MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes

Rafael Misoczki[1] and Jean-Pierre Tillich[1] and
Nicolas Sendrier[1] and Paulo S. L. M. Barreto[2]

[1] Project SECRET, INRIA-Rocquencourt, France
[2] Escola Politécnica, Universidade de São Paulo, Brazil

Abstract. Cryptography based on coding theory is believed to resist to quantum attacks (all cryptosystems based on factoring/discrete logarithm can be quantum attacked in polynomial time). The McEliece cryptosystem is the oldest code-based cryptosystem and its security relies on two problems: the indistinguishability of the code family and the hardness of decoding random linear codes. The former is usually the weakest one. The main drawback of this cryptosystem regards its huge public-keys. Recently, several attempts to reduce its key-size have been proposed. Almost all of them were successfully broken due to the additional algebraic structure used to reduce the keys. In this work, we propose McEliece variants from Moderate Density Parity-Check codes. These codes are LDPC codes of higher density than what is usually adopted for telecommunication solutions. We show that our proposal strongly strengthens the security against distinguishing attacks and also provides extremely compact-keys. Under a reasonable assumption, MDPC codes reduce the distinguishing problem to decoding a linear code and thus the security of our proposal relies only on a well studied coding-theory problem. Furthermore, using a quasi-cyclic structure, we provide the smallest public-keys for code-based cryptosystem. For 80-bits of security, the public-key has only 4800 bits. In summary, this represents the most competitive code-based cryptosystem ever proposed and is a strong alternative for traditional cryptography.

Keywords: post-quantum cryptography, code-based cryptography, coding-theory, LDPC codes.

1 Introduction

Code-Based Cryptography. In [38], Peter Shor showed that all cryptosystems based on the hardness of factoring or taking a discrete logarithm can be attacked in polynomial time with a quantum computer (see [10] for an extensive report). This threatens most if not all public-key cryptosystems deployed in practice, such as RSA [35] or DSA [24]. Cryptography based on coding theory, on the other hand, is believed to resist quantum attacks and is therefore considered as a viable replacement for those schemes in future applications. Yet, independently of their so-called "post-quantum" nature, code-based cryptosystems offer other benefits even for present-day applications due to their excellent algorithmic efficiency, which is up to several orders of complexity better than traditional schemes.

The first code-based cryptosystem is the McEliece cryptosystem [27], originally proposed using Goppa codes. Its security is based on two assumptions, the indistinguishability of the Goppa code family and the hardness of decoding a generic linear code. It is namely proved in [14] that if an adversary is not able to distinguish a Goppa code from a random code, then he is challenged to decode a generic linear code, a problem proved to be NP-complete [8]. However in [16] a distinguisher for Goppa codes of high rate (like those originally suggested for CFS signature [14] and for some realistic secure parameters of McEliece cryptosystems) is presented. Although this fact does not represent an effective attack, it would be more satisfactory to use other code families which would permit to ensure the completeness of such security reduction.

Although efficient, this cryptosystem suffers from an extremely large key size. There is a way to reduce considerably the key size which consists in choosing codes with a large automorphism group, such as quasi-cyclic codes [20]. It has been followed by several other proposals such as [28, 7]. The structural algebraic attack proposed in [17] succeeds in breaking many of them with the exception of the dyadic scheme based on binary Goppa codes proposed in [28]. It makes use of the fact that the underlying codes which are alternant codes come with an algebraic structure which allows a cryptanalysis consisting in setting up an algebraic system and then solving it with Gröbner bases techniques. Several particular features of the algebraic system make this attack feasible: the system is bihomogeneous and bilinear and most importantly the quasi-cyclic or the quasi-dyadic structure of these schemes allows a drastic reduction of the number of unknowns in the system. This kind of attack is exponential in nature in the case at hand, and can therefore be prevented rather easily by choosing more conservative parameters. However, again, it might be desirable to avoid this kind of algebraic attacks by suggesting other code families which would thwart completely this approach.

Related work. Low-Density Parity Check (LDPC) codes [21] would be a good candidate for achieving such a goal. These are codes with no algebraic structure, they just meet a very simple combinatorial property : they admit a sparse parity-check matrix. They admit efficient iterative decoding through algorithms based on Belief Propagation. They have been repeatedly suggested for the McEliece scheme [29, 4, 5, 3, 2]. The very first proposal [29] analyzes the use of simple LDPC codes in the original setup of McEliece: the private-key is the sparse parity-check matrix H of row weight w of a code $\mathcal{C}$, which allows for efficient decoding and the public-key is a public generator matrix $G' = S \cdot G \cdot P$ of a code $\mathcal{C}'$, where S is a scrambling matrix, G is a generator matrix for $\mathcal{C}$ and P is a permutation matrix. Unfortunately, looking for low weight codewords in the dual of the code $\mathcal{C}'$ leads to an attack which recovers a sparse parity-check matrix, allowing the adversary to decode successfully. In [3], a proposal to fix this problem is suggested. It consists in using a sparse matrix S and in replacing the permutation matrix P by a sparse invertible matrix Q of some small constant row weight m. The dual of $\mathcal{C}'$ has codewords of weight $\leq wm$ and they would allow to decode successfully, however for well chosen parameters w and m, finding such codewords is hard. Nevertheless, the unfortunate choices for Q and S allowed to cryptanalyze successfully the scheme in [31]. In [2], using a dense matrix S and with a more general matrix Q the variant seems to be immune against the attack suggested in [31]. Furthermore using a quasi-cyclic structure, it is possible to achieve compact keys. For 80-bits of security, the authors suggest public-keys in quasi-cyclic form, composed by 3 rows of 4 blocks of circulant matrices 4032×4032. This implies in a public-key of $3 \times 4 \times 4032 = 48384$ bits.

Our contribution. Our first observation in this paper is that changing the permutation matrix P into a more general matrix is not necessary for using an LDPC code in the McEliece cryptosystem. It is indeed possible to take an LDPC code there and avoid all message recovery attacks (using standard decoding algorithms) and key recovery attacks (aiming at finding low weight codewords in the dual of the public code) by choosing the parameters carefully. To avoid the second attack, the length and the row weight of the secret sparse parity-check matrix are just chosen to be large enough. For instance, for a rate $\frac{1}{2}$ code and for 80 bits of security, we chose the secret parity-check matrix to be of size 4800×9600 and rows with about 90 non zero entries (whereas the LDPC codes which are used in practice for error correcting purposes have much lower row weights, which are typically less than 10). We call them MDPC codes (which stands for Moderate Parity Check Codes) to insist on the fact that they admit a parity-check which is only moderately sparse. Notice that this terminology has already been proposed in the communications theory literature before for the very same concept [32]. The authors showed there that certain quasi-cyclic MDPC codes may perform well at moderate lengths for correcting a rather large number of errors by using a variation of the standard belief propagation taking advantage of the quasi-cyclic structure. However in our case, for the very large code lengths we choose in our scheme, the error correction performance degrade significantly when compared to standard LDPC codes. For instance, we correct for the aforementioned example only 84 errors, whereas any decent LDPC code of this length and rate would correct about 700-800 errors. Despite this fact, this number of errors is still large enough so that standard decoding algorithms for correcting errors in a generic linear code are thwarted by such parameters. More generally, for any security level and code rate, it is possible to set up the parameters in such a McEliece cryptosystem, namely length, code rate, weight of the rows in the secret parity-check matrix and number of errors so that standard attacks completely fail.

Our contribution is not only to observe that standard attacks on McEliece cryptosystems based on LDPC codes can be avoided by moving from LDPC codes to MDPC codes and choosing

the parameters appropriately, we also give a quite satisfactory security reduction to a well studied problem, namely decoding a linear code. For achieving this purpose, we only use a very natural assumption, namely that distinguishing an MDPC code from a random linear code amounts to be able to answer the question “does the dual code contain codewords of weight w?”, where w is the row weight chosen for the secret parity-check matrix of the code (the rows of this matrix belong to the dual code by definition). This provides a strong argument in favor of the security of this new scheme. This reduction to a single problem should be compared with the situation that we have right now for the McEliece cryptosystem based on Goppa codes:
(1) there is no security reduction to a single problem in this case, what is proven right now is that an attacker which is able to attack such a system is either able to decode a random linear code or is able to distinguish a Goppa code from a random code.
(2) The latter problem can now be solved for certain rates [16] and should not be considered as a hard problem in general.

For choosing our parameters, we have considered the most recent Information Set Decoding variant [6]. Using a non-asymptotic analysis, this algorithm gives the lowest work-factors for decoding random linear codes. We also present an analysis based on [37], taking into account the possible gains obtained by an adversary when multiple instances and solutions of the decoding problem are available. This is exactly what happens for MDPC codes. For the unstructured MDPC variant, our parameters lead to huge public-keys. However, using a quasi-cyclic structure, our proposal achieves extremely compact-keys. For instance, only 4800 bits for 80 bits of security. In summary, this represents the most competitive code-based cryptosystem ever proposed: the structure of the code family relies only on the existence of (possibly quasi-cyclic) low weight codewords (progresses in finding such a structure in general would represent a major breakthrough in coding-theory), efficient decoding can be achieved using LDPC decoding algorithms and finally it benefits from extremely compact keys.

2 Preliminaries

We gather here a few basic definitions which are used in this paper.

Definition 1 (Hamming distance and weight). *The Hamming weight (or simply weight) of a vector $x \in \mathbb{F}_2^n$ is the number $\mathsf{wt}(x)$ of its nonzero components. The Hamming distance (or simply distance) $d_h(x, y)$ between two vectors $x, y \in \mathbb{F}_2^n$ is the number of coordinates where they differ, i.e. $d_h(x, y) = \mathsf{wt}(x - y)$.*

Definition 2 (Linear codes). *A binary (n, r)-linear code $\mathcal{C}$ of length n, dimension k and codimension $r = n - k$, is a k-dimensional vector subspace of $\mathbb{F}_2^n$. The* rate *is defined by the ratio k/n. It is spanned by the rows of a matrix $G \in \mathbb{F}_2^{k\times n}$, called a* generator matrix *of the code. Equivalently, it is the kernel of a matrix $H \in \mathbb{F}_2^{r\times n}$, called a* parity-check matrix *of the code. The* dual *$\mathcal{C}^\perp$ of $\mathcal{C}$ is the linear code spanned by the rows of any parity-check matrix of $\mathcal{C}$.*

Definition 3 (Quasi-cyclic code). *An (n, r)-linear code is quasi-cyclic (QC) if there is some integer n_0 such that every cyclic shift of a codeword by n_0 places is again a codeword.*

When $n = n_0 p$, for some integer p, it is possible and convenient to have the generator and parity check matrix being composed by $p \times p$ circulant blocks. We call in this case such a code a quasi-cyclic code of order p. Note that a circulant block is completely described by its first row (or column) and that the algebra of $p \times p$ binary circulant matrices is isomorphic to the algebra of polynomials modulo $x^p - 1$ over $\mathbb{F}_2$.

Definition 4 (Matrix density). *The density of a matrix $H \in \mathbb{F}_2^{r\times n}$ is the average number of ones per row in H, that is*

$$\frac{\textit{number of entries equal to ones in the parity-check matrix}}{r}.$$

Definition 5 (Low-density parity-check code family). *A family $(\mathcal{C}_n)_{n\geq 0}$ of codes is said to be a low-density parity check code family if all these codes admit parity-check matrices of some bounded density $O(1)$.*

As explained in the introduction, LDPC codes which are used in practice have typically densities which are less than 10. In our case, we will be interested in codes with larger densities (with densities ranging between 90 and 644 for security parameters between 80 and 256 bits). Moreover as the security parameter goes to infinity, the density of the parity-check matrix will increase, but stays small when compared to the length of the code (it will scale like $O(\sqrt{n \log n})$). We call such codes MDPC codes. More formally, we define such codes by

Definition 6 (Moderate-density parity-check code family). *A family $(\mathcal{C}_n)_{n \geq 0}$ of codes is said to be a moderate-density parity check code family if all these codes admit parity-check matrices of density which are negligible in front of the codelength (say of the form $o(n_i)$ where n_i is the length of C_i).*

It will also be convenient to bring in the following definition

Definition 7 ((n, r, w)-code). *An (n, r, w)-code is a linear code of length n, codimension r which admits a parity check matrix with constant row weight w.*

When it is also quasi-cyclic and is LDPC or MDPC, we call such a code an (n, r, w)-quasi-cyclic low/moderate-density parity-check (QC-LDPC/QC-MDPC) code.

3 Moderate Density Parity-Check McEliece variants

In this section, we present two new McElice variants: one based on MDPC codes and another one on QC-MDPC codes. The first one benefits from the absence of any code structure but comes at the price of huge keys. The second one uses a quasi-cyclic structure to obtain very compact-keys.

(n, r, w)-MDPC code construction. A random (n, r, w)-MDPC code is easily generated by picking a random $r \times n$ matrix with rows of weight w. With overwhelming probability this matrix is of full rank and the rightmost $r \times r$ block is always invertible after possibly swapping a few columns.

(n, r, w)-QC-MDPC code construction. As in [4] (where the case of quasi-cyclic LDPC codes was considered) we are specially interested in (n, r, w)-QC-MDPC codes where $n = n_0 p$ and $r = p$. Basically, we pick one random word of length $n = n_0 p$ and weight w. This word will be the first row of an $r \times n$ matrix formed by n_0 circulant blocks H_i of size $p \times p$ and row weight w_i, such that $w = \sum_{i=0}^{n_0-1} w_i$. Therefore the matrix has the form $H = [H_0|H_1|\ldots|H_{n_0-1}]$. The other rows are obtained from $r - 1$ quasi-cyclic shifts.

A generator matrix G in row reduced echelon form can be easily derived from the H_i's blocks. Assuming H_{n_0-1} is non-singular (this particularly implies w_{n_0-1} odd, otherwise the rows of H_{n_0-1} would sum up to 0):

$$G = \left[\begin{array}{c|c} & (H_{n_0-1}^{-1} \cdot H_0)^T \\ & (H_{n_0-1}^{-1} \cdot H_1)^T \\ \mathrm{I} & \\ & \vdots \\ & (H_{n_0-1}^{-1} \cdot H_{n_0-2})^T \end{array} \right]$$

The performance of iterative decoding based on belief propagation does not depend only on the density of the parity-check matrix but also in how this weight is spread. For instance, cycles of length 4 in the Tanner graph associated to the code can sometimes be prejudicial for decoding capability. In [4], a construction based on random difference families avoids such a problem at the price of adding an algebraic relation on how the weight is distributed. For the codelengths and densities proposed in this work, it is very difficult (and impossible when the length becomes too large) to avoid 4-cycles and we choose to use a random construction. However, we wish to point out that despite the fact that there are four cycles in our construction, the analysis of belief propagation (which assumes that there are no such cycles) seems to be accurate in our situation. This random construction has also the benefit of supporting the security reduction presented in 4.1.

Therefore our scheme can be described as follows:

Key-Generation.

1. Generate a parity-check matrix $H \in \mathbb{F}_2^{r\times n}$ of a t-error-correcting (n, r, w)-MDPC or (n, r, w)-QC-MDPC code, as described above.
2. Generate its corresponding generator matrix $G \in \mathbb{F}_2^{(n-r)\times n}$ in row reduced echelon form.

The public key is G and the private key is H.

Encryption. To encrypt $m \in \mathbb{F}_2^{(n-r)}$ into $x \in \mathbb{F}_2^n$:

- Randomly select $e \in \mathbb{F}_2^n$ of $wt(e) \leq t$.
- Compute $x \leftarrow mG + e$.

Decryption. Let Ψ_H be a t-error-correcting LDPC decoding algorithm equipped with the knowledge of H. To decrypt $x \in \mathbb{F}_2^n$ into $m \in \mathbb{F}_2^{(n-r)}$,

- Compute $mG \leftarrow \Psi_H(mG + e)$.
- Extract the plaintext m from the first $(n - r)$ indices of mG.

Note that this description gets rid of the usual scrambling matrix S and permutation matrix P [3]. A folklore reasoning has given some security function to those matrices. However it is enough that the public-key does not reveal any useful information for decoding. Note also that the use of a CCA-2 security-conversion, like [19] and [23], allows for G in systematic-form, without bringing any security-flaw.

4 Security Assessment

The security assessment of our proposal is divided in two parts: its security reduction and the practical security assessment.

4.1 Security reduction

By security reduction, we mean a proof that an adversary able to attack the scheme is able solve some (presumably hard) algorithmic problem with a similar computational effort.

We consider the same four parameters as in the previous section, n the code length, r the code co-dimension, w the row weight and t the error correcting capability. Let $\mathcal{F}_{n,r,w}$ denote a family of codes which can be either (n, r, w)-MDPC or (n, r, w)-QC-MDPC. We assume the public key is a parity check matrix of some code in $\mathcal{F}_{n,r,w}$, we denote $\mathcal{K}_{n,r,w}$ the key space and $\mathcal{H}_{n,r} \supset \mathcal{K}_{n,r,w}$ the *apparent* key space. In the MDPC case $\mathcal{H}_{n,r}$ is the set of all full rank matrices in $\mathbb{F}_2^{r\times n}$ while in the quasi-cyclic case $\mathcal{H}_{n,r}$ is restricted to block circulant matrices. All the statements in this section are valid in both cases.

Generic Reduction. Let $\mathcal{S}_n(0, t)$ denote the sphere centered in zero of radius t in the Hamming space $\mathbb{F}_2^n$ and let Ω denote the probability space consisting of the sample space $\mathcal{H}_{n,r} \times \mathcal{S}_n(0, t)$ equipped with a uniform distribution. We define:

Definition 8. ***Distinguisher.*** *A program $\mathcal{D} : \mathcal{H}_{n,r} \longrightarrow \{0, 1\}$ is a (T, ϵ)-distinguisher for $\mathcal{K}_{n,r,w}$ (vs. $\mathcal{H}_{n,r}$) if it runs in time at most T and the* advantage of $\mathcal{D}$ for $\mathcal{K}_{n,r,w}$

$$Adv(\mathcal{D}, \mathcal{K}_{n,r,w}) = |Pr_\Omega(\mathcal{D}(H) = 1|H \in \mathcal{K}_{n,r,w}) - Pr_\Omega(\mathcal{D}(H) = 1)|$$

is greater than ϵ.

[3] This kind of McEliece description can also be found in [36] and in [12], for instance.

Decoder. *A program $\phi : \mathcal{H}_{n,r} \times \mathbb{F}_2^r \longrightarrow \mathcal{S}_n(0,t)$ is a (T,ϵ)-decoder for $(\mathcal{H}_{n,r}, t)$ if it runs in time at most T and its* success probability

$$Succ(\phi) = Pr_{\Omega}(\phi(H, eH^T) = e)$$

is greater than ϵ.

Adversary. *A program $\mathcal{A} : \mathcal{H}_{n,r} \times \mathbb{F}_2^n \longrightarrow \mathcal{S}_n(0,t)$ is a (T,ϵ)-adversary against $\mathcal{K}_{n,r,w}$-Niederreiter if it runs in time at most T its* success probability

$$Succ(\mathcal{A}, \mathcal{K}_{n,r,w}) = Pr_{\Omega}(\mathcal{A}(H, eH^T) = e | H \in \mathcal{K}_{n,r,w})$$

is greater than ϵ.

As in [36] the reduction is stated for the Niederreiter scheme [30]. An adversary against $\mathcal{K}_{n,r,w}$-McEliece could be defined as a program $\mathcal{H}_{n,r} \times \mathbb{F}_2^n \to \mathbb{F}_2^k \times \mathcal{S}_n(0,t)$ with $k = n - r$ the code dimension. It is a simple matter, first remarked in [25], to prove that this adversary is equivalent to the Niederreiter adversary but the probability space Ω would require a larger sample set $\mathcal{H}_{n,r} \times \mathbb{F}_2^k \times \mathcal{S}_n(0,t)$ which would make all the statements and proofs more cumbersome. Below, the proposition from [36] which supports the security reduction.

Proposition 1. *Given the security parameters (n, r, w) and t, if there exists a (T, ϵ)-adversary against $\mathcal{K}_{n,r,w}$-Niederreiter, then there exists either a $(T, \epsilon/2)$-decoder for $(\mathcal{H}_{n,r}, t)$ or a $(T + O(n^2), \epsilon/2)$-distinguisher for $\mathcal{K}_{n,r,w}$ vs. $\mathcal{H}_{n,r}$.*

Proof. Let $\mathcal{A} : \mathcal{H}_{n,r} \times \mathbb{F}_2^r \to \mathcal{S}_n(0,t)$ be a (T, ϵ)-adversary against $\mathcal{K}_{n,r,w}$-Niederreiter. We define the following distinguisher:

```
$\mathcal{D}$: input $H \in \mathcal{H}_{n,r}$.
    $e \leftarrow \mathcal{S}_n(0,t)$ //pick randomly and uniformly
    if $(\mathcal{A}(H, eH^{\mathrm{T}}) = e)$ then return $1$ else return $0$.
```

We have

$$\begin{aligned} \mathrm{Pr}_{\Omega}(\mathcal{D}(H) = 1) &= \mathrm{Pr}_{\Omega}(\mathcal{A}(H, eH^{\mathrm{T}}) = e) \\ &= \mathrm{Succ}(\mathcal{A}) \\ \mathrm{Pr}_{\Omega}(\mathcal{D}(H) = 1 | H \in \mathcal{K}_{n,r,w}) &= \mathrm{Pr}_{\Omega}(\mathcal{A}(H, eH^{\mathrm{T}}) = e | H \in \mathcal{K}_{n,r,w}) \\ &= \mathrm{Succ}(\mathcal{A}, \mathcal{K}_{n,r,w}) \end{aligned}$$

thus $Adv(\mathcal{D}, \mathcal{K}_{n,r,w}) = |Succ(\mathcal{A}, \mathcal{K}_{n,r,w}) - Succ(\mathcal{A})|$ and particularly:

$$Adv(\mathcal{D}, \mathcal{K}_{n,r,w}) + Succ(\mathcal{A}, \mathcal{K}_{n,r,w}) \geq Succ(\mathcal{A})$$

Since $Succ(\mathcal{A}, \mathcal{K}_{n,r,w}) \geq \epsilon$, we either have $Adv(\mathcal{C}, \mathcal{K}_{n,r,w})$ or $Succ(\mathcal{A})$ greater or equal to $\epsilon/2$ (recall that both are positive). The running time of $\mathcal{D}$ is equal to the running time of $\mathcal{A}$ increased by the cost for picking e and computing the product eH^{T}, which cannot exceed $O(n^2)$. So either $\mathcal{A}$ is a (T, ϵ)-decoder for $(\mathcal{H}_{n,r}, t)$ or $\mathcal{D}$ is a $(T + O(n^2), \epsilon/2)$-distinguisher for $\mathcal{K}_{n,r,w}$. □

z A distinguisher for $\mathcal{K}_{n,r,w}$ *vs.* $\mathcal{H}_{n,r}$ and a decoder for $(\mathcal{H}_{n,r}, t)$ provide a solution respectively to the two following problems

Problem 1 (Code distinguishing problem).
Parameters: $\mathcal{K}_{n,r,w}$, $\mathcal{H}_{n,r}$.
Instance: a matrix $H \in \mathcal{H}_{n,r}$.
Question: is $H \in \mathcal{K}_{n,r,w}$?

Problem 2 (Computational syndrome decoding problem).
Parameters: $\mathcal{H}_{n,r}$, an integer $t > 0$.
Instance: a matrix $H \in \mathcal{H}_{n,r}$ and a vector $s \in \mathbb{F}_2^r$.
Problem: find a vector $e \in \mathcal{S}_n(0,t)$ such that $eH^{\mathrm{T}} = s$.

So, from Proposition 1, it will be enough to assume that none of those problems can be solved efficiently to insure that no efficient adversary against the scheme exists.

Reduction for MDPC Codes. We introduce an additional problem, which consists in deciding the existence of words of given weight in a given linear code. Note that the code that we consider below has a *generator matrix* $H \in \mathcal{H}_{n,r}$, it is thus the dual of a code in $\mathcal{F}_{n,r,w}$.

Problem 3 (Codeword existence problem).
Parameters: $\mathcal{H}_{n,r}$, an integer $w > 0$.
Instance: a matrix $H \in \mathcal{H}_{n,r}$.
Question: is there a codeword of weight w in the code of generator matrix H?

Ideally, we would like to replace the Problem 1 by the Problem 3 in the security reduction statement (Proposition 1). Unfortunately, this would introduce, to replace the advantage of a distinguisher, the quantity

$$Adv(\mathcal{E}, \mathcal{K}_{n,r,w}) = |Pr_{\Omega}(\mathcal{E}(H) = 1 | H \in \mathcal{K}_{n,r,w}) - Pr_{\Omega}(\mathcal{E}(H) = 1)|$$

($\mathcal{E}$ denotes a program deciding the of the existence of a word weight w in a given code) which is not directly related to the hardness of Problem 3. We would reach our purpose if the following conjecture was true.

Conjecture 1 *Solving Problem 1 for parameters $(\mathcal{H}_{n,r}, \mathcal{K}_{n,r,w})$ is not easier than solving Problem 3 for the parameters $(\mathcal{H}_{n,r}, w)$.*

Of course we wish the statement to be as tight as possible, but it would be satisfactory if "not easier" only meant "up to a polynomial factor". Within this conjecture we could modify the reduction to a claim that the $\mathcal{K}_{n,r,w}$-McEliece scheme is at least as hard as either Problem 2 and Problem 3. Now if we remark[4] that the Problem 3 is polynomially equivalent to its associate computational problem:

Problem 4 (Codeword finding problem).
Parameters: $\mathcal{H}_{n,r}$, an integer $w > 0$.
Instance: a matrix $H \in \mathcal{H}_{n,r}$.
Problem: find a codeword of weight w in the code of generator matrix H.

and that this Problem 4 is polynomially equivalent to syndrome decoding (Problem 2) we may then produce strong security statements.

Security statements: (assuming Conjecture 1)

- Breaking the MDPC variant of McEliece or Niederreiter is not easier than solving the syndrome decoding problem in a random linear code.
- Breaking the QC-MDPC variant of McEliece or Niederreiter is not easier than solving the syndrome decoding problem in a random quasi-cyclic linear code.

4.2 Practical security

In this section, we analyze the various scenarios of attacks against the proposed scheme. Key attacks aim either at recovering the secret decoder or simply distinguish the public key from a random matrix (what invalidates the security reduction). Message attacks try to decode one particular message considered as a noisy codeword.

Consider the system as an instantiation of the McEliece (or Niederreiter) scheme with an (n, r, w)-MDPC code, possibly quasy-cyclic, correcting t errors. We denote $\mathcal{C}$ the hidden MDPC code defined by the public key (a generator matrix of $\mathcal{C}$ for McEliece or a parity check matrix of $\mathcal{C}$ for Niederreiter). We claim that the best attacks for each scenario are:

- *Key distinguishing attack:* exhibit one codeword of $\mathcal{C}^{\perp}$ of weight w.
- *Key recovery attack:* exhibit r codewords of $\mathcal{C}^{\perp}$ of weight w.
- *Decoding attack:* decode t errors in an $(n, n-r)$-linear code.

[4] See the appendices for the proofs.

For all those attacks we have to solve either the codeword finding problem or the computational syndrome decoding problem. For both those problems and for the considered parameters the best technique is information set decoding (ISD) [33]. In today's state-of-the-art the best variants derive from Stern's collision decoding algorithm [39]. There have been numerous contributions and improvements [15, 13, 11, 18, 9] until the recent asymptotic improvements [26, 6].

For selecting our parameters, we have analyzed all of them and the variant [6], here denoted by $1 + 1 = 0$ ISD variant, gives slightly lower workfactors. For a more curious reader, we leave in Appendix C the closed formula of our non-asymptotic analyses of $1 + 1 = 0$ ISD variant[5].

Besides decoding workfactor computations, there is a novelty related to the practical security assessment of our proposal. The probability of finding low weight codewords in MDPC codes is bigger than what is usually assumed. For example, when we want to find *one* low weight codeword in the dual of an MDPC code (like in the distinguishing problem), we must notice that there exist *several* solutions (at least those codewords which compose the sparse parity-check matrix), not only one as usually considered. Therefore next we discuss about the possible extra gains obtained by an adversary attacking MDPC or QC-MDPC codes, i.e. when multiple instances/solutions of the decoding problem indeed exist.

Impact of multiple instances and multiple solutions. We denote by $\mathrm{WF}_{\mathrm{isd}}(n, r, t)$ the cost for decoding t errors in a binary linear code of length n and codimension r when there is a single solution of the problem. It is also the cost for finding a word of weight t in a binary linear code of the same length and codimension. Our analyses is based on [37] [6], also mentioned by *Decoding One Out of Many* setting (DOOM), where it is analyzed the workfactor gain when multiple instances are attacked simultaneously and the adversary is satisfied with a solution for a single of those instances.

Information Set Decoding algorithms are iterative and can be roughly described as follows. At each iteration a Gaussian elimination is performed on the parity-check matrix and two lists of partial syndromes are produced. Each element of the intersection of those lists has a chance to produce the solution. Both lists have a certain size L (in Appendix C, it is denoted by S_0) which depends on the particular variant and on optimal parameters. Each iteration has a probability P to produce the solution which also which depends on the optimal parameters. *When the parameters are optimal* the workfactor $\mathrm{WF}_{\mathrm{isd}}(n, r, t)$ is equal, up to a small factor, to the ratio L/P.

When the problem has several solutions, say N_s, the probability of success P will increase by a factor N_s (as long as $N_s P \ll 1$). When several instances, say N_i, are treated simultaneously the list size L will increase at most by a factor $\sqrt{N_i}$. The square root derives from the fact that all variants of collision decoding make use of the birthday paradox: if the search space increases by a factor N_i, the complexity increases by a factor $\sqrt{N_i}$. In short, the gain obtained by the adversary when multiple instances/solutions are allowed is $N_s/\sqrt{N_i}$ [7].

Key Distinguishing Attack. To distinguish a public key from a random matrix it is enough to produce a word of weight w in the dual code $\mathcal{C}^{\perp}$. In this scenario we apply ISD to the all-zero syndrome and the problem has r solutions (the r rows of the sparse parity check matrix). Then we have $N_s = r$ and $N_i = 1$ and the distinguishing attack costs

$$\mathrm{WF}_{\mathrm{dist}}(n, r, w) = \frac{\mathrm{WF}_{\mathrm{isd}}(n, n - r, w)}{r}.$$

In the quasi-cyclic case there is no obvious speedup and the distinguishing attack has the same cost as above.

[5] This is part of an unpublished more general work in progress.

[6] The possibility to exploit multiple instances, say N, to gain a factor of order $\sqrt{N}$ was studied for Dumer's algorithm [15].

[7] In general, the real gain is in fact slightly smaller because the optimal parameters are not the same with multiple instances or with only one (see the detailed analysis in [37]).

Key Recovery Attack. To recover a decoder and thus the secret key it is enough to recover all (or almost all) the low weight parity check equations. All ISD variants are randomized and thus we can make r independent calls to a codeword finding algorithm. Each call costs on average $\frac{\mathrm{WF}_{\mathrm{isd}}(n,n-r,w)}{r}$ because there are r codewords of weight w. Therefore on average, recovering almost all equations will cost

$$\mathrm{WF}_{\mathrm{reco}}(n,r,w) = \mathrm{WF}_{\mathrm{isd}}(n,n-r,w).$$

In the quasi-cyclic case, any word of low weight will provide the sparse matrix (the sparse parity check matrix is the concatenation of several $r \times r$ circulant blocks) and thus the key recovery attack is not more expensive than the key distinguishing attack.

$$\mathrm{WF}^{\mathrm{QC}}_{\mathrm{reco}}(n,r,w) = \mathrm{WF}^{\mathrm{QC}}_{\mathrm{dist}}(n,r,w) = \frac{\mathrm{WF}_{\mathrm{isd}}(n,n-r,w)}{r}.$$

Decoding Attack. In the MDPC (*i.e.* non quasi-cyclic) case, the message security is related to the hardness of decoding t errors in a seemingly random binary linear code of length n and codimension r

$$\mathrm{WF}_{\mathrm{dec}}(n,r,t) = \mathrm{WF}_{\mathrm{isd}}(n,r,t).$$

In the quasi-cyclic case, any cyclic shift of the target syndrome $s \in \mathbb{F}_2^r$ provides a new instance whose solution is equal to the one of the original syndrome, up to a block-wise cyclic shift. The number of instances and the number of solutions are thus $N_i = N_s = r$. Therefore a factor $\sqrt{r}$ (at most) is gained.

$$\mathrm{WF}^{\mathrm{QC}}_{\mathrm{dec}}(n,r,t) \geq \frac{\mathrm{WF}_{\mathrm{isd}}(n,r,t)}{\sqrt{r}}.$$

	MDPC	QC-MDPC
Key distinguishing	$\frac{1}{r}\mathrm{WF}_{\mathrm{isd}}(n,n-r,w)$	$\frac{1}{r}\mathrm{WF}_{\mathrm{isd}}(n,n-r,w)$
Key recovery	$\mathrm{WF}_{\mathrm{isd}}(n,n-r,w)$	$\frac{1}{r}\mathrm{WF}_{\mathrm{isd}}(n,n-r,w)$
Decoding	$\mathrm{WF}_{\mathrm{isd}}(n,r,t)$	$\frac{1}{\sqrt{r}}\mathrm{WF}_{\mathrm{isd}}(n,r,t)$

Table 1. Best attacks for code-based encryption schemes using t-error correcting (n,r,w)-MDPC (or QC-MDPC) codes

A Final Remark on Practical Security. For the parameter selection presented in Section 6, we have considered the non-asymptotic analyses of the $1+1=0$ ISD variant (which provides the lowest workfactors for our parameters), *decreased* by the possible gains obtained from the multiplicity of solutions/instances, as explained above. Note that the complex structure of the $1+1=0$ ISD variant (an increased number of initial lists, pairs of non-disjoint lists and the probability of overlapped positions) might prejudice the maximal gain claimed for DOOM: $N_s/\sqrt{N_i}$. But since the difference of the $1+1=0$ ISD variant work-factor to the work-factor of less complex variants (which achieve the maximal gain for DOOM) is marginal, it is reasonable to assume a secure lower bound for decoding attacks considering the workfactor of the $1+1=0$ ISD variant decreased by the optimal gain for DOOM.

5 Decoding Algorithm and Error Correction Capability of MDPC codes

In this section, we present a discussion about decoding MDPC codes. Our approach is to use the same decoding framework available for LDPC codes.

Decoding algorithm. There are several decoding algorithms available for LDPC codes with different features. All these algorithms are iterative and provide an error correction capability which linearly increases along with the codelength and which decreases as the parity-check matrix density increases. Basically, we can divide them in two groups. The first is an iterative bit-flipping algorithm [21] which flips bits locally in the code and hopes to converge in this way to the right solution. It is quite fast but corrects less errors than Gallager's belief propagation algorithm [21] also called Sum-Product algorithm [22]. However, in our case where we use MDPC codes instead of LDPC codes, using the first kind of algorithm seems more appropriate: the gain in decoding complexity more than outweighs the slight improvement in performance of the second algorithm.

The estimation of error correction capability for LDPC or MDPC codes is a hard task. In general, two steps are needed. The first one is theoretical and probabilistic, providing what is known as the waterfall threshold. From this value, reliable decoding can be expected, i.e. it is *possible* to achieve correct decoding when the codelength goes to infinity. Then a second step based on the exhaustive decoding simulation is adopted, refining this estimation. This provides a failure decoding probability. Thus a valid approach for determining the error correction capability for such codes is to evaluate the initial waterfall threshold through exhaustive simulation and decrease it until a negligible decoding failure rate is verified.

In Appendix A, we describe a way to compute the waterfall threshold for the bit-flipping algorithm, as presented in [21]. Since this algorithm does not achieve the best error correction capability, this analyses can be used as a lower bound for the error correction capability of more elaborate decoding algorithms.

5.1 Dealing with decoding failures.

An important remark regarding the use of MDPC codes in cryptography refers to its probabilistic decoding nature, i.e. these algorithms admit a probability of decoding failure. For cryptography purposes, this problem must be addressed and below we present three approaches to deal with this problem.

The first approach is choosing a number of errors conservatively smaller than the theoretical threshold, implying in negligible decoding failure rates. This is a problem usually faced in error-correcting codes applications. A traditional approach adopted in this scenario consists in scaling this decoding failure rate to be smaller than the failure rate of the machine where the system is deployed. In general, this is enough to enable practical applications.

Disregarding some precautions in the setup, it would be interesting to deal with this (very) unlikely events on the fly. Thus we present the second approach to deal with this problem. It comes from the fact that the error decoding capability estimation for the Bit-Flipping algorithm gives a highly conservative bound for error correction capability in general. We can always resort to more sophisticated algorithms, which benefit from better error correction capability, at the price of more involved decoding algorithms, reducing the overall decoding failure probability to extremely small values.

The third approach refers to the use of a CCA-2 secure conversion. In short, a *CCA2*-secure conversion, like [19], uses hash functions and random sequences to ensure the indistinguishability of the encrypted messages. Thus, when the application allows the following scenario, a simple and naive approach can be used to address this issue: requesting a new encryption for messages with decoding failure. Since the encrypted messages behave like random sequences, the adversary would not be able to extract any information from this redundancy.

6 Practical application

In this section, we provide practical parameters and discuss about the particularities regarding the practical application of our scheme. Table 2 summarizes the parameters for our quasi-cyclic variant, the most relevant for practical applications. For each security level, we propose three parameter sets, for $n_0 = 2$, $n_0 = 3$ and $n_0 = 4$, leading to different code rates: 1/2, 2/3, 3/4, respectively.

Security. For the security assessment, we consider the non-asymptotic analysis of the $1+1=0$ ISD variant decreased by the possible gains obtained from the multiple solutions/instances of the decoding problem, as explained in 4.2. For example, regarding the parameters $n_0 = 2$, $n = 9600$, $r = 4800$, $w = 90$, $t = 84$, our analysis of the $1+1=0$ ISD variant gives a cost of $2^{87.16}$ for decoding attacks and $2^{92.70}$ for key-recovery attacks. Decreasing them by a factor of $\sqrt{4800}$ and 4800, respectively, give final work-factors of $2^{81.04}$ and $2^{80.47}$.

Error correction capability. Regarding the error correction capability of MDPC codes, we consider the estimation for the bit flipping algorithm. Note that this estimation can be seen as a lower bound for more elaborate decoding algorithms. Choosing values conservatively below this threshold and verifying them with simulation, we have selected parameters that achieve decoding failure rates below 10^{-7} for the QC-MDPC variant. Note that, for the same parameters, the MDPC variant might present a worse error correction capability due to the non-regularity of the column weights, but significant improvements can be obtained with a slightly increased code length.

Key size. The public key-size is given by $(n-r)$ for the QC-MDPC variant and by $r(n-r)$ for the MDPC variant. In practice, the MDPC variant obtains huge keys whilst the QC-MDPC allows for extremely compact keys. For $n_0 = 2$, we achieved the smallest key-sizes. Note that increasing n_0 provides better code rates at the price of less compact key sizes. Table 3 provides a comparison of the key-sizes of our proposal and the potential [8] key size of QC-LDPC variant proposed in [2], the key size of the Quasi-Dyadic Goppa McEliece variant [28] and the original McEliece scheme using update parameters provided in [11]. The column r also gives the syndrome size in bits.

Complexity efficiency. There is no novelty in complexity efficiency regarding the MDPC and QC-MDPC McEliece variants. For the key-generation, both MDPC and QC-MDPC variants depend only on the generation of random word(s). The encryption reduces to the computation of a matrix-vector product plus an addition of vectors. Many optimizations have already been proposed for the quasi-cyclic case [2]. Regarding the decryption, since we are dealing with denser codes, a worse decoding algorithmic performance is expected. However this does not represent a problem in practice. We used the bit flipping algorithm, which is very simple and for an (n, r, w)-MDPC code has complexity of order $\lambda r w^2$, where λ is the necessary number of iterations until the syndrome converges to a zero-vector. In general and in the case at hand, the algorithm converges very quickly and λ is negligible in comparison with n and w. Note also that the nature of the algorithm allows for parallelized operations. For a practical example, we achieved decryption timings of a few milliseconds for the parameters of 80-bits of security in a non-optimized C++ implementation running at an Intel Xeon CPU @3.20GHz. We prefer to omit these timings since serious optimizations should lead to much better results.

Table 2. Suggested parameters. Syndrome and key size given in bits.

Level security	n_0	n	r	w	t	QC-MDPC key-size
80	2	9600	4800	90	84	4800
80	3	10752	3584	153	53	7168
80	4	12288	3072	220	42	9216
128	2	19712	9856	142	134	9856
128	3	22272	7424	243	85	14848
128	4	25088	6272	340	68	18816
256	2	65536	32768	274	264	32768
256	3	67584	22528	465	167	45056
256	4	81920	20480	644	137	61440

[8] In [2], the authors did not consider the fact that using a CCA-2 secure conversion it is allowed to have public-keys in systematic form.

Table 3. Key-size comparison. Key-sizes given in bits.

Level security	QC-MDPC	QC-LDPC [2]	QD-Goppa [28]	Goppa [11]
80	4800	12096	20480	460 647
128	9856	–	32768	1 537 536
256	32768	–	65536	7 667 855

Scaling of the parameters for very large security. Our system can be scaled to meet arbitrarily large security requirements. It is rather straightfoward to prove that the number of errors which can be corrected by the bit flipping algorithm is of order $\frac{n(1+o(1))\ln(w(1-R))}{4w}$ where n is the codelength, w the density of the parity-check matrix, R is the rate of the code. Message recovery attacks and key recovery attacks are of the same order of complexity in this case when w is chosen of the form $(1+o(1))\sqrt{\frac{n\ln n\ln(1-R)}{\ln R}}$. Chosing an $(n,(1-R)n,w)$-code with w of this form allows to reach arbitrarily large security when n goes to infinity.

7 Conclusion

In this work, we propose two McEliece variants from Moderate Density Parity-Check codes. These codes are LDPC codes of higher density than what is usually adopted for telecommunication solutions. The McEliece cryptosystem is the oldest code-based cryptosystem and its security relies on two problems: the indistinguishability of the code family and the hardness of decoding random linear codes. The former is usually the weakest one.

Under the reasonable assumption that distinguishing a (quasi-cyclic) MDPC code from a (quasi-cyclic) random linear code amounts to be able to answer the question “does the dual code contain codewords of weight w?”, our proposal reduces the distinguishing problem to decoding a (quasi-cyclic) linear code and thus its security relies only on a well studied coding-theory problem.

The main drawback of code-based cryptosystems is the huge public-keys. Recently, several attempts to reduce its key-size have been proposed. Almost all of them were successfully broken due to the additional algebraic structure used to reduce the keys. Our variants are based on MDPC codes and another one on Quasi-Cyclic MDPC codes. The first one benefits from the absence of any code structure but comes at the price of huge keys. Using a quasi-cyclic structure, we provide the smallest public-keys for code-based cryptosystem so far. For 80-bits of security, the public-key has only 4800 bits.

Regarding its complexity efficiency, for the key-generation, both MDPC and QC-MDPC variants depend only on the generation of random word(s). The encryption is the computation of a matrix-vector product plus an addition of vectors. For the decryption, we suggest the use of an algorithm from the usual LDPC decoding framework. The Bit-Flipping algorithm is very simple and achieves very low complexity. In summary, this represents the most competitive code-based cryptosystem ever proposed and is a strong alternative for traditional cryptography.

References

1. V. L. Arlazarov, E. A. Dinic, M. A. Kronrod, and I. A. Faradzev. On economical construction of the transitive closure of a directed graph. *Soviet Mathematics—Doklady*, 11(5):1209 – 1210, 1970.
2. M. Baldi, M. Bodrato, and F. 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, Berlin, Heidelberg, 2008. Springer-Verlag.
3. M. Baldi and F. Chiaraluce. Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 2591 –2595, june 2007.
4. M. Baldi, F. Chiaraluce, and R. Garello. On the usage of quasi-cyclic low-density parity-check codes in the McEliece cryptosystem. In *Proceedings of the First International Conference on Communication and Electronics (ICEE’06)*, pages 305–310, October 2006.
5. M. Baldi, F. Chiaraluce, R. Garello, and F. Mininni. Quasi-cyclic low-density parity-check codes in the McEliece cryptosystem. In *Communications, 2007. ICC ’07. IEEE International Conference on*, pages 951 –956, june 2007.

6. A. Becker, A. Joux, A. May, and A. Meurer. Decoding random binary linear codes in $2^{n/20}$: How 1+1=0 improves information set decoding. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 520–536. Springer, 2012.
7. T. P. Berger, P.-L. Cayrel, P. Gaborit, and A. Otmani. Reducing key length of the McEliece cryptosystem. In B. Preneel, editor, *Progress in Cryptology – Africacrypt'2009*, volume 5580 of *Lecture Notes in Computer Science*, pages 77–97. Springer, 2009.
8. E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *Information Theory, IEEE Transactions on*, 24(3):384 – 386, may 1978.
9. D. Bernstein, T. Lange, and C. Peters. Smaller decoding exponents: Ball-collision decoding. In P. Rogaway, editor, *Advances in Cryptology CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 743–760. Springer Berlin / Heidelberg, 2011. 10.1007/978-3-642-22792-942.
10. D. J. Bernstein, J. Buchmann, and E. Dahmen, editors. *Post-Quantum Cryptography.* Springer-Verlag, 2009.
11. D. J. Bernstein, T. Lange, and C. Peters. Attacking and defending the McEliece cryptosystem. In *Proceedings of the 2nd International Workshop on Post-Quantum Cryptography*, PQCrypto '08, pages 31–46, Berlin, Heidelberg, 2008. Springer-Verlag.
12. B. Biswas and N. Sendrier. Mceliece cryptosystem implementation: Theory and practice. In J. Buchmann and J. Ding, editors, *Post-Quantum Cryptography*, volume 5299 of *Lecture Notes in Computer Science*, pages 47–62. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-88403-3-4.
13. A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *Information Theory, IEEE Transactions on*, 44(1):367 –378, Jan. 1998.
14. N. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology – Asiacrypt'2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 157–174, Gold Coast, Australia, 2001. Springer.
15. I. Dumer. On minimum distance decoding of linear codes. In *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory*, pages 50–52, Moscow, 1991.
16. J.-C. Faugère, V. Gauthier, A. Otmani, L. Perret, and J.-P. Tillich. A distinguisher for high rate McEliece cryptosystems. In *ITW 2011*, pages 282–286, Paraty, Brazil, Oct. 2011.
17. J.-C. Faugère, A. Otmani, L. Perret, and J.-P. Tillich. Algebraic cryptanalysis of McEliece variants with compact keys. In H. Gilbert, editor, *Advances in Cryptology – Eurocrypt'2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 279–298. Springer, 2010.
18. M. Finiasz and N. Sendrier. Security bounds for the design of code-based cryptosystems. In M. Matsui, editor, *Advances in Cryptology – Asiacrypt 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 88–105. Springer, 2009.
19. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology – CRYPTO'1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, Gold Coast, Australia, 1999. Springer.
20. P. Gaborit. Shorter keys for code based cryptography. In *International Workshop on Coding and Cryptography – WCC'2005*, pages 81–91, Bergen, Norway, 2005. ACM Press.
21. R. G. Gallager. *Low-Density Parity-Check Codes.* M.I.T. Press, 1963.
22. J. Hagenauer, E. Offer, and L. Papke. On the inherent intractability of certain coding problems (corresp.). *Information Theory, IEEE Transactions on*, 42(2):429 – 445, march 1996.
23. K. Kobara and H. Imai. Semantically secure mceliece public-key cryptosystems -conversions for mceliece pkc -. In K. Kim, editor, *Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 19–35. Springer Berlin / Heidelberg, 2001. 10.1007/3-540-44586-2-2.
24. D. Kravitz. Digital signature algorithm. US patent 5231668, July 1991.
25. Y. X. Li, R. H. Deng, and X. M. Wang. On the equivalence of mceliece's and niederreiter's public-key cryptosystems. *Information Theory, IEEE Transactions on*, 40(1):271 –273, jan 1994.
26. A. May, A. Meurer, and E. Thomae. Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In D. Lee and X. Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 107–124. Springer, 2011.
27. R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *Deep Space Network Progress Report*, 44:114–116, Jan. 1978.
28. R. Misoczki and P. S. L. M. Barreto. Compact McEliece keys from Goppa codes. In *Selected Areas in Cryptography*, pages 376–392, 2009.
29. C. Monico, J. Rosenthal, and A. Shokrollahi. Using low density parity check codes in the McEliece cryptosystem. In *IEEE International Symposium on Information Theory – ISIT'2000*, page 215, Sorrento, Italy, 2000. IEEE.
30. H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.

31. A. Otmani, J. Tillich, and L. Dallot. Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes. *Special Issues of Mathematics in Computer Science*, 3(2):129–140, Jan. 2010.
32. S. Ouzan and Y. Be'ery. Moderate-density parity-check codes. *CoRR*, abs/0911.3262, 2009.
33. E. Prange. The use of information sets in decoding cyclic codes. *Information Theory, IRE Transactions on*, 8(5):5–9, september 1962.
34. T. Richardson and R. Urbanke. *Modern Coding Theory.* Cambridge University Press, 2008.
35. R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.
36. N. Sendrier. On the use of structured codes in code based cryptography. In S. Nikova, B. Preneel, and L. Storme, editors, *Coding Theory and Cryptography III*, Contactforum, pages 59–68. Koninklijke Vlaamse Academie van België voor Wetenschaeppen en Kunsten, 2009.
37. N. Sendrier. Decoding one out of many. In B.-Y. Yang, editor, *Post-Quantum Cryptography*, volume 7071 of *Lecture Notes in Computer Science*, pages 51–67. Springer Berlin / Heidelberg, 2011. 10.1007/978-3-642-25405-5-4.
38. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.
39. J. Stern. A method for finding codewords of small weight. In G. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 1989.

A Computing the threshold for the Bit-Flipping algorithm

A way for estimating this threshold is considering the probability of a bit be in error after a given number of iterations of the decoding algorithm. When such probability converges to zero, reliable error correction can be achieved. Below we discuss the weak bound presented in [21] based on this probability.

We denote by P_i the probability of a bit be in error after i iterations of the decoding algorithm. When we assume that the code length is infinite and that there are no cycles of length less than or equal to $2i$ in the Tanner graph associated to the parity-check matrix, this probability does not depend on a particular position [34]. These conditions can be somehow relaxed and a finite analysis of the decoding process can be obtained [34], but this is beyond the scope of this paper (furthermore, these values might also be refined through exhaustive simulation).

We denote by H the parity-check matrix of an (n, r, w)-MDPC code. Suppose we are verifying the convergence of P_i when messages containing t errors are received (thus $P_0 = \frac{t}{n}$). To describe how p_i evolves, we have to introduce some additional notation. Let m be the total number of entries equal to 1 in H. Let m_i be the total number of entries of H which are equal to 1 which appear in a column of weight i and define $\lambda_i \overset{\text{def}}{=} \frac{m_i}{m}$. Notice that m_i is also equal to i times the number of columns of weight i in H. In the quasi-cyclic case, note that $m = rw$ and $m_i = \sum_{j=0}^{n_0-1} w_j^2 \mathbf{1}_{w_j=i}$, where $\mathbf{1}_{w_j=i}$ stands for the indicator of the event $w_j = i$ (i.e. it is equal to 1 if $w_j = i$ and 0 otherwise). With this notation we have

$$\begin{aligned} p_{i+1} = p_0 - p_0 \sum_d \lambda_d \sum_{l=b_d}^{d-1} \binom{d-1}{l} \left[\frac{1+(1-2p_i)^{w-1}}{2}\right]^l \left[\frac{1-(1-2p_i)^{w-1}}{2}\right]^{d-l-1} \\ +(1-p_0) \sum_d \lambda_d \sum_{l=b_d}^{d-1} \binom{d-1}{l} \left[\frac{1-(1-2p_i)^{w-1}}{2}\right]^l \left[\frac{1+(1-2p_i)^{w-1}}{2}\right]^{d-l-1} \end{aligned}$$

The integer b_d is chosen as an integer between $d-1$ and $d/2$ which aims at minimizing the function p_{i+1}. Similarly to the case of a constant column weight equal to d which is treated in [21] we choose it as the smallest integer for which the following expression holds:

$$\frac{1-p_0}{p_0} \leq \left[\frac{1+(1-2p_i)^{w-1}}{1-(1-2p_i)^{w-1}}\right]^{2b_d-d+1}$$

Therefore the *threshold* of an (n, r, w)-MDPC code for the Bit-Flipping algorithm is the maximal integer t such that $p_0 = t/n$ and p_i converges to 0.

B Equivalence of Various Coding Problems

B.1 Codeword Existence and Codeword Finding

Let $\mathcal{G}_{n,k}$ denote a subset of $\mathbb{F}_2^{k\times n}$ composed of full rank matrices, a matrix $G \in \mathcal{G}_{n,k}$ is the generator matrix of some binary linear code $\mathcal{C}$ of length n and dimension k. For any $1 \leq i \leq n$, we denote $\mathcal{C}_i$ the code shortened in i, that is

$$\mathcal{C}_i = \{c = (c_1, \dots, c_n) \in \mathcal{C} \mid c_i = 0\}.$$

We will denote by G_i a generator matrix of $\mathcal{C}_i$. We wish to prove that the following two problems are equivalent.

Problem 3 (Codeword existence problem).
Parameters: $\mathcal{G}_{n,k}$, an integer $w > 0$.
Instance: a matrix $G \in \mathcal{G}_{n,k}$.
Question: is there a codeword of weight w in the code of generator matrix G?

Problem 4 (Codeword finding problem).
Parameters: $\mathcal{G}_{n,k}$, an integer $w > 0$.
Instance: a matrix $G \in \mathcal{G}_{n,k}$.
Problem: find a codeword of weight w in the code of generator matrix G.

Proof. (Sketch) We assume we have a solution to Problem 3, that is a program $\mathcal{E} : \mathcal{G}_{n,k} \to \{0, 1\}$ such that $\mathcal{E}(G) = 1$ if and only if there exists a word of weight w in the code spanned by G. The following program called on input G such that $\mathcal{E}(G) = 1$

```
$\mathcal{A}$: input $G \in \mathcal{G}_{n,k}$
  for $i$ from 1 to $n$ while $G$ has a rank $> 1$
    if $\mathcal{E}(G_i) = 1$ then $G \leftarrow G_i$        // false at most $w$ times
  return the first row of $G$
```

will return a word of weight w in the code spanned by G. It calls the program $\mathcal{E}$ at most n times. Conversely a solution to Problem 4 obviously provides a solution to Problem 3.

B.2 Codeword Finding and Syndrome Decoding

We switch to parity check matrices. Let $\mathcal{H}_{n,r}$ denote a subset of $\mathbb{F}_2^{r\times n}$ composed of full rank matrices, a matrix $H \in \mathcal{H}_{n,r}$ is the parity check matrix of some binary linear code $\mathcal{C}$ of length n and dimension $k = n - r$. We rewrite the "Codeword Finding Problem" in this setting.

Problem 4 (Codeword finding problem).
Parameters: $\mathcal{H}_{n,r}$, an integer $w > 0$.
Instance: a matrix $H \in \mathcal{H}_{n,r}$.
Problem: find a codeword of weight w in the code of parity check matrix H.

We recall the "Syndrome Deoding Problem".

Problem 2 (Computational syndrome decoding problem).
Parameters: $\mathcal{H}_{n,r}$, an integer $w > 0$.
Instance: a matrix $H \in \mathcal{H}_{n,r}$ and a vector $s \in \mathbb{F}_2^r$.
Problem: find a vector $e \in \mathcal{S}_n(0, w)$ such that $eH^{\mathrm{T}} = s$.

We claim that those two problems are equivalent up to a polynomial factor.

Proof. (Sketch)

1. Let us assume that we have a program $\mathcal{B}$ which solves the Problem 4 for parameters $(\mathcal{H}_{n+1,r}, w + 1)$, we define the following program

```
$\mathcal{A}$: input $H \in \mathcal{H}_{n,r}$, $s \in \mathbb{F}_2^r$
  $H' \leftarrow (H \mid s)$      // $s$ serves as $n+1$-th row of $H'$
  $e \leftarrow \mathcal{B}(H')$      // $e = (e_1, \dots, e_n, e_{n+1})$
  if $e_{n+1} = 1$ then return $(e_1, \dots, e_n)$ else FAIL
```

If $w+1$ is smaller than the minimum distance of the code of parity check matrix H, the call $\mathcal{A}(H)$ will never fail. This provides a solution to Problem 2 with parameters $(\mathcal{H}_{n,r}, w)$.

2. Conversely, let us assume that we have a program $\mathcal{A}$ which solves the Problem 2 for parameters $(\mathcal{H}_{n,r+1}, w)$

```
$\mathcal{B}$: input $H \in \mathcal{H}_{n,r}$
  $(g_1, \dots, g_k) \leftarrow$ a basis of $\mathcal{C}$ // where $\mathcal{C}$ is the code of parity check matrix $H$
  for $j$ from 1 to $n$
    $H' \leftarrow$ parity check matrix of $\bigoplus_{i \neq j} \langle g_i \rangle$      // subcode of $\mathcal{C}$ without $g_j$
    if $\mathcal{A}(H', g_j H'^{\mathrm{T}}) \neq$ FAIL then
      $z \leftarrow \mathcal{A}(H', g_j H'^{\mathrm{T}})$
      return $z + g_j$
  FAIL      // $\mathcal{A}$ fails to decode for all $j$
```

If there exists a codeword of weight w, the decoder $\mathcal{A}$ will succeed for at least one value of j. The above program provide a solution to Problem 4 for parameters $(\mathcal{H}_{n,r}, w)$.

□

C Computing the work-factor of the $1+1=0$ ISD variant [6].

Let $H \in \mathbb{F}_2^{r \times n}$, $s \in \mathbb{F}_2^r$ and $k = n - r$. We are interested in finding a vector $e \in \mathbb{F}_2^n$ of weight w such that $He^T = s$. Equivalently we want to find a linear combination of w columns of H which when added to s gives a 0-vector. Below we briefly describe the algorithm proposed in [6] for solving this problem. The algorithm is divided in two steps: the setup and the search step. The former consists in randomly permute the columns of H and proceed with a partial Gaussian elimination on the rows of H. More precisely, let l be an optimal algorithm parameter, we compute the matrix:

$$H' = \left[\begin{array}{c|c} \begin{matrix} I^{(r-l)\times(r-l)} \\ 0^{l\times(r-l)} \end{matrix} & Q^{r\times(k+l)} \end{array} \right]$$

where I stands to an identity block and 0 to a zero block. The second step depends on the algorithm parameter $p < w$. The value p defines the error pattern of the sought error vector. Then we will looking for vectors of weight $w-p$ in the first $r-l$ positions and p in the last $k+l$ positions. A valid strategy for finding solutions is: compute all possible linear combinations of p columns in Q and select those one which sums up to a vector coinciding in the last l positions with the the syndrome. We have found a solution when the sum of such combination plus the syndrome gives a vector of weight $w-p$. Note that the sum of each combination plus the syndrome gives a vector of weight 0 in the last l positions. Thus the weight of each combination plus the syndrome will be concentrated in the first $r-l$ positions. When this part has weight exactly $w-p$, we can add the $w-p$ columns from the identity part of H' which erase these positions. In summary, we have selected $w-p$ columns from the first $r-l$ columns of H' plus p columns from the last $k+l$ columns of H', therefore it is a solution.

An improvement is achieved using a Meet-In-The-Middle strategy. It is convenient to compute two lists $\mathcal{L}_1$, $\mathcal{L}_2$ of all possible linear combinations of $p/2$ columns in Q, instead of computing all possible linear combinations of p columns in Q, taking advantage from the Birthday Paradox. Then we select the sums $\{a + b | a \in \mathcal{L}_1, b \in \mathcal{L}_2\}$ which have weight exactly p. Note that the fact of $\mathcal{L}_1$ and $\mathcal{L}_2$ be not disjoint might lead to multiple representations of the same solution. The main improvement presented in [6] is that they allow elements in $\mathcal{L}_1$ and $\mathcal{L}_2$ of weight $p/2 + \epsilon$, for some small integer ϵ. This generalizes the previous approaches. Basically they are considering also the case when ϵ positions of a are erased by ϵ positions of b (i.e. $1+1=0$ for binary codes), which still gives a sum of weight p. Actually, the authors propose to apply this strategy not only once. This leads to an algorithm which can be divided in 4 layers, we label it from 3 (the initial) until 0 (the final layer). The third layer has 4 pairs of two disjoint lists each one. The second layer has two pairs of lists. The first layer has one pair and the layer 0 has the final list. Next we describe the algorithm along with the cost for each step.

Let p, l, p_1, p_2, ϵ_1, ϵ_2, r_1, r_2 be optimal algorithm parameters such that: $p_1 = p/2 + \epsilon_1$, $p_2 = p_1/2 + \epsilon_2$ and $l > r_1 > r_2$. In the third and initial layer, we produce 4 pairs of 2 disjoint lists each one. Each list has the linear combination of $p_2/2$ columns of Q. Thus the size of each list is: $S_3 = \binom{(k+l)/2}{p_2/2}$. We develop the discussion for a pair of lists $\mathcal{L}_{3,1}$ and $\mathcal{L}_{3,2}$, but the same apply for the other three pairs.

For the second layer, we select all sums $\{a + b \mid a \in \mathcal{L}_{3,1},\ b \in \mathcal{L}_{3,2}\}$ of weight $p_2 = p_1/2 + \epsilon_2$ and which coincide with the syndrome in the last r_2 positions. Thus the size of each list is: $S_2 = \frac{(S_3)^2}{2^{r_2}}$. Let the result be $\mathcal{L}_{2,1}$ and consider $\mathcal{L}_{2,2}$ be the merge from another pair in the third layer.

For the first layer, we select all sums $\{a + b | a \in \mathcal{L}_{2,1}, b \in \mathcal{L}_{2,2}\}$ of weight $p_1 = p/2 + \epsilon_1$ and which coincide with the syndrome in the last r_1 positions. Since all elements already coincide in the last r_2 positions, and $r_1 > r_2$, we have to discard only $2^{r_1 - r_2}$ from all possibilities obtained from $\mathcal{L}_{2,1} \times \mathcal{L}_{2,2}$. Thus the cost of merging these lists is $C_2 = \frac{(S_2)^2}{2^{r_1 - r_2}}$. Since $\mathcal{L}_{2,1}$ and $\mathcal{L}_{2,2}$ are not disjoint, we can obtain multiple representations of the same partial solution. We should proceed with only one representation for each solution. The rate of distinct solutions can be measured:

$$\mu_2 = \frac{\binom{k+l}{\epsilon_2}\binom{k+l-\epsilon_2}{p_2-\epsilon_2}\binom{k+l-p_2}{p_2-\epsilon_2}}{\binom{k+l}{p_2}^2}$$

The maximal size of this list is $S_1^{max} = \frac{\binom{k+l}{p_1}}{2^{r_1}}$. Thus the size of the list of distinct solutions is $S_1 = \min\left(\mu_2 C_2, S_1^{max}\right)$. Let the result be $\mathcal{L}_{1,1}$ and consider $\mathcal{L}_{1,2}$ be the result from the other pair in the second layer. Finally, we select all sums $\{a + b | a \in \mathcal{L}_{1,1}, b \in \mathcal{L}_{1,2}\}$ of weight p and which coincide with the syndrome in the last l positions. Since all elements already coincide in the last r_1 positions, and $l > r_1$, we have to discard only 2^{l-r_1} from all possibilities obtained from $\mathcal{L}_{1,1} \times \mathcal{L}_{1,2}$. Thus the cost of merging these lists is $C_1 = \frac{(S_1)^2}{2^{l-r_1}}$. Again, since $\mathcal{L}_{1,1}$ and $\mathcal{L}_{1,2}$ are not disjoint, we can obtain multiple representations of the same solution. We should consider only one representation for each solution. The rate of distinct solutions can be measured:

$$\mu_1 = \frac{\binom{k+l}{\epsilon_1}\binom{k+l-\epsilon_1}{p_1-\epsilon_1}\binom{k+l-p_1}{p_1-\epsilon_1}}{\binom{k+l}{p_1}^2}$$

The maximal size of the final list is $S_0^{max} = \frac{\binom{k+l}{p}}{2^l}$. Thus the size of the final list of distinct solutions is $S_0 = \min\left(\mu_1 C_1, S_0^{max}\right)$. Considering the cost for the Gaussian elimination as $K_0 = \frac{(n+1)(n-k)}{\log_2(n+1)}$ [1] and the cost of merging two lists being twice the cost of building a list (we use coefficients $K_1 = 1$ and $K_2 = 2$ to make this adjustment), the cost of each iteration (an attempt of the algorithm in finding a solution) is:

$$WF^{\text{iteration}}(n, r, w, p, l, r_1, r_2, \epsilon_1, \epsilon_2, p_1, p_2) = K_0 + 8S_3K_1 + 4C_3K_2 + 2C_2K_2 + C_1K_2$$

The number of iterations that the algorithm must perform until find a solution depends on the probability of finding an error vector with the sought error pattern: vectors of weight $w - p$ in the first $r - l$ positions and p in the last $k + l$ positions. This probability is

$$P(n, r, w, p, l, r_1, r_2, \epsilon_1, \epsilon_2, p_1, p_2) = \frac{\binom{n-k-l}{w-p}\binom{k+l}{p}\frac{S_0}{S_0^{max}}}{\binom{n}{w}} = \frac{\binom{n-k-l}{w-p}S_0 2^l}{\binom{n}{w}}$$

Therefore our estimation of cost for the $1 + 1 = 0$ *ISD variant* [6] given l, p, r_1, r_2, ϵ_1, ϵ_2, p_1, p_2 is:

$$\begin{aligned} WF(n, r, w, p, l, r_1, r_2, \epsilon_1, \epsilon_2, p_1, p_2) &= P^{-1} \cdot WF^{\text{iteration}}(n, k, w, p, l, r_1, r_2, \epsilon_1, \epsilon_2, p_1, p_2) \\ &= P^{-1}(K_0 + 8S_3K_1 + 4C_3K_2 + 2C_2K_2 + C_1K_2). \end{aligned} \tag{1}$$

There are several ways for choosing the parameters l, p, r_1, r_2, ϵ_1, ϵ_2, p_1, p_2. With some heuristic approaches, we succeeded to find good parameters, providing lower work-factors than what is obtained for the other ISD variants. However for the parameters presented in Section 6 these values are still quite close from what is obtained for much simpler ISD variants.