Side Channel Key Recovery Attacks on QC-MDPC Codes

Matthieu Lequesne

Internship report – MPRI M2

Side Channel Key Recovery Attacks on QC-MDPC Codes

Matthieu Lequesne,
supervised by Nicolas Sendrier,
Inria Paris, équipe-projet Secret

March – August, 2017

Contents

Summary

General Context

In 1994, Peter Shor proposed an efficient algorithm to factor integers using a quantum computer. Considering the recent advances in quantum computing, this becomes a real threat for most cryptosystems based on number therory, including the widely used RSA. Therefore, it is urgent to start using cryptosystems resistant to attacks using quantum computers. Such cryptosystems are called post-quantum cryptosystems.

The National Institute of Standards and Technology (NIST) announced in 2016 a call for standardization of post-quantum cryptosystems. Four possibilities are often mentioned as good candidates: cryptosystems based on error correcting codes, lattices, hash functions and multivariate quadratic equations. The approach that we study here is based on codes. Error correcting codes have been extensively studied in the 20^{th} century for their use in telecommunication systems. In a paper from 1978, McEliece proposed to use them for cryptographic purposes.

McEliece's proposal is still valid but requires very long public keys. Here, we will study a variant of McEliece's idea using a family of codes named Quasi-Cyclic Medium Density Parity Check (QC-MDPC) codes, proposed in 2013, whose quasi-cyclic structure allows shorter key sizes. This will be an important criteria for the NIST competition.

Problem Studied

The QC-MDPC codes are very recent and have not been studied widely. In order to make the QC-MDPC cryptosystem a competitive proposal for the NIST competition, we need to understand their possible weaknesses in order to prevent different kinds of attacks.

In this work, we will focus specifically on side-channel attacks. In 2016, Guo, Johansson and Stankovski exploited a weakeness in the original decoding algorithm to propose a key-recovery attack. The decoding algorithm has been improved but their attack is actually linked to the quasi-cyclic structure, which is inherent to this type of codes. It is therefore crucial to deeply understand the origin of their attack, and all other ways to exploit the flaw, to propose adequate countermeasures.

Proposed Contributions

Our first goal was to understand the origin of Guo's attack. We traced the flaw back to a specific parameter, the syndrome weight. We proposed a new attack specifically observing this parameter and derived from it a timing attack using the tuned version of the decoding algorithm. To our knowledge, this is the first timing attack on this QC-MDPC codes. Our theoretical analysis of the flaw gives a bound that will allow us to determine parameters to resist attacks that would be too costly to implement in practice.

Arguments Supporting Their Validity

This work contains theoretical parts as well as results of experiments. All described attacks were implemented successfully. The experimental parts confirm the theoretical results, hence supporting their validity. On the graphs in the appendix, we can see very clearly several lines that correspond to the linearity of the average syndrome weight in terms of multiplicity, just as expected with the theoretical analysis. Then, theoretical results from information theory provide lower bounds on the number of decryption needed to perform the attack. This bound is coherent with the experiments. The fact that the attacks were realised in practice on a simple personal computer also shows that their cost is not high. This proves that it is important to prevent them.

Still, this work relies on the assumption that one can measure precisely the syndrome weight or the number of rounds of the decoding algorithm. In the case of a physical attack on a decryption device, if one were to measure a parameter such as the power consumption to access the syndrome weight, or the time to access the number of rounds, this measures would not be completely precise. But it would be easy to adapt our work for a given model of noise, and fault-tolerant reconstruction algorithms are also discussed in this paper.

Summary and Future Work

Our paper studies a general flaw inherent to the quasi-cyclic structure of codes. Most post-quantum cryptosystem face the same problem of long key size and several of them use a quasi-cyclic structure to solve this issue. It is therefore interesting to try to find similar attacks exploiting the same kind of flaw on other cryptosystems.

This work proposes the first timing attack on QC-MDPC codes and proposes analytical bounds useful for countermeasures. It is now possible to derive the right parameters to ensure appropriate security bounds, depending the performance of the decoding algorithm. Especially, this work emphasizes the need for a better decoding algorithm havign both a small failure rate and requiring constant time. The choice of the decoding algorithm will be essential to make the QC-MDPC a competitive proposal for the NIST.

Finally, the algorithmical question of the reconstruction of the key from its spectrum also requires further study, especially the case of fault-tolerant reconstruction.

1 Introduction

1.1 Quantum computers and post-quantum cryptography

Most cryptosystems rely on the difficulty of two mathematical problems: factoring integers and computing discrete logarithms. In 1994, Peter Shor proved that both these problems can be efficiently solved using quantum computers [21].

Although there are no practical quantum computers yet, many believe it's only a matter of decades before they become a reality. This threatens most public-key cryptosystems, especially the widely used RSA. There is a need to develop post-quantum cryptosystems, that is cryptosystems that remain secure against an adversary equiped with a quantum computer, and we should start using such quantum-resistant systems rapidly if we want to ensure long-term security. The National Institute of Standards and Technology (NIST) announced in 2016 a call for standardization of post-quantum cryptosystems [7].

Four families of cryptosystems are often mentioned as potential candidates: cryptosystems based on error correcting codes, lattices, hash functions and multivariate quadratic equations. All of these are based on mathematical problems that are expected to remain hard even in the presence of a quantum computer.

1.2 Previous related work

In this paper, we propose a side-channel attack on the QC-MDPC code-based public-key cryptosystem [18].

The first code-based cryptosystem was proposed in 1978 by McEliece [17]. The McEliece scheme makes use of a family of structured codes. The original form used Goppa codes and remains unbroken. But this version requires very large public keys (about 500 kbit). Several variants proposed to use other families of codes that would lead to shorter keys but most of them turned out to be unsecure.

In 2013, one new variant was proposed using quasi-cyclic (QC) moderate density parity-check (MDPC) codes [18]. QC-MDPC codes use much shorter keys (about 5 kbit). Indeed, the quasi-cyclic nature of the matrices allows to fully describe them by their first row. This choice appears promising and the QC-MDPC scheme was recomended for further study by the report "Initial Recommandation of long-term secure post-quantum systems" of the European initiative PQCRYPTO [2]. Some hardware implementations of this scheme were achieved in [15] and [22].

The decryption algorithm of the QC-MDPC scheme is a variant of Gallager's bit flipping algorithm [13]. It's an iterative algorithm which appears to be very basic at first sight and is rather easy to implement on small devices. The problem is that this algorithm is subject to decryption failure with non-negligeable probability. The algorithm proposed in the original paper [18] has a *decoding failure rate* (DFR) of 10^{-7}.

In [14], Guo, Johansson and Stankovski exploited this DFR and managed to successfully recover the key by analyzing the error patterns that made the decryption fail. They found out that these error patterns are correlated in some way with the key. They introduce a new definition, the *distance spectrum*, to caracterize the correlation. Although they successfully use this correlation to

perform their attack, they don't explain why error patterns correlated in such a way are more prone to cause decryption failure.

The original QC-MDPC scheme is designed to be secure in the chose plaintext attack model (CPA) and is not secure against chosen ciphertext attacks (CCA). But it can be easily converted to resist adaptative chosen ciphertext attacks (CCA2) using semantically secure conversions [16]. The attack of [14] recovers the key within minutes in the CPA model but requires $2^{39.7}$ operations in the CCA model.

In another paper [10], Fabsic *et al.* use the same idea to attack the QC-LDPC McEliece cryptosystem. This scheme had been propose by Baldi in [3], some vulnerability were shown in [19], and it inspired the design of the QC-MDPC scheme. The decryption algorithm used in QC-LDPC is quite similar to the one in QC-MDPC but takes soft decisions. This attack demonstrates that the ideas used in [14] to attack QC-MDPC can be used on a soft-decision algorithm. Indeed, both schemes share the quasi-cyclic form.

Attempts were made to turn the iterating decoder into a constant-time decoder. This is the case of QcBits proposed by Chou [8]. QcBits is CCA-secure and claims a DFR of 10^{-8}. Guo's attack on QcBits would require $2^{55.3}$ operations [14] but has not been performed in practice. Another form of side-channel attack was recently proposed against QcBits: in [20], Rossi *et al.* use differential power analysis against the syndrome computation to recover one half of the private key, and then manage to recover the whole key, proving the need to use masking techniques in the implementation of the decoding algorithm.

Since the first publication of the QC-MDPC scheme, efforts have been made to tune the decoding algorithm, especially exploring the different ways to fix the thresholds in order to reduce the DFR [6].

1.3 Our contribution

Indeed, this quasi-cyclic nature induces some properties that can be used to recover information on the key. We analyze the origin of the flaw in the distribution of the syndrome weight induced by the quasi-cyclic structure and explain how to turn it into a side-channel attack by monitoring the syndrome weight. This also explains the attack of [14]. Then we successfully exploit this flaw to recover the key via a timing attack on the decryption algorithm. To our knowledge, this is the first timing attack on this scheme.

Throughout this paper, we always consider the most efficient decoding algorithm so far, that is using the recommendations of [6]. This is the first attack involving this tuned decoder with lower DFR. Our attacks are directly performed in the CCA model. Unless mention of the contrary, we always use the parameters proposed in [18] for 80-bits security.

1.4 Paper roadmap

In Section 2, we recall some useful definitions from coding theory and describe the McEliece cryptosystem. In Section 3, we present the QC-MDPC scheme and its decoding algorithm in details. In Section 4, we identify the origin of the flaw that will lead to attacks. In Section 5, we describe our side-channel attack on the syndrome weight. In Section 6, we present the results and analyse the attack. In Section 7 we explain how to turn this in a timing attack.

Finally, in Section 9, we propose countermeasures and discuss future research directions.

2 Background

2.1 Preliminaries in coding theory

Let us recall some basics of coding theory which are used in this paper.

Definition 2.1 (Hamming weight)**.** The Hamming weight $\mathsf{w}(x)$ of a vector $x \in \mathbb{F}_2^n$ is the number of it's non-zero components.

Definition 2.2 (Linear codes)**.** A binary $[n, k]$-linear code $\mathcal{C}$ is a linear subspace of $\mathbb{F}_2^n$ of dimension k.

Definition 2.3 (Generator matrix)**.** A matrix $G \in \mathbb{F}_2^{k \times n}$ is a generator matrix of the binary $[n, k]$-linear code $\mathcal{C}$ if its rowspan is $\mathcal{C}$, that is if $\mathcal{C} = \left\{G \cdot x \,|\, x \in \mathbb{F}_2^k\right\}$.

Definition 2.4 (Parity-check matrix)**.** A matrix $H \in \mathbb{F}_2^{(n-k) \times n}$ is a parity-check matrix of the binary $[n, k]$-linear code $\mathcal{C}$ if $\mathcal{C}$ is the kernel of H, that is if $\mathcal{C} = \{x \in \mathbb{F}_2^n \,|\, H \cdot x^\intercal = 0\}$.

Note that a code can be represented by different generator or parity-check matrices.

Definition 2.5 (Syndrome)**.** The syndrome $s \in \mathbb{F}_2^{n-k}$ of a vector $e \in \mathbb{F}_2^n$ is $s = H \cdot e^\intercal$.

Definition 2.6 (Quasi-cyclic code)**.** A binary $[n, k]$-linear code $\mathcal{C}$ 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 $n = n_0 k$, it is possible and convenient to represent the generator and parity-check matrices composed by $k \times k$ circulant blocks. A circulant block is completely described by its first row, which is useful to reduce key size, and the algebra of $k \times k$ binary circulant matrices is isomorphic to the algebra of polynomials modulo $X^k - 1$ over $\mathbb{F}_2$, which makes computations efficient.

2.2 The McEliece Cryptosystem

McEliece's idea [17] is to use a structured linear code with an efficient decoding algorithm but display its generator matrix in a base where it appears random. The (modified) generator matrix is serves as the public key. To encrypt, transform the message in codewords and add random errors. The legitimate recipient is the only one to know the efficient decoding algorithm (the secret key) using the structure of the code.

The security of this scheme relies on two assumptions: the hardness of general decoding (*the message security*) and the pseudorandomness of the family of codes (*the key security*). If an adversary wants to decrypt without the secret key, it has either to try to decode a random linear code, hence breaking the hardness of general decoding assumption, or it has to recognize the hidden structure of the code and hence distinguish the code from a random code, which breaks the pseudorandomness assumption.

Generic decoding is a known NP-complete problem [4] and is believed to be hard on average [1]. But the pseudorandomness assumption depends on the family of codes.

The McEliece cryptosystem was the first code-based cryptosystem and its original form proposes the uses of Goppa codes, which is still considered as secure. After almost four decades, no efficient algorithm was found to distinguish generator matrices of Goppa codes from random matrices (except when the code rate is close to 1 [11], which is not relevant in the case of McEliece). Several proposals were made using other families of codes but most of them were broken. The main drawback of the original McEliece cryptosystem using Goppa codes is that it requires large public keys (the generator matrix, around 500 kbit for 80-bits security), unfit for practical purposes.

In 2013, a new version of the McEliece cryptosystem was proposed: the QC-MDPC-scheme [18].

3 The QC-MDPC Encryption Scheme

3.1 QC-MDPC codes

Definition 3.1 (LDPC and MDPC codes)**.** An (n, r, w)-LDPC or MDPC code is a binary $[n, n-r]$-linear code which admits a parity-check matrix of constant row weight w.

LDPC and MDPC codes have the same definition. The name depends on the choice of w: LDPC codes have a small constant row weight (usually $w \leq 10$) while MDPC codes have a row weight that scales in $w = O(\sqrt{n})$. The LDPC codes were first introduced for telecommunication purposes and inspired the idea of MDPC codes. MDPC codes have worse error-correction capacity than LDPC codes, but are better for cryptographical use. Actually, MDPC codes are the first family of codes designed specifically for cryptographic purposes and not for telecommunication. These codes can take a quasi-cyclic form, in this case, they are called (n, r, w)-QC-LDPC or MDPC codes.

For the sake of simplicity, in the rest of this paper we will only consider QC-MDPC codes whose generator matrix is composed of two circulant blocks ($H = [H_0 \,|\, H_1] \in \mathbb{F}_2^{r \times n}$ with $n = 2r = 2k$) as proposed in [18].

3.2 Encryption scheme

We denote t the number of errors that will be added to encrypt the message. It needs to be chosen so that the bit flipping decoder can correct t errors for an (n, w, r)-QC-MDPC code with $n = 2r$. Usually this involves $tw = O(n)$ and as $w = O(\sqrt{n})$ we have $t = O(\sqrt{n})$. We will denote d an odd number that will be the row weight of each of the two circulant matrices, hence we will have $w = 2d$. We take r prime to prevent attacks exploiting non-prime quasi-cyclicity such as [12].

Hence, we have parameters $r, d, t \in \mathbb{N}$ such that r prime, d odd and $2d \sim t \sim \sqrt{2r}$.

3.2.1 Key Generation

Generate two vectors $h_0, h_1 \in \mathbb{F}_2^r$ such that $\mathsf{w}(h_0) = \mathsf{w}(h_1) = d$. Each vector defines a circulant matrix whose lines are its shifts. The parity-check matrix of the code will be $H = [H_0 \,|\, H_1] \in \mathbb{F}_2^{r \times n}$, the matrix formed by concatenating the two circulant blocs. The associated generator matrix in systematic form is $G = [I \,|\, (H_1^{-1} H_0)^\intercal]$ which is also circulant. G is the public key and H is the private key[1].

3.2.2 Encryption

To encrypt the message $m \in \mathbb{F}_2^r$, generate a random vector $e \in \mathbb{F}_2^n$ such that $\mathsf{w}(e) = t$. The ciphertext is $c = mG + e \in \mathbb{F}_2^n$.

3.2.3 Decryption

Use the bit flipping iterative decoder to correct the errors in c and get $c - e = mG$. As G is in systematic form, the plaintext m corresponds to the first half of the codeword.

3.2.4 Parameters

Parameters suggested in [18]:

Level security	r	d	t	public key size
80	4801	45	84	4801
128	9857	71	134	9857
256	32771	137	264	32771

3.3 The iterative decoding algorithm

The original paper on MDPC codes [18] proposes to use a hard decision version of Gallager's bit flipping algorithm for decoding LDPC codes [13]. The main idea is the following. At each iteration, the algorithm computes the number of unsatisfied parity-check equations associated to each bit. Each bit that is involved in $\geq b$ unsatisfied equations is flipped, for b some threshold, and the syndrome is recomputed. This repeats until the syndrome becomes zero (in practice, the algorithm stops after fixed number of iterations and this is considered a decoding failure).

Here, $h^{(i)} \in \mathbb{F}_2^r$ denotes the i^{th} column of H. The scalar product $\langle s, h^{(i)} \rangle$ is computed in $\mathbb{Z}$ and corresponds to the number of unsatisfied parity-check equations involving the position i.

The choice of the threshold b is crucial for the algorithm and represents the main difference between the variants of the bit flipping algorithms. Gallager [13] proposes to use different (precomputed) thresholds for each iteration. For his scheme QcBits [8], Chou developed a constant-time algorithm using such thresholds and six iterations. The thresholds are obtained through experimental results. Another approach, recomanded in [18], is to take for b the maximum

[1] Note that the first line of $H_1^{-1} H_0$ is enough to describe G, hence the public key has size r and the private hey is described by (h_0, h_1) of size $n = 2r$.

Algorithm 1 Iterative bit flipping decoding algorithm

Input: $c = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_2^n$, $H = (h^{(0)}, \ldots, h^{(n-1)}) \in \mathbb{F}_2^{n \times r}$
  $s \leftarrow H \cdot c^\intercal$
  while $s \neq 0$ **do**
    for $i = 0, \ldots, n-1$ **do**
      if $\langle s, h^{(i)} \rangle \geq b$ **then**
        $c_i \leftarrow c_i \oplus 1$
    $s \leftarrow H \cdot c^\intercal$
  return c

value of the counters minus some constant (around 5). [6] propose an optimization of the fixed threshold approach, choosing the values of b for each iteration depending on the syndrome weight at the time. This algorithm gives the best results so far.

The attack of [14] was performed against Gallager's decoder with fix precomputed threshold values (decoder $\mathcal{B}$ in the implementation [22]). In our attack, we will consider the best algorithm so far (in terms of decoding failure), following the proposal of [6].

4 The Flaw

4.1 Correlation

Our attack is based of the fact that the average syndrome weight is slightly different if the relative position of bits set to "1" in the key and the error are correlated.

For the sake of simplicity, in this section, we will consider a parity-check matrix made of one single circulant bloc in $H \in \mathbb{F}_2^{r \times r}$ instead of two. We will see later that the practical results are the same. We denote $h \in \mathbb{F}_2^r$ the first row of the matrix H. The variable t still denotes the weight of the error e, though in the scheme above this error is slit between the two halfs of the error vector.

4.1.1 Without any information

First, let's suppose that we don't have any information about the key. For a random key vector h of size r and weight d and a random error vector e of size r and weight t, denote $f(r, d, t, b)$ the probability that the scalar product in $\mathbb{F}_2$ be of parity b:

$$f(r, d, t, b) := \mathbb{P}(\langle h, e \rangle = b) = \sum_{i=0,\ i \equiv b[2]} \frac{\binom{d}{i}\binom{r-d}{t-i}}{\binom{r}{t}}.$$

The average syndrome weight of an error e and a parity-check matrix generated by cyclic shifts of h is r times this value (see [5, page 91]), that is:

$$|H \cdot e^\intercal| = r \cdot f(r, d, t, 1).$$

4.1.2 If two consecutive key bits are set to "1"

Now, suppose the key vector h has ℓ times two consecutive bits set to "1". Let's observe the shifts of the vector:

$$\mathsf{shift}(h) = \boxed{1}\boxed{1}\boxed{u, |u| = d-2} \quad \ell \text{ times}$$

$$\mathsf{shift}(h) = \boxed{1}\boxed{0}\boxed{u, |u| = d-1} \quad d-\ell \text{ times}$$

$$\mathsf{shift}(h) = \boxed{0}\boxed{1}\boxed{u, |u| = d-1} \quad d-\ell \text{ times}$$

$$\mathsf{shift}(h) = \boxed{0}\boxed{0}\boxed{u, |u| = d} \quad r-2d+\ell \text{ times.}$$

Suppose that the first two bits of the error vector are set to "1", that is:

$$e = \boxed{1}\boxed{1}\boxed{u, |u| = t-2}.$$

With this extra assumption on the form of h and e, the average syndrome weight of e with respect to the the parity-check matrix H generated by cyclic shifts of h can now be approximated[2] by:

$$\begin{array}{rcccl} |H \cdot e^{\mathsf{T}}| & = & \ell & f(r-2, d-2, t-2, 1) & \\ & + & 2(d-\ell) & f(r-2, d-1, t-2, 0) & (1) \\ & + & (r-2d+\ell) & f(r-2, d, t-2, 1). & \end{array}$$

Contrary to the previous result, this is an approximation. Indeed, this result doesn't take into account the covariance between different lines. Previously we were averaging on all the lines and the covariance was therefore null, while here the fact that we group the rows depending on the value of the first two bits breaks the symetry. Still, we will see that the approximation is close to the real value and we can neglect the correction term for the rest of the reasoning.

4.1.3 Exploiting the flaw

Suppose that we only consider error patterns starting with two consecutive bits set to "1", the syndrome weight is expected to be slightly different on average, depending on ℓ the number of times two consecutive bits are set to "1" in the key vector h. Moreover, the expected value depends linearly of ℓ. Therefore, if we observe enough values of the syndrome weight, we can recover the value of ℓ.

4.2 The distance spectrum

In fact, this observation is not only true for two consecutive bits. The same results is obtained if you shift the columns. Hence, for any distance d such there exist two bits at distance d set to "1" in the error patern, the expected syndrome weight will vary depending on the presence or absence of two bits set to "1" at distance d in the key vector.

This observation leads to the definition of the distance spectrum, proposed in [14]:

[2]Indeed, this model assumes that the "rest" of the vector (denoted u) is random for each shift, it doesn't take into account the covariance between the bits. In the previous case, the covariance was zero because we were averaging on all the possibilities.

Definition 4.1 (Distance Spectrum)**.** The distance spectrum of a vector $h \in \mathbb{F}_2^r$, denoted $\Delta(h)$, is the set of distances δ such that there exist two bits of h at distance δ with value "1". The distance are counted cyclically.

$$\Delta(h) = \left\{ \delta : 1 \leq \delta \leq \lfloor \frac{r}{2} \rfloor, \exists (i,j), \begin{array}{l} 0 \leq i < j < r, \\ h_i = h_j = 1, \\ \min\{j-i, r-(j-i)\} = \delta \end{array} \right\}.$$

Example 4.1. Consider a case where $r = 10$ and $d = 3$. If $h = 1001000001$ then $\Delta(h) = \{1, 3, 4\}$.. The first and last bits of h are neighbours (again, the distance is counted cyclically), which gives the 1 in the spectrum. The first and fourh bits are at distance 3. The fourth and last are at distance 4.

4.3 Reconstructing the key

4.3.1 Unicity

How much information does the spectrum give on the key? Given a distance spectrum, can we recover the key? First, remark that all cyclic shifts of the key lead to the same spectrum. But this is not an issue since they generate the same code. Also, two symetric vectors (that is h and $\tilde{h}$ such that $\tilde{h}_i = h_{r-i}$) have the same spectrum. Hence, the key can only be reconstructed up to a rotation and symetry.

And still, it's possible to find two different vectors with the same spectrum. For example, take $r = 10$ and $w = 4$. The keys $h_1 = 1101100000$ and $h_2 = 1101000100$ are neither symetric nor cyclic shifts of the other but we have $\Delta(h_1) = \Delta(h_2) = \{1, 2, 3, 4\}$.

Still, the spectrum gives very strong constrains on the form of the key. In practice, for the usual parameters of MDPC codes, the spectrum defines uniquely the key up to symetry, though we don't have a proof for this statement.

4.3.2 Reconstruction algorithm

Guo *et al.* propose a simple backtrack algorithm to list all possible keys matching the spectrum (up to the symetry class) in [14]. Start from a null vector. Assign the first bit to one, as well as the d^{th} bit, where d is the smallest distance in the spectrum. Then try to flip a third bit to one and test if the distances created with the other "1" bits are all in the spectrum. If they don't, test the next position. If they do, try to add a fourth bit. Keep goint until you have reached the expected weight, then this gives a potential solution, backtrack if needed. This is detailed in Algorithm 2.

This algorithm is efficient on average: Guo claims that an unoptimized version runs in 144 seconds on average, worst case in 49 minutes.

4.3.3 Making use of the negative information

This algorithm relies on the fact that if we set a new bit i to "1" in the key, then for each bit j that was already set to "1", the distance $j - i$ must be in the spectrum. But the converse in also true: when we set a new bit i to "1", for all distances δ that are not in the spectrum, we know for sure that the bits at distance δ of i have to be set to "0" and we should never try to flip them.

Algorithm 2 Key recovery from distance spectrum (1)

Input: distance spectrun Δ, partial secret key h, current depth l.
Output: recovered secret key h or message "No such key exists".

```
if l = w then return h
for all potential key bits i do
    for all distances to key bit i exist in Δ do
        Add key bit i to secret key h
        Make recursive call with parameters (Δ, h, l + 1)
        if recursive call finds solution h then
            if h is the secret key then return h
        Remove key bit i from secret key h
return "No such key exists"
```

For the parameters of MDPC codes, approximately one third of the possible distances appear in the spectrum. Hence, this negative information is twice as important at the positive one. Therefore, we propose an upgraded version of Guo's algorithm taking this into account.

The algorithm works as follows:

- Put all bits in the *unknown* position
- Flip a bit to 1, check if all the new distances to known 1-bits are in Δ.
- If so, flip to 0 all bits at a distance $\delta \notin \Delta$.
- If not, try the next bit, backtrack if none works.
- Stop when the weight of 1-bits is equal to d.

This algorithm is more efficient than the previous one. But in both cases, the algorithm finds a solution rather quickly on average but some cases are really slow (when the initial branching is wrong). One possible improvement would be to randomize it (there is no reason to flip the bits in a particular order) and abort and restart the algorithm if no solution is found after some time (around one minute). This should improve the average complexity.

4.3.4 Reconstructing with partial knowledge

This algorithm takes the distance spectrum as input and is supposed to return the key. But in the attack, the distance spectrum is obtained through measurement and might contain some errors. An interesting question is to try to recover the proper key even if the spectrum lacks some information. For example, suppose that for some distances, the measurement is not clear enough to decide with certitude whether or not the distance is in the spectrum. The key-reconstruction algorithm could take as input a variant of the spectrum specifying which distances appear for sure in the spectrum, which ones don't appear, and that it's unclear for the other distances. The previous algorithm can easily be adapted to work with such input.

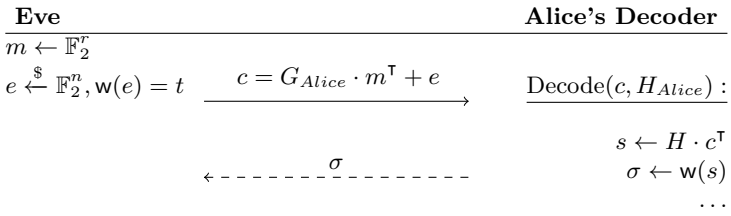
We tried this with examples using the usual 80-bits security parameters. It worked most of the time when 90% of the spectrum values were known but seemed to fail when only 80% were sure. Still, this is a very basic approach and some improvements could certainly be made introducing soft "confidence factors" for each distance, and taking account the expected multiplicity in the spectrum.

5 The Attack

5.1 Attack Model

The scenario for our attack is the following: Eve has access to Alice's decoding device. She sends messages encrypted using the QC-MDPC scheme described in 3.2 and Alice's public key. Whenever the device decodes a message sent by Eve, she can measure to the weight of the syndrome. This assumption is realistic, since the first step of any decoding algorithm is to compute the syndrome of the message to decode. Suppose that Eve has some physical access to the device, she could deduce the weight of the syndrome by measuring some physical value (e.g. power consumption).

This is a theoretical side-channel attack on the weight of the syndrome. We don't focus on the way Eve gets access to this information. From this, Eve will try to recover the distance spectrum Alice's private key, and then reconstruct the key.



This is the first side-channel key-recovery attack on QC-MDPC codes. In [14], the authors proposed a *reaction attack*, measuring only the failures of the decoding algorithm.

We suppose that Alice's error paterns are randomly generated. This model is the equivalent of a *chosen plaintext attack* in code-based cryptography. In fact, Alice doesn't have to even chose a message and can only send the error, since the choice of the plaintext has no influence on the decoding: by definition, the codeword is in the kernel of the parity-check matrix and hence doesn't change the syndrome. Indeed, in the scheme, semantically secure conversions ensure that the error patterns are random [16]. If we allow Alice to chose the error patterns, this will only make the attack easier as in [14].

5.2 Attack Description

Our goal is to compute the distance spectrum of Alice's private key. For each distance δ between 1 and $\lfloor \frac{r}{2} \rfloor$ we want to distinguish if $\delta \in \Delta h_{Alice}$. As we have seen in 4.1.3, for each distance $\delta \in \Delta(e)$, the expected average weight of the syndrome $\sigma = \mathsf{w}(s)$, where $s = H_{Alice} \cdot c^\intercal = H_{Alice} \cdot e^\intercal$, is expected to be different if $\delta \in \Delta(h_{Alice})$.

Hence, the idea is, for each distance δ, to compute the average value of the syndrome weight σ for error paterns e such that $\delta \in \Delta(e)$. The error paterns are generated randomly and each error e can be used to obtain information on all the distances in its spectrum. This leads to the following algorithm:

Algorithm 3 Computing the distance spectrum

Input: N the size of the sample, oracle access to the decoder

```
SyndromeCount ← (0, ..., 0)
OccurenceCount ← (0, ..., 0)
Δ ← (0, ..., 0)
for 0 ≤ i ≤ N − 1 do
    e ←$ F_2^n, w(e) = t
    σ ← OracleDecoder(e)
    for δ ∈ Δ(e) do
        SyndromeCount[δ] += σ
        OccurenceCount[δ] += 1
for 1 ≤ δ ≤ ⌊r/2⌋ do
    if SyndromeCount[δ]/OccurenceCount[δ] < threshold then
        Δ[δ] ← 1
return Δ
```

6 Results and Analysis

6.1 Results

The spectrum recovery algorithm was first tried on a simplified version of the scheme using only one block, in order to compare to the expected behaviour. The result is striking. Using the usual parameters for 80-bits security, with one hundred thousant samples, the spectrum appears very clearly and we can even see the multiplicities, that is, distances that appear several times in the key, see Fig. 1. When pushing to one billion samples, there is no room for confusion (see Fig. 2 in appendix).

When performing the same experiment on the real QC-MDPC scheme with two blocks, we obtain similar results. The attack is performed on each block separatly, that is for each error pattern, we added the syndrome weight to the counters of all distances present in the first half of the error to recover the spectrum of the first block. Because there is no correlation between the two halfs of the error pattern, the presence of the second block acts as a random noise added to the syndrome weight. Hence the only difference is that we need more samples to reduce the variance and distinguish well which distances are in the key spectrum. Note that it is possible to compute the spectrum of both blocks at the same time, so there is no use to double the number of samples to recover the second block. For the usual parameters, with half a million samples, we can recover the spectrum (see Fig. 3 in appendix).

This was also performed the attack when another error is added to the syndrom, like in the Ouroboros scheme [9] (with an additional error of weight $3d$). Again, this only adds random noise and we can recover the spectrum with around one million samples on the usual 80-bits security parameters (see Fig. 4 in appendix).

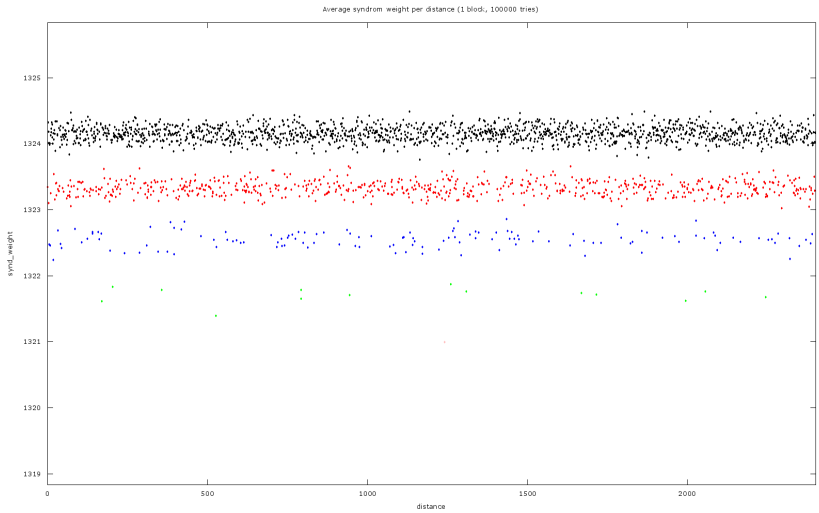


Figure 1: Average syndrome weight per distance, 10^5 samples, 1 block. The color of the distances indicate their multiplicity in the key spectrum (black = 0, red = 1, blue = 2, green = 3)

6.2 Analysis

6.2.1 Multiplicity

In order to analyse more precisely the results, we need to define some more tools involving the multiplicity, that is taking into account the fact that some distances may appear more than once.

Definition 6.1 (Distance Spectrum with multiplicity)**.** The distance spectrum with multiplicity of a vector $h \in \mathbb{F}_2^r$, denoted $\Delta^\mu(h)$, is a vector of $\mathbb{N}^{\lfloor \frac{r}{2} \rfloor}$ such that for every distance $1 \leq \delta \leq \lfloor \frac{r}{2} \rfloor$, its δ^{th} component $\Delta^\mu(h)[\delta]$ is the number of existing sets of two bits of h at distance δ with value "1". The distance are counted cyclically.

Example 6.1. Consider a case where $r = 10$ and $d = 4$. If $h = 0011000011$ then $\Delta^\mu(h) = [2, 0, 1, 2, 1]$.

Let's now fix the distance δ for the rest of this section and suppose that we want to determine wether or not δ is in the spectrum of the key.

Let's denote $\bar{\sigma}_\ell$ the expected average syndrome weight when the distance δ appears in the error and appears ℓ times in the key. More precisely:

Definition 6.2 (Average syndrome weight with multiplicity)**.** Let $\mathcal{D}_\ell$ denote the following set:

$$\mathcal{D}_\ell := \left\{ (h, e) \in \mathbb{F}_2^r \times \mathbb{F}_2^r \,|\, \mathsf{w}(h) = d, \mathsf{w}(e) = t, \delta \in \Delta(e), \tilde{\Delta}(h)[\delta] = \ell \right\}.$$

The average syndrome weight with multiplicity $\bar{\sigma}_\ell$ is the expectency of the syndrome weight for a uniform distribution of (h, e) over $\mathcal{D}_\ell$:

$$\bar{\sigma}_\ell := \mathbb{E}_{(h,e)\sim\mathcal{U}(\mathcal{D}_\ell)} \left[\mathsf{w}(H \cdot e^\intercal) \right].$$

From the equation (1) in section 4.1.2 we know that we can approximate $\bar{\sigma}_\ell$ by:

$$\begin{array}{rcll} \bar{\sigma}_\ell & = & \ell & f(r-2,d-2,t-2,1) \\ & + & 2(d-\ell) & f(r-2,d-1,t-2,0) \\ & + & (r-2d+\ell) & f(r-2,d,t-2,1). \end{array}$$

with $f(r,d,t,b) := \sum_{i=0,\ i\equiv b[2]} \frac{\binom{d}{i}\binom{r-d}{t-i}}{\binom{r}{t}}$

6.2.2 Comparison with measured values

The values of $\bar{\sigma}_\ell$ correspond to the different clusters that we can see on the graph. According to the approximation, the value of $\bar{\sigma}_\ell$ is linear in the multiplicity: $\bar{\sigma}_0 - \bar{\sigma}_\ell = \ell \cdot (\bar{\sigma}_0 - \bar{\sigma}_1)$. Indeed, we can see on Fig. 1 (and even better on Fig. 2 with more samples) that observed average values form lines at regular interval.

With the usual parameters for 80-bits security, we obtain $\bar{\sigma}_0 = 1324.0$ and $\bar{\sigma}_1 = 1323.6$.

When comparing the values to those measured on Fig. 1 and 2, we can see that the measured $\bar{\sigma}_0$ is slightly lower than the approximated value, and on the contrary $\bar{\sigma}_1$ is slightly higher. This error is due to the approximation that neglects the covariance. When performing the same experiment on parameters for LDPC codes, where the covariance is much smaller, the measures correspond exactly to the computed values.

As a consequence, the real distance $\bar{\sigma}_0 - \bar{\sigma}_1$ is smaller than the one computed using equation (1). Hence, the theoretical analysis gives an interesting bound on the relative distance $\varepsilon = \frac{\bar{\sigma}_0 - \bar{\sigma}_1}{\bar{\sigma}_0}$: $\varepsilon_{\text{measured}} < \varepsilon_{\text{computed}}$.

6.2.3 Hypothesis testing

Indeed, this ε is the value of interest. The goal is to decide whether or not the distance δ is in the distance spectrum of h. This problem is actually a simple question of hypothesis testing, and the most relevant test is just decide if the observed value of the average syndrome weight is closer to $\bar{\sigma}_0$ or to $\bar{\sigma}_1$.

The threshold for the decision test at the end of Algorithm 3 is hence to be taken as $\frac{\bar{\sigma}_0 + \bar{\sigma}_1}{2}$.

There is a lot of litterature about hypothesis testing, and in particular a theorem from Chernoff [**?**], according to which, in such a scenario of repeated Bernouilli trials, one needs $\geq \frac{1}{\varepsilon^2}$ trials for the decision test to be relevent, where ε is the distance between the two outcomes.

This means that to decide if the distance δ is in the spectrum or not, we need roughly $\frac{1}{\varepsilon^2}$ decoding tests involving an error pattern containing the distance δ. In the attack, the error patterns are chosen randomly and a given distance appears with frequency α where:

$$\alpha := \mathbb{P}(d \in \Delta(e)) = 1 - \frac{\prod_{k=0}^{t-1}(n-3k)}{\prod_{k=0}^{t-1}(n-k)}.$$

For usual 80-bits security, $\alpha \sim 0.34$.

Hence as we can compute α and give a bound on ε, according to Chernoff's result, we need at least $\frac{1}{\varepsilon^2} \cdot \frac{1}{\alpha \cdot r}$ decoding queries to recover the distance spectrum

of the key, which gives a lower bound of $N \geq 6800$ for the usual parameters, which is reasonable considering the results obtained with 10^5 trials on Fig. 1 where we can already distinguish the spectrum.

7 Timing Attack

Now that we know that the syndrome weight leaks information, any parameter correlated to this quantity could be used for a side channel attack. An interesting parameter that is often easy to measure is the time.

The decoding algorithm for QC-MDPC codes is an iterative algorithm with no termination proof. The number of rounds needed to correct the errors varies. This has been studied by in [6]. For the usual parameters for 80 bits of security, the algorithm usually corrects the error in 3 rounds, but some instances need 4, 5 or even more iterations. Usual implementations abort after a certain number of rounds (around 10), this is what was used for the attack in [14].

This motivated us to try to perform a timing attack. The scenario is the same as previously, but instead of observing the syndrome weight, Eve can measure the number of iterations needed to decode her message. To obtain the spectrum, Eve uses the exact same data collection algorithm: for every distance in the spectrum, she computes the average number of iterations needed to correct an error containing this distance. On average, the number of iterations is slightly lower when the distance appears in the key's spectrum. This works well and it's possible to recover fully the distance spectrum with 100 million samples on 80-bits security QC-MDPC scheme (see Fig. 5 in appendix).

This is the first timing attack on the QC-MDPC scheme. It indicates that we need constant-time decoding, such as for QcBits [8], but with a more efficient decoding algorithm.

8 Conclusions and future work

In section 4.1.3 we show that the QC-MDPC scheme has a flaw based on its inherent quasi-cyclic nature. This flaw affects the distribution of the weight of the syndrome.

Any implementation of the scheme should make sure that this value is masked, as well as all other parameters correlated with it. This includes for example the number of steps of the decoding process. Therefore, a constant-time decoder is needed to prevent timing attacks (like in section 7), with a small enough failure rate.

The best way to prevent any attack of this form is to make sure, using a theoretical analysis and Chernoff's bound such as in section 6.2, that the direct attack on the syndrome weight (studied in section 5) requires too many samples to recover the spectrum. Hence, as this is the origin of the flaw, other side-channel attacks on different (more noisy) parameters exploiting the same idea will be even more costly.

This level of security can be ensured through a proper choice of parameters. It's also possible to add in the specification of the system a bound on the allowed number of decryption queries (just like for signatures the system specifies the number of allowed signatures using the same key).

In order to do so, we need to understand more deeply the algorithmic problem of reconstruction of the key from a partial spectrum. Indeed, we could choose proper parameters to prevent the recovery of the full spectrum, but if it's actually possible to recover the key from only half of the spectrum's values, there is still an attack. This problem should be studied further.

Many other proposals for post-quantum cryptography face the same problems of long key size and suggest to use a quasi-cyclic structure to have shorter keys. We believe some of them might be exposed to the same kind of attacks and intend to study the possibility to adapt the idea of this attack to other cryptosystems.

Remerciements

Je tiens à remercier Nicolas Sendrier qui m'a encadré durant ce stage et a accepté d'être mon directeur de thèse. Ses conseils réguliers me sont toujours précieux et je suis très heureux de pouvoir faire mes premiers pas en recherche sous sa tutelle. Je veux également remercier Anne Canteaut qui a pris le temps de discuter avec moi lorsque je cherchais un stage et m'a orientée vers Nicolas.

Je remercie l'ensemble de l'équipe Secret qui m'a chaleureusement accueilli. C'est agréable de travailler au quotidien dans une si bonne ambiance. Merci particulièrement à Julia qui m'a précédée sur ce sujet et m'a laissé tous ses articles et à Thomas qui m'a plusieurs fois orienté vers les bonnes références.

Je remercie mes collègues du bureau *Tapdance*, Ferdinand, Mathilde et Valentin, pour nos discussions enflammées autour du tableau.

Enfin, je tiens à remercier tous les enseignants que j'ai rencontré cette année au sein du MPRI. Grâce a eux j'ai pu en quelques mois avoir une vue d'ensemble de la recherche actuelle en cryptographie et approfondir plusieurs sujets avant de choisir mon sujet de thèse.

A Experimental Results

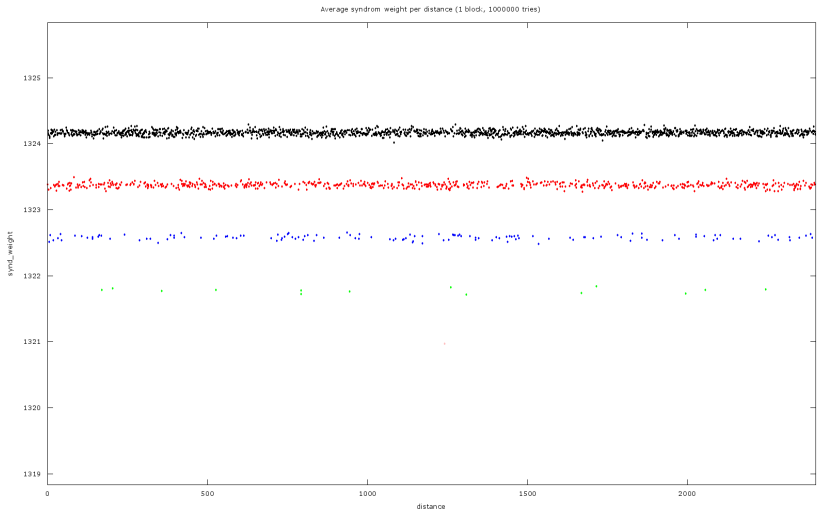


Figure 2: Average syndrome weight per distance, 10^6 samples, 1 block.

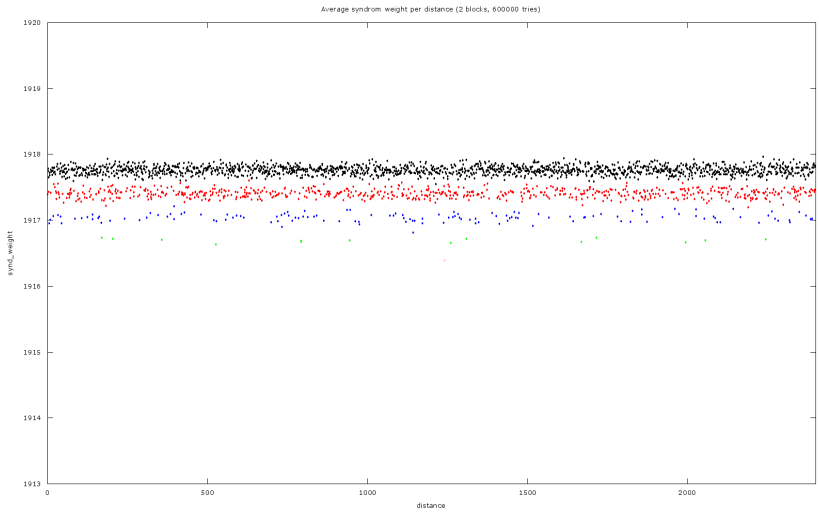


Figure 3: Average syndrome weight per distance, $6 \cdot 10^5$ samples, 2 blocks.

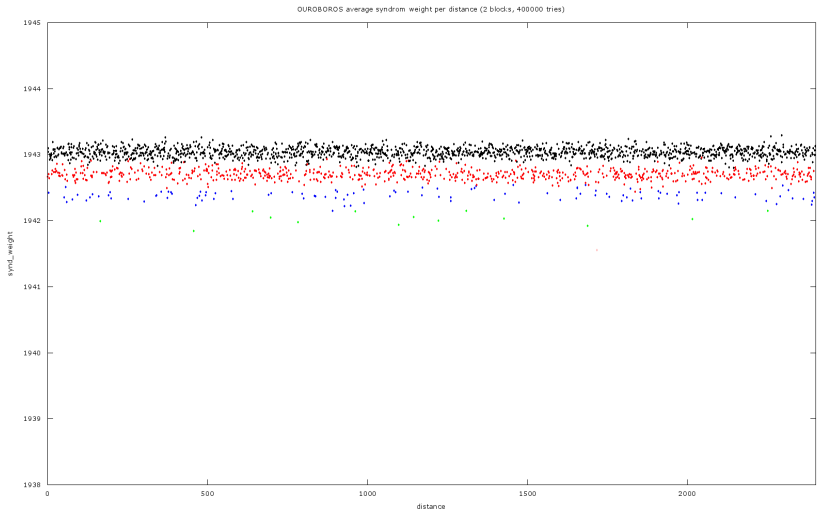


Figure 4: Average syndrome weight per distance on Ouroboros scheme, $4 \cdot 10^5$ samples, 1 block.

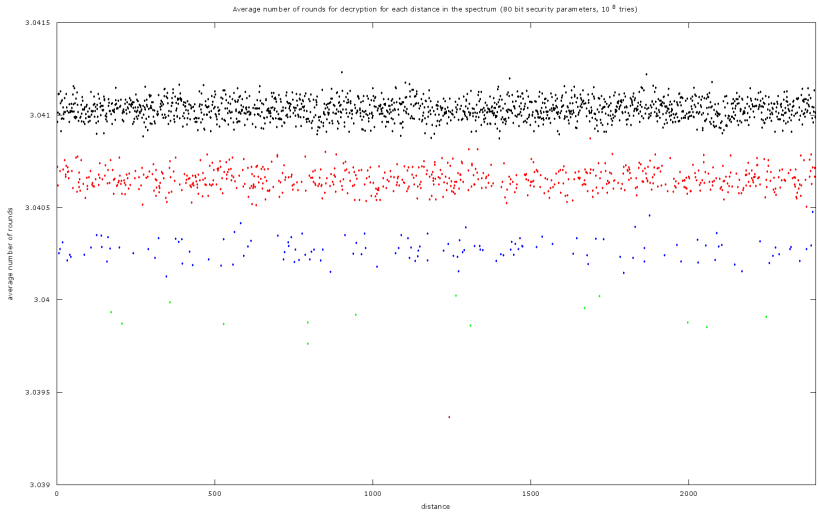


Figure 5: Average number of iterations to decode per distance, 10^8 samples

B Bibliography

[1] M. Alekhnovich. More on average case vs approximation complexity. *Computational Complexity*, 20(4):755–786, 2011.

[2] D. Augot, L. Batina, D. J. Bernstein, J. Bos, J. Buchmann, W. Castryck, O. Dunkelman, T. Güneysu, S. Gueron, A. Hülsing, T. Lange, M. S. E. Mohamed, C. Rechberger, P. Schwabe, N. Sendrier, F. Vercauteren, and B.-Y. Yang. Initial recommendations of long-term secure post-quantum systems, 2015.

[3] M. Baldi, M. Bodrato, and F. Chiaraluce. A new analysis of the mceliece cryptosystem based on QC-LDPC codes. In *Security and Cryptography for Networks, 6th International Conference, SCN 2008, Amalfi, Italy, September 10-12, 2008. Proceedings*, pages 246–262, 2008.

[4] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Trans. Information Theory*, 24(3):384–386, 1978.

[5] J. Chaulet. *Étude de cryptosystèmes à clé publique basés sur les codes MDPC quasi-cycliques*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2017.

[6] J. Chaulet and N. Sendrier. Worst case QC-MDPC decoder for mceliece cryptosystem. In *IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016*, pages 1366–1370, 2016.

[7] L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone. Report on post-quantum cryptography, 2016.

[8] T. Chou. Qcbits: Constant-time small-key code-based cryptography. In *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, pages 280–300, 2016.

[9] J. Deneuville, P. Gaborit, and G. Zémor. Ouroboros: A simple, secure and efficient key exchange protocol based on coding theory. In *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, pages 18–34, 2017.

[10] T. Fabsic, V. Hromada, P. Stankovski, P. Zajac, Q. Guo, and T. Johansson. A reaction attack on the QC-LDPC mceliece cryptosystem. In *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, pages 51–68, 2017.

[11] J. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, and J. Tillich. A distinguisher for high rate mceliece cryptosystems. In *2011 IEEE Information Theory Workshop, ITW 2011, Paraty, Brazil, October 16-20, 2011*, pages 282–286, 2011.

[12] P. Fouque and G. Leurent. Cryptanalysis of a hash function based on quasi-cyclic codes. In *Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings*, pages 19–35, 2008.

[13] R. G. Gallager. Low-density parity-check codes. *M.I.T. Press*, 1963.

[14] Q. Guo, T. Johansson, and P. Stankovski. A key recovery attack on MDPC with CCA security using decoding errors. In *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, pages 789–815, 2016.

[15] S. Heyse, I. von Maurich, and T. Güneysu. Smaller keys for code-based cryptography: QC-MDPC mceliece implementations on embedded devices. In *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, pages 273–292, 2013.

[16] K. Kobara and H. Imai. Semantically secure mceliece public-key cryptosystems-conversions for mceliece PKC. In *Public Key Cryptography, 4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001, Cheju Island, Korea, February 13-15, 2001, Proceedings*, pages 19–35, 2001.

[17] R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 44:114–116, Jan. 1978.

[18] R. Misoczki, J. Tillich, N. Sendrier, and P. S. L. M. Barreto. Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. In *Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul, Turkey, July 7-12, 2013*, pages 2069–2073, 2013.

[19] A. Otmani, J. Tillich, and L. Dallot. Cryptanalysis of two mceliece cryptosystems based on quasi-cyclic codes. *CoRR*, abs/0804.0409, 2008.

[20] M. Rossi, M. Hamburg, M. Hutter, and M. E. Marson. A side-channel assisted cryptanalytic attack against qcbits. *IACR Cryptology ePrint Archive*, 2017:596, 2017.

[21] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134, 1994.

[22] I. von Maurich, T. Oder, and T. Güneysu. Implementing QC-MDPC mceliece encryption. *ACM Trans. Embedded Comput. Syst.*, 14(3):44:1–44:27, 2015.