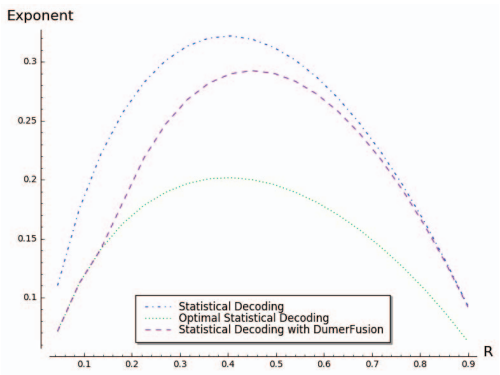

Figure 4: Asymptotic exponents of naive statistical decoding and with the use of optimal `DumerFusion` and optimal/optimistic statistical decoding for $\tau = H^{-1}(1-R)$

References

[1] A. Barg. Complexity issues in coding theory. *Electronic Colloquium on Computational Complexity*, Oct. 1997.

[2] A. Becker, A. Joux, A. May, and A. Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1+1=0$ improves information set decoding. In *Advances in Cryptology - EUROCRYPT 2012*, Lecture Notes in Comput. Sci. Springer, 2012.

[3] T. Debris-Alzard and J.-P. Tillich. Statistical decoding. preprint, arXiv:1701.07416, Jan. 2017.

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

[5] M. P. C. Fossorier, K. Kobara, and H. Imai. Modeling bit flipping decoding based on nonorthogonal check sums with application to iterative decoding attack of McEliece cryptosystem. *IEEE Trans. Inform. Theory*, 53(1):402–411, 2007.

[6] M. E. Ismail and P. Simeonov. Strong asymptotics for Krawtchouk polynomials. *Journal of Computational and Applied Mathematics*, pages 121–144, 1998.

[7] A. A. Jabri. A statistical decoding algorithm for general linear block codes. In B. Honary, editor, *Cryptography and coding. Proceedings of the 8th IMA International Conference*, volume 2260 of *Lecture Notes in Comput. Sci.*, pages 1–8, Cirencester, UK, Dec. 2001. Springer.

[8] A. May, A. Meurer, and E. Thomae. Decoding random linear codes in $O(2^{0.054n})$. In D. H. Lee and X. Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Comput. Sci.*, pages 107–124. Springer, 2011.

[9] A. May and I. 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 *Lecture Notes in Comput. Sci.*, pages 203–228. Springer, 2015.

[10] R. Overbeck. Statistical decoding revisited. In R. S.-N. Lynn Batten, editor, *Information security and privacy : 11th Australasian conference, ACISP 2006*, volume 4058 of *Lecture Notes in Comput. Sci.*, pages 283–294. Springer, 2006.

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

[12] J. Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *Lecture Notes in Comput. Sci.*, pages 106–113. Springer, 1988.