Bit-flipping Decoder Failure Rate Estimation for (v,w)-regular Codes

Alessandro Annechini and Alessandro Barenghi and Gerardo Pelosi *Member, IEEE*
Department of Electronics, Information and Bioengineering - DEIB
Politecnico di Milano, Milano, Italy
Email: alessandro.annechini@mail.polimi.it, alessandro.barenghi@polimi.it, gerardo.pelosi@polimi.it

***Abstract*—Providing closed form estimates of the Decoding Failure Rates (DFR) of iterative decoder for low- and moderate-density parity check codes has attracted significant interest in the research community over the years. This interest has raised due to the use of iterative decoders in post-quantum cryptosystems, where the desired DFRs are impossible to estimate via Monte Carlo simulations. In this work, we propose a new technique to provide accurate estimates of the DFR of a two-iterations (parallel) bit-flipping decoder, which is also employable for cryptographic purposes. In doing so, we successfully tackle the estimation of the bit-flipping probabilities at the first and second decoder iteration, and provide a fitting estimate for the syndrome weight distribution. We numerically validate our results, providing comparisons of the modeled and simulated weight of the syndrome, incorrectly-guessed error bit distribution at the end of the first iteration, and two-iteration DFR, both in the floor and waterfall regime. Finally, we apply our method to estimate the DFR of LEDAcrypt, a post-quantum cryptosystem, improving by factors larger than 2^{70}, with respect to the previous estimation techniques.**

I. Introduction

This work focuses on binary codes with block length n and redundancy r with a $r\times n$ parity-check matrix having all rows and columns with a constant number of set bits, referred to as (v,w)-regular binary codes, where w and v denote the number of 1's in any row and any column of the parity-check matrix, respectively. Among binary Low Density Parity Check (LDPC) codes, originally studied by Gallager [1], are (v,w)-regular binary codes that admit a sparse parity-check matrix, amenable to linear time decoding algorithms, exhibiting column and row weights in the range of $O(\log(n))$ [2]. Increasing the number of non null elements in each row of the parity-check matrix up to the range of $O(\sqrt{n\log(n)})$, the codes are also known as Moderate Density Parity Check Codes (MDPC) [3], [4]. The iterative (parallel) bit-flipping decoding algorithm proposed by Gallager [1] can be applied also to a generic (v,w)-regular binary code. The said decoding process takes as input the parity-check matrix $H=[h_{i,j}]$, $i\in\{0,\ldots,r-1\}$, $j\in\{0,\ldots,n-1\}$ of a code, and the value of a syndrome s of an error affected codeword, $\widetilde{c}=c+e$, where e is an unknown $1\times n$ error vector with Hamming weight t, while c is a legit codeword: $s{=}H(c+e)^T=He^T$. After each iteration, the algorithm updates the value of the syndrome to be used for the next iteration to match the equality $s=H(\bar{e}\oplus e)^T$, and terminates as soon as $s{=}0$, indicating that $\bar{e}{=}e$, or after a predetermined maximum number of iterations yielding a decoding failure. The initial value of $\bar{e}$ is the null vector $0_{1\times n}$. Each algorithm iteration is split up in three phases. In the first phase, it computes the inner product between the bit vector s and the bits in each column of H, considering them as integers, to obtain a quantity known as the "unsatisfied parity-check [equation count]" (upc) bound to the j-th bit position in the error vector, and stores such a value in a variable $\mathtt{upc}_j$, $j\in\{0,\ldots,n-1\}$. In the second phase, a threshold $\mathtt{th}\in\left\{\lceil\frac{v+1}{2}\rceil,\ldots,v\right\}$ is either looked up from a set of predetermined values $\{\mathtt{th}_1\mathtt{th}_2,\ldots\}$, each of which associated to a specific iteration, or computed as a function of the current value of the syndrome and of the current iteration count. In the third phase, the algorithm evaluates for each $j\in\{0,\ldots,n{-}1\}$, if $\mathtt{upc}_j$ is greater than the threshold $\mathtt{th}$, and in the affirmative case it flips the value in $\bar{e}_j$ (i.e., $\bar{e}_j\leftarrow\bar{e}_j\oplus 1$, by adding 1 to $\bar{e}_j$ modulo 2) and updates the syndrome by adding (modulo 2) to it (bitwise) the j-th column of H (i.e., $s\leftarrow s\oplus H_{:,j}$). When fixing a predetermined maximum number of iterations, the bit-flipping algorithm can fail to recover the error vector thus, the decoding failure rate (DFR) is the most important figure of merit for the code and the decoding algorithm of choice. One context where closed form estimates for the DFR are remarkably important is the design of post-quantum cryptosystems. In this context, both LDPC and MDPC codes are employed: BIKE [5], a current fourth-round candidate in NIST's standardization process, and LEDAcrypt [6]–[10] employ quasi-cyclic LDPC/MDPC codes where the error vector is randomly generated (with proper weight) to be the confidential message (password) transmitted from the sending to the receiving endpoint. Whenever a decoding failure takes place during the decryption, information regarding the private key of the cipher is leaked to an (active) attacker [11]. To attain security against active attackers both cryptosystems require the DFR of the employed codes to be below 2^{-128}, for a decoder of choice. To this end, BIKE relies on an extrapolation of the behaviour of its own variant of the iterative bit-flipping decoder [12] at lower values of DFR, while LEDAcrypt employs a canonical two-iteration (parallel) bit-flipping decoder for which it combines a first-iteration model [13], [14] with a code-specific lower bound for the correction capability of the second iteration [15], [16]. In the line of research related to the study of DFR estimates, J. Chaulet reports in [17] an estimate of the distribution of the syndrome weights for QC-MDPC codes and models the

probability of parity check equations to be unsatisfied at the first iteration of the parallel bit-flipping decoder. Chaulet notes that error vectors with remarkably low syndrome weight tend to be decoded with lower probability. In an affine line of work, [18], [19] observes that errors vectors having regularities such as runs of ones are less frequently decoded by QC-MDPC iterative decoders. The authors of [20] and [21] observed that, while indeed harder to decode, the error vectors represent a relatively low amount of the overall non-decodable errors. Going in a different direction, J-P. Tillich [15] provided a code-specific technique to determine the maximum weight of an error guaranteed to be corrected by a single iteration bit-flipping decoder on a QC-LDPC/QC-MDPC code, proving that the DFR falls exponentially for MDPC codes as the code length increases under an assumption on the code density.

Contribution. We describe a DFR estimation technique for two-iterations (parallel) bit flipping decoder for (v, w)-regular LDPC/MDPC codes, such as the ones of [5], [6]. We provide a closed form estimate of the syndrome weight distribution, improving on [17], and a technique to assess the number of mismatched bits between the error vector estimate and the actual error vector values after the first iteration, improving the accuracy with respect to [13], through direct counting arguments, diverging from the Markov chain approach of [18], [19]. Finally, we propose a technique to estimate the mismatches between the error estimate and the actual error after the second iteration, obtaining an estimate for the two-iterations DFR. We validate our results with extensive numerical simulations (C code available at [22]), while we provide an extended version of this work at [23].

II. Decoding failure rate model

In the following, we make this assumption: *the rows of H are independently and uniformly random drawn from the set of binary vectors having length n and w asserted bits, acknowledging it as an approximation, as the weight of the parity-check matrix columns will be kept constant to v*. We share this assumption with the original paper by Gallager [1]. In the following, $\text{BIN}(\mathtt{tr}, \mathtt{spr}, \mathtt{ns})$ indicates the probability mass function of obtaining $\mathtt{ns}$ success events out of $\mathtt{tr}$ independent events with a success probability of $\mathtt{spr}$.

Modeling Syndrome Weight Distribution. Denote as $(e, s)_l$, $l \in \{0, 1, 2 \ldots, t\}$, a pair of values: an error vector e and its corresponding syndrome $s = He$ both indexed by the weight of the error vector l. Consider a given syndrome and its corresponding error vector with weight t as the last pair in the sequence $(e, s)_0, (e, s)_1, (e, s)_2, \ldots, (e, s)_t$, where $(e, s)_0$ is the null error vector and its (null) syndrome, while $(e, s)_l$, $l \geq 1$, denotes a pair with an error vector that includes the same set bits of the error vector in $(e, s)_{l-1}$ plus an additional set bit that is uniformly randomly placed in one out of the $n-(l-1)$ available positions. The syndrome value in $(e, s)_l$, $l \geq 1$, thus differs from the one in $(e, s)_{l-1}$ for the bitwise addition of a column of the parity-check matrix H.

We model the Hamming weight of each syndrome in the sequence as a discrete random variable $\mathcal{W}_l$ bound to a probability mass function $\Pr(\mathcal{W}_l = y)$, with $l \in \{0, \ldots, t\}$, $y \in \{0, \ldots, r\}$, represented as an array $\mathtt{wp}_{(l)} = [\mathtt{wp}_{(l),0}, \ldots, \mathtt{wp}_{(l),x}, \ldots, \mathtt{wp}_{(l),r}]$.

Starting from the distribution of the weight of the syndrome of a null error vector, $\mathtt{wp}_{(0)} = [1, 0, \ldots, 0]$, the random variable $\mathcal{W}_t$ associated to the weight of the syndrome at hand coincides with final state of discrete-time non-homogeneous Markov chain with $r+1$ states. Such a Markov chain is uniquely defined by $\mathtt{wp}_{(0)}$ and the transition matrices $\mathtt{P}_{(l)} = [p_{x,y,l}]_{x,y \in \{0,\ldots,r\}}$. Specifically, the distribution of each random variable $\mathcal{W}_l$ is derived through the following vector-matrix multiplication $\mathtt{wp}_{(l)} = \mathtt{wp}_{(l-1)} \cdot \mathtt{P}_{(l)}$, with $l \in \{1, \ldots, t\}$, where each transition probability $p_{x,y,l} = \Pr(\mathcal{W}_l = y | \mathcal{W}_{l-1} = x)$ is a function of the starting and ending weight of the syndrome as well as of the step l considered along the chain. We denote as $\mathcal{F}_l \in \{0, \ldots, \min(w, l)\}$ the discrete random variable modeling the count of flips of any single bit of the syndrome of an error vector with weight l, during the computation of the syndrome itself.

The probability mass function of $\mathcal{F}_l$ follows an hypergeometric distribution $\phi_l(f, l) = \Pr(\mathcal{F}_l = f) = \frac{\binom{w}{f}\binom{n-w}{l-f}}{\binom{n}{l}}$. Indeed, the l set positions in the error vector select l positions in any single row of H, which in turn corresponds to a syndrome bit. Whenever one of such selected row positions contains one of the w set bits out of n, the syndrome bit corresponding to the row at hand is flipped. Following any syndrome bit along the sequence $(e, s)_0, \ldots, (e, s)_{l-1}, (e, s)_l, \ldots$, we note that $\Pr(\mathcal{F}_l = f+1 \,|\, \mathcal{F}_{l-1} = f) = \frac{w-f}{n-l}$, while the event $\mathcal{F}_{l-1} = f$ implies that the syndrome bit is either clear or set, depending on f being either even or odd, respectively.

Thus the probability $\pi_{\substack{l-1 \to l \\ \mathtt{flip}\, 0 \to 1}}(l)$ of flipping at step l any syndrome bit cleared at step $l-1$, depends on the value of the step, and is derived as $\frac{\sum_f \Pr(\mathcal{F}_l = f+1 | \mathcal{F}_{l-1} = f) \Pr(\mathcal{F}_{l-1} = f)}{\sum_f \Pr(\mathcal{F}_{l-1} = f)}$, where the f ranges over the even values in $\{0, \ldots, \min(w, l)\}$. Analogously, the probability $\pi_{\substack{l-1 \to l \\ \mathtt{flip}\, 1 \to 0}}(l)$ of flipping at step l any syndrome bit that was set at step $l-1$, also depends on l and is derived applying the same formula with $f \in \{0, \ldots, \min(w, l)\}$ ranging over odd values:

$$\pi_{\substack{l-1 \to l \\ \mathtt{flip}\, 0 \to 1}}(l) = \left(\sum_{f \atop 0,\mathtt{even}}^{\min(l,w)} \left(\frac{w-f}{n-l} \cdot \phi_l(f, l) \right) \right) \Big/ \left(\sum_{f \atop 0,\mathtt{even}}^{\min(l,w)} \phi_l(f, l) \right)$$

$$\pi_{\substack{l-1 \to l \\ \mathtt{flip}\, 1 \to 0}}(l) = \left(\sum_{f \atop 0,\mathtt{odd}}^{\min(l,w)} \left(\frac{w-f}{n-l} \cdot \phi_l(f, l) \right) \right) \Big/ \left(\sum_{f \atop 1,\mathtt{odd}}^{\min(l,w)} \phi_l(f, l) \right)$$

Analyzing the change of the syndrome weight from step $l-1$ to step l, we derive the probability mass function $p_{x,y,l} = \Pr(\mathcal{W}_l = y | \mathcal{W}_{l-1} = x)$, with $x, y \in \{0, \ldots, r\}$. At each step, v bits (the column weight of H) of the syndrome flip (out of r). Thus, the weight y of the syndrome at step l is obtained from flipping up a clear bits out of $r-x$, and flipping down $v-a$ set bits out of x, for all admissible a, i.e., $a \in \{\max\{0, v-x)\}, \ldots, \min\{r-x, v\}\}$. The weight of the syndrome after the l-th step is $y = x + a - (v - a)$, or equivalently $r - y = r - x - a + (v - a)$, from which we derive $a = \frac{x-y+v}{2}$. Given $\mathcal{W}_{l-1} = x$ and a value for

a, define two events: $\mathtt{E}_{1,a}$: a bits are flipped up in $r-x$ flip trials; $\mathtt{E}_{2,a}$: $v-a$ bits are flipped down in x flip trials. The probability mass function $\Pr(\mathtt{E}_{1,a})=\varphi(x,a,l)$ describes the probability of flipping up a bits of the syndrome at step $l-1$, while $\Pr(\mathtt{E}_{2,a})=\psi(x,a,l)$ describes the probability of flipping down $v-a$ bits of the at the step $l-1$. Note that, depending on the weight of the syndrome at step $l-1$, there are $\binom{r-x}{a}$ possibile patterns for the said flip-ups and $\binom{x}{v-a}$ possibile patterns for the said flips down. We thus have that $\varphi(x,a,l) = \textsc{Bin}(r-x, \pi_{\substack{l-1\to l\\ \mathtt{flip}\,0\to 1}}(l), a)$ while $\psi(x,a,l) = \textsc{Bin}(x, \pi_{\substack{l-1\to l\\ \mathtt{flip}\,1\to 0}}(l), v-a)$.

Given a specific value for a and $\mathcal{W}_{l-1}=x$, the event modelling the v flips in the transition of the syndrome weight from x to $y{=}x{+}2a{-}v$, from step $l-1$ to step l, is $\mathtt{E}_{1,a}\cap\mathtt{E}_{2,a}$. The event is bound to the probability mass function $\Pr(\mathtt{E}_{1,a})\cdot\Pr(\mathtt{E}_{2,a}) = \varphi(x,a,l)\cdot\psi(x,a,l)$ since $\mathtt{E}_{1,a}$ and $\mathtt{E}_{2,a}$ are independent (they take place on disjoint sets of bits).

Considering all the admissible values for a, the probability mass function $\Pr(\mathcal{W}_{l-1}=x)=\omega(x,l)$, models the probability of moving to any valid syndrome weight y at step l, is: $\omega(x,l)=\sum_{i=\max\{0,v-x\}}^{\min\{r-x,v\}}\left(\varphi(x,i,l)\cdot\psi(x,i,l)\right)$.
Thus, the transition probability $p_{x,y,l}{=}\Pr(\mathcal{W}_l=y|\mathcal{W}_{l-1}=x)$ can be written as a function of the step count l, and of both the starting and ending weights of the syndrome.

$$p_{x,y,l}=\begin{cases}1, & l=1, x=0, y=v\\ \rho(x,y,l), & \begin{array}{l}l\geq 2\\ \max(0,x-v)\leq y\leq \min(x+v,r)\\ y\equiv_2 (x+v)\end{array}\\ 0, & \text{otherwise}\end{cases}$$

with $\rho(x,y,l)=\frac{\varphi(x,\frac{x-y+v}{2},l)\cdot\psi(x,\frac{x-y+v}{2},l)}{\omega(x,l)}$, where $\pi_{\substack{l-1\to l\\ \mathtt{flip}\,1\to 0}}(l)$ is not defined in the special case $l=1, x=0, y=v$. The value $p_{0,v,1}=1$ is justified as a null syndrome (i.e. with $x=0$) will deterministically turn into a weight v syndrome when a column of H is added. The algorithmic procedure deriving the distribution of the weight of the syndrome of an error vector with weigth t, $\Pr(\mathcal{W}_t=y)$ is available in the extended version of this paper at [23].

First Iteration of a Bit Flipping Decoder. We define a parity-check equation as $\sum_{j=0}^{n-1}h_{i,j}e_j = s_i$, where e_j are the unknowns, $h_{i,j}$ the coefficients and s_i the constant known term. The equation is *satisfied* if $s_i{=}0$, *unsatisfied* if $s_i{=}1$. The (parallel) bit flipping decoding algorithm iteratively estimates the most likely value $\bar{e}$ of the error vector e, given s and H, starting from the initial value $\bar{e}=0_{1\times n}$.

We estimate the distribution of the random variable $\mathcal{E}_{(\mathtt{iter})}$ modeling the Hamming weight of $\bar{e}\oplus e$ after the $\mathtt{iter}$-th iteration of the decoding algorithm, i.e. the number of mismatches between $\bar{e}\oplus e$. The probability mass function $\Pr(\mathcal{E}_{(\mathtt{iter})}=\mathrm{d})$, $\mathrm{d}{\in}\{0,\dots,n\}$, will be considered only with $\mathtt{iter}>0$, because $\Pr(\mathcal{E}_{(0)}=t){=}1$, before the beginning of the decoding algorithm, when $\bar{e}{=}0$. The probability mass function $\Pr(\mathcal{E}_{(\mathtt{iter})}=\mathrm{d})$ will be obtained, using the distribution of the weigth of the syndrome of an error vector of weight t, (i.e., $\mathcal{W}_t=y$), as follows: $\Pr(\mathcal{E}_{(\mathtt{iter})}=\mathrm{d}) = \sum_{y=0}^{r}\left(\Pr\left(\mathcal{E}_{(\mathtt{iter})}=\mathrm{d}\mid\mathcal{W}_t=y\right)\Pr(\mathcal{W}_t=y)\right)$.

From now on, the goal of the analysis is going to be the estimate of the probability $\Pr\left(\mathcal{E}_{(\mathtt{iter})}=\mathrm{d}\mid\mathcal{W}_t=y\right)$, $y{=}\mathtt{wt}(s)$: $\mathtt{iter}{=}1$ in this subsection, and $\mathtt{iter}{=}2$ in the next one. Furthermore, for the sake of brevity, in the definition of any event and in the formulas of probability mass functions we are going to omit any reference to the weight of the syndrome.

In the analysis of the first iteration of the decoding algorithm, we denote as $\mathcal{S}_i$, $i{\in}\{0,\dots,r\}$, the random variables modeling the value taken by the i-th bit of the syndrome, s_i, at the beginning of each iteration of the decoding algorithm. Therefore, for each unsatisfied and satisfied parity-check equation, the probability to observe a clear or set constant term is: $\Pr(\mathcal{S}_i=0)=\frac{r-y}{r}$, $\Pr(\mathcal{S}_i=1)=\frac{y}{r}$, respectively.

We denote as $\mathtt{E}_{(i,j),0}$ or $\mathtt{E}_{(i,j),1}$ the event of an error bit being either clear or set, respectively, in a position j captured by one of the w set coefficients of the row $h_{i,:}$ in the i-th parity-check equation, i.e.: $\mathtt{E}_{(i,j),0}{=}\{h_{i,j}=1 \text{ and } e_j=0\}$, and $\mathtt{E}_{(i,j),1}{=}\{h_{i,j}=1 \text{ and } e_j=1\}$, respectively.

Observe that, using $\mathcal{F}_t$ from the previous section as the random variable modeling the count of bitflips determining s_i, for any i, in the computation $s=He^T$, the probabilities $\Pr(\mathcal{S}_i=0)$ and $\Pr(\mathcal{S}_i=1)$ can be written also as: $\Pr(\mathcal{S}_i=0) = \Pr\left(\bigcup_{f=0,\mathtt{even}}^{\min(t,w)}(\mathcal{F}_t=f)\right) = \sum_{f=0,\mathtt{even}}^{\min(t,w)}\Pr(\mathcal{F}_t=f)$, $\Pr(\mathcal{S}_i=1) = \Pr\left(\bigcup_{f=1,\mathtt{odd}}^{\min(t,w)}(\mathcal{F}_t=f)\right) = \sum_{f=1,\mathtt{odd}}^{\min(t,w)}\Pr(\mathcal{F}_t=f)$, where the last step is mutuated by the fact that the events $(\mathcal{F}_t=f)$ are disjoint. Note that $\mathcal{F}_t=f$ implies that there are f variables of the i-th parity-check equation which are both set to 1 and have a corresponding coefficient set to 1 (out of the w set ones present in the equation). Here, $\Pr(\mathcal{F}_t=f)$ is a shorthand for $\Pr(\mathcal{F}_t=f\,|\,\mathcal{W}_t=y)$, and the formula for the probability mass function of $\Pr(\mathcal{F}_t=f)$ reported in the previous section cannot be applied anymore. The complete derivation of $\Pr(\mathcal{F}_t=f\,|\,\mathcal{W}_t=y)$, is available in [23].

Consider now, $p_{\mathtt{unsat}|0}=\Pr\left(\mathcal{S}_i=1\mid\mathtt{E}_{(i,j),0}\right)$, that is the probability that, given an error variable $e_j{=}0$, the constant term s_i of the i-th parity-check equation, where the variable is involved with a coefficient $h_{i,j}{=}1$, is $s_i{=}1$. Since the rows of the parity-check matrix are assumed to be independent and with the same weight, we have that $p_{\mathtt{unsat}|0}$ is independent from the index of the parity-check equation i:

$$p_{\mathtt{unsat}|0}=\frac{\Pr\left(\mathtt{E}_{(i,j),0}\,|\,\mathcal{S}_i{=}1\right)\Pr(\mathcal{S}_i{=}1)}{\Pr\left(\mathtt{E}_{(i,j),0}\,|\,\mathcal{S}_i{=}1\right)\Pr(\mathcal{S}_i{=}1)+\Pr\left(\mathtt{E}_{(i,j),0}\,|\,\mathcal{S}_i{=}0\right)\Pr(\mathcal{S}_i{=}0)}$$

The factors $\Pr\left(\mathtt{E}_{(i,j),0}\mid\mathcal{S}_i=0\right)$, $\Pr\left(\mathtt{E}_{(i,j),0}\mid\mathcal{S}_i=1\right)$ are:

$$\Pr\left(\mathtt{E}_{(i,j),0}\mid\mathcal{S}_i=0\right) = \frac{\Pr\left(\mathtt{E}_{(i,j),0}\cap\mathcal{S}_i{=}0\right)}{\Pr(\mathcal{S}_i{=}0)} = \frac{\sum_{f=0,\mathtt{even}}^{\min(t,w)}\Pr\left(\mathtt{E}_{(i,j),0}\,|\,\mathcal{F}_t{=}f\right)\Pr(\mathcal{F}_t{=}f)}{\sum_{f=0,\mathtt{even}}^{\min(t,w)}\Pr(\mathcal{F}_t{=}f)}$$

$$\Pr\left(\mathtt{E}_{(i,j),0}\mid\mathcal{S}_i=1\right) = \frac{\Pr\left(\mathtt{E}_{(i,j),0}\cap\mathcal{S}_i{=}1\right)}{\Pr(\mathcal{S}_i{=}1)} = \frac{\sum_{f=1,\mathtt{odd}}^{\min(t,w)}\Pr\left(\mathtt{E}_{(i,j),0}\,|\,\mathcal{F}_t{=}f\right)\Pr(\mathcal{F}_t{=}f)}{\sum_{f=1,\mathtt{odd}}^{\min(t,w)}\Pr(\mathcal{F}_t{=}f)}.$$

Note that $\Pr\left(\mathtt{E}_{(i,j),0}\mid\mathcal{F}_t=f\right){=}\frac{w-f}{w}$, since knowing $\mathcal{F}_t=f$ for the i-th parity-check equation implies that there are $w-f$ variables e_j, among the w ones with their coefficient $h_{i,j}=1$, which are equal to zero. The

expression for $p_{\texttt{unsat}|1}$ can be obtained with a derivation analogous to the one for $p_{\texttt{unsat}|0}$ (full derivation in [23]). We derive the distribution $\mathcal{U}_j$ modelling $\texttt{upc}_j$ variables computed at each decoder iteration. In particular, we model $\Pr(\mathcal{U}_j = u \mid e_j = 0)$ and $\Pr(\mathcal{U}_j = u \mid e_j = 1)$ through a counting argument (fully detailed in [23]), obtaining $\Pr(\mathcal{U}_j = u \mid e_j = 0) = \textsc{Bin}(v, p_{\texttt{unsat}|0}, u)$ and $\Pr(\mathcal{U}_j = u \mid e_j = 1) = \textsc{Bin}(v, p_{\texttt{unsat}|1}, u)$.

We then obtain $p_{\texttt{flip}|0} = \Pr(\texttt{upc}_j \geq \texttt{th} | e_j = 0)$, that is the probability that the decoder flips the j-th error estimate value $\bar{e}_j$, when $e_j = 0$. $p_{\texttt{flip}|0}$ is to the union of all events where the $\texttt{upc}_j$ value is equal or greater than the threshold $\texttt{th}$: $p_{\texttt{flip}|0} = \sum_{a=\texttt{th}}^{v} \Pr(\mathcal{U}_j = a \mid e_j = 0)$. We denote with $p_{\neg\texttt{flip}|0}$ its complement $1 - p_{\texttt{flip}|0}$, i.e., the decoder decides not to flip the error estimate bit. Analogously, we obtain $p_{\texttt{flip}|1}$ (the probability that the decoder flips an error estimate bit, assuming that the corresponding error vector bit is set) as $p_{\texttt{flip}|1} = \sum_{a=\texttt{th}}^{v} \Pr(\mathcal{U}_j = a \mid e_j = 1)$, denoting as $p_{\neg\texttt{flip}|1} = 1 - p_{\texttt{flip}|1}$.

Denote with $\bar{e}_{(i)}$ the value of $\bar{e}$ after the bit flips of the i-th iteration have been applied. Given the previous probabilities, we consider the event $\texttt{E}_{(\texttt{d}_+)} = |(\{0, \ldots, n-1\} \setminus \text{Supp}(e)) \cap \text{Supp}(\bar{e}_{(1)})| = \texttt{d}_+$, that is $\texttt{d}_+$ flips of $\bar{e}$ happen on the $n - t$ positions where $e_j = 0$, and the event $\texttt{E}_{(\texttt{d}_-)} = |\text{Supp}(e) \cap \text{Supp}(\bar{e}_{(1)})| = \texttt{d}_-$, that is $\texttt{d}_-$ flips happen on the t positions where $e_j = 1$ has a set bit. We have $\delta_+(\texttt{d}_+) = \Pr(\texttt{E}_{(\texttt{d}_+)}) = \textsc{Bin}(n - t, p_{\texttt{flip}|0}, \texttt{d}_+)$, and $\delta_-(\texttt{d}_-) = \Pr(\texttt{E}_{(\texttt{d}_-)}) = \textsc{Bin}(t, p_{\texttt{flip}|1}, \texttt{d}_-)$.

Having obtained closed formulas for $\delta_+(\texttt{d}_+)$ and $\delta_-(\texttt{d}_-)$, we observe that the events $\texttt{E}_{(\texttt{d}_+)}$ and $\texttt{E}_{(\texttt{d}_-)}$ act on disjoint subsets of the bits of e and are thus independent. We are thus able to obtain $\Pr(\mathcal{E}_{(1)} = \texttt{d} \mid \mathcal{W}_t = y)$ considering the set $\mathbf{D}$ of pairs $(\texttt{d}_+, \texttt{d}_-)$ such that $\texttt{d}{=}t{-}\texttt{d}_-{+}\texttt{d}_+$: $\Pr(\mathcal{E}_{(1)} = \texttt{d} \mid \mathcal{W}_t = y) = \Pr\left(\bigcup_{(\texttt{d}_+,\texttt{d}_-)\in\mathbf{D}} (\texttt{E}_{(\texttt{d}_+)} \cap \texttt{E}_{(\texttt{d}_-)})\right) = \sum_{(\texttt{d}_+,\texttt{d}_-)\in\mathbf{D}} \delta_+(\texttt{d}_+) \cdot \delta_-(\texttt{d}_-)$ The conditioning on the value of $\mathcal{W}_t$ is embedded in the fact that $(\mathcal{W}_t = y)$ is employed to derive $\Pr(\mathcal{S}_i = 0)$ and $\Pr(\mathcal{S}_i = 1)$, which are needed to obtain $p_{\texttt{flip}|0}$ and $p_{\texttt{flip}|1}$.

Second Iteration of the Bit Flipping Decoder. We partition the bits of $\bar{e}_{(1)}$ into four classes. Each class is labeled with a pair (a, b), where, for each bit $\bar{e}_j$ belonging to the class, we have $a = e_j, b = e_j \oplus \bar{e}_{(1),j}$. From now on, we denote as $\mathbf{J}_{a,b}$ with $a, b{\in}\{0, 1\}$ the sets of positions of the bits in the class (a, b). We estimate the probabilities of flips being applied to the bits in each of the four (a, b) classes considering their values obtained after the first iteration $\bar{e}_{(1)}$, which we denote as $p_{\texttt{flip}|00}$, $p_{\texttt{flip}|01}$, $p_{\texttt{flip}|10}$, and $p_{\texttt{flip}|11}$, derive as a function of the cardinalities of the sets, $|\mathbf{J}_{0,1}| = \epsilon_{01}$ and $|\mathbf{J}_{1,1}| = \epsilon_{11}$. Note that, from the analysis of the first iteration, we are able to compute $\Pr(\epsilon_{01} = \texttt{d}_+)$ as $\delta_+(\texttt{d}_+)$ and $\Pr(\epsilon_{11} = t - \texttt{d}_-) = \delta_-(\texttt{d}_-)$.

Once we obtain $p_{\texttt{flip}|00}$, $p_{\texttt{flip}|01}$, $p_{\texttt{flip}|10}$, and $p_{\texttt{flip}|11}$, we derive the probability of performing a correct decoding at the second iteration as $\Pr(\mathcal{E}_{(2)} = 0 | \mathcal{W}_t = y)$, combining together the said probabilities, as follows:

$$\sum_{(\epsilon_{01},\epsilon_{11})\in\{0,\ldots,n-t\}\times\{0,\ldots,t\}} \Big(\delta_+(\epsilon_{01}) \cdot \delta_-(t - \epsilon_{11}) \cdot (1 - p_{\texttt{flip}|00}(\epsilon_{01}, \epsilon_{11}))^{n-t-\epsilon 01} \cdot p_{\texttt{flip}|01}(\epsilon_{01}, \epsilon_{11})^{\epsilon_{01}} \cdot (1 - p_{\texttt{flip}|10}(\epsilon_{01}))^{t-\epsilon_{11}} \cdot p_{\texttt{flip}|11}(\epsilon_{01}, \epsilon_{11}))^{\epsilon_{11}}\Big).$$

Finally, we derive the Decoding Failure Rate (DFR) after the second decoder iteration as $\text{DFR} = 1 - \Pr(\mathcal{E}_{(2)} = 0) = 1 - \sum_{y=0}^{r} \Pr(\mathcal{E}_{(2)} = 0 | \mathcal{W}_t = y) \Pr(\mathcal{W}_t = y)$.

We note that our result can be combined with the ones from [13], [15]: to do so, given the specific matrix H required by [13], [15] compute the weight which an iteration is guaranteed to correct, τ, and subsequently obtain $\Pr(\mathcal{E}_{(2)} = 0 | \mathcal{W}_t = y)$ excluding from the weighted sum employed to derive it all the terms resulting in an amount of mismatches after the first iteration $\leq \tau$.

We now provide a summary of the steps to derive $p_{\texttt{flip}|00}$, while the full derivations are available in [23]. We start by computing the probability $p_{\texttt{flip}|0,\texttt{OneEqSat}}$ of flipping up a bit of $\bar{e}$, given that it appears in a satisfied parity-check equation and the corresponding error bit value is 0, that is $p_{\texttt{flip}|0,\texttt{OneEqSat}} = \Pr(\texttt{upc}_j \geq \texttt{th} \mid e_j = 0, h_{i,j} = 1, s_i = 0)$ is obtained as $p_{\texttt{flip}|0,\texttt{OneEqSat}} = \sum_{a=th}^{v-1} \textsc{Bin}(v{-}1, p_{\texttt{unsat}|0}, a)$, i.e., the probability of the union of the events where $a \geq \texttt{th}$ parity-check equations are unsatisfied, knowing that one out of the v parity-check equations is satisfied, hence reducing the number of trials to $v{-}1$. Analogously, $p_{\texttt{flip}|0,\texttt{OneEqUnsat}}{=}\Pr(\texttt{upc}_j \geq \texttt{th} \mid e_j = 0, h_{i,j} = 1, s_i = 1)$ as $p_{\texttt{flip}|0,\texttt{OneEqUnsat}} = \sum_{a=th-1}^{v-1} \textsc{Bin}(v - 1, p_{\texttt{unsat}|0}, a)$, and $p_{\neg\texttt{flip}|1,\texttt{OneEqSat}} = \sum_{a=0}^{\texttt{th}-1} \textsc{Bin}(v - 1, p_{\texttt{unsat}|1}, a)$, while $p_{\neg\texttt{flip}|1,\texttt{OneEqUnsat}} = \sum_{a=0}^{\texttt{th}-2} \textsc{Bin}(v - 1, p_{\texttt{unsat}|1}, a)$.

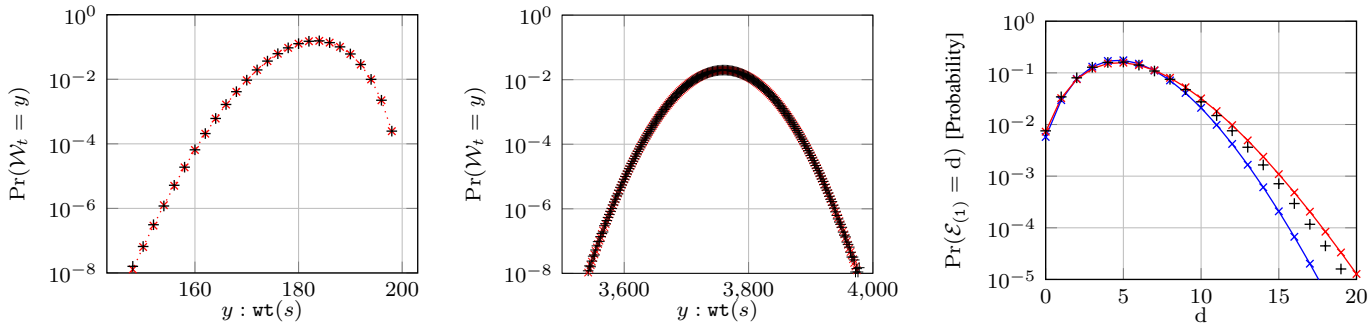
We model with $\mu(\texttt{nsat}, \texttt{nunsat}, a, \epsilon_{01}, \epsilon_{11})$ the probability that, as a result of the flips from the first iteration, among the v parity check equations which involve a given position in $\mathbf{J}_{0,0}$, out of the $v{-}a$ satisfied ones $\texttt{nsat}$ become unsatisfied, while, out of the a unsatisfied ones $\texttt{nunsat}$ stay unsatisfied; we derive, from the previous quantities, the probabilities that parity checks involving bits in $\mathbf{J}_{0,0}$ become unsatisfied ($p_{00|\texttt{BecomeUnsat}}$) or stay unsatisfied ($p_{00|\texttt{StayUnsat}}$). We have $\mu(\texttt{nsat}, \texttt{nunsat}, a) = \textsc{Bin}(v - a, p_{00|\texttt{BecomeUnsat}}(\epsilon_{01}, \epsilon_{11}), \texttt{nsat}) \cdot \textsc{Bin}(a, p_{00|\texttt{StayUnsat}}(\epsilon_{01}, \epsilon_{11}), \texttt{nunsat})$. Finally, we obtain $p_{\texttt{flip}|00}$ as a function of ϵ_{01} and ϵ_{11}, $p_{\texttt{flip}|00}(\epsilon_{01}, \epsilon_{11})$:

$$p_{\texttt{flip}|00}(\epsilon_{01}, \epsilon_{11}) = \sum_{a=0}^{\texttt{th}_{(1)}-1} \Pr(\mathcal{U}_j = a, j \in \mathbf{J}_{0,0}) \cdot \left(\sum_{\texttt{nsat}=0}^{v-a} \sum_{\substack{\texttt{nunsat} = \\ \max(0,\, \texttt{th}_{(2)} - \texttt{nsat})}}^{a} \mu(\texttt{nsat}, \texttt{nunsat}, a, \epsilon_{01}, \epsilon_{11}) \right)$$

$p_{\texttt{flip}|01}(\epsilon_{01}, \epsilon_{11}), p_{\texttt{flip}|10}(\epsilon_{01}, \epsilon_{11})$ and, $p_{\texttt{flip}|11}(\epsilon_{01}, \epsilon_{11})$ are obtained through an analogous line of reasoning (in [23]).

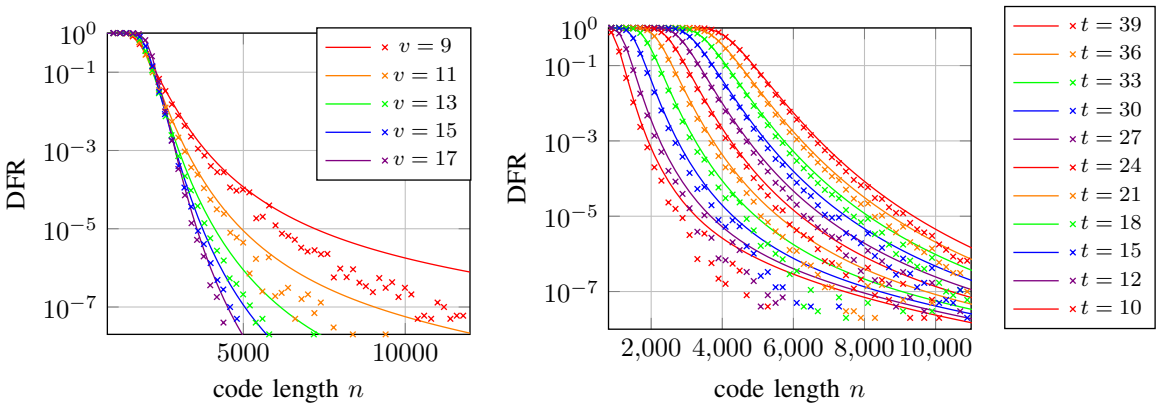
III. Numerical validation

In this section, we provide numerical validations of our results. The numeric simulations were run on two Dell Pow-



(a) $n = 4400, k = 2200, v = 11, t = 18$ (b) $n_0 = 4, p = 13397, v = 83, t = 66$ (c) $n_0 = 4, p = 13397, v = 83, t = 66$

Fig. 1. Numerical validation of the model of syndrome weight distribution, picking a communication-grade code parameter set (a) and a cryptography-grade code parameter set (b); Distribution of $\mathrm{d} = \mathtt{wt}(\bar{e}_{(1)} \oplus e)$ (c). Numerical results obtained with 10^9 random syndrome samples. ×— depicts our estimate, + reports the numerical simulations, ×— depicts the estimates from [6]



(a) $v \in \{9, 11, 13, 15, 17\}, t = 18$ (b) $t \in \{10, \ldots, 39\}, v = 11$

Fig. 2. Two iterations DFR values for $(v, 2v)$-regular LDPC codes with rate $\frac{k}{n} = \frac{1}{2}$, parallel decoder employing majority thresholds, i.e., $\mathtt{th1} = \mathtt{th2} = \lceil \frac{v+1}{2} \rceil$. Each data point obtained performing either 10^8 decoding actions, or reaching 100 decoding failures. Solid lines are the model, crosses are numerical simulations

erEdge R630 nodes, each one endowed with two Intel Xeon CPU E5-2698 v4 (20 cores/40 threads each), and a Dell PowerEdge R7425 equipped with two AMD Epyc 7551 (32 cores/64 threads each), taking around 50k core-hours. The memory footprint of the simulations was small (<200MiB).

Figure 1 reports the comparison between our modeled (red) and the numerically estimated (black) distribution of syndrome weights $\Pr(\mathcal{W}_t = y)$ for a small $(11-22)$ regular LDPC code with code length $n = 4400$ and error weight $t = 18$ (Figure 1a), and for the code with rate $\frac{3}{4}$ employed in the LEDAcrypt specification [6] (Figure 1b), for NIST security category 1. In both cases, our estimation technique provides a very good match for the numerically simulated probabilities. Figure 1c reports the distribution of d, the weight of $e \oplus \bar{e}_{(1)}$, comparing our model (red) ad the one employed in [6] (blue) with numerical data (black): our model provides a closer fit to the sample distribution of d, w.r.t. the one employed in [6]. This improvement in fitness results in a significant improvement on the value of the expected DFR for cryptographic-grade parameters in LEDACrypt (NIST category 1 parameters in table). Parameters designed for a 2^{-64} DFR, actually provide a better (smaller) DFR than the one required to achieve security against active attackers (2^{-128}). Finally, Figures 2a and 2b compare numerical DFR values of a two-iterations parallel bit-flipping decoder with our estimation technique, while varying either the code density (left) or the number of errors (right) over the code length range $n \in \{1200, \ldots, 12000\}$. We provide a reliable estimate of the waterfall region, and a conservative estimate for the floor region of the codes. More numerical results are available in the extended version at [23].

Acknowledgements. This work was carried out with partial financial support of the Italian MUR PRIN 2022 project POINTER ID-2022M2JLF2 and project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU. We are grateful to Datacloud (https://datacloud.polimi.it/) for providing 50k core hours for simulations.

n	k	v	t	τ	LEDAcrypt DFR	Our DFR
46742	23371	71	130	10	2^{-64}	2^{-147}
48201	32134	79	83	9	2^{-64}	2^{-139}
53588	40191	83	66	8	2^{-64}	2^{-134}

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, 1962. [Online]. Available: https://doi.org/10.1109/TIT.1962.1057683

[2] S. Litsyn and V. Shevelev, "On ensembles of low-density parity-check codes: Asymptotic distance distributions," *IEEE Trans. Inf. Theory*, vol. 48, no. 4, pp. 887–908, 2002. [Online]. Available: https://doi.org/10.1109/18.992777

[3] S. Ouzan and Y. Be'ery, "Moderate-Density Parity-Check Codes," *CoRR*, vol. abs/0911.3262, 2009. [Online]. Available: http://arxiv.org/abs/0911.3262

[4] R. Misoczki, J. Tillich, N. Sendrier, and P. S. L. M. Barreto, "MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes," in *Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul, Turkey, July 7-12, 2013*. IEEE, 2013, pp. 2069–2073. [Online]. Available: https://doi.org/10.1109/ISIT.2013.6620590

[5] N. Aragon, P. S. L. M. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Ghosh, S. Gueron, T. Güneysu, C. A. Melchor, R. Misoczki, E. Persichetti, J. Richter-Brockmann, N. Sendrier, J.-P. Tillich, V. Vasseur, and G. Zémor, "BIKE: Bit Flipping Key Encapsulation. Round 4 Submission," [Online] Available: https://bikesuite.org/files/v5.0/BIKE_Spec.2022.10.10.1.pdf, 2023.

[6] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini, "LEDAcrypt - version 3.0 Specification," [Online] Available: https://www.ledacrypt.org/documents/LEDAcrypt_v3.pdf, 2023.

[7] ——, "LEDAcrypt: QC-LDPC Code-Based Cryptosystems with Bounded Decryption Failure Rate," in *Code-Based Cryptography - 7th International Workshop, CBC 2019, Darmstadt, Germany, May 18-19, 2019, Revised Selected Papers*, ser. Lecture Notes in Computer Science, M. Baldi, E. Persichetti, and P. Santini, Eds., vol. 11666. Springer, 2019, pp. 11–43. [Online]. Available: https://doi.org/10.1007/978-3-030-25922-8_2

[8] ——, "LEDAkem: A Post-quantum Key Encapsulation Mechanism Based on QC-LDPC Codes," in *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings*, ser. Lecture Notes in Computer Science, T. Lange and R. Steinwandt, Eds., vol. 10786. Springer, 2018, pp. 3–24. [Online]. Available: https://doi.org/10.1007/978-3-319-79063-3_1

[9] A. Barenghi and G. Pelosi, "Constant weight strings in constant time: a building block for code-based post-quantum cryptosystems," in *Proceedings of the 17th ACM International Conference on Computing Frontiers, CF 2020, Catania, Sicily, Italy, May 11-13, 2020*, M. Palesi, G. Palermo, C. Graves, and E. Arima, Eds. ACM, 2020, pp. 132–141. [Online]. Available: https://doi.org/10.1145/3387902.3392630

[10] ——, "Constant weight strings in constant time: a building block for code-based post-quantum cryptosystems," in *Proceedings of the 17th ACM International Conference on Computing Frontiers, CF 2020, Catania, Sicily, Italy, May 11-13, 2020*, M. Palesi, G. Palermo, C. Graves, and E. Arima, Eds. ACM, 2020, pp. 132–141. [Online]. Available: https://doi.org/10.1145/3387902.3392630

[11] Q. Guo, T. Johansson, and P. S. Wagner, "A Key Recovery Reaction Attack on QC-MDPC," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1845–1861, 2019. [Online]. Available: https://doi.org/10.1109/TIT.2018.2877458

[12] N. Sendrier and V. Vasseur, "On the decoding failure rate of QC-MDPC bit-flipping decoders," in *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers*, ser. Lecture Notes in Computer Science, J. Ding and R. Steinwandt, Eds., vol. 11505. Springer, 2019, pp. 404–416. [Online]. Available: https://doi.org/10.1007/978-3-030-25510-7_22

[13] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini, "A Failure Rate Model of Bit-flipping Decoders for QC-LDPC and QC-MDPC Code-based Cryptosystems," in *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications, ICETE 2020 - Volume 2: SECRYPT, Lieusaint, Paris, France, July 8-10, 2020*, P. Samarati, S. D. C. di Vimercati, M. S. Obaidat, and J. Ben-Othman, Eds. ScitePress, 2020, pp. 238–249. [Online]. Available: https://doi.org/10.5220/0009891702380249

[14] ——, "Analysis of In-Place Randomized Bit-Flipping Decoders for the Design of LDPC and MDPC Code-Based Cryptosystems," in *E-Business and Telecommunications - 17th International Conference on E-Business and Telecommunications, ICETE 2020, Online Event, July 8-10, 2020, Revised Selected Papers*, ser. Communications in Computer and Information Science, M. S. Obaidat and J. Ben-Othman, Eds., vol. 1484. Springer, 2020, pp. 151–174. [Online]. Available: https://doi.org/10.1007/978-3-030-90428-9_7

[15] J. Tillich, "The Decoding Failure Probability of MDPC Codes," in *2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018*. IEEE, 2018, pp. 941–945. [Online]. Available: https://doi.org/10.1109/ISIT.2018.8437843

[16] P. Santini, M. Battaglioni, M. Baldi, and F. Chiaraluce, "Analysis of the Error Correction Capability of LDPC and MDPC Codes Under Parallel Bit-Flipping Decoding and Application to Cryptography," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4648–4660, 2020. [Online]. Available: https://doi.org/10.1109/TCOMM.2020.2987898

[17] J. Chaulet, "Etude de cryptosystèmes à clé publique basés sur les codes MDPC quasi-cycliques. (study of public key cryptosystems based on quasi-cyclic MDPC codes)," Ph.D. dissertation, Pierre and Marie Curie University, Paris, France, 2017. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01599347

[18] V. Vasseur, "Post-quantum cryptography: a study of the decoding of QC-MDPC codes. (Cryptographie post-quantique : étude du décodage des codes QC-MDPC)," Ph.D. dissertation, University of Paris, France, 2021. [Online]. Available: https://tel.archives-ouvertes.fr/tel-03254461

[19] ——, "QC-MDPC codes DFR and the IND-CCA security of BIKE," *IACR Cryptol. ePrint Arch.*, p. 1458, 2021. [Online]. Available: https://eprint.iacr.org/2021/1458

[20] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini, "Performance Bounds for QC-MDPC Codes Decoders," in *Code-Based Cryptography - 9th International Workshop, CBCrypto 2021, Munich, Germany, June 21-22, 2021 Revised Selected Papers*, ser. Lecture Notes in Computer Science, A. Wachter-Zeh, H. Bartz, and G. Liva, Eds., vol. 13150. Springer, 2021, pp. 95–122. [Online]. Available: https://doi.org/10.1007/978-3-030-98365-9_6

[21] S. Arpin, T. R. Billingsley, D. R. Hast, J. B. Lau, R. A. Perlner, and A. Robinson, "A Study of Error Floor Behavior in QC-MDPC Codes," in *Post-Quantum Cryptography - 13th International Workshop, PQCrypto 2022, Virtual Event, September 28-30, 2022, Proceedings*, ser. Lecture Notes in Computer Science, J. H. Cheon and T. Johansson, Eds., vol. 13512. Springer, 2022, pp. 89–103. [Online]. Available: https://doi.org/10.1007/978-3-031-17234-2_5

[22] A. Annechini, A. Barenghi, and G. Pelosi, "C implementation of the 2-iteration dfr estimate and accessory numerical results," https://crypto.deib.polimi.it/DFR_codebase.zip, 2024.

[23] ——, "Bit-flipping Decoder Failure Rate Estimation for (v,w)-regular Codes," https://arxiv.org/abs/2401.16919, 2024.