# The decoding failure probability of MDPC codes

Jean-Pierre Tillich

# The decoding failure probability of MDPC codes

Jean-Pierre Tillich *

January 11, 2018

**Abstract**

Moderate Density Parity Check (MDPC) codes are defined here as codes which have a parity-check matrix whose row weight is of order the square root of the length $n$ of the code. They can be decoded like LDPC codes but they decode much less errors than LDPC codes: the number of errors they can decode in this case is of order the square root of $n$. Despite this fact they have been proved very useful in cryptography for devising key exchange mechanisms [BGG$^+$17]. They have also been proposed in McEliece type cryptosystems. However in this case, the parameters that have been proposed in [MTSB13] were broken in [GJS16]. This attack exploits the fact that the decoding failure probability is non-negligible. We show here that this attack can be thwarted by choosing the parameters in a more conservative way. We first show that such codes can decode with a simple bit-flipping decoder any pattern of $O\left(\frac{\sqrt{n}\log\log n}{\log n}\right)$ errors. This avoids the previous attack at the cost of significantly increasing the key size of the scheme. We then show that under a very reasonable assumptions the error probability after decoding decays almost exponentially with the codelength with just two iterations of bit-flipping. With an additional assumption it has even been proved that it decays exponentially with an unbounded number of iterations and show that in this case the increase of the key size which is required for resisting to the [GJS16] attack is only moderate.

## 1 Introduction

Virtually all the public key cryptography used in practice today can be attacked in polynomial time by a quantum computer [Sho94]. Even if such a quantum computer does not exist yet, finding viable solutions which would be resistant to a quantum computer is expected to be a lengthy process. This is one of the reasons why the NIST has recently launched a process for standardizing public key cryptographic algorithms that would be safe against a quantum adversary. Code-based cryptography is believed to be quantum resistant and is therefore considered as a viable solution. The McEliece system [McE78] based on binary Goppa codes, which is almost as old as RSA, is a public key cryptosystem that falls into this category. It has withstood all cryptanalyses up to now. It is well known to provide extremely fast encryption and fast decryption [BS08, BCS13], but has large public keys, about 200 kilobytes for 128 bits of security and slightly less than one megabyte for 256 bits of security [BLP08].

There have been many attempts to decrease the key size of this system. They are either based on codes with a better error correction capacity such as generalized Reed-Solomon codes [Nie86], algebraic geometry codes [JM96], a certain kind of non binary Goppa codes (called

*Inria, SECRET Project, 2 Rue Simone Iff 75012 Paris Cedex, France, Email: {jean-pierre.tillich}@inria.fr. Part of this work was supported by the Commission of the European Communities through the Horizon 2020 program under project number 645622 PQCRYPTO.

wild Goppa codes or wild Goppa codes incognito) [BLP10, BLP11], convolutional codes [LJ12, GSJB14] or polar codes [SK14] or on more structured codes, such as codes with a non trivial automorphism group; for instance quasi-cyclic codes [Gab05, BC07, BBC08, BCGO09, MTSB13] or quasi–dyadic and quasi–monoidic Goppa codes [MB09, BLM11]. Using codes with better error correction capacity reduces the key size because of the following phenomenon : the size of the public key in a McEliece cryptosystem is generally the size needed to store a systematic generator matrix of the code used in it, that is $R(1-R)n^2 \log_2 q$ bits for a code of rate $R$ and length $n$ over $\mathbb{F}_q$. The attacks using generic decoding techniques [Ste88, BJMM12] have an exponential complexity which is of the form $e^{\alpha(q,R)t}$ where $t$ is the number of errors that the code can correct. The security of the cryptosystem is generally measured against this attack and a better error-correction capacity implies a better dependency of the key size in terms of the complexity of the generic decoding attack. With the second method, the key size is decreased directly because the public generator matrix is a quasi–cyclic code for instance. When the circulant blocks that form the generator matrix of the corresponding code are of size $p$ this allows to decrease the key size by a multiplicative factor $p$, whereas the best decoding on quasi-cyclic codes have roughly the same complexity as the best generic decoding on unstructured codes.

However, it has turned out that most of the aforementioned schemes allowed key recovery attacks that could not be mounted on the original Goppa codes [SS92, Wie10, MS07, FM08, FOPT10, OTD10, CGG$^+$14, LT13, CMCP14, COT14, FPdP14, COTG15, BCD$^+$16]. But some of them remain unbroken by direct key attacks, namely those relying on Moderate Density Parity Check (MDPC) codes [MTSB13] and their cousins [BBC08], the original binary Goppa codes of [McE78] and their non-binary variants as proposed in [BLP10, BLP11]. The family of Moderate Density Parity Check codes (MDPC) codes is particularly interesting since (i) the decryption algorithm is extremely simple and is based on a extremely simple bit flipping decoding algorithm, (ii) direct attacks on key directly really amount to a problem of the same nature as decoding a linear code. This can be used to give a security proof [MTSB13]. However this security proof does not take into account the decoding failure probability. This is not not necessarily a problem in a setting where the scheme is used to devise ephemeral keys [BGG$^+$17, ABB$^+$17]. However, in security models where an attacker is allowed to query the decryption oracle many times, this can be a problem as observed by [GJS16] which showed how to attack the parameters proposed in [MTSB13]. This attack really exploits the non negligible decoding failure probability of the MDPC codes chosen in [MTSB13]. If this probability were as low as $2^{-\lambda}$ where $2^{\lambda}$ is the complexity of the best attack that the scheme has to sustain, then this would not be a problem and the security proof of [BGG$^+$17] could be used to show the security of the scheme under this stronger attacking model. This raises the issue whether or not the error probability of MDPC codes can be made arbitrarily small.

We tackle this issue by giving several different answers to this issue. We study in depth this question in the regime which is particularly interesting for these cryptographic applications, namely when the weight of the parity-check equations is of order $O(\sqrt{n})$ where $n$ is the length of the MDPC code. We define in the whole article MDPC codes in this way

**Definition 1** (MDPC code)**.** *Let $\alpha$ be a positive real number. An $\alpha$ MDPC code is a binary linear code that admits a parity check matrix whose rows are all of weight $\leq \alpha\sqrt{n}$. When we do not specify $\alpha$, we implicitly assume that $\alpha = 1$. In the case where this parity-check matrix have rows of a same weight $w$ and columns of a same weight $v$, we say that the parity-check matrix is of type $(v, w)$. By some abuse of terminology, we will also call the corresponding code a code of type $(v, w)$.*

We will decode these codes with a simple bit-flipping decoding algorithm. One round of decoding is just majority-logic decoding based on a sparse parity-check matrix of the code.

When we perform just one round of bit-flipping we call this decoder a majority-logic decoder. Recall that a majority logic decoder based on a certain parity-check matrix computes for all bits the number $u_i$ of parity-checks that involve the bit $i$ that are are unsatisfied. Let $n_i$ be the number of parity-checks involving bit $i$. If for a bit $i$ we have $u_i > n_i/2$ (i.e if a strict majority of such parity-checks is violated) the bit gets flipped. We will assume here that the computation of the $v_i$'s so that flipping one bit does not affect the other $v_j$'s. In other words the decoder works as follows when we perform $t$ iterations Usually, majority-logic decoding is performed by taking

**Algorithm 1** Bit-flipping decoder

**for all** $i \in \{1, \ldots, n\}$ **do**
  $n_i \leftarrow \#\{j \in \{1, \ldots, r\} : h_{ji} = 1\}$
**for** $a = 1$ to $t$ **do**
  **for all** $i \in \{1, \ldots, n\}$ **do**
    $u_i \leftarrow \#\{j \in \{1, \ldots, r\} : h_{ji} = 1,\ \sum_\ell h_{j\ell} y_\ell = 1 \pmod 2\}$
  **for all** $i \in \{1, \ldots, n\}$ **do**
    **if** $u_i > n_i/2$ **then**
      $y_i \leftarrow 1 - y_i$

for each bit $i$ a subset of the parity-checks that involve it that do not intersect each other in other positions and perform majority-voting based on this subset. The analysis of the majority-logic decoder is based on this assumption. We proceed differently here by taking into account all parity-checks that involve a given bit. A crucial quantity will play an important role, namely

**Definition 2** (maximum column intersection)**.** *Let* $\mathbf{H} = (h_{ij})_{\substack{1 \le i \le r \\ 1 \le j \le n}}$ *be a binary matrix. The intersection number of two different columns* $j$ *and* $j'$ *of* $\mathbf{H}$ *is equal to the number of rows* $i$ *for which* $h_{ij} = h_{ij'} = 1$*. The* maximum column intersection *of* $\mathbf{H}$ *is equal to the maximum intersection number of two different columns of* $\mathbf{H}$*.*

The point is that it is readily verified (see Subsection 2.1) that

**Proposition 1.** *Consider a code with a parity check matrix for which every column has weight at least* $v$ *and whose maximum column intersection is* $s$*. Performing majority-logic decoding based on this matrix (i.e. Algorithm 1 with* $t = 1$*) corrects all errors of weight* $\le \lfloor \frac{v}{2s} \rfloor$*.*

The point is that for MDPC codes the maximum column intersection is really small. We namely prove that for a natural random MDPC code model, the maximum intersection number of the parity-check matrix defining the MDPC code is typically of order $\Omega\left(\frac{\log n}{\log \log n}\right)$. Computing the maximum intersection number can obviously be performed in polynomial time and this allows us to give a randomized polynomial time algorithm for constructing MDPC codes of length $n$ and fixed rate $R \in [0, 1)$ that correct any pattern of $\Omega\left(\frac{\sqrt{n} \log \log n}{\log n}\right)$ errors with the majority-logic decoder decoder.

For these codes we prove here that

i. A code that admits a parity-check matrix of type $(v, w)$ and maximum column intersection $s$ corrects at least errors with the majority-logic decoder based on this parity-check matrix.

ii There is a randomized polynomial time algorithm for constructing MDPC codes of length $n$ and fixed rate $R \in [0, 1)$ that correct any pattern of $\Omega\left(\frac{\sqrt{n} \log \log n}{\log n}\right)$ errors with the majority-logic decoder decoder.

iii. Under a reasonable assumption on the first round of the bit-flipping decoder, the same MDPC codes correct $\Omega(\sqrt{n})$ errors with two iterations of a bit-flipping decoder with decoding failure probability of order $e^{-\Omega\left(\frac{n \log\log n}{\log n}\right)}$.

iv. Under an additional assumption .

It should be noted that under an additional assumption on the subsequent iterations of the bit-flipping decoder, it has been proved in [ABB+17] that MDPC codes correct $\Omega(\sqrt{n})$ errors by performing an unbounded number of bit-flipping iterations with probability of error $e^{-\Omega(n)}$. We also provide some concrete numbers to show that it is possible to construct MDPC codes that avoid completely the [GJS16] attack and for which it is possible to provide a security proof in strong security models with a significant key size overhead when compared to the parameters proposed in [MTSB13] if we want to stay in the no-error scenario, with a reasonable overhead if we make the assumption of the point [iii.] above, and very moderate overhead if we make the assumptions of [iii.] and [iv.].

**Notation.** We denote by $h(x)$ the entropy (in nats) of a Bernoulli random variable of parameter $x$, that is $h(x) \stackrel{\text{def}}{=} -x \ln x - (1-x)\ln(1-x)$.

# 2 Majority-logic decoding and its performance for MDPC codes

We start this section by proving Proposition 1, then show that for typical MDPC codes the intersection number is small and that this allows to construct efficiently MDPC codes that correct all patterns of $O\left(\frac{\sqrt{n}\log\log n}{\log n}\right)$ errors.

## 2.1 Proof of Proposition 1

Let us first recall this proposition.

**Proposition 1.** *Consider a code with a parity check matrix for which every column has weight at least $v$ and whose maximum column intersection is $s$. Performing majority-logic decoding based on this matrix (i.e. Algorithm 1 with $t = 1$) corrects all errors of weight $\leq \lfloor \frac{v}{2s} \rfloor$.*

*Proof.* We denote by $\mathbf{H} = (h_{ij})_{\substack{1 \leq i \leq r \\ 1 \leq j \leq n}}$ the parity-check matrix we use for performing majority-logic decoding. For $i$ in $\{1, \ldots, r\}$ denote by $E_i$ the subset of positions $\ell$ which are in error and in the support of the $i$-th parity check equation (i.e. $h_{i\ell} = 1$). We number the parity-check equations of the code from 1 to $r$. We consider now what happens to $y_j$ in the algorithm. There are two cases to consider.

**Case 1:** $y_j$ is erroneous. We can upper-bound the number $s_j$ of satisfied parity-check equations involving this bit by the number of parity-check equations involving this bit whose support contains at least 2 errors. We consider now the graph $G_j$ which is a bipartite graph associated to $j$ which is constructed as follows. Its set of vertices is the union of the set $A_j$ of positions different from $j$ which are in error and the set $B_j$ of parity-check equations that involve the position $j$ and whose support contains at least 2 errors. There is an edge between a position $\ell$ in $A_j$ and parity-check equation $i$ in $B_j$ if and only if the parity-check equation involves $\ell$, that is $h_{i\ell} = 1$. Let $e_j$ be the number of edges of $G_j$ and let $n_j =$ be the number of parity-check

equations involving $j$. We observe now that

$$\begin{aligned} s_j &\leq \#\{i : h_{ij} = 1, |E_i| \geq 2\} && (1)\\ &\leq e_j \\ &\leq s\#A_j && (2)\\ &\leq s(t-1) \\ &\leq s\left(\lfloor \frac{v}{2s} \rfloor - 1\right) \\ &< v/2. \end{aligned}$$

(1) is just the first observation whereas (2) follows from the fact the degree in $G_j$ of any vertex is at most $s$ by the assumption on the maximum intersection number of $\mathbf{H}$. Since $v/2 \leq n_j/2$ it follows that the majority-logic decoder necessarily flips the bit and therefore corrects the corresponding error.

**Case 2:** there is no error in position $j$. We can upper-bound the number $u_j$ of unsatisfied positions in a similar way. This time we consider the graph $G'_j$ whose vertex set is the union of $A'_j$ which is the set of positions which are in error and $B'_j$ the set of parity-check equations involving $j$ and whose support contains this time at least one error. We put an edge between a position $\ell$ in $A'_j$ and parity-check equation $i$ in $B'_j$ if and only if the parity-check equation involves $\ell$. Let $e'_j$ be the number of edges of $G'_j$. Similarly to what we did we observe now that

$$\begin{aligned} u_j &\leq \#\{i : h_{ij} = 1, |E_i| \geq 1\} \\ &\leq e'_j \\ &\leq s\#A_j \\ &\leq st \\ &\leq s\left(\lfloor \frac{v}{2s} \rfloor\right) && (3)\\ &\leq v/2 \\ &\leq n_j/2. \end{aligned}$$

In other words we will not flip this bit. □

## 2.2 A random model for MDPC codes of type $(v, w)$

There are several ways to build random MDPC codes of type $(v, w)$. The one which is used in cryptography [BBC08, MTSB13, DGZ17, BGG+17, ABB+17, BBC+17] is to construct them as quasi-cyclic codes. Our proof technique can also be applied to this case, but since there are several different types of construction of this kind, so that we have to adapt our proof technique to each of those, we will consider a more general model here. It is based on Gallager's construction of LDPC codes [Gal63]. We will construct an $r \times n$ random parity-check matrix of type $(v, w)$ by assuming that $n$ is a multiple of $w$ ($n = n'w$), $r$ is a multiple of $v$ ($r = r'v$) and that $rw = nv$ (this condition is necessary in order to obtain a matrix of type $(v, w)$). Let $\mathbf{P}_{n,w}$ be a matrix of size $n' \times n$ constructed as follows

$$\mathbf{P}_{n,w} = \mathbf{I}_{n'} \otimes \mathbf{1}_w = \begin{pmatrix} \mathbf{1}_w & \mathbf{0}_w & \dots & \dots & \mathbf{0}_w \\ \mathbf{0}_w & \mathbf{1}_w & \mathbf{0}_w & \cdots & \mathbf{0}_w \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & \mathbf{0}_w \\ \mathbf{0}_w & \dots & \dots & \mathbf{0}_w & \mathbf{1}_w \end{pmatrix}.$$

where $\mathbf{I}_{n'}$ denotes the identity matrix of size $n'$, $\mathbf{1}_w$ a row vector of length $w$ whose entries are all equal to 1, that is $\mathbf{1}_w = \underbrace{(1 \dots 1)}_{w \text{ times}}$, $\mathbf{0}_w$ a row vector of length $w$ whose entries are all equal to 0. We then choose $v$ permutations of length $n$ at random and they define a parity-check matrix $\mathbf{H}(\pi_1, \dots, \pi_v)$ of size $r \times n$ of type $(v, w)$ as

$$\mathbf{H}(\pi_1, \dots, \pi_v) = \begin{pmatrix} \mathbf{P}_{n,w}^{\pi_1} \\ \mathbf{P}_{n,w}^{\pi_2} \\ \dots \\ \mathbf{P}_{n,w}^{\pi_v} \end{pmatrix},$$

where $\mathbf{P}_{n,w}^{\pi_i}$ denotes the matrix $\mathbf{P}_{n,w}$ whose columns have been permuted with $\pi_i$. We denote by $\mathcal{D}_{r,n,v,w}$ the associated probability distribution of binary matrices of size $r \times n$ and type $(v, w)$ we obtain when the $\pi_i$'s are chosen uniformly at random.

## 2.3 The maximum intersection number of matrices drawn according to $\mathcal{D}_{r,n,v,w}$

The maximum intersection number of matrices drawn according to $\mathcal{D}_{r,n,v,w}$ turns out to be remarkably small when $w$ and $v$ are of order $\sqrt{n}$, it is namely typically of order $O\left(\frac{\log n}{\log \log n}\right)$. To prove this claim we first observe that

**Lemma 1.** *Consider a matrix $\mathbf{H}$ drawn at random according to the distribution $\mathcal{D}_{r,n,v,w}$. Take two arbitrary columns $j$ and $j'$ of $\mathbf{H}$ and let $n_{jj'}$ be the intersection number of $j$ and $j'$. We have for all $t \in \{0, \dots, v\}$*

$$\mathbb{P}(n_{jj'} = t) = \binom{v}{t} \left(\frac{w-1}{n-1}\right)^t \left(1 - \frac{w-1}{n-1}\right)^{v-t}.$$

*Proof.* Recall that $\mathbf{H} = (h_{ij})_{\substack{1 \le i \le r \\ 1 \le j \le n}}$ is of the form

$$\mathbf{H}(\pi_1, \dots, \pi_v) = \begin{pmatrix} \mathbf{P}_{n,w}^{\pi_1} \\ \mathbf{P}_{n,w}^{\pi_2} \\ \dots \\ \mathbf{P}_{n,w}^{\pi_v} \end{pmatrix},$$

for some permutations $\pi_1, \pi_2, \dots, \pi_v$ chosen uniformly at random in $S_n$. A row $i$ of $\mathbf{H}$ is called a coincidence if and only if $h_{ij} = h_{ij'}$. There is obviously one coincidence at most in each of the blocks $\mathbf{P}_{n,w}^{\pi_\ell}$. We claim now that the probability of a coincidence in each of these blocks is $\frac{w-1}{n-1}$. To verify this consider the row $i$ of block $\mathbf{P}_{n,w}^{\pi_\ell}$ which is such that $h_{ij} = 1$. The probability that there is a coincidence for this block is the probability that $h_{ij'} = 1$ which amounts to the fact that $\pi_\ell(j')$ takes its values in a subset of $w-1$ values among $n-1$ possible values. All of these $n-1$ are equiprobable. This shows the claim. Since the coincidences that occur in the blocks are all independent (since the $\pi_i$'s are independent) we obtain the aforementioned formula. □

We use this to prove the following result

**Proposition 2.** *Let $\alpha$ and $\beta$ be two constants such that $0 < \alpha < \beta$. Assume we draw a parity-check matrix $\mathbf{H}$ at random according to the distribution $\mathcal{D}_{r,n,v,w}$ where we assume that both $v$ and $w$ satisfy $\alpha\sqrt{n} \le v < w \le \beta\sqrt{n}$. Then for any $\varepsilon > 0$ the the maximum intersection number of $\mathbf{H}$ is smaller than $(2+\varepsilon)\frac{\log n}{\log \log n}$ with probability $1 - o(1)$ as $n$ tends to infinity.*

*Proof.* Let is number the columns of $\mathbf{H}$ from 1 to $n$. For $i$ and $j$ in $\{1,\ldots,n\}$ two different columns of $\mathbf{H}$ we denote by $E_{i,j,t}$ the event that the intersection number of $i$ and $j$ is $\geq t$. Let $E_t$ be the probability that the maximum intersection number is larger than or equal to $t$. By the union bound, and then Lemma 1 we obtain

$$\begin{aligned}
\mathbb{P}(E_t) &= \mathbb{P}\left(\bigcup_{1\leq i<j\leq n} E_{i,j,t}\right) \\
&\leq \sum_{1\leq i<j\leq n} \mathbb{P}(E_{i,j,t}) \\
&\leq n^2 \sum_{s=t}^{v} \binom{v}{s} \left(\frac{w-1}{n-1}\right)^s \left(1-\frac{w-1}{n-1}\right)^{v-s}
\end{aligned}$$

From this we deduce

$$\mathbb{P}(E_t) \leq n^2 \sum_{s=t}^{v} \frac{v^v}{s^s(v-s)^{v-s}} \left(\frac{w-1}{n-1}\right)^s \left(1-\frac{w-1}{n-1}\right)^{v-s}$$

where we use the well known upper-bound $\binom{v}{s} \leq e^{vh(s/v)} = \frac{v^v}{s^s(v-s)^{v-s}}$. This allows to write

$$\begin{aligned}
\mathbb{P}(E_t) &\leq n^2 \sum_{s=t}^{v} \frac{v^v}{s^s(v-s)^{v-s}} \left(\frac{w}{n}\right)^s \qquad (4) \\
&\leq n^2 \sum_{s=t}^{v} \frac{v^{v-s}}{(v-s)^{v-s}} \left(\frac{v\cdot w}{s\cdot n}\right)^s \\
&\leq n^2 \sum_{s=t}^{v} \left(1+\frac{s}{v-s}\right)^{v-s} \left(\frac{v\cdot w}{s\cdot n}\right)^s \\
&\leq n^2 \sum_{s=t}^{v} \left(\frac{e\cdot v\cdot w}{s\cdot n}\right)^s \\
&\leq n^2 \sum_{s=t}^{v} \left(\frac{e\beta^2}{s}\right)^s
\end{aligned}$$

Choose now $t \geq (2+\varepsilon)\frac{\ln n}{\ln\ln n}$ for some $\varepsilon > 0$. When $n$ is large enough, we have that $\frac{e\beta^2}{t} < 1$. In such a case we can write

$$\begin{aligned}
\mathbb{P}(E_t) &\leq n^2 \sum_{s=t}^{v} \left(\frac{e\beta^2}{t}\right)^s \\
&\leq \frac{n^2\left(\frac{e\beta^2}{t}\right)^t}{1-\frac{e\beta^2}{t}} \\
&\leq n^2 \left(\frac{K}{t}\right)^t
\end{aligned}$$

for some constant $K > 0$. This implies that

$$\begin{aligned}\mathbb{P}(E_t) &\leq n^2 e^{(2+\varepsilon)\frac{\ln n}{\ln\ln n}\ln\left(\frac{K\ln\ln n}{\gamma \ln n}\right)} \\ &\leq e^{\varepsilon \ln n + \frac{\gamma \ln n \ln\left(\frac{K\ln\ln n}{\gamma}\right)}{\ln\ln n}} \\ &= o(1)\end{aligned}$$

as $n$ tends to infinity. □

This together with Proposition 1 implies directly the following corollary

**Corollary 1.** *There exists a randomized algorithm working in expected polynomial time outputting for any designed rate $R \in (0,1)$ an MDPC code of rate $\geq R$ of an arbitrarily large length $n$ and parity-check equations of weight $\Theta(\sqrt{n})$ that corrects all patterns of errors of size less than $\gamma \frac{\sqrt{n}\ln\ln n}{\ln n}$ for $n$ large enough, where $\gamma > 0$ is some absolute constant.*

*Proof.* The randomized algorithm is very simple. We choose $n$ to be a square $n = w^2$ for some integer $w$ and let $v \stackrel{\text{def}}{=} \lfloor (1-R)w \rfloor$ and $r \stackrel{\text{def}}{=} \frac{nv}{w}$. Then we draw a parity-check matrix $\mathbf{H}$ at random according to the distribution $\mathcal{D}_{r,n,v,w}$. The corresponding code has clearly rate $\geq R$. We compute the maximum column intersection of $\mathbf{H}$. This can be done in time $O\left(wn^2\right)$. If this column intersection is greater than $(2+\varepsilon)\frac{\ln n}{\ln\ln n}$ we output the corresponding MDPC code, if not we draw at random $\mathbf{H}$ again until finding a suitable matrix $\mathbf{H}$. By Proposition 1 we know that such a code can correct all patterns of at most $\left\lfloor \frac{\alpha\sqrt{n}\ln\ln n}{(4+\varepsilon)\ln n}\right)\rfloor$ errors. This implies the corollary. □

# 3 Analysis of two iterations of bit-flipping

We derived in the previous section a condition ensuring that one round of bit-flipping corrects all the errors. We will now estimate the probability that performing one round of bit-flipping corrects enough errors so that another round of bit-flipping will correct all remaining errors. To analyze the first round of decoding we will model the bit-flipping algorithm by a binomial distribution. More precisely, consider an MDPC code of type $(v,w)$ and length $n$. The noise model is the following: an error of weight $t$ was chosen uniformly at random and added to the codeword of the MDPC code. For $i \in \{1,\ldots,n-t\}$, let $E_i^0$ be the Bernouilli random variable which is equal to 1 iff the $i$-th position that was not in error initially is in error after the first round of iterative decoding. We also denote by $U_i^0$ the counter $u_j$ associated to the $i$-th position which was not in error. $U_i^0$ is the sum $\sum_{j=1}^{v} V_{ij}^0$ of $v$ Bernouilli random variables $V_{ij}^0$ associated to the $v$ parity-check equations involving this bit. A Bernoulli-random variable $V_{ij}^0$ is equal to 1 if and only the corresponding parity-check is equal to 1. Note that by definition of the bit-flipping decoder

$$E_i^0 = \mathbf{1}_{\{U_i^0 > v/2\}}$$

Similarly, for $i \in \{1,\ldots,t\}$ we denote by $E_i^1$ the Bernoulli random variable that is equal to 1 iff the $i$-th bit that was in error initially stays in error after the first round of Algorithm 1. We also define the $U_i^1$'s and the $V_{ij}^1$'s similarly. In this case

$$E_i^1 = \mathbf{1}_{\{U_i^0 \leq v/2\}}.$$

Let us bring in for $b \in \{0,1\}$:

$$p_b \stackrel{\text{def}}{=} \mathbb{P}(V_{ij}^b = 1). \tag{5}$$

It is clear that these probabilities do not depend on $i$ and $j$ and that this definition is consistent. It is (essentially) proved in [ABB+17] that

**Lemma 2.** *Assume that $w = O(\sqrt{n})$ and $t = O(\sqrt{n})$. Then*

$$p_b = \frac{1}{2} - (-1)^b \varepsilon \left( \frac{1}{2} + O\left( \frac{1}{\sqrt{n}} \right) \right), \tag{6}$$

*where $\varepsilon \stackrel{def}{=} e^{-\frac{2wt}{n}}$.*

We will recall a proof of this statement in the appendix. We will now make the following assumption that simplifies the analysis

**Assumption 1.** *When we use Algorithm 1 on an MDPC code of type $(v, w)$, we assume that*

- *for all $i \in \{1, \ldots, n-t\}$ the counters $U_i^0$ of Algorithm 1 are distributed like sums of $v$ independent Bernoulli random variables of parameter $q_0$ at the first iteration and the $E_i^0$'s are independent;*
- *for all $i \in \{1, \ldots, t\}$ the counters $U_i^1$ of Algorithm 1 are distributed like sums of $v$ independent Bernoulli random variables of parameter $q_1$ at the first iteration and the $E_i^1$'s are independent.*

To analyze the behavior of Algorithm 1 we will use the following lemma which is just a slight generalization of Lemma 6 in [ABB+17]

**Lemma 3.** *Under Assumption 1 used for an MDPC code of type $(v, w)$ and when the error is chosen uniformly at random among the errors of weight $t$, we have for all $(b, i) \in \{0\} \times \{1, \ldots, n-t\} \cup \{1\} \times \{1, \ldots, t\}$,*

$$\mathbb{P}(E_i^b = 1) = O\left( \frac{(1-\varepsilon^2)^{v/2}}{\sqrt{v}\varepsilon} \right),$$

*where $\varepsilon \stackrel{def}{=} e^{-\frac{2wt}{n}}$.*

Because this quantity does not depend on $i$ we will therefore denote

$$q_b \stackrel{\text{def}}{=} \mathbb{P}(E_i^b = 1).$$

For the ease of reading the proof of this lemma is also recalled in the appendix. We let

$$\begin{aligned} S_0 &\stackrel{\text{def}}{=} E_1^0 + \cdots + E_{n-t}^0 \\ S_1 &\stackrel{\text{def}}{=} E_1^1 + \cdots + E_t^1 \end{aligned}$$

$S_0$ is the number of errors that were introduced after one round of iterative decoding coming from flipping the $n-t$ bits that were initially correct. Similarly $S_1$ is the number of errors that are left after one round of iterative decoding coming from not flipping the $t$ bits that were initially incorrect. Let $S \stackrel{\text{def}}{=} S_0 + S_1$, which represents the total number of errors that are left after the first round of iterative decoding. We quantify the probability that this quantity does not decay enough by the following Theorem which holds under Assumption 1.

**Theorem 1.** *Provided that Assumption 1 holds, we have for an MDPC code of type $(v, w)$ where $v = \Theta(\sqrt{n})$ and $w = \Theta(\sqrt{n})$:*

$$\mathbb{P}(S \geq t') \leq \frac{1}{\sqrt{t'}} e^{\frac{t'v}{4}\ln\left(1-\varepsilon^2\right) + \frac{t'}{8}\ln(n) + O\left(t'\ln(t'/t)\right)}.$$

From this theorem we deduce that

**Corollary 2.** *Provided that Assumption 1 holds, we can construct in expected polynomial time arbitrarily for any designed rate $R \in (0,1)$ an MDPC code of rate $\geq R$ of an arbitrarily large length $n$ and parity-check equations of weight $\Theta(\sqrt{n})$ large MDPC codes where the probability of error $P_e$ after two iterations of bit-flipping is upper-bounded by $e^{-\Omega\left(n\frac{\ln\ln n}{\ln n}\right)}$ when there are $t = \Theta(\sqrt{n})$ errors.*

*Proof.* We use the construction given in the proof of Corollary 1 to construct an MDPC code of type $(v, w)$ of length $n = w^2$ and with $v \stackrel{\text{def}}{=} \lfloor (1-R)w \rfloor$ that allows to correct all patterns of errors of size less than $\gamma \frac{\sqrt{n}\ln\ln n}{\ln n}$ for $n$ large enough, where $\gamma > 0$ is some absolute constant with just one round of the bit-flipping decoder of Algorithm 1. Then we use Theorem 1 to show that with probability upper-bounded by $e^{-\Omega\left(n\frac{\ln\ln n}{\ln n}\right)}$ there remains at most $\gamma \frac{\sqrt{n}\ln\ln n}{\ln n}$ errors after one round of Algorithm 1. This proves the corollary. □

In [ABB+17] there is an additional assumption which is made which is that the probability of error is dominated by the probability that the first round of decoding is not able to decrease the number by some mutiplicative factor $\alpha$. With the notation of thi section, this assumption can be described as follows.

**Assumption 2.** *There exists some constant $\alpha > 0$ such that the probability of error $P_{err}$ for an unbounded number of iterations of Algorithm 1 is upper-bounded by $\mathbb{P}(S \geq \alpha t)$ where $S$ is the number of errors that are left after the first round of Algorithm 1 and $t$ is the initial number of errors.*

With this additional assumption (Assumption 1 is actually also made) it is proven in [ABB+17] that the probability of errors decays exponentially when $t = \Theta(\sqrt{n})$. This is actually obtained by a slightly less general version of Theorem 1 (see [ABB+17, Theorem 1]).

# 4 Choosing the length in order to have a negligible probability of error

# A Proof of Lemma 3

Let us first recall this lemma.

**Lemma 3.** *Under Assumption 1 used for an MDPC code of type $(v, w)$ and when the error is chosen uniformly at random among the errors of weight $t$, we have for all $(b, i) \in \{0\} \times \{1, \ldots, n-t\} \cup \{1\} \times \{1, \ldots, t\}$,*

$$\mathbb{P}(E_i^b = 1) = O\left(\frac{(1-\varepsilon^2)^{v/2}}{\sqrt{v}\varepsilon}\right),$$

*where $\varepsilon \stackrel{def}{=} e^{-\frac{2wt}{n}}$.*

To prove this lemma we will use

# B The Kullback-Leibler divergence

The proofs of the results proved in the appendix use the Kullback-Leibler divergence (see see for instance [CT91]) and some of its properties what we now recall.

**Definition 3. Kullback-Leibler divergence**
*Consider two discrete probability distributions* $\mathbf{p}$ *and* $\mathbf{q}$ *defined over a same discrete space* $\mathcal{X}$. *The Kullback-Leibler divergence between* $\mathbf{p}$ *and* $\mathbf{q}$ *is defined by*

$$D(\mathbf{p}\|\mathbf{q}) = \sum_{x\in\mathcal{X}} p(x)\ln\frac{p(x)}{q(x)}.$$

*We overload this notation by defining for two Bernoulli distributions* $\mathcal{B}(p)$ *and* $\mathcal{B}(q)$ *of respective parameters* $p$ *and* $q$

$$D(p\|q) \stackrel{def}{=} D(\mathcal{B}(p)\|\mathcal{B}(q)) = p\ln\left(\frac{p}{q}\right) + (1-p)\ln\left(\frac{1-p}{1-q}\right).$$

*We use the convention (based on continuity arguments) that* $0\ln\frac{0}{p} = 0$ *and* $p\ln\frac{p}{0} = \infty$.

We will need the following approximations/results of the Kullback-Leibler divergence

**Lemma 4.** *For any* $\delta \in (-1/2, 1/2)$ *we have*

$$D\left(\frac{1}{2}\middle\|\frac{1}{2}+\delta\right) = -\frac{1}{2}\ln(1-4\delta^2). \tag{7}$$

*For constant* $\alpha \in (0,1)$ *and* $\delta$ *going to* $0$ *by staying positive, we have*

$$D(\alpha\|\delta) = -h(\alpha) - \alpha\ln\delta + O(\delta). \tag{8}$$

*For* $0 < y < x$ *and* $x$ *going to* $0$ *we have*

$$D(x\|y) = x\ln\frac{x}{y} + x - y + O\left(x^2\right). \tag{9}$$

*Proof.* Let us first prove (7).

$$\begin{aligned} D\left(\frac{1}{2}\middle\|\frac{1}{2}+\delta\right) &= \frac{1}{2}\ln\frac{1/2}{1/2+\delta} + \frac{1}{2}\ln\frac{1/2}{1/2-\delta} \\ \mathbb{P} &= -\frac{1}{2}\ln(1+2\delta) - \frac{1}{2}\ln(1-2\delta) \\ &= -\frac{1}{2}\ln(1-4\delta^2). \end{aligned}$$

To prove (8) we observe that

$$\begin{aligned} D(\alpha\|\delta) &= \alpha\ln\left(\frac{\alpha}{\delta}\right) + (1-\alpha)\ln\left(\frac{1-\alpha}{1-\delta}\right) \\ &= -h(\alpha) - \alpha\ln\delta - (1-\alpha)\ln(1-\delta) \\ &= -h(\alpha) - \alpha\ln\delta + O(\delta). \end{aligned}$$

For the last estimate we proceed as follows

$$\begin{aligned} D(x\|y) &= x\ln\frac{x}{y}+(1-x)\ln\frac{1-x}{1-y} \\ &= x\ln\frac{x}{y}-(1-x)\left(-x+y+O\left(x^2\right)\right) \\ &= x\ln\frac{x}{y}+x-y+O\left(x^2\right). \end{aligned}$$

□

The Kullback-Leibler appears in the computation of large deviation exponents. In our case, we will use the following estimate which is well known and which can be found for instance in [BGT11]

**Lemma 5.** *Let $p$ be a real number in $(0,1)$ and $X_1,\dots X_n$ be $n$ independent Bernoulli random variables of parameter $p$. Then, as $n$ tends to infinity:*

$$\mathbb{P}(X_1+\dots X_n\geq\tau n) = \frac{(1-p)\sqrt{\tau}}{(\tau-p)\sqrt{2\pi n(1-\tau)}}e^{-nD(\tau\|p)}(1+o(1)) \text{ for } p<\tau<1, \quad (10)$$

$$\mathbb{P}(X_1+\dots X_n\leq\tau n) = \frac{p\sqrt{1-\tau}}{(p-\tau)\sqrt{2\pi n\tau}}e^{-nD(\tau\|p)}(1+o(1)) \text{ for } 0<\tau<p. \quad (11)$$

# C Proof of Lemma 2

Recall first this lemma.

**Lemma 2.** *Assume that $w=O(\sqrt{n})$ and $t=O(\sqrt{n})$. Then*

$$p_b=\frac{1}{2}-(-1)^b\varepsilon\left(\frac{1}{2}+O\left(\frac{1}{\sqrt{n}}\right)\right), \quad (6)$$

*where $\varepsilon\stackrel{def}{=}e^{-\frac{2wt}{n}}$.*

Before giving the proof of this lemma, observe $\mathbb{P}(V_{ij}^b=1)$ can be viewed as the probability that the $j$-th parity check equation involving a bit $i$ gives an incorrect information about bit $i$. This is obtained through the following lemma.

**Lemma 6.** *Consider a word $\mathbf{h}\in\mathbb{F}_2^n$ of weight $w$ and an error $\mathbf{e}\in\mathbb{F}_2^n$ of weight $t$ chosen uniformly at random. Assume that both $w$ and $t$ are of order $\sqrt{n}$: $w=O(\sqrt{n})$ and $t=O(\sqrt{n})$. We have*

$$\mathbb{P}_{\mathbf{e}}(\langle\mathbf{h},\mathbf{e}\rangle=1)=\frac{1}{2}-\frac{1}{2}e^{-\frac{2wt}{n}}\left(1+O\left(\frac{1}{\sqrt{n}}\right)\right).$$

**Remark 1.** *Note that this probability is in this case of the same order as the probability taken over errors $\mathbf{e}$ whose coordinates are drawn independently from a Bernoulli distribution of parameter $t/n$. In such a case, from the piling-up lemma [Mat93] we have*

$$\begin{aligned} \mathbb{P}_{\mathbf{e}}(\langle\mathbf{h},\mathbf{e}\rangle=1) &= \frac{1-\left(1-\frac{2t}{n}\right)^w}{2} \\ &= \frac{1}{2}-\frac{1}{2}e^{w\ln(1-2t/n)} \\ &= \frac{1}{2}-\frac{1}{2}e^{-\frac{2wt}{n}}\left(1+O\left(\frac{1}{\sqrt{n}}\right)\right). \end{aligned}$$

The proof of this lemma will be done in the following subsection. Lemma 2 is a corollary of this lemma since we have

$$p_b = \mathbb{P}(\langle \mathbf{h}, \mathbf{e} \rangle = 1 | e_1 = b). \tag{12}$$

## C.1 Proof of Lemma 6

The proof involves properties of the Krawtchouk polynomials. We recall that the (binary) Krawtchouk polynomial of degree $i$ and order $n$ (which is an integer), $P_i^n(X)$ is defined for $i \in \{0, \cdots, n\}$ by:

$$P_i^n(X) \stackrel{\text{def}}{=} \frac{(-1)^i}{2^i} \sum_{j=0}^{i} (-1)^j \binom{X}{j} \binom{n-X}{i-j} \quad \text{where } \binom{X}{j} \stackrel{\text{def}}{=} \frac{1}{j!} X(X-1)\cdots(X-j+1). \tag{13}$$

Notice that it follows on the spot from the definition of a Krawtchouk polynomial that

$$P_k^n(0) = \frac{(-1)^k \binom{n}{k}}{2^k}. \tag{14}$$

Let us define the bias $\delta$ by

$$\delta \stackrel{\text{def}}{=} 1 - 2\mathbb{P}_{\mathbf{e}}(\langle \mathbf{h}, \mathbf{e} \rangle = 1).$$

In other words $\mathbb{P}_{\mathbf{e}}(\langle \mathbf{h}, \mathbf{e} \rangle = 1) = \frac{1}{2}(1-\delta)$. These Krawtchouk polynomials are readily related to $\delta$. We first observe that

$$\mathbb{P}_{\mathbf{e}}(\langle \mathbf{h}, \mathbf{e} \rangle = 1) = \frac{\sum_{\substack{j=1 \\ j \text{ odd}}}^{w} \binom{t}{j}\binom{n-t}{w-j}}{\binom{n}{w}}.$$

Moreover by observing that $\sum_{j=0}^{w} \binom{t}{j}\binom{n-t}{w-j} = \binom{n}{w}$ we can recast the following evaluation of a Krawtchouk polynomial as

$$\begin{aligned}
\frac{(-2)^w}{\binom{n}{w}} P_w^n(t) &= \frac{\sum_{j=0}^{w} (-1)^j \binom{t}{j}\binom{n-t}{w-j}}{\binom{n}{w}} \\
&= \frac{\sum_{\substack{j=0 \\ j \text{ even}}}^{w} \binom{t}{j}\binom{n-t}{w-j} - \sum_{\substack{j=1 \\ j \text{ odd}}}^{w} \binom{t}{j}\binom{n-t}{w-j}}{\binom{n}{w}} \\
&= \frac{\binom{n}{w} - 2\sum_{\substack{j=1 \\ j \text{ odd}}}^{w} \binom{t}{j}\binom{n-t}{w-j}}{\binom{n}{w}} \\
&= 1 - 2\mathbb{P}_{\mathbf{e}}(\langle \mathbf{h}, \mathbf{e} \rangle = 1) \\
&= \delta.
\end{aligned} \tag{15}$$

To simplify notation we will drop the superscript $n$ in the Krawtchouk polynomial notation. It will be chosen as the length of the MDPC code when will use it in our case. An important lemma that we will need is the following one.

**Lemma 7.** *For all $x$ in $\{1, \ldots, t\}$, we have*

$$\frac{P_w(x)}{P_w(x-1)} = \left(1 + O\left(\frac{1}{n}\right)\right) \frac{n - 2w + \sqrt{(n-2w)^2 - 4w(n-w)}}{2(n-w)}.$$

*Proof.* This follows essentially from arguments taken in the proof of [MS86][Lemma 36, §7, Ch. 17]. The result we use appears however more explicitly in [KL95][Sec. IV] where it is proved that if $x$ is in an interval of the form $\left[0, (1-\alpha)\left(n/2 - \sqrt{w(n-w)}\right)\right]$ for some constant $\alpha \in [0,1)$ independent of $x$, $n$ and $w$, then

$$\frac{P_w(x+1)}{P_w(x)} = \left(1 + O\left(\frac{1}{n}\right)\right) \frac{n - 2w + \sqrt{(n-2w)^2 - 4w(n-w)}}{2(n-w)}.$$

For our choice of $t$ this condition is met for $x$ and the lemma follows immediately. □

We are ready now to prove Lemma 6.

*Proof of Lemma 6.* We start the proof by using (15) which says that

$$\delta = \frac{(-2)^w}{\binom{n}{w}} P_w^n(t).$$

We then observe that

$$\begin{aligned}
\frac{(-2)^w}{\binom{n}{w}} P_w^n(t) &= \frac{(-2)^w}{\binom{n}{w}} \frac{P_w^n(t)}{P_w^n(t-1)} \frac{P_w^n(t-1)}{P_w^n(t-2)} \cdots \frac{P_w^n(1)}{P_w^n(0)} P_w^n(0) \\
&= \frac{(-2)^w}{\binom{n}{w}} \left(\left(1 + O\left(\frac{1}{n}\right)\right) \frac{n - 2w + \sqrt{(n-2w)^2 - 4w(n-w)}}{2(n-w)}\right)^t P_w^n(0) \text{ (by Lemma 7)} \\
&= \left(1 + O\left(\frac{1}{n}\right)\right)^t \left(\frac{n - 2w + \sqrt{(n-2w)^2 - 4w(n-w)}}{2(n-w)}\right)^t \text{ (by (14))} \\
&= e^{t \ln\left(\frac{1-2\omega+\sqrt{(1-2\omega)^2-4\omega(1-\omega)}}{2(1-\omega)}\right)} \left(1 + O\left(\frac{t}{n}\right)\right) \text{ where } \omega \stackrel{\text{def}}{=} \frac{w}{n} \\
&= e^{t \ln\left(\frac{1-2\omega+1-4\omega+O\left(\omega^2\right)}{2(1-\omega)}\right)} \left(1 + O\left(\frac{t}{n}\right)\right) \\
&= e^{t \ln\left(\frac{1-3\omega+O\left(\omega^2\right)}{1-\omega}\right)} \left(1 + O\left(\frac{t}{n}\right)\right) \\
&= e^{-2t\omega+O\left(\frac{tw^2}{n^2}\right)} \left(1 + O\left(\frac{t}{n}\right)\right) \\
&= e^{-\frac{2wt}{n}} \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right),
\end{aligned}$$

where we used at the last equation that $t = O(\sqrt{n})$ and $w = O(\sqrt{n})$. □

# D Proof of Lemma 3

Let us first recall this lemma.

**Lemma 3.** *Under Assumption 1 used for an MDPC code of type $(v, w)$ and when the error is chosen uniformly at random among the errors of weight $t$, we have for all $(b, i) \in \{0\} \times \{1, \ldots, n-t\} \cup \{1\} \times \{1, \ldots, t\}$,*

$$\mathbb{P}(E_i^b = 1) = O\left(\frac{(1-\varepsilon^2)^{v/2}}{\sqrt{v}\varepsilon}\right),$$

*where* $\varepsilon \stackrel{def}{=} e^{-\frac{2wt}{n}}$.

*Proof.* For $(b,i) \in \{0\} \times \{1,\ldots,n-t\} \cup \{1\} \times \{1,\ldots,t\}$, let $X_i$ be independent Bernoulli random variables of parameter $p_b$. From Assumption 1 we have

$$\begin{aligned}\mathbb{P}(E_i^0 = 1) = q_0 &= \mathbb{P}(\sum_{i=1}^{v} X_i^0 > v/2)\\ \mathbb{P}(E_i^1 = 1) = q_1 &= \mathbb{P}(\sum_{i=1}^{v} X_i^1 \leq v/2).\end{aligned}$$

By using Lemma 5 we obtain for $q_0$

$$\begin{aligned}q_0 &\leq \frac{(1-p_0)\sqrt{\frac{1}{2}}}{(\frac{1}{2}-p_0)\sqrt{2\pi v(1-\frac{1}{2})}} e^{-vD\left(\frac{1}{2}\|p_0\right)}\\ &\leq \frac{(1-p_0)\sqrt{2}}{\sqrt{\pi v}\varepsilon\left(1+O\left(1/\sqrt{n}\right)\right)} e^{-vD\left(\frac{1}{2}\|\frac{1}{2}-\frac{1}{2}\varepsilon\left(1+O\left(1/\sqrt{n}\right)\right)\right)} \quad (16)\\ &\leq \frac{(1-p_0)\sqrt{2}}{\sqrt{\pi v}\varepsilon\left(1+O\left(1/\sqrt{n}\right)\right)} e^{\frac{v\left(\ln(1-\varepsilon^2)+O\left(\frac{1}{\sqrt{n}}\right)\right)}{2}} \quad (17)\\ &\leq O\left(\frac{(1-\varepsilon^2)^{v/2}}{\sqrt{v}\varepsilon}\right) \quad (18)\end{aligned}$$

Whereas for $q_1$ we also obtain

$$\begin{aligned}q_1 &\leq \frac{p_1\sqrt{\frac{1}{2}}}{(p_1-\frac{1}{2})\sqrt{2\pi v\frac{1}{2}}} e^{-vD\left(\frac{1}{2}\|p_1\right)} \quad (19)\\ &\leq O\left(\frac{(1-\varepsilon^2)^{v/2}}{\sqrt{v}\varepsilon}\right) \quad (20)\end{aligned}$$

□

# E Proof of Theorem 1

We are ready now to prove Theorem 1. We first recall it.

**Theorem 1.** *Provided that Assumption 1 holds, we have for an MDPC code of type* $(v,w)$ *where* $v = \Theta(\sqrt{n})$ *and* $w = \Theta(\sqrt{n})$*:*

$$\mathbb{P}(S \geq t') \leq \frac{1}{\sqrt{t'}} e^{\frac{t'v}{4}\ln\left(1-\varepsilon^2\right)+\frac{t'}{8}\ln(n)+O\left(t'\ln(t'/t)\right)}.$$

*Proof.*

$$\begin{aligned}\mathbb{P}(S \geq t') &\leq \mathbb{P}(S_0 \geq t'/2 \cup S_1 \geq t'/2)\\ &\leq \mathbb{P}(S_0 \geq t'/2) + \mathbb{P}(S_1 \geq t'/2)\end{aligned}$$

By Assumption 1, $S_0$ is the sum of $n-t$ Bernoulli variables of parameter $q_0$. By applying Lemma 5 we obtain

$$\begin{aligned}\mathbb{P}(S_0 \geq t'/2) &\leq \frac{(1-q_0)\sqrt{\frac{t'}{2(n-t)}}}{(\frac{t'}{2(n-t)}-q_0)\sqrt{2\pi(n-t)(1-\frac{t'}{2(n-t)})}} e^{-(n-t)D\left(\frac{t'}{2(n-t)}\middle\|q_0\right)} \\ &\leq O\left(\frac{1}{\sqrt{t'}} e^{-(n-t)D\left(\frac{t'}{2(n-t)}\middle\|q_0\right)}\right) \end{aligned} \tag{21}$$

We observe now that

$$D\left(\frac{t'}{2(n-t)}\middle\|q_0\right) \geq D\left(\frac{t'}{2(n-t)}\middle\|O\left(\frac{(1-\varepsilon^2)^{v/2}}{\sqrt{v}\varepsilon}\right)\right) \tag{22}$$

where we used the upper-bound on $q_0$ coming from Lemma 3 and the fact that $D(x\|y) \geq D(x\|y')$ for $0 < y < y' < x < 1$. By using this and Lemma 4, we deduce

$$\begin{aligned} D\left(\frac{t'}{2(n-t)}\middle\|q_0\right) &\geq \frac{t'}{2(n-t)}\ln\left(\frac{t'}{2(n-t)}\right) - \frac{t'}{2(n-t)}\ln\left(O\left(\frac{(1-\varepsilon^2)^{v/2}}{\varepsilon\sqrt{v}}\right)\right) + O\left(\frac{t'}{2(n-t)}\right) \\ &\geq \frac{t'}{2(n-t)}\ln\left(\frac{t'\sqrt{v}}{n}\right) - \frac{t'v}{4(n-t)}\ln\left(1-\varepsilon^2\right) + O\left(\frac{t'}{n}\right) \\ &\geq \frac{t'}{2(n-t)}\ln\left(\frac{t\sqrt{v}}{n}\right) + \frac{t'}{2(n-t)}\ln(t'/t) - \frac{t'v}{4(n-t)}\ln\left(1-\varepsilon^2\right) + O\left(\frac{t'}{n}\right) \\ &\geq -\frac{t'}{8(n-t)}\ln n - \frac{t'v}{4(n-t)}\ln\left(1-\varepsilon^2\right) + O\left(\frac{t'\ln(t'/t)}{n}\right). \end{aligned}$$

By plugging this expression in (21) we obtain

$$\mathbb{P}(S_0 \geq t'/2) \leq \frac{1}{\sqrt{t'}} e^{\frac{t'v}{4}\ln\left(1-\varepsilon^2\right) + \frac{t'}{8}\ln(n) + O\left(t'\ln(t'/t)\right)}$$

On the other hand we have

$$\begin{aligned}\mathbb{P}(S_1 \geq t'/2) &\leq \frac{(1-q_1)\sqrt{\frac{t'}{2t}}}{(\frac{t'}{2t}-q_1)\sqrt{2\pi t(1-\frac{t'}{2t})}} e^{-tD\left(\frac{t'}{2t}\middle\|q_1\right)} \\ &\leq O\left(\frac{1}{\sqrt{t'}} e^{-tD\left(\frac{t'}{2t}\middle\|q_1\right)}\right) \end{aligned} \tag{23}$$

Similarly to what we did above, by using the upper-bound on $q_1$ of Lemma 3 and $D(x\|y) \geq D(x\|y')$ for $0 < y < y' < x < 1$, we deduce that

$$D\left(\frac{t'}{2t}\middle\|q_1\right) \geq D\left(\frac{t'}{2t}\middle\|O\left(\frac{(1-\varepsilon^2)^{v/2}}{\varepsilon\sqrt{v}}\right)\right)$$

By using this together with Lemma 4 we obtain

$$\begin{aligned} D\left(\frac{t'}{2t}\middle\|q_1\right) &\geq -h(t'/2t) - \frac{t'}{2t}\ln\left(O\left(\frac{(1-\varepsilon^2)^{v/2}}{\varepsilon\sqrt{v}}\right)\right) + O\left(\frac{(1-4\varepsilon^2)^{v/2}}{\varepsilon\sqrt{v}}\right) \\ &\geq -\frac{t'v}{4}\ln\left(1-\varepsilon^2\right) + \frac{t'}{8t}\ln n + O\left(\frac{t'}{t}\ln(t'/t)\right). \end{aligned}$$

By using this lower-bound in (23), we deduce

$$\mathbb{P}(S_1 \geq t'/2) \quad \leq \quad \frac{1}{\sqrt{t'}} e^{\frac{t'v}{4}\ln\left(1-\varepsilon^2\right)+\frac{t'}{8}\ln(n)+O\left(t'\ln(t'/t)\right)}.$$

□

# References


[ABB+17] Nicolas Aragon, Paulo Barreto, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Shay Gueron, Tim Güneysu, Carlos Aguilar Melchor, Rafael Misoczki, Edoardo Persichetti, Nicolas Sendrier, Jean-Pierre Tillich, and Gilles Zémor. BIKE. first round submission to the NIST post-quantum cryptography call, November 2017.

[BBC08] Marco Baldi, Marco Bodrato, and Franco Chiaraluce. A new analysis of the McEliece cryptosystem based on QC-LDPC codes. In *Proceedings of the 6th international conference on Security and Cryptography for Networks*, SCN '08, pages 246–262. Springer-Verlag, 2008.

[BBC+17] Marco Baldi, Alessandro Barenghi, Franco Chiaraluce, Gerardo Pelosi, and Paolo Santini. LEDAkem. first round submission to the NIST post-quantum cryptography call, November 2017.

[BC07] Marco Baldi and Franco Chiaraluce. Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, pages 2591–2595, Nice, France, June 2007.

[BCD+16] Magali Bardet, Julia Chaulet, Vlad Dragoi, Ayoub Otmani, and Jean-Pierre Tillich. Cryptanalysis of the McEliece public key cryptosystem based on polar codes. In *Post-Quantum Cryptography2016*, LNCS, pages 118–143, Fukuoka, Japan, February 2016.

[BCGO09] Thierry P. Berger, Pierre-Louis Cayrel, Philippe Gaborit, and Ayoub Otmani. Reducing key length of the McEliece cryptosystem. In Bart Preneel, editor, *Progress in Cryptology - AFRICACRYPT 2009*, volume 5580 of *LNCS*, pages 77–97, Gammarth, Tunisia, June 21-25 2009.

[BCS13] Daniel J. Bernstein, Tung Chou, and Peter Schwabe. Mcbits: Fast constant-time code-based cryptography. In Guido Bertoni and Jean-Sébastien Coron, editors, *Cryptographic Hardware and Embedded Systems - CHES 2013*, volume 8086 of *LNCS*, pages 250–272. Springer, 2013.

[BGG+17] Paulo S. L. M. Barreto, Shay Gueron, Tim Güneysu, Rafael Misoczki, Edoardo Persichetti, Nicolas Sendrier, and Jean-Pierre Tillich. CAKE: code-based algorithm for key encapsulation. In *Cryptography and Coding - 16th IMA International Conference, IMACC 2017, Oxford, UK, December 12-14, 2017, Proceedings*, volume 10655 of *LNCS*, pages 207–226. Springer, 2017.

[BGT11] Céline Blondeau, Benoît Gérard, and Jean-Pierre Tillich. Accurate estimates of the data complexity and success probability for various cryptanalyses. *Des. Codes Cryptogr.*, 59(1-3):3–34, 2011.

[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.

[BLM11] Paulo Barreto, Richard Lindner, and Rafael Misoczki. Monoidic codes in cryptography. In *Post-Quantum Cryptography 2011*, volume 7071 of *LNCS*, pages 179–199. Springer, 2011.

[BLP08] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the McEliece cryptosystem. In *Post-Quantum Cryptography 2008*, volume 5299 of *LNCS*, pages 31–46, 2008.

[BLP10] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Wild McEliece. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *LNCS*, pages 143–158, 2010.

[BLP11] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Wild McEliece Incognito. In Bo-Yin Yang, editor, *Post-Quantum Cryptography 2011*, volume 7071 of *LNCS*, pages 244–254. Springer Berlin Heidelberg, 2011.

[BS08] Bhaskar Biswas and Nicolas Sendrier. McEliece cryptosystem implementation: theory and practice. In Johannes Buchmann and Jintai Ding, editors, *Post-Quantum Cryptography, Second International Workshop, PQCrypto 2008, Cincinnati, OH, USA, October 17-19, 2008, Proceedings*, volume 5299 of *LNCS*, pages 47–62. Springer, 2008.

[CGG+14] Alain Couvreur, Philippe Gaborit, Valérie Gauthier-Umaña, Ayoub Otmani, and Jean-Pierre Tillich. Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes. *Des. Codes Cryptogr.*, 73(2):641–666, 2014.

[CMCP14] Alain Couvreur, Irene Márquez-Corbella, and Ruud Pellikaan. A polynomial time attack against algebraic geometry code based public key cryptosystems. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT 2014*, pages 1446–1450, June 2014.

[COT14] Alain Couvreur, Ayoub Otmani, and Jean-Pierre Tillich. Polynomial time attack on wild McEliece over quadratic extensions. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 17–39. Springer Berlin Heidelberg, 2014.

[COTG15] Alain Couvreur, Ayoub Otmani, Jean-Pierre Tillich, and Valérie Gauthier-Umaña. A polynomial-time attack on the BBCRS scheme. In J. Katz, editor, *Public-Key Cryptography - PKC 2015*, volume 9020 of *LNCS*, pages 175–193. Springer, 2015.

[CT91] Thomas M. Cover and Joy A. Thomas. *Information Theory*. Wiley Series in Telecommunications. Wiley, 1991.

[DGZ17] Jean-Christophe Deneuville, Philippe Gaborit, and Gilles Zémor. Ouroboros: A simple, secure and efficient key exchange protocol based on coding theory. In *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, volume 10346 of *LNCS*, pages 18–34. Springer, 2017.

[FM08] Cédric Faure and Lorenz Minder. Cryptanalysis of the McEliece cryptosystem over hyperelliptic curves. In *Proceedings of the eleventh International Workshop on Algebraic and Combinatorial Coding Theory*, pages 99–107, Pamporovo, Bulgaria, June 2008.

[FOPT10] Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. Algebraic cryptanalysis of McEliece variants with compact keys. In *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 279–298, 2010.

[FPdP14] Jean-Charles Faugère, Ludovic Perret, and Frédéric de Portzamparc. Algebraic attack against variants of McEliece with Goppa polynomial of a special form. In *Advances in Cryptology - ASIACRYPT 2014*, volume 8873 of *LNCS*, pages 21–41, Kaoshiung, Taiwan, R.O.C., December 2014. Springer.

[Gab05] Philippe Gaborit. Shorter keys for code based cryptography. In *Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005)*, pages 81–91, Bergen, Norway, March 2005.

[Gal63] Robert G. Gallager. *Low Density Parity Check Codes*. M.I.T. Press, Cambridge, Massachusetts, 1963.

[GJS16] Qian Guo, Thomas Johansson, and Paul Stankovski. A key recovery attack on MDPC with CCA security using decoding errors. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016*, volume 10031 of *LNCS*, pages 789–815, 2016.

[GSJB14] Danilo Gligoroski, Simona Samardjiska, Håkon Jacobsen, and Sergey Bezzateev. McEliece in the world of Escher. IACR Cryptology ePrint Archive, Report2014/360, 2014. `http://eprint.iacr.org/`.

[JM96] Heeralal Janwa and Oscar Moreno. McEliece public key cryptosystems using algebraic-geometric codes. *Des. Codes Cryptogr.*, 8(3):293–307, 1996.

[KL95] Gil Kalai and Nathan Linial. On the distance distribution of codes. *IEEE Trans. Inform. Theory*, 41(5):1467–1472, September 1995.

[LJ12] Carl Löndahl and Thomas Johansson. A new version of McEliece PKC based on convolutional codes. In *Information and Communications Security, ICICS*, volume 7168 of *LNCS*, pages 461–470. Springer, 2012.

[LT13] Grégory Landais and Jean-Pierre Tillich. An efficient attack of a McEliece cryptosystem variant based on convolutional codes. In P. Gaborit, editor, *Post-Quantum Cryptography'13*, volume 7932 of *LNCS*, pages 102–117. Springer, June 2013.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology - EUROCRYPT'93*, volume 765 of *LNCS*, pages 386–397, Lofthus, Norway, May 1993. Springer.

[MB09] Rafael Misoczki and Paulo Barreto. Compact McEliece keys from Goppa codes. In *Selected Areas in Cryptography*, Calgary, Canada, August 13-14 2009.

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

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

[MS07] Lorenz Minder and Amin Shokrollahi. Cryptanalysis of the Sidelnikov cryptosystem. In *Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 347–360, Barcelona, Spain, 2007.

[MTSB13] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, pages 2069–2073, 2013.

[Nie86] Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.

[OTD10] Ayoub Otmani, Jean-Pierre Tillich, and Léonard Dallot. Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes. *Special Issues of Mathematics in Computer Science*, 3(2):129–140, January 2010.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In S. Goldwasser, editor, *FOCS*, pages 124–134, 1994.

[SK14] Sujan Raj Shrestha and Young-Sik Kim. New McEliece cryptosystem based on polar codes as a candidate for post-quantum cryptography. In *2014 14th International Symposium on Communications and Information Technologies (ISCIT)*, pages 368–372. IEEE, 2014.

[SS92] Vladimir Michilovich Sidelnikov and S.O. Shestakov. On the insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Math. Appl.*, 1(4):439–444, 1992.

[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.

[Wie10] Christian Wieschebrink. Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. In *Post-Quantum Cryptography 2010*, volume 6061 of *LNCS*, pages 61–72. Springer, 2010.