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. Recently, several variants of the McEliece cryptosystem based on low-density parity-check (LDPC) codes have been proposed [1,2,3,4,5]. When combined with quasi-cyclic structure, these proposals provide much smaller key sizes than the original McEliece cryptosystem. LDPC codes are characterized by the existence of low weight dual codewords, used to perform an efficient iterative decoding. In order to avoid attacks aimed at recovering such codewords, these last proposals suggested to replace the permutation matrix used by McEliece by a matrix of small constant row and column weight, in order to increase the dual codeword weight. In this paper, we introduce the moderate density parity-check codes (MPDC, for short), which provide a better decoding process than the aforementioned LDPC variants. It also recovers the possibility to use permutation equivalent private and public codes. As a result, we present two new McEliece variants (one using quasi-cyclic MDPC codes and other employing generic MDPC codes). One of the main benefits of our variants is that both key-recovery and message decoding attacks boil down to the same coding-theory problem: low weight codeword finding. Therefore we present a security reduction much closer to the general decoding problem than any other code-based encryption scheme. Regarding each variant separately, while the QC-MDPC variant is mainly focused on allowing smaller public keys (e.g., for 80-bits of security, only 4800 bits), the MDPC variant further reduces the ways for structural attacks. Finally, we evaluate several kind of attacks, resulting in practical parameters quite competitive to conventional cryptography.

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

1 Introduction

Code-Based Cryptography. Code-based cryptography has been attracting growing interest since the work of Peter Shor [6], who 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 [7] for an extensive report). This threatens most if not all public-key cryptosystems deployed in practice such as RSA [8] or DSA [9]. 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 cryptosystem whose security relies on the intractability of certain computational problems in coding theory is the McEliece cryptosystem [10], originally proposed by Robert McEliece in 1978 using Goppa codes. More precisely, its security is based on two assumptions, the pseudo-randomness of the Goppa code family and the hardness of decoding a generic linear code. It is namely proved in [11] that if an attacker 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 [12]. However in [13] a distinguisher for Goppa codes of high rate (like those originally suggested for CFS signature [11] and for some realistic secure parameters of McEliece cryptosystem) is presented. Although this fact does not represent an effective attack, it would be appropriate to use other families of codes, in order to ensure the completeness of such security reduction.

Besides, 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 [14]. It has been followed by several other proposals such as [15,16]. The structural algebraic attack proposed in [17] succeeds to break many of them (the binary case for quasi-dyadic Goppa codes [15] was not affected). 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 though, leading to its ineffectiveness against the dyadic scheme based on binary Goppa codes proposed in [15], for instance. It is also likely that this kind of cryptanalysis can be prevented for all these schemes by choosing more conservative parameters. However, it might be desirable to avoid this kind of algebraic attacks by suggesting other code families which would thwart completely this approach.

Cryptographic Schemes Based on LDPC Codes. Low-Density Parity Check (LDPC) codes would be a good candidate for achieving such a goal. These are just codes with a sparse parity-check matrix. They have a very efficient iterative decoding algorithm suggested by Gallager in [18] which makes use of the sparsity of the parity-check matrix and have no algebraic structure which could be used in an algebraic cryptanalysis. They correct a very large fraction of errors with a low complexity decoding algorithm, getting extremely close to the Shannon limit. It has been repeatedly suggested to replace the Goppa codes by LDPC codes in the McEliece scheme. Unfortunately, it is a folklore result that it is easy to recover its sparse parity-check matrix by looking for low weight codewords in the dual code. This in turn can be used for decoding the public code even without knowing the secret key. A way of avoiding attacks of this kind was suggested in [4] and consisted in replacing the permutation matrix in the McEliece cryptosystem by a sparse invertible matrix Q of some small constant row and column weight m. The public code is not an LDPC code anymore because the row weights of the secret LDPC code are multiplied by this factor of m and generic algorithms for finding low weight codewords become useless in this case. Moreover, the secret LDPC code, the matrix Q and a scrambling matrix S (also needed for the cryptosystem) can be chosen to be quasi-cyclic, reducing its key sizes and encoding complexity. However, the unfortunate choices of Q and S of [4] allowed to cryptanalyze successfully the scheme in [19]. A more general way of choosing Q and S was suggested in the variants proposed in [5,20]. It seems to be immune against the attack suggested in [19] and displays similar reductions of the key sizes. For instance whereas for 80 bits of security, public key sizes of about 600000 bits are suggested in [21] in a McEliece scheme relying on Goppa codes, this can be reduced to about 49000 bits with the scheme of [5] relying on quasi-cyclic LDPC codes.

Our Contribution. Our first observation in this paper is that the attack on the McEliece variant based on LDPC codes mentioned in [1] can be avoided by moving from LDPC codes to MDPC codes (standing for Moderate Density Parity Check codes). Roughly speaking, a MDPC code is nothing but a code which admits a parity-check matrix which is moderately sparse (say that its rows contain each only several hundreds of “1”s). Such codes correct much less errors than LDPC codes when used with Gallager's decoding algorithm but there is a way to choose the parameters of the MDPC code such that both key recovery and message recovery attacks stay infeasible. The number of errors and the length of the code can indeed be chosen to be sufficiently large so that decoding algorithms for generic linear codes become too complex whereas the sufficiently large row weight of the secret parity-check matrix does not allow low weight codeword finding algorithms to be successful either. Compared to the aforementioned work [5,20], for which there is right now no security reduction, there is one in our case. More precisely by using the technique of [11], it can be proved that an attacker breaking this new scheme is either able to solve the decoding problem of a generic linear code or is able to distinguish an MDPC code from a random linear code. It is quite natural to consider that the only way of distinguishing an MDPC code from a random code is by being able to find the codewords in the dual of the public code which are of moderate weight (these are precisely the rows of the secret parity-check matrix which is used in the decoding process). If we make such an hypothesis, since algorithms for decoding a random linear code and finding low weight codewords are basically of the same nature and complexity, the security of our scheme only relies on the hardness of decoding a generic linear code. This provides a strong argument in favor of the security of this new scheme.

There is also a quasi-cyclic version of our scheme which reduces significantly the key sizes. We also provide a security assessment by estimating precisely the complexity of the best known decoding algorithms and low weight codeword finding algorithms in the general case and also in the quasi-cyclic case. We carefully take into account the natural applicability of the attack decoding one out of many, presented in [22], which has advantages when multiple instances and solutions of the syndrome decoding problem are available. This is exactly what happens for MDPC codes. We use this analysis to propose several sets of parameters. It should be mentioned that the quasi-cyclic case achieves public keys of only 4800 bits for 80 bits of security against the aforementioned algorithms. This even improves the key sizes obtained by quasi-dyadic Goppa codes [15]. The MDPC structure further reduces the venues for mounting structural attacks. Regarding the LDPC McEliece variant [5], our scheme also compares quite favorably for instance in these observations:

- It is not clear whether or not the more complex structure of their scheme really boils down to decoding a generic quasi-cyclic code or to finding low weight codewords in a quasi-cyclic code.
- The public key size that we obtain in our case are significantly smaller for a same security. This is due to several reasons. A first reduction of the key size comes from the fact that our public key can be chosen to be systematic. A second reduction comes from the fact that the way the public code of [5] is constructed and decoded leads actually to an MDPC code which has a worse iterative decoding process than in our case. This means that we can purposely add more errors than those proposals, or equivalently we can use a shorter length code to correct the same amount of errors. This is explained in more detail in Section 3.
- The private key size is reduced, since the transformation matrix Q does not exist anymore.

Organization. In Section 2 we describe basic concepts and the previous McEliece variant based on LDPC codes [5]. In Section 3 we describe the new variants. In Section 4 we assess their security, giving a security reduction. In Section 5 we discuss about their practical security, resulting in practical parameters. We conclude in Section 6.

2 Basic Concepts

In this section, we review some basic concepts from coding theory. All the codes, vectors, matrices used in this paper are binary.

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 $(n - r)$ and codimension r, is an $(n - r)$-dimensional vector subspace of $\mathbb{F}_2^n$. It is spanned by the rows of a matrix $G \in \mathbb{F}_2^{(n-r)\times n}$, 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* minimal distance *of a linear code is the smallest weight of a non-zero codeword. The* dual *$\mathcal{C}^\perp$ of $\mathcal{C}$ is the linear code spanned by the rows of any parity-check matrix of $\mathcal{C}$. The* dual distance *of a code is the minimal distance of its dual.*

Definition 3 (Gilbert-Varshamov distance). *The Gilbert-Varshamov distance $d_0(n, r)$ (when there is no ambiguity, simply d_0) is defined as the largest integer such that*

$$\sum_{i=0}^{d_0-1} \binom{n}{i} \leq 2^r.$$

Definition 4 (Low-density parity-check code). *An (n, r, w)-low-density parity-check (LDPC) code is a linear code of length n and codimension r which admits a parity check matrix whose rows have weight w much smaller than the dual distance expected for a (n, r)-linear code.*

Definition 5 (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_0 divides the length n of the code, it is possible (and convenient) to reorder the positions of a quasi-cyclic code of length n in such a way that it has a generator or parity check matrix[3] which is formed by $p \times p$ circulant blocks, where $p = n/n_0$. We say in this case that the quasi-cyclic code is of order p. Note that a circulant block is completely described by its first row (or column). We are interested in (n, r, w)-QC-LDPC codes, where $n = n_0 p$, $r = p$, for some integers n_0 and p. The parity-check matrix which is chosen to be quasi-cyclic of order p takes the form

$$H = [H_0|H_1| \dots |H_{n_0-1}]$$

Each block H_i is a $p \times p$ circulant matrix with row/column weight w_i, such that $w = \sum_{i=0}^{n_0-1} w_i$:

$$H_i = \begin{bmatrix} h_0 & h_1 & h_2 \dots & h_{p-1} \\ h_{p-1} & h_0 & h_1 \dots & h_{p-2} \\ \vdots & \vdots & \vdots \ddots & \vdots \\ h_1 & h_2 & h_3 \dots & h_0 \end{bmatrix}$$

The generator matrix G in row reduced echelon form can be derived directly from the H_i's. We assume that H_{n_0-1} is non-singular, implying that w_{n_0-1} has to be odd (otherwise the rows of H_{n_0-1} would add up to 0), and G can be produced as follows:

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

being composed by a $(n - r) \times (n - r)$ identity matrix, followed by a column of $(n_0 - 1)$ binary circulant blocks of size $p \times p$. For the multiplication and inversion of the blocks H_i, recall that the algebra of $p \times p$ binary circulant matrices is isomorphic to the algebra of polynomials modulo $x^p - 1$ over $\mathbb{F}_2$, which means that we can do it efficiently. Below we give the description of the proposal presented in [5], in order to compare with our new approach.

The QC-LDPC McEliece Variant [5]. In this scheme, the private code $\mathcal{C}$ and the public code $\mathcal{C}'$ are not permutation equivalent, as in the original proposal of McEliece cryptosystem. The dual code of $\mathcal{C}$ contains low weight codewords, while the dual of $\mathcal{C}'$ does not. This difference avoids the effectiveness of low weight codeword finding algorithms applied on its dual public code. This is achieved through the use of the matrix Q which has row and column weight m, as explained below.

– **Key Generation**:
  1. Generate the sparse parity-check matrix $H \in \mathbb{F}_2^{r \times n}$ of an (n, r, w)-QC-LDPC code $\mathcal{C}$.
  2. Generate an invertible dense block-circulant matrix $S \in \mathbb{F}_2^{(n-r)\times(n-r)}$.
  3. Generate an invertible block-circulant matrix $Q \in \mathbb{F}_2^{n \times n}$ of row and column weight m.
  4. Compute the generator matrix $G \in \mathbb{F}_2^{(n-r)\times n}$ in block-circulant systematic form of $\mathcal{C}$.
  - Private key is the triple (S, H, Q).
  - Public key is $G' = S^{-1} G Q^{-1}$, a generator matrix in $\mathbb{F}_2^{(n-r)\times n}$ of a code $\mathcal{C}'$.

– **Encryption**:

[3] This statement holds by generalizing the notion of a generator matrix and a parity-check matrix a little bit by allowing these matrices not to have full rank: a (generalized) generator matrix is just a matrix whose rows span the code, and a parity-check matrix a matrix whose rows span the dual code.

1. Select randomly $e \in \mathbb{F}_2^n$, such that $wt(e) \leq t'$.
2. Compute: $x = mG' + e$.

- **Decryption**:
  1. Compute: $x' = xQ = mS^{-1}G + eQ$.
  2. Decode $t = mt'$ errors in x', in order to obtain $m' = mS^{-1}$.
  3. Compute $m = m'S = mS^{-1}S$.

We have to stress the strong security dependence of such cryptosystem on the transformation matrix Q. During the decryption step, e multiplies this matrix, propagating the number of errors from t' to $t = mt'$. Next we discuss about such particularity.

3 Moderate Density Parity-Check McEliece Variants

Our proposal comes from simple but not trivial observations over the cryptosystem proposed in [5]:

1. The private code $\mathcal{C}$ has an error correction capability very close to mt', whilst the public code $\mathcal{C}'$ has an error correction capability much higher than t'.
2. When increasing t' by λ errors, this aforementioned scheme must provide a private code able to correct $m\lambda$ more errors, due to the propagation of the errors performed by the matrix Q.

These two facts imply that we have much more freedom to increase t' (the number of errors added during the encryption process) using a code with higher density, as exemplified through the public code of this scheme. Furthermore, using a reasonable denser code, its dual will not contain low weight codewords that allow key-recovery attacks. The Figure 1 summarizes this situation for the private code $\mathcal{C}$ of parity-check matrix H of parameters: $p = 4096$, $n_0 = 3$, $w = 39$, $m = 7$ (as suggested in [20] for 80-bits of security) and its respective public code $\mathcal{C}'$ which admits a valid parity-check matrix $H' = HQ$. In order to compare these codes, we use threshold values from where reliable LDPC decoding can be expected. In Appendix A, a way to compute them is explained.

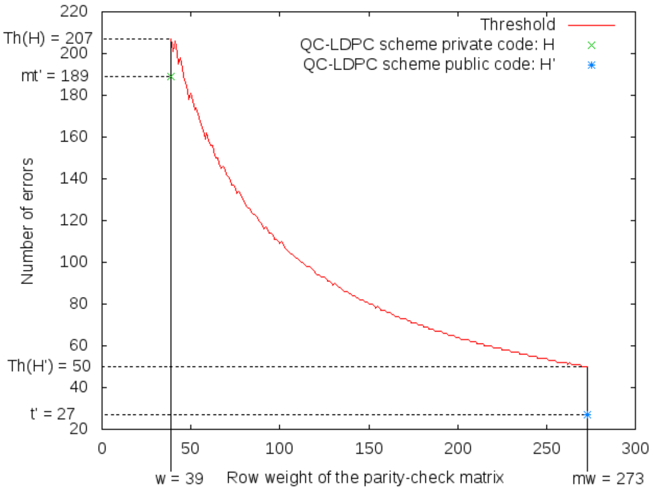

Fig. 1. Threshold for different row weights of the parity-check matrix

In this example, the parity-check matrix of the private code has row weight $w = 39$ and its threshold for error correction capability is estimated in 207 errors. In order to use the private code for decoding, we must have $mt' \leq \text{Th(H)}$. Suppose we want to increase the number of errors t' by λ. For this parameter set, in order to have $m(t' + \lambda) \leq \text{Th(H)}$, we can choose $0 < \lambda \leq 2$. On the other hand, using the code $\mathcal{C}'$ for decoding, we must have $t' \leq \text{Th(H')}$. Therefore, if we want to increase the number of errors t' by λ, we would be able to choose $0 < \lambda \leq 23$ without invalidate: $t' + \lambda \leq 50$. In summary, the use of denser LDPC codes allows for bigger values of errors added during the encryption step than the aforementioned scheme.

Therefore in this work we replace the LDPC codes (which have been disguised by the matrix Q) by the simpler use of a moderate density parity-check code. This approach ensures a better decoding procedure and for well chosen parameters resists to all kind of attacks against code-based cryptosystems. Basically, this moderate density must be high enough to avoid key recovery attacks, while still allows decoding for a secure amount of errors. Achieving these requirements, besides providing a better decoding process, our proposal reduces the private key size, discards the multiplication by Q during the decryption process, moves the underlying problem much closer to the general decoding problem than any other code-based cryptosystem (this feature is discussed in Section 4), recovers the completeness of the McEliece security reduction (when compared with high rate Goppa codes) and, for the quasi-cyclic case, it also obtains the smallest public-keys for cryptosystems based on binary codes (e.g., for 80-bits of security, only 4800 bits).

3.1 Scheme description

The difference between the MDPC and the QC-MDPC variants only lies in the family where the code is sampled from. Therefore the encryption and decryption processes are always the same and the differences regarding the code generation are described below.

Definition 6 (MDPC/QC-MDPC code). *An (n, r, w)-MDPC code is a linear code of length n and codimension r which admits a parity check matrix whose rows have weight w. If in addition this code is quasi-cyclic, we will speak of an (n, r, w)-QC-MDPC code.*

(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. For the quasi-cyclic case, we use basically the same construction as explained in Section 2, with a few relaxed constraints in order to support our security reduction presented in Section 4. Basically, we pick one random word of length $n = n_0 p$ (using the previous notation) and weight w. This word will form the first row of an $r \times n$ matrix (with $r = p$) formed by n_0 circulant $p \times p$ blocks.

General remarks about the code construction. For the quasi-cyclic case, unlike [2], where a construction of QC-LDPC codes based on random difference families is suggested, we follow here a more general route. Our QC-LDPC codes have a row and column weight much higher (they are MDPC codes rather than LDPC codes) and for our specific parameter sets (presented in Section 5) we can afford to have some cycles of length 4 in the associated Tanner graph without strongly compromising the iterative decoding performance. This means that with a random construction we achieve an enough practical error correcting capability. Besides, we do not require as in the last proposals each one of the n_0 blocks to have an identical row (and column) weight, although values quite close to w/n_0 are mostly expected on average. Indeed, applications that check the uniform weight distribution among the blocks (i.e. ensure the weight w splits equally in n_0 sets) can be more convenient in practice, but in order to hold our security reduction we leave it as general as possible. A special attention should be given to the last block H_{n_0-1}, which must have its first row with odd weight, in order to be invertible. If it is not invertible, we swap positions or pick a new word, carrying on this process until we find an invertible H_{n_0-1}. It is important to stress that these remarks do not affect the generality of our code generation on average, a necessary condition to support our security reduction presented in Section 4.

Key generation: Let $\mathcal{F}_{n,r,w}$ be a family of t-error correcting (n, r, w)-MDPC or (n, r, w)-QC-MDPC codes.

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

- Public key: G.
- Private key: H.

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

1. Select randomly $e \in \mathbb{F}_2^n$, such that $wt(e) \leq t$.
2. Compute $x = mG + e$.

Decryption: To decrypt $x \in \mathbb{F}_2^n$ into $m \in \mathbb{F}_2^{(n-r)}$:

1. Using H, apply an LDPC decoding algorithm to $mG + e$ in order to retrieve mG.
2. Extract the plaintext m from the first $(n - r)$ indices of mG.

In both cases (MDPC and QC-MDPC) we generate a sparse generator matrix whose rightmost block is invertible, allowing a public key in row reduced echelon form. As we can see, there is no matrix Q and the difference between the private description of the code H and the public row reduced echelon form G lies in their densities. While the private description of H is sparse with row weight w, its public description G is formed by a $(n-r) \times (n-r)$ identity block followed by a dense block of size $(n-r) \times r$. It is quite intuitive to think in the low weight codeword finding problem in order to attack our scheme. This fact is well evaluated in Sections 4 and 5. Note that the *dual* version of the McEliece cryptosystem, the Niederreiter cryptosystem, can also be used. It features the same security reduction, and thus the choice only depends on the user's requirements. However, with MDPC codes the error capability is relatively small and the message expansion between the cleartext and the ciphertext can be relatively high. For the parameters given in Section 5 and for the Niederreiter scheme, the expansion factor runs from 7 (for 80 bits of security) to 14 (for 256 bits of security).

The Importance of a CCA2-Secure Conversion. As pointed out in [23], the original McEliece cryptosystem is malleable and susceptible to *adaptative chosen ciphertext attacks* (CCA2 for short). A mere resend attack, as described in [24], presents a real threat. A *CCA2*-secure conversion, like [25], avoid such attacks, through a suitable usage of hash functions and random sequences, ensuring the indistinguishability of the encrypted messages. Besides the message resend attack resistance, this property impacts over two different aspects of our proposal (and over [4] and [5], where this fact was not taken into account): its key size and how to deal with decoding failures. The public key size for the [4] QC-LDPC Mceliece variant, and subsequently in [5], were overestimated. The authors demand to store the whole public generator matrix G. However, as pointed out in [21], if a *CCA2*-secure conversion is used there is no threat in having G in systematic form, storing only its nontrivial part. Besides the systematic form, when using QC-MDPC codes, it is also possible to reduce the key size by a factor of p, since each block $p \times p$ of G can be completely described by its first row alone due to its cyclicity. Regarding decoding issues, LDPC decoding algorithms are probabilistic and assume a (quite low) decoding failure rate. However, we must be able to deal with this situation without expose the cryptosystem to any flaw. A simple and naive approach to address such issue is to request a new encryption for messages with decoding failure. However, as exemplified by the resend attack, this approach can not be used standalone and a *CCA2*-secure conversion must be applied together.

4 Security Reduction

In [26], a reductional security proof for the Niederreiter cryptosystem [27] is explained. It holds for the McEliece cryptosystem as well, since their security are equivalent [28]. It mainly depends on two assumptions regarding the difficulty in distinguishing and decoding a linear code, both in the average case. Using the same approach, we present a security reduction of our MDPC/QC-MDPC variants, which is much closer to the general decoding problem than any other code-based cryptosystem. For a matter of simplicity we argue regarding the Niederreiter cryptosystem.

Notations:

$\mathcal{F}_{n,r,w}$: a family of (n, r, w)-MDPC or (n, r, w)-QC-MDPC codes.
$\mathcal{K}_{n,r,w}$: the public key space of $\mathcal{F}_{n,r,w}$.
$\mathcal{H}_{n,r}$: the apparent key space:
MDPC case: $\{H \mid H \in \mathbb{F}_2^{r\times n}$ of full rank$\}$.
QC-MDPC case: $\{H \mid H \in \mathbb{F}_2^{r\times n}$ of full rank, H is block-circulant$\}$.
$\mathcal{S}_n(0, t)$: the sphere centered in zero of radius t of $\mathbb{F}_2^n$.
Ω : the uniformly distributed sample space $\mathcal{H}_{n,r} \times \mathcal{S}_n(0, t)$.

Note that $\mathcal{K}_{n,r,w} \subset \mathcal{H}_{n,r}$. For the sake of simplicity, when there is no ambiguity we omit the parameters in $\mathcal{F}_{n,r,w}$, $\mathcal{K}_{n,r,w}$, $\mathcal{H}_{n,r}$ and denote it simply by $\mathcal{F}$, $\mathcal{K}$, $\mathcal{H}$.

4.1 Underlying problems

The main idea in proposing a security reduction is to evaluate what the two possibilities to break a cryptosystem (in the case of an encryption scheme, either decrypt a message or recover its private-key) can imply to well established mathematical problems. Regarding the particular case of code-based cryptosystem, the key-recovery attack could mean recover the private structure of the code, but more conservatively we can consider that it means *at least* the ability of the attacker in distinguishing the code. Regarding the message attack, it refers to the ability of correcting errors in a linear code.

Definition 7 (Code distinguishing problem).

Parameters: $\mathcal{K}_{n,r,w}$, $\mathcal{H}_{n,r}$
Instance: a matrix $H \in \mathcal{H}_{n,r}$
Question: is $H \in \mathcal{K}_{n,r,w}$*?*

Definition 8 (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* $eH^T = s$

The decision problem associated to the syndrome decoding problem is NP-complete [12]. NP-completeness only implies worst case complexity while for cryptographic purposes, we would rather need average case complexity.

4.2 Security reduction structure

Programs which are able to solve the code distinguishing problem, the computational syndrome decoding problem and break the Niederreiter cryptosystem can be defined as follows.

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

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

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:

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

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 and:

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

Each one of these programs (distinguisher, decoder and adversary) is efficient if the ratio T/ϵ is small, upper bounded by a polynomial in n. If there is no efficient distinguisher then the set $\mathcal{K}_{n,r,w}$ is *pseudorandom*, since there is no easy way to decide whether or not a matrix in $\mathcal{H}_{n,r}$ is or is not a valid public-key. The existence of an efficient decoder is related to the difficulty of the bounded decoding problem in the average case. The absence of an efficient adversary means that the Niederreiter scheme is difficult to break in average when the error and the key are chosen randomly and uniformly in $\mathcal{S}_n(0,t) \times \mathcal{K}_{n,r,w}$. Below we present the proposition of [26] which ensures the validity of such security reduction.

Proposition 1. *We fix the security parameters (n, r, w) and t and denote by $\mathcal{K}_{n,r,w}$ the public-key space. 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 $\mathcal{K}_{n,r,w}$.*

Proof: Let $\mathcal{A} : \mathcal{H}_{n,r} \times \mathbb{F}_2^r \rightarrow \mathcal{S}_n(0,t)$ 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

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

then

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

which implies

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

We know that $Succ(\mathcal{A}, \mathcal{K}_{n,r,w}) \geq \epsilon$. Therefore we either have $Adv(\mathcal{C}, \mathcal{K}_{n,r,w})$ or $Succ(\mathcal{A})$, which are both positive, greater or equal to $\epsilon/2$. 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 $\mathcal{K}_{n,r,w}$.

□

It exactly means that the system is an one-way encryption scheme when the two following two assumptions hold.

Assumption 1 ($(\mathcal{K}, \mathcal{H})$-indistinguishability) *The code distinguishing problem is hard on average for the parameters $(\mathcal{K}, \mathcal{H})$. In other words, distinguishing a public key in $\mathcal{K}$ from a matrix randomly chosen in $\mathcal{H}$ is difficult on average.*

Assumption 2 ($(\mathcal{H}, t)$-decoding hardness) *The computational syndrome decoding problem is hard on average for parameters $(\mathcal{H}, t)$. That is decoding t errors in a code of parity check matrix $H \in \mathcal{H}$ is difficult on average.*

Since the distinguisher, decoder and adversary programs are valid for both MDPC and QC-MDPC case, thus holding Proposition 1, we therefore need to state the validity of Assumptions 1 and 2 for these codes.

4.3 Hardness of decoding and distinguishing MDPC/QC-MDPC codes in the average case

The Assumption 2 is the general decoding problem, believed to be hard on average based on decades of active research. Therefore, no improvement is expected for MDPC codes, even when it is limited to the quasi-cyclic case.

Regarding Assumption 1, the discussion deserves more attention, particularly after a recent result over the distinguishability of Goppa codes [13], where an efficient distinguisher for high rate Goppa codes is presented. The main idea of such distinguisher boils down to an algebraic technique proposed to attack McEliece variants with compact keys [17], where it is suggested to recover the private code structure by solving an algebraic system. This system can then be transformed into a linearized system which has much more equations than unknowns for high rate Goppa codes. Despite this fact, the solution space of the linearized system is extremely large in the case of Goppa codes whereas it is of dimension 0 for random codes. This allows to distinguish high rate Goppa codes from random codes.

For MDPC codes, in contrast, there is no known distinguisher so far. They are generated randomly, solely restricting the rows of their parity-check matrix to have low weight. Therefore it is reasonable to assume that the only remarkable property useful to distinguish them refers to the existence of a few codewords of low weight in their dual code. The problem of finding codewords with a specific weight can be described generically as follows.

Definition 9 (Codeword weight problem).

Parameters: $\mathcal{H}$ *, an integer* $W > 0$
Instance: *a matrix* $H \in \mathcal{H}$
Problem: *is there a codeword of weight* W *in the code of generator matrix* H*?*

Note that we are searching for dual codewords, i.e. codewords in the linear code of length n and codimension $n - r$ spanned by H. When these code parameters are fixed, the maximal cost to solve such problem is obtained when W is close to the Gilbert-Varshamov distance d_0 and decreases when $W < d_0$ and $W > d_0$. Figure 2, based on [7, page 111], shows this behavior through the cost of information set decoding technique, for fixed length and dimension, varying the sought weight W.

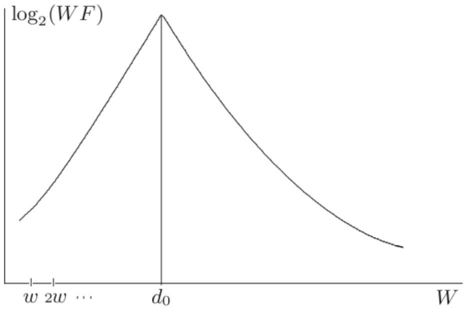

Fig. 2. Information set decoding cost for a fixed length and dimension when the error W varies.

At first glance, we would be interested in codewords of sought weight W equal to the weight w of the rows of the MDPC sparse parity-check matrix (for practical parameters, it is a value smaller than d_0). However someone can argue that finding codewords of weight $W = \lambda w$, for a small integer λ, can be a better approach in order to distinguish MDPC codes. Such approach can be used as a distinguisher because with overwhelming probability those codewords are the result of the addition of λ rows of the sparse parity-check matrix, with the benefit of an increased number of possible codewords: there exist around $\binom{r}{\lambda}$ codewords of weight λw for each codeword of weight w. On the other hand, the difficulty to find low weight codewords of weight $W = \lambda w$, such that $w < \lambda w < d_0$, increases dramatically, as showed in Figure 2. In practice, the increasing on the hardness of finding codewords of weight λw is much more relevant than what is gained due to the existence of more codewords of such weight.

For weight λw greater than the Gilbert-Varshamov distance, i.e $w < d_0 < \lambda w$, this hardness decreases. However this approach cannot be used as a distinguisher, because such codewords are already expected on average, not necessarily being the addition of a few rows of the sparse parity-check matrix. Therefore the best strategy for distinguishing MDPC codes is indeed to find codewords of weight exactly w, the row weight of the sparse parity-check matrix.

Conjecture 1 *1. Distinguishing an (n, r, w)-MDPC code from a random code is not computationally easier than deciding the existence of a codeword of weight w in a code of length n and dimension r.*

2. Distinguishing an (n, r, w)-QC-MDPC code from a random QC code is not computationally easier than deciding the existence of a codeword of weight w in a QC code of length n and dimension r.

The codeword weight problem is old in coding theory and when restricted to low weights is very close to the syndrome decoding. In fact they are considered to be equally difficult, since all known algorithms that solve one of these problems can also solve the other and with the same complexity, as explained next. Let $\mathcal{C}$ be a $(n, n-r)$-linear code of minimum distance d, generator matrix $G \in \mathbb{F}_2^{(n-r)\times n}$ and parity-check matrix $H \in \mathbb{F}_2^{r\times n}$. The minimum distance is usually unknown, but for most binary linear codes it is very close to the Gilbert-Varshamov distance. A message $m \in \mathbb{F}_2^{n-r}$ is encoded and noised by an error vector $e \in \mathcal{S}_n(0, \lfloor d/2 \rfloor)$ resulting in $x \in \mathbb{F}_2^n$, $x = mG + e$. In order to retrieve m from x, we should be able to correct up to $\lfloor d/2 \rfloor$ errors in $\mathcal{C}$. Alternatively, we can use the codeword weight problem. Suppose that we take the code $\mathcal{C}' = \mathcal{C} \oplus e$, which has a generator matrix: $G' = \begin{bmatrix} G \\ e \end{bmatrix}$. If we are able to find minimum weight codewords in $\mathcal{C}'$, then we are able to find e, because $wt(e) \leq d$ and all other codewords of $\mathcal{C}'$ which are linearly independent from e have weight greater than d, considering the minimum distance of $\mathcal{C}$. Therefore decode $e \in \mathcal{S}_n(0, \lfloor d/2 \rfloor)$ in a code $\mathcal{C}$ with minimum distance d can be achieved finding minimum weight codewords in $\mathcal{C}'$. This allows us to assume the validity of the following assumption, which replaces Assumption 1 for our scheme.

Assumption 3 ($(\mathcal{H}, w)$-codeword weight hardness) *The codeword weight problem is hard on average for parameters $(\mathcal{H}, w)$. In other words, deciding the existence of a word of weight w in a code of generator matrix $H \in \mathcal{H}$ is difficult on average.*

This assumption is very close to the decoding hardness assumption and is strongly believed to hold, also in the quasi-cyclic case. Therefore, assuming Conjecture 1 and under Assumptions 2 and 3, we can state that the encryption primitive of the MDPC/QC-MDPC variant of McEliece or Niederreiter is OWE (One Way Encryption) and any desired security level can be achieved with a proper semantically secure conversion.

Security statements:

1. Breaking the MDPC variant of McEliece or Niederreiter is not easier than decoding or deciding the existence of low weight codewords in a random linear code.
2. Breaking the QC-MDPC variant of McEliece or Niederreiter is not easier than decoding or deciding the existence of low weight codewords in a random quasi-cyclic linear code.

5 Practical Security

In this section, we describe 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 (which invalidates the security reduction). Message attacks try to decode one particular message considered as a noisy codeword.

The system is 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.

5.1 Cost Analysis of the Best Known Attacks

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 algorithm is information set decoding (ISD) [29]. In today's state-of-the-art the best variants derive from Stern's collision decoding algorithm [30]. There have been numerous contributions and improvements [31,32,33,34,35] until the recent asymptotic improvements [36,37].

We will denote $\mathrm{WF}_{\mathrm{isd}}(n, r, t)$ 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. In the above notation, we consider the cost of the best variant whichever it may be.

We also mention the *Decoding One Out of Many* setting (DOOM), presented in [22], which measures the worfactor gain when multiple instances are attacked simultaneously and the attacker is content with a solution for a single of those instances. This last attack has to be considered for our parameters since, when using MDPC/QC-MDPC codes, the attacker can indeed has or assumes the existence of multiple instances and solutions.

At first, we discuss about the cost of the variants of collision decoding, which are iterative algorithms. At each iteration a Gaussian elimination is performed on the public key 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 which depends of the particular variant and of various parameters to optimize. Each iteration has a probability P to produce the solution. This probability also depends on the various 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. This complexity analysis, though very crude, will help us in the next paragraph to understand how the workfactor evolves when the number of solutions and instances are greater than 1.

Impact of Multiple Instances and Multiple Solutions. 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 by a factor[4] $\sqrt{N_i}$. In the original proposal of DOOM we have $N = N_i = N_s$ and attacking N instances will gain a factor $N_s/\sqrt{N_i} = \sqrt{N}$. The real gain is in fact slightly smaller (see the detailed analysis in [22]) because the optimal parameters are not the same with multiple instances or with only one.

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). Refering to the previous paragraph, we have $N_s = r$ and $N_i = 1$. 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.

[4] 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 $\sqrt{N_i}$

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 r. 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 normal (*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 $N_i = N_s = r$. Using DOOM we gain a factor $\sqrt{r}$ (at most).

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

	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 DOOM and ISD. The possiblity to exploit multiple instances, say N, to gain a factor of order $\sqrt{N}$ was studied for Dumer's algorithm [31], it probably applies to ball-collision decoding [35]. The very last variants [36,37] involve a more complex tree structure in the algorithm which might decrease the efficiency of DOOM. Since nothing conclusive is known on the subject, we have applied a conservative approach and assumed for the parameter selection that a $\sqrt{N}$ gain was always possible.

5.2 Practical Parameters

The procedure to select practical parameters involves a few trade-offs, between security issues and error correction capability. Basically, for a given code length n and a level of security γ, we use the best algorithm in practice for low weight codeword finding [31] taking into account the gains showed in Table 1, in order to estimate the number of errors t and the row weight w. If such (n,r,w)-MDPC code is not able to correct t errors, then the length is increased (say by 100, for instance). This procedure quickly converges because the workfactor is not much affected by changing n in such order of magnitude, whilst the error correction capability of MDPC codes is rather improved.

Regarding the security aspects, we have to stress that we are not considering the scenario where the attacker really possesses multiple instances of the syndrome decoding problem. We are only considering the specific MDPC code structure which naturally allows the application of such attack. For the application of our variants where the attacker has indeed access to multiple instances, parameter adjustments must be done in order to achieve the desirable level of security.

Regarding the error correction of MDPC codes, as discussed in Appendix A, it can be estimated through thresholds obtained from theoretical computation and refined through practical simulation. For our parameters, we select the number of errors to be corrected departing from the theoretical threshold and then evaluating their practical error correction capability through exhaustive simulation until achieving a suitable low failure rate for Gallager A algorithm. We managed to achieve rates below 10^{-7} for the quasi-cyclic variant. The MDPC variant (not quasi-cyclic) might present a worse error correction capability due to the different column weights, but significant improvements can be obtained with small increasings on the code length. It is important to stress however that a small probability of decoding failure for the Gallager A algorithm ensures a highly conservative bound for error correction capability in general, since we can always resort to the more sophisticated Gallager B algorithm, which has an increased error correction capability, reducing then the overall failure probability to negligible rates. However, this (very small) chance of decoding failure has to be handled and it is addressed in the discussion of *CCA-2* conversions in 3.1.

The Table 2 presents the suggested parameters for our scheme. The quasi-cyclic variant allows for extremely compact public key sizes (equal to $n - r$ bits, equivalent to the message size). For 80-bits of security, it achieves public-key sizes of only 4800 bits. This represents a reduction of 76.56% from the Quasi-Dyadic McEliece variant [15] and of 98.95%, if compared with the standard version of McEliece using its updated parameters [33]. The purely MDPC variant reduces the venues for mounting structural attacks and also ensures higher security against decoding attacks, since there is no gain using the DOOM attack, by the price of much bigger public keys (equal to $r(n - r)$ bits).

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

Level security	n	w	t	r (syndrome size)	Key size (QC-MDPC)	Key size (MDPC)
80	9600	90	84	4800	4800	23 040 000
80	10752	153	53	3584	7168	25 690 112
80	12288	220	42	3072	9216	28 311 552
128	19712	142	134	9856	9856	97 140 736
128	22272	243	85	7424	14848	110 231 552
128	25088	340	68	6272	18816	118 013 952
256	65536	274	264	32768	32768	1 073 741 824
256	67584	465	167	22528	45056	1 015 021 568
256	81920	660	137	20480	61440	1 258 291 200

6 Conclusion

In this paper, we show that moderate density parity-check codes are suitable for cryptographic purposes. We propose two new McEliece variants based on MDPC and on QC-MDPC codes. Our approach simplifies the idea presented in [5], discarding any transformation matrices used so far. The main benefit of our approach is the possibility to provide a security reduction much closer to the general decoding problem than any other public key code-based cryptosystem proposed so far. The general security reduction for McEliece and Niederreiter cryptosystems relies on two problems: the pseudo-randomness of the employed family of code and decoding a linear code. The pseudo-randomness, in fact the key security, is often the weak spot. Even for Goppa codes, if the problem is considered hard, it does not inspire as much confidence as the decoding problem. In this paper we managed to relate the key security to a hard decoding problem. Under a plausible conjecture, our proposals are secure if decoding, or finding low weight codewords, is difficult in a random linear code or in a random quasi-cyclic linear code. Those hardness assumptions are commonly admitted as true.

The quasi-cyclic variant also allows for very compact public-keys, being equal to the message size. For 80 bits of security, it features a public key of 4800 bits, the smallest ever proposed for a public-key encryption scheme based on binary codes. It represents around one percent of the public-keys suggested in [33] and less than one fourth of those suggested for quasi-dyadic Goppa codes [15]. Our proposal is also quite competitive to the widely employed RSA cryptosystem [8], with the advantage of being much more efficient, a feature inherent to code-based cryptosystems. On the other hand, the purely MDPC variant has even less structure, which further reduces the venues for mounting structural attacks. Additionally, by using MDPC codes, instead of Goppa codes, the McEliece cryptosystem might also become more robust against implementation attacks [38,39].

Acknowledgements

. . .

References

1. Monico, C., Rosenthal, J., Shokrollahi, A.: Using low density parity check codes in the McEliece cryptosystem. In: IEEE International Symposium on Information Theory – ISIT'2000, Sorrento, Italy, IEEE (2000) 215
2. Baldi, M., Chiaraluce, F., Garello, R.: 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). (2006) 305–310
3. Baldi, M., Chiaraluce, F., Garello, R., Mininni, F.: Quasi-cyclic low-density parity-check codes in the McEliece cryptosystem. In: Communications, 2007. ICC '07. IEEE International Conference on. (2007) 951 –956
4. Baldi, M., Chiaraluce, F.: Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes. In: Information Theory, 2007. ISIT 2007. IEEE International Symposium on. (2007) 2591 –2595
5. Baldi, M., Bodrato, M., Chiaraluce, F.: 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, Berlin, Heidelberg, Springer-Verlag (2008) 246–262
6. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5) (1997) 1484–1509
7. Bernstein, D.J., Buchmann, J., Dahmen, E., eds.: Post-Quantum Cryptography. Springer-Verlag (2009)
8. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2) (1978) 120–126
9. Kravitz, D.: Digital signature algorithm. US patent 5231668 (1991)
10. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. Deep Space Network Progress Report **44** (1978) 114–116
11. Courtois, N., Finiasz, M., Sendrier, N.: How to achieve a McEliece-based digital signature scheme. In: Advances in Cryptology – Asiacrypt'2001. Volume 2248 of Lecture Notes in Computer Science., Gold Coast, Australia, Springer (2001) 157–174
12. Berlekamp, E., McEliece, R., van Tilborg, H.: On the inherent intractability of certain coding problems (corresp.). Information Theory, IEEE Transactions on **24**(3) (1978) 384 – 386
13. Faugère, J.C., Gauthier, V., Otmani, A., Perret, L., Tillich, J.P.: A distinguisher for high rate McEliece cryptosystems. In: ITW 2011, Paraty, Brazil (2011) 282–286
14. Gaborit, P.: Shorter keys for code based cryptography. In: International Workshop on Coding and Cryptography – WCC'2005, Bergen, Norway, ACM Press (2005) 81–91
15. Misoczki, R., Barreto, P.S.L.M.: Compact McEliece keys from Goppa codes. In: Selected Areas in Cryptography. (2009) 376–392
16. Berger, T.P., Cayrel, P.L., Gaborit, P., Otmani, A.: Reducing key length of the McEliece cryptosystem. In Preneel, B., ed.: Progress in Cryptology – Africacrypt'2009. Volume 5580 of Lecture Notes in Computer Science., Springer (2009) 77–97
17. Faugère, J.C., Otmani, A., Perret, L., Tillich, J.P.: Algebraic cryptanalysis of McEliece variants with compact keys. In Gilbert, H., ed.: Advances in Cryptology – Eurocrypt'2010. Volume 6110 of Lecture Notes in Computer Science., Springer (2010) 279–298
18. Gallager, R.G.: Low-Density Parity-Check Codes. M.I.T. Press (1963)

19. Otmani, A., Tillich, J., Dallot, L.: Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes. Special Issues of Mathematics in Computer Science **3**(2) (2010) 129–140
20. Baldi, M., Bianchi, M., Chiaraluce, F.: Security and complexity of the McEliece cryptosystem based on QC-LDPC codes (2012) arXiv:1109.5827.
21. Biswas, B., Sendrier, N.: McEliece cryptosystem implementation: Theory and practice. In: PQCrypto. (2008) 47–62
22. Sendrier, N.: Decoding one out of many. In Yang, B.Y., ed.: Post-Quantum Cryptography. Volume 7071 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2011) 51–67 10.1007/978-3-642-25405-5-4.
23. Overbeck, R., Sendrier, N.: Code-based cryptography. In Bernstein, D., Buchmann, J., Dahmen, E., eds.: Post-Quantum Cryptography. Springer (2009) 95–145
24. Berson, T.A.: Failure of the McEliece public-key cryptosystem under message-resend and related-message attack. In: Advances in Cryptology CRYPTO 97 Proceedings, Springer-Verlag (1997) 213–220
25. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Advances in Cryptology – CRYPTO'1999. Volume 1666 of Lecture Notes in Computer Science., Gold Coast, Australia, Springer (1999) 537–554
26. Sendrier, N.: On the use of structured codes in code based cryptography. In Nikova, S., Preneel, B., Storme, L., eds.: Coding Theory and Cryptography III. Contactforum. Koninklijke Vlaamse Academie van België voor Wetenschaeppen en Kunsten (2009) 59–68
27. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory **15**(2) (1986) 159–166
28. Li, Y.X., Deng, R.H., Wang, X.M.: On the equivalence of mceliece's and niederreiter's public-key cryptosystems. Information Theory, IEEE Transactions on **40**(1) (1994) 271 –273
29. Prange, E.: The use of information sets in decoding cyclic codes. Information Theory, IRE Transactions on **8**(5) (1962) 5–9
30. Stern, J.: A method for finding codewords of small weight. In Cohen, G., Wolfmann, J., eds.: Coding Theory and Applications. Volume 388 of Lecture Notes in Computer Science., Springer (1989) 106–113
31. Dumer, I.: On minimum distance decoding of linear codes. In: Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory, Moscow (1991) 50–52
32. Canteaut, A., Chabaud, F.: 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) (1998) 367 –378
33. Bernstein, D.J., Lange, T., Peters, C.: Attacking and defending the McEliece cryptosystem. In: Proceedings of the 2nd International Workshop on Post-Quantum Cryptography. PQCrypto '08, Berlin, Heidelberg, Springer-Verlag (2008) 31–46
34. Finiasz, M., Sendrier, N.: Security bounds for the design of code-based cryptosystems. In Matsui, M., ed.: Advances in Cryptology – Asiacrypt 2009. Volume 5912 of Lecture Notes in Computer Science., Springer (2009) 88–105
35. Bernstein, D., Lange, T., Peters, C.: Smaller decoding exponents: Ball-collision decoding. In Rogaway, P., ed.: Advances in Cryptology CRYPTO 2011. Volume 6841 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2011) 743–760 10.1007/978-3-642-22792-942.
36. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In Lee, D., Wang, X., eds.: Advances in Cryptology - ASIACRYPT 2011. Volume 7073 of LNCS., Springer (2011) 107–124
37. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How 1+1=0 improves information set decoding. In Pointcheval, D., Johansson, T., eds.: Advances in Cryptology - EUROCRYPT 2012. Volume 7237 of LNCS., Springer (2012) 520–536
38. Shoufan, A., Strenzke, F., Molter, H., Stttinger, M.: A timing attack against Patterson algorithm in the McEliece PKC. In Lee, D., Hong, S., eds.: Information, Security and Cryptology ICISC 2009. Volume 5984 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 161–175
39. Strenzke, F.: A timing attack against the secret permutation in the McEliece PKC. In Sendrier, N., ed.: Post-Quantum Cryptography. Volume 6061 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2010) 95–107
40. Richardson, T., Urbanke, R.: Modern Coding Theory. Cambridge University Press (2008)

A LDPC/MDPC error correction capability estimation

The error correction capability of LDPC codes increases linearly with the code length and decreases with the density of its parity-check matrix. The LDPC decoding algorithms are iterative and can be summarized in two groups. The first one is devoted to achieve a very fast decryption process, at the price of a decreased error correction capability, exemplified through the Gallager A algorithm. At every step of the decoding process each bit of the code is chosen to be either 0 or 1. The second one is the opposite, being less efficient algorithmically but able to correct more errors, exemplified by Gallager's B algorithm. It consists in passing soft information during the decoding process (here a probability that a certain bit is either equal to 0 rather than a hard decision on the bit). Both algorithms A and B were presented by Robert Gallager in his seminal work [18]. For our case, using MDPC codes, a significant decrease in the error correction capability is expected (but as we have presented, this degradation is not an obstacle to obtain secure parameters and efficient decoding).

In general, the estimation of error correction capability for LDPC codes is always a hard task, but two approaches can be used to circumvent this issue. The first one is based on theoretical and probabilistic argumentation, providing what is known as the waterfall threshold for a LDPC code. This value represents the number of errors from which reliable decoding can be expected. However there is no strict guarantee, for a number of errors below this threshold, that a negligible decoding failure rate will be reached. In this sense, a second and naïve approach is also usually adopted. Through exhaustive simulation it is possible to determine the decoding failure probability for a specified number of errors. Having said that, the approach to choose the number of errors to be corrected by our codes is to start from the theoretical bound (waterfall threshold) and decrease it until the exhaustive simulation attests a negligible decoding failure rate.

Regarding the theoretical argumentation, in [18] a weak bound on the probability of decoding failure is provided, represented through a recursive function p_i. To calculate this quantity it is assumed that

– the length of the code is infinite,
– there are no cycles of length less than or equal to $2i$ in the Tanner graph [40].

These conditions can be somehow relaxed and a finite analysis of the decoding process can be obtained [40], but this is beyond the scope of this paper. Here p_i is the probability that a certain bit is incorrectly estimated at the i-th step of iterative decoding in Gallager's algorithm A. This probability does not depend on a particular position with the aforementioned hypotheses. A necessary and sufficient condition ensuring that iterative decoding typically converges to the right solution when the length of the code is large enough [18,40] is that the sequence p_i should converge to 0.

p_0 represents the probability that a bit is incorrect at the initial stage of the decoding algorithm and is equal for our cryptographic application to

$$p_0 \overset{\text{def}}{=} \frac{t}{n}.$$

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 the parity-check matrix H of the MDPC code. 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 λ_i by

$$\lambda_i \overset{\text{def}}{=} \frac{m_i}{m}$$

Notice that m_i is also equal to i times the number of columns of weight i in H. In the quasi-cyclic case, notice that H has the following block form

$$H = [H_0|H_1|\dots|H_{n_0-1}],$$

where H_i is a $p \times p$ matrix of constant row/column weight equal to some quantity w_i where $w_0 + w_1 + \cdots + w_{n_0-1} = w$. In this case

$$m = rw$$

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 [18] we choose it as the smallest integer for which the following expression holds:

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

The *threshold* for iterative decoding is defined as the supremum of the value of p_0's for which p_i converges to 0. Since LDPC decoding is probabilistic, even with a number of errors below such threshold there is a probability that the algorithm does not successfully decode. However, exhaustive simulation can be used, in order to estimate such decoding failure probability. Therefore, to ensure a good error correcting capability in practice, a good approach is to chose the number of errors to be corrected to start from the theoretical threshold value and then simulate its performance. If the decoding failure probability is not satisfactory, the number of errors is decreased until a negligible failure rate is achieved.