Two algorithms to improve the reaction attack on the QC-MDPC McEliece

Thales Bandiera Paiva[1]*, **Routo Terada**[1]†

[1]Instituto de Matemática e Estatística – Universidade de São Paulo (USP)

{tpaiva,rt}@ime.usp.br

Abstract. *In 2016, a reaction attack on the QC-MDPC McEliece scheme was presented at Asiacrypt by Guo et al.. This attack exploits one aspect that was not considered in the scheme's security reduction: the probability of a decoding failure to occur is lower when the secret key and the error used for encryption share certain properties, which were called spectrums. By detecting decoding failures, the attacker can obtain information on the spectrum of the secret key and then use this information to reconstruct the key. To improve the efficiency of the attack, we propose two different key reconstruction algorithms that are more efficient and use less information on the secret key than Guo's et al. one. Furthermore, both algorithms can be trivially parallelized.*

1. Introduction

In 2013, a new variant of the McEliece scheme that uses quasi-cyclic moderate-density parity-check codes (QC-MDPC) was presented by Misoczki et al. [Misoczki et al. 2013]. This variant promises extremely compact public keys of only 4801 bits for a security level of 80 bits, and has an apparently strong security reduction. The European initiative PQCRYPTO, which supports the development of post-quantum cryptography, considered this variant as a serious candidate for a post-quantum secure standard in the 2015 revision of its paper with recommendations for post-quantum secure systems [Augot et al. 2015].

Until the end of 2016, the QC-MDPC has not suffered critical attacks. However at Asiacrypt 2016 Guo, Johansson, and Stankovski [Guo et al. 2016] presented an efficient reaction attack for key recovery on the QC-MDPC. This attack is based on the fact that QC-MDPC decoders can fail. When a decoding failure occurs, the receiver asks the sender to resend the message, which hopefully, will be encrypted with an error pattern that the decoder will be able to correct. The main observation of the authors is that the probability that the decoder fails when correcting the error $\mathbf{e}$ is significantly smaller when $\mathbf{e}$ and the secret key share some certain properties.

In the McEliece scheme and its variants, Alice's secret key is a linear code $\mathcal{G}$ capable of correcting t errors with high probability, for which Alice knows an efficient decoder. Her public key is the generator matrix of $\mathcal{G}$, possibly scrambled, in such a form that it is unfeasible for an attacker to use this matrix for building an efficient decoder for $\mathcal{G}$. To send Alice a message $\mathbf{m}$, we encode $\mathbf{m}$ using her public key and add t intentional errors to this encoding, giving us $\mathbf{c}$. We now can safely send her $\mathbf{c}$, because Alice is the only one that has an efficient decoder for $\mathcal{G}$, which she uses to obtain $\mathbf{m}$ from $\mathbf{c}$.

*Supported by Capes grant number 00.889.834/0001-08.

†Supported by FAPESP grant number 2015/01587-0.

Guo's et al. [Guo et al. 2016] attack is done in two parts. In the first part, an attacker Eve sends a number of ciphertexts to her target Alice, and records for which error vectors the decoder failed or succeeded, to obtain some information on Alice's secret key, called its spectrum. The second part is the key reconstruction, where Eve, without interacting any further with Alice, builds Alice's secret key with the information gathered in the first part.

1.1. Motivation

We address the following three problems of Guo's et al. reconstruction algorithm:

1. it cannot recover the secret key when the information about it is incomplete;
2. the number of operations needed to find the key grows very fast with respect to the security level;
3. it is recursive in nature, and it is not obvious how to parallelize it, neither how much one gains by doing it.

By developing more efficient key reconstruction algorithms, an attacker can recover the secret key using less interaction with the secret key holder. Therefore improvements in key reconstruction algorithms significantly affect the secure lifetime of a secret key, and the parameters for different security levels.

1.2. Original contributions

Our work presents two contributions, which are key reconstruction algorithms that are more efficient, both asymptotically and in practice, and use less information on the secret key than needed by Guo's et al. one. Furthermore, both algorithms can benefit from parallel implementations. The MSc dissertation on which this paper is based, entitled "Melhorando o ataque de reação contra o QC-MDPC McEliece", is available at `www.ime.usp.br/~tpaiva/msc`.

The first algorithm is a simple randomized extension of Guo's et al. algorithm. For comparison purposes, the randomized algorithm can be 3 orders of magnitude faster than Guo's et al. one when they are given the complete spectrum of the secret key.

The second algorithm, which we consider to be the main contribution of this work, is significantly different from Guo's et al. algorithm. It uses a linear relation among some components of the secret key. Even with around 50% less information on the spectrum than needed by Guo's et al. algorithm, our algorithm runs faster. Its complexity is $O\left(r^4\right)$, where r is the codimension of the QC-MDPC code. This complexity is a significant improvement with respect to the other algorithms, for which the complexity increases rather drastically when higher security levels are considered. A paper describing this algorithm and comparing it with Guo's et al. algorithm was accepted for publication on IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences [Paiva and Terada 2018].

2. The QC-MDPC McEliece scheme

This section presents an overview on the QC-MDPC McEliece [Misoczki et al. 2013] scheme and the reaction attack against it. The QC-MDPC McEliece is a variant of the McEliece scheme that uses QC-MDPC codes, which are formally defined next.

Definition 1 (QC-MDPC) *An (n, r, w)-QC-MDPC code is a quasi-cyclic linear code of length n, co-dimension r which divides n, that has a sparse parity-check matrix $\mathbf{H}$ with row weights $w = O\left(\sqrt{n \log n}\right)$, and formed by $n_0 = n/r$ cyclic blocks.*

The parameters suggested by Misoczki et al. [Misoczki et al. 2013] are shown in Table 1. In this table, n_0, n, r, w are the parameters of the QC-MDPC code as defined above, and t is the maximum number of errors it can decode with a sufficiently high probability. The suggested parameters for the QC-MDPC McEliece entail extremely small keys when compared with the key size of hundreds of megabytes for the original McEliece scheme [Bernstein et al. 2013].

Table 1. Suggested QC-MDPC parameters for some security levels [Misoczki et al. 2013]

Security	n_0	n	r	w	t	Key size in bits
80	2	9602	4801	90	84	4801
128	2	19714	9857	142	134	9857
256	2	65542	32771	274	264	32771

We describe how to generate keys pairs for the QC-MDPC McEliece with the security level λ. Let n_0, n, r, and w be parameters supporting this security level λ, which can be taken from Table 1. First generate a random QC-MDPC code by choosing at random a binary vector $\mathbf{h}$ from $\mathbb{F}_2^n = \{0,1\}^n$ such that the weight[1] of $\mathbf{h}$ is w. Break[2] $\mathbf{h}$ into $n_0 = 2$ equal parts $\mathbf{h} = [\,\mathbf{h}_0 \,|\, \mathbf{h}_1\,]$. Then build the parity-check matrix $\mathbf{H} = [\,\mathbf{H}_0 \,|\, \mathbf{H}_1]$, where each $\mathbf{H}_i$ is the cyclic matrix with $\mathbf{h}_i$ as its first row. It is required that the block $\mathbf{H}_{n_0-1}$ is invertible. If it is not, restart the key generation procedure by picking another $\mathbf{h}$ at random. We now build the generator matrix as $\mathbf{G} = \left[\, \mathbf{I} \,\middle|\, \left(\mathbf{H}_1^{-1} \cdot \mathbf{H}_0\right)^T \,\right]$, and one can check that $\mathbf{G}\mathbf{H}^T = \mathbf{0}$ with a simple evaluation. Since each $\mathbf{H}_i$ is cyclic, each product $\left(\mathbf{H}_{n_0-1}^{-1}\mathbf{H}_i\right)^T$ is also cyclic. The secret key is the matrix $\mathbf{H}$ and the public key is the matrix $\mathbf{G}$. Both matrices admit compact representation because of their cyclic blocks.

To encrypt a vector $\mathbf{m}$ of $\mathbb{F}_2^{(n-r)}$, first encode $\mathbf{m}$ obtaining $\hat{\mathbf{c}} = \mathbf{m}\mathbf{G}$. Then add a random vector $\mathbf{e}$ of weight t to $\hat{\mathbf{c}}$ to get the ciphertext $\mathbf{c} = \hat{\mathbf{c}} + \mathbf{e}$.

Since the ciphertext is just a corrupted codeword, decrypting $\mathbf{c}$ is equivalent to correct the errors from $\mathbf{c}$. This is achieved with iterative decoders [Gallager 1962], which can be of two types: based on either hard or soft decision. Soft decision based decoders have better error correction capability, but are less efficient. It is usual to decode QC-MDPC codes with hard decision based decoders, because the main problem here is not efficient communication, but secure communication. Therefore one does not need to correct a lot of errors, but only enough to make the scheme secure.

The reaction attack is based on the observation that the probability of a failure to occur when decoding a vector $\mathbf{c} = \hat{\mathbf{c}} + \mathbf{e}$ is lower when the blocks of $\mathbf{h}$ share some similarities with the blocks of the error $\mathbf{e}$. The important structural information of the

[1] The (Hamming) weight of a vector is the number of its non-null entries.

[2] For simplicity, we only consider the case $n_0 = 2$, but our results can easily be extended for $n_0 \geq 3$.

blocks of the secret vector $\mathbf{h}$ which an attacker can recover from decoding failures is called the spectrum, and it is defined next.

Definition 2 *Let* $\mathbf{v} = [v_1, v_2, \ldots, v_r]$ *be an element of* $\mathbb{F}_2^r$. *Then the spectrum of* $\mathbf{v}$ *is the set* $\sigma(\mathbf{v}) = \{\text{dist}_r(i, j) : i \neq j,\ v_i = 1,\ \text{and}\ v_j = 1\}$, *where* $\text{dist}_r(i, j)$ *denotes the circular distance*[3] *between the positions* i *and* j *in a vector of length* r.

The attack is done in two parts. In the first, the attacker collects information about the key by sending challenge ciphertexts to the secret key holder Alice, and recording Alice's reactions when he tries to decode these ciphertexts. The first part is the only part where the attacker needs to interact with the secret key holder. In the second part, the attacker tries to reconstruct the key using the information on the spectrums of the blocks of $\mathbf{h}$ previously collected. Given the spectrum of some of the blocks of $\mathbf{h}$, say $\sigma(\mathbf{h}_0)$, the reconstruction algorithm presented by Guo et al. is a simple pruned depth-first search in a tree. Its main problem is that, to run efficiently, it needs a lot of information on the spectrum of the key, which requires lots of interactions with the secret key holder. Further, being a depth-first search, it can be stuck in an unfruitful branch of the tree for too long.

The next sections present two algorithms for the second part of the attack.

3. A randomized key reconstruction algorithm

Our first algorithm, which is given as Algorithm 1, is a simple randomized extension of Guo's et al. key reconstruction algorithm. Instead of performing a depth-first search for the key, at each level of the search tree, the algorithm chooses the next node at random.

Algorithm 1: Randomized variant of the key reconstruction algorithm

Data: n, r, w, t parameters of the QC-MDPC to be attacked
$\hat{w}$ the weight of $\mathbf{h}_0$
s_0 a distance inside $\sigma(\mathbf{h}_0)$
D_0 a set of distances which are not in $\sigma(\mathbf{h}_0)$
Result: V the support of a rotation of $\mathbf{h}_0$

```
1  begin
2      do
3          V ← {1, 1 + s_0}
4          F_2 ← {i ∈ [⌊r/2⌋] − V : dist_r(i, v) ∉ D_0 ∀v ∈ V}
5          l ← 2
6          while |V| < ŵ e |F_l| > 0 do
7              p ← a random element from F_l
8              V ← V ∪ {p}
9              F_l ← F_l − {p}
10             F_{l+1} ← {i ∈ F_l : dist_r(i, v) ∉ D_0 ∀v ∈ V}
11             l ← l + 1
12     while V is not the support of a rotation of h_0;
13     return V
```

[3]The circular distance between two indexes is simply the minimum number of steps to get from one position to another supposing the vector is represented by a circular array.

We now give a brief description of the algorithm. Parameters s_0, which must be a distance inside the spectrum of $\mathbf{h}_0$, and D_0, which is a set of distances outside the spectrum of $\mathbf{h}_0$, are obtained by the spectrum recovery algorithm, as described by Guo et al. [Guo et al. 2016]. At each iteration, the algorithm starts with the set $V = \{1, s_0+1\}$ and tries to complete it with $\hat{w} - 2$ indexes. To complete the support vector V, the algorithm chooses at random an index inside the auxiliary set F_l, which contains, for each level l, the possible positions to complete the support. That is, F_l consists of all the elements of $\{1, \ldots, r\}$ which are not in V, and whose circular distance to any index in V is not in D_0.

4. An iterative key reconstruction algorithm

Consider the public generator matrix of a QC-MDPC code with $n_0 = 2$, which is given as $\mathbf{G} = \left[\mathbf{I} \,\middle|\, \left(\mathbf{H}_1^{-1} \cdot \mathbf{H}_0\right)^T \right]$. Let $\mathbf{B} = \mathbf{H}_1^{-1}\mathbf{H}_0$ be the transpose of the right block of the public generator matrix. Our main idea is to explore the relation $\mathbf{h}_1\mathbf{B} = \mathbf{h}_0$, where $\mathbf{h}_1$ and $\mathbf{h}_0$ are corresponding lines of the matrices $\mathbf{H}_1$ and $\mathbf{H}_0$, respectively. Let Z_0 and Z_1 be sets of indexes of some of the null entries of $\mathbf{h}_0$ and $\mathbf{h}_1$, respectively. Denote by $\mathbf{B}^{Z_0}$ the matrix consisting of the columns of $\mathbf{B}$ whose indexes are in Z_0, which makes $\mathbf{h}_1\mathbf{B}^{Z_0} = \mathbf{0}$.

We can discard the entries of $\mathbf{h}_1$ whose indexes are in Z_1, if we discard the corresponding columns in $\mathbf{B}^{Z_0}$. Let Z_1' be the complement of Z_1 with respect to the possible indexes of $\mathbf{h}_1$. Then $\mathbf{h}_1^{Z_1'}\mathbf{B}^{Z_0}_{Z_1'} = \mathbf{0}$, where $\mathbf{h}_1^{Z_1'}$ is the vector consisting of the columns of $\mathbf{h}_1$ whose indexes are in Z_1', and $\mathbf{B}^{Z_0}_{Z_1'}$ is the matrix consisting of the lines from $\mathbf{B}^{Z_0}$ whose indexes are in Z_1'. In other words, $\mathbf{h}_1^{Z_1'}$ is in the left kernel of the matrix $\mathbf{B}^{Z_0}_{Z_1'}$. Then we can compute the kernel matrix of $\mathbf{B}^{Z_0}_{Z_1'}$ and hope to find $\mathbf{h}_1^{Z_1'}$ in one of its columns.

Let D_0 and D_1 be sets of distances which are not in the spectrum of $\mathbf{h}_0$ and $\mathbf{h}_1$, respectively. Suppose we know that s_0 is in $\sigma(\mathbf{h}_0)$, and we also know that the distances $d_1, \ldots, d_l$ in D_0. Letting $*$ denote unknown entries, we know that there must exist a shift of $\mathbf{h}_0$ which has the following format

$$[\underbrace{0 * \ldots * \overbrace{1 * \ldots * 0}^{s_0} * \ldots *}_{d_1 \quad d_1} \underbrace{0 * \ldots * 1}_{d_1} * \ldots * \underbrace{0}_{d_1} * \ldots *].$$

Further, if we consider all other d_i, it is possible to know the positions of a hopefully large number of zeros in this shift of $\mathbf{h}_0$, which will be the set Z_0. Notice that distance s_0 that is in the spectrum $\sigma(\mathbf{h}_0)$ with high probability, one can take the distance with least probability of failure estimated by the spectrum recovery algorithm. An analogous construction can be made for Z_1.

Using the construction above, we can obtain sets Z_0 and Z_1 which consist of positions of non-null entries of some circular shifts of $\mathbf{h}_0$ and $\mathbf{h}_1$, respectively. The problem is that these shifts might not be by the same amount of positions, thus with high probability $\mathbf{h}_1^{Z_1'}$ is not in the kernel of $\mathbf{B}^{Z_0}_{Z_1'}$. To deal with this, we fix Z_1' and iterate through the circular rotations of the indexes in Z_0. When the shift of Z_0 corresponds to the shift of Z_1, $\mathbf{h}_1^{Z_1'}$ can be computed.

We now put everything together to give a full description of our algorithm to reconstruct the key as Algorithm 2. Since the algorithm is not recursive and uses common operations, it is straightforward to compute its complexity as $O(r^4)$.

Algorithm 2: Proposed key recovery algorithm

Data: r, w parameters of the QC-MDPC code to be broken
D_0, D_1 sets of distances not in the $\sigma(\mathbf{h}_0)$ and $\sigma(\mathbf{h}_1)$, respectively
s_0, s_1 distances in the spectrum of $\mathbf{h}_0$ and $\mathbf{h}_1$, respectively
$\mathbf{B}$ the right block of the public generator matrix

Result: $\mathbf{h}_1$ which is some line of the matrix $\mathbf{H}_1$, or $\perp$ if $\mathbf{h}_1$ could not be found

$Z'_1 \leftarrow \{i \in [r] : \text{dist}(1, i) \notin D_1 \text{ and } \text{dist}(s_1 + 1, i) \notin D_1\}$
$\mathbf{B}_{Z'_1} \leftarrow$ rows of $\mathbf{B}$ whose indexes are in Z'_1
for $p = 0$ *to* $r - 1$ **do**
  $Z_0 \leftarrow \{i + p \mod r : \text{dist}(1, i) \in D_0 \text{ or } \text{dist}(s_0 + 1, i) \in D_0\}$
  $\mathbf{B}^{Z_0}_{Z'_1} \leftarrow$ columns of $\mathbf{B}^{Z'_1}$ whose indexes are in Z_0
  $\mathbf{K} \leftarrow$ left kernel matrix of $\mathbf{B}^{Z_0}_{Z'_1}$
  if $\dim \mathbf{K} = 1$ **then**
    $\mathbf{v} \leftarrow$ the only row in $\mathbf{K}$
    if $\text{weight}(\mathbf{v}) \leq w$ **then**
      $\mathbf{h}_1 \leftarrow \mathbf{0} \in \mathbb{F}_2^r$
      for *each* i = *1 to* $|\mathbf{v}|$ **do**
        $\mathbf{h}_1[Z_1[i]] \leftarrow \mathbf{v}[i]$
      return $\mathbf{h}_1$
return $\perp$

It is important to note that the lemma above does not say anything about the probability of the algorithm finding the key. It only states that the algorithm runs in $O(r^4)$, whether it finds the key or not. The probability that it finds the key is the probability that the kernel of $\mathbf{B}^{Z_0}_{Z'_1}$ has dimension 1, which happens when Z_0 and Z_1 are sufficiently large. Under the assumption that the matrix $\mathbf{B}^{Z_0}_{Z'_1}$ behaves somewhat like a random matrix, it is easy to show that the probability that the algorithm finds the key is lower bounded by $1 - 2^{r-|Z_1|-|Z_0|}$.

5. Performance of the proposed algorithms

We implemented both algorithms in C language and run it using an Intel i7 870 Lynnfield CPU, at a 2.93GHz clock frequency, and 8GB of RAM. The implementation of the randomized algorithm is trivial, without any optimizations. For the iterative algorithm, we used the M4RI library [Albrecht, M. and Bard, G. 2012] for the kernel matrix computation. The source code is available at `www.ime.usp.br/~tpaiva/msc`.

Table 2 shows the performance of Guo's et al. algorithm[4] and both of our proposed algorithms when they have access to the full spectrum of the code. We can see that the

[4]We implemented Guo's et al. algorithm with a minor optimization [Fabšič et al. 2017] which cuts in half the expected number of paths needed to recover the key.

randomized algorithm outperforms the other two by a large margin, while Guo's et al. algorithm is the slowest one. In the following tables, when entries are marked with *, it means that it was used a simple variant[5] of the iterative algorithm more suited for the cases where there are abundant information on the spectrums.

Table 2. Performance comparison of the algorithms when *full* spectrums are known, for different security parameters

Security level λ	n_0	Average running time of Guo's et al. algorithm	Average running time of the randomized algorithm	Average running time of the iterative algorithm
80	2	71.18s	0.01s	0.72s*
128	2	$\sim$ 24 days	0.36s	11s*
256	2	-	33s	141s*

However we are really interested in how the algorithms perform when they only know a fraction of the distances outside the spectrums. The average of the observed running times of our algorithms when they know a fraction of the distances outside the spectrums is shown in Table 3. Guo's et al. algorithm is not considered in this table because it takes too long to finish when partial information on the spectrums is considered.

Table 3. Performance comparison of our algorithms algorithm when *partial* spectrums are known, for different security parameters

Security level λ	n_0	Fraction of known distances outside the spectrum	Average running time of the randomized algorithm	Average running time of the iterative algorithm
80	2	80%	1.1s	1s*
80	2	62.93%	1h	1.3s*
80	2	45.63%	-	41s
128	2	87.36%	4m	8.6s*
128	2	77.28%	50m	10s*
128	2	49.58%	-	14m
256	2	94.84%	20m	2m12s*
256	2	89.99%	1h40m	2m24s*
256	2	52.34 %	-	15h34m

6. Conclusion

This work presents two new algorithms for key reconstruction from the spectrums of the rows of the secret matrix. Both algorithms are more efficient than Guo's et al. one, and can be trivially parallelized. But their main feature is that they can run with less information on the private key, which means an attacker needs to interact significantly less with the private key holder. We ran simulations[6] considering the CCA2 setting for the 80 bits

[5]The variant consists in taking $s_0 = 0$.

[6]The simulations used the HPC resources provided by the Technology Superintendence of USP.

security parameters with $n_0 = 2$. The results were that, with 29M interactions on average, and 60M in the worst case, we were able to successfully recover the key using either of the proposed algorithms. This is a significant improvement on the 200M decoding trials reported by Guo et al..

The first algorithm is a randomized variant of Guo's et al.. Even though the randomized algorithm is much more efficient, since it is also based on a search tree, it suffers from scalability issues just like Guo's et al. algorithm when the amount of information on the spectrum gets lower. The second algorithm is iterative and it is based on a linear relation among blocks of the secret key. This avoids the exponential loss in performance when there are less information on the secret key spectrum.

An immediately interesting future work is to protect the QC-MDPC McEliece against the reaction attack. As discussed by Guo et al. [Guo et al. 2016], a conservative way to protect the scheme is to develop decoders that fail with probability close to $2^{-\lambda}$ when using parameters for the security level λ. The problem is that these decoders are far from today's technology, and it is not clear if we can achieve this level of precision, and even if we could, at what cost in performance. We believe it is more realistic to develop decoding algorithms which fail with probability independent of the similarity between the spectrums and the error vector. Even if it fails with a non-negligible probability, the information on the spectrum would be protected.

References

Albrecht, M. and Bard, G. (2012). *The M4RI Library – Version 20121224*. The M4RI Team.

Augot, D., Batina, L., Bernstein, D. J., Bos, J., Buchmann, J., Castryck, W., Dunkelman, O., Güneysu, T., Gueron, S., and Hülsing, A. (2015). Initial recommendations of long-term secure post-quantum systems.

Bernstein, D. J., Chou, T., and Schwabe, P. (2013). McBits: fast constant-time code-based cryptography. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 250–272. Springer.

Fabšič, T., Hromada, V., Stankovski, P., Zajac, P., Guo, Q., and Johansson, T. (2017). A reaction attack on the QC-LDPC McEliece cryptosystem. In *International Workshop on Post-Quantum Cryptography*, pages 51–68. Springer.

Gallager, R. (1962). Low-density parity-check codes. *IRE Transactions on information theory*, 8(1):21–28.

Guo, Q., Johansson, T., and Stankovski, P. (2016). A key recovery attack on MDPC with CCA security using decoding errors. In *22nd Annual International Conference on the Theory and Applications of Cryptology and Information Security (ASIACRYPT)*.

Misoczki, R., Tillich, J. P., Sendrier, N., and Barreto, P. S. L. M. (2013). MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In *IEEE International Symposium on Information Theory*, pages 2069–2073. IEEE.

Paiva, T. B. and Terada, R. (2018). Improving the efficiency of a reaction attack on the QC-MDPC McEliece (to appear in 2018). *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*.